The Hitchhiker's Guide to Online Anonymity

(Or "How I learned to start worrying and love privacy")

Version 0.7.4 (draft), February 2021 (work in progress, some parts are incomplete) by AnonymousPlanet.

Remember to check frequently for new versions of this guide.

This guide is a non-profit open-source initiative, licensed under Creative Commons Attribution 4.0 International (cc-by-4.0).

Find it online at https://anonymousplanet.github.io/
Latest PDF version (best format for the best readability) of this guide at:
https://anonymousplanet.github.io/thgtoa/guide.pdf
Alternative password (weak password) protected PDF version of this guide at:
https://anonymousplanet.github.io/thgtoa/guide-p.pdf (password is **thgtoa**)
Feel free to submit issues using GitHub Issues at: https://github.com/AnonymousPlanet/thgtoa/issues
Feel free to discuss ideas using GitHub Discussions at: https://github.com/AnonymousPlanet/thgtoa/discussions

This guide does mention and even recommend some commercial services in some sections (such as VPN and hosting providers) but is not endorsed/sponsored by any of them in any way. There are no referral links and no commercial ties with any of these providers.

Donations are welcome at https://anonymousplanet.github.io/thgtoa/donations.html or directly by sending Monero to this address:

"monero:4549BGJrEPBfpiPRL9CVGzGMgJnC1Dzf8EXLVfY8Ukrnj7LzkTV611dGf9tuQHiSQjbixsNWiffNiV5fPB3LkyF7UXi3vwQ" **(please do verify the checksum and gpg signature of this file for authenticity, this is explained in the README of the repository if you don't know how to do that)**.

Follow me on Twitter https://twitter.com/AnonyPla (can't guarantee this account will stay up for long tho)
An automated mirror of the repository of this guide is available at https://gitlab.com/AnonymousPlanet/thgtoa in case of issue with GitHub.

There are several ways you could read this guide:

- You want to understand the current state of online privacy and anonymity not necessarily get too technical about it: Just read the Introduction:, Requirements:, Understanding some basics of how some information can lead back to you and how to mitigate those: and A final editorial note: sections.
- You want to do the above but also learn how to remove some online information about you: Just read the above and add the Removing some traces of your identities on search engines and various platforms:
- You want to do the above and create online anonymous identities online safely and securely: Read the whole guide.

Table of Contents

Introduction: **6**

Requirements: **8**

Understanding some basics of how some information can lead back to you and how to mitigate those: **8**

Your IP address: *8*

Your DNS requests: *8*

Your IMEI and IMSI (and by extension, your phone number): *12*

Your Wi-Fi or Ethernet MAC address: *13*

Your Bluetooth MAC address: *13*

Your Operating Systems and Apps telemetry services: *14*

The WIFIs and Bluetooth devices around you: *14*

Your Metadata including your Geo-Location: *15*

Your Smart devices in general: *16*

Your Devices can be tracked even when offline: *16*

Your RFID enabled devices: *16*

Your Files Properties/Metadata: *17*

Malware hidden in your files/documents: *18*

Watermarking: *18*

Pictures/Videos/Audio: 18

Printing Watermarking: 19

Your Pixelized/Blurred Information: *19*

Your "Anonymized" Tor/VPN traffic: *20*

Your Crypto currencies transactions: *23*

Exploits in your apps and services: *23*

Your CPU: *23*

Malicious/Rogue Wi-Fi Access Points: *24*

Malicious USB devices: *25*

Your Cloud backups/sync services: *25*

Your Digital Fingerprint and Footprint: *26*

Your Clues about your Real Life and OSINT: *26*

Your Browser and Device Fingerprints: *26*

Your Face, Voice, Biometrics and Pictures: *27*

Phishing and Social Engineering: *28*

Local Data Leaks and Forensics: *29*

No logging but logging anyway policies: *29*

Some Advanced targeted techniques: *30*

Notes: *31*

General Preparations: **32**

Picking your route: *32*

Timing limitations: 32
Budget/Material limitations: 32
Skills: 32
Adversaries (threats): 32

Steps for all routes: 33
Get an anonymous Phone number: 34
Get an USB key: 36
Find some safe places with decent public WIFI: 36

The TAILS route: 36
Plausible Deniability with the TAILS route: 37

Steps for all other routes: 38
Get a dedicated laptop for your sensitive activities: 38
Some laptop recommendations: 38
Bios/UEFI/Firmware Settings of your laptop: 39
Physically Tamper protect your laptop: 41

The Whonix route: 41
Picking your Host OS (the OS installed on your laptop): 41
Linux Host OS: 45
MacOS Host OS: 47
Windows Host OS: 48
Virtualbox on your Host OS: 56
Pick your connectivity method: 57
Get an anonymous VPN: 59
Whonix: 59
A note on Virtualbox Snapshots: 60
Download Virtualbox and Whonix utilities: 61
Virtualbox Hardening recommendations: 61

Tor over VPN: 62

Whonix Virtual Machines: 62

Pick your guest workstation Virtual Machine: 62
If you can use Tor: 62
If you cannot use Tor: 62
Linux Virtual Machine (Whonix or Linux): 62
Windows 10 Virtual Machine: 63
MacOS Virtual Machine: 65
KeePassXC: 66
VPN client installation (cash/Monero paid): 67
Final steps: 67

The Qubes Route: 67
Pick your connectivity method: 67
Get an anonymous VPN: 70
Installation: 70
Lid Closure Behavior: 70
Setup the VPN ProxyVM: 71
KeePassXC: 74

Creating your anonymous online identities: 75

Understanding the methods used to prevent anonymity and verify identity: 75
Captchas: 75
Phone verification: 76
E-Mail verification: 76
User details checking: 76
Proof of ID verification: 76
IP Filters: 77

Browser and Device Fingerprinting: 77
Human interaction: 78
User Moderation: 78
Behavioral Analysis: 78
Financial transactions: 78
Sign-in with some platform: 78
Live Face recognition and biometrics (again): 79
Manual reviews: 80

Getting Online: 80
Creating new identities: 81
Overview: 82
Maintenance tasks: 90

Backing-up your work securely: 90

Offline Backups: 90
Selected Files Backups: 90
Full Disk/System Backups: 91

Online Backups: 93

Covering your tracks: 94

Understanding HDD vs SSD: 94
Wear-Leveling. 95
Trim Operations: 95
Garbage Collection: 96
Conclusion: 96

How to securely wipe your whole Laptop/Drives if you want to erase everything: 97
Linux (all versions including Qubes OS): 98
Windows: 99
MacOS: 99

How to securely delete specific files/folders/data on your HDD/SDD and Thumb drives: 100
Windows: 100
Linux (non Qubes OS): 102
Linux (Qubes OS): 103
MacOS: 104

Some additional measures against forensics: 105
Removing Metadata from Files/Documents/Pictures: 106
TAILS: 107
Whonix: 107
MacOS: 107
Linux (Qubes OS): 109
Linux (non-Qubes): 109
Windows: 110

Communicating sensitive information to various organizations: 113

Removing some traces of your identities on search engines and various platforms: 114
Google: 115
Bing: 115
DuckDuckGo: 115
Yandex: 115
Qwant: 115
Yahoo Search: 115
Baidu: 115
Wikipedia: 116
Internet Archive: 116

Some low tech old school tricks: **116**

Hidden communications in plain sight: *116*

How to spot if someone has been searching your stuff: *117*

Some last OPSEC thoughts: **117**

If you think you got burned: **118**

If you have some time: *118*

If you have no time: *118*

A final editorial note: **118**

Appendix A: Windows Installation **119**

Installation: *119*

Privacy Settings: *119*

Appendix B: Windows Additional Privacy Settings **120**

Appendix C: Windows Installation Media Creation **121**

Appendix D: Using System Rescue to securely wipe an SSD drive. **121**

Appendix E: Clonezilla **122**

Appendix F: Diskpart **122**

Appendix G: Safe Browser **122**

If you can use Tor: *122*

If you cannot use Tor: *123*

Appendix H: Windows Cleaning Tools **123**

Appendix I: Using ShredOS to securely wipe an HDD drive: **124**

Windows: *124*

Linux: *124*

Appendix J: Manufacturer tools for Wiping HDD and SSD drives: **124**

Tools that provide a boot disk for wiping from boot: *124*

Tools that provide only support from running OS (for external drives). *125*

Appendix K: Considerations for using external SSD drives **125**

Windows: *125*

Trim Support: 125

ATA/NVMe Operations (Secure Erase/Sanitize): 125

Linux: *125*

Trim Support: 125

ATA/NVMe Operations (Secure Erase/Sanitize): 125

MacOS: *126*

Trim Support: 126

ATA/NVMe Operations (Secure Erase/Sanitize): 126

Appendix L: Creating a mat2-web guest VM for removing metadata from files **126**

Appendix M: BIOS/UEFI options to wipe disks in various Brands **128**

Appendix N: Warning about smartphones and smart devices. **128**

Appendix O: Get an anonymous VPN **128**

Cash-Paid VPN (preferred): *128*

Self-hosted VPN on a VPS paid with Monero (for skilled users familiar with Linux): *129*

Appendix P: Accessing the internet as safely as possible when Tor and VPNs are not an option **129**

Appendix Q: Using long range Antenna to connect to Public Wi-Fis from a safe distance: **130**

Appendix R: Installing a VPN on your VM or Host OS. **132**

Appendix S: Check your network for surveillance/censorship using OONI **133**

Appendix T: Checking files for malware **133**

Acknowledgements: **134**

Introduction:

TLDR for the whole guide: "A strange game. The only winning move is not to play."[1]

Making a social media account with a pseudonym or artist/brand name is easy. And it's enough is most use cases to protect your identity as the next George Orwell. There are plenty of people using pseudonyms all over Facebook/Instagram/Twitter/LinkedIn/TikTok/Snapchat/Reddit/… But the vast majority of those are anything but anonymous and can easily be traced to their real identity by your local cops, random people within the OSINT[2] (Open-Source Intelligence) community and trolls[3] on 4chan[4].

This is a good thing as most criminals/trolls are not really tech savvy and will be identified with ease. But this is also a bad thing as most political dissidents, human rights activists and whistleblowers can also be tracked rather easily.

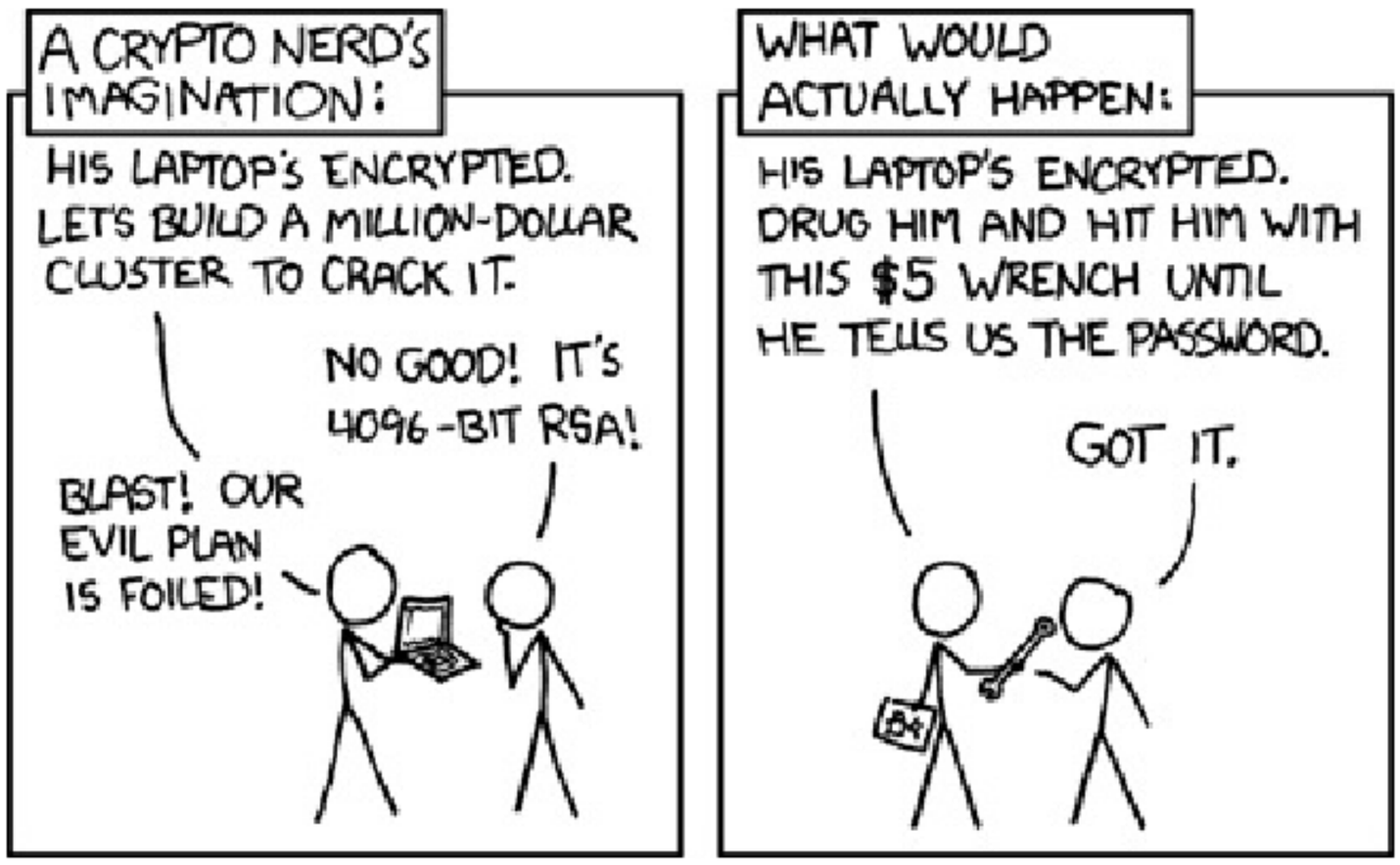
This updated guide aims to provide introduction to various de-anonymization techniques, tracking techniques, id verification techniques and optional guidance to creating and maintaining **reasonably** anonymous identities online including social media accounts safely.

It's important to understand that the purpose of this guide is anonymity and not just privacy but many of the guidance you'll find here will also help you improve your privacy even if you're not interested in anonymity. There is an important overlap in techniques and tools used for privacy and anonymity but they differ at some point:

- **Privacy is about people knowing who you are but not knowing what you are doing.**
- **Anonymity is about people knowing what you are doing but not knowing who you are.**[5]

Will this guide help you protect yourself from the NSA, the FSB, Mark Zuckerberg or the Mossad if they're out to find you? Probably not … Mossad will be doing "Mossad things" [6] and will probably find you no matter how hard you try to hide[7].

You have to consider your threat model[8] before going further.



(Illustration by xkcd.com, licensed under CC BY-NC 2.5)

Will this guide help you protect your privacy from OSINT researchers like Bellingcat[9] , Doxing[10] trolls on 4chan[11] and others that have no access to the NSA toolbox? More likely. Tho I wouldn't be so sure about 4chan.

It's also important to understand this guide is the humble result of years of experience, learning and testing from a single individual (myself) and that many of those systems that aim to prevent anonymity are opaque proprietary closed-source systems. Many of those guidelines are based on experience, on referenced studies and recommendations by other people and projects. These experiences take a lot of time, resources and are sometimes far from being scientific. There might be some wrong or outdated information in this guide too because I'm not omniscient (feel free to report any using GitHub Issues). **Your mileage may vary (a lot). Use at your own risk.**

You might think this guide has no legitimate use but there are many[12,13,14,15] such as:

- Evading Censorship
- Evading Oppression
- Evading Unlawful Government Surveillance
- Whistle Blowing
- Journalism
- Legal Practice
- Activism

This guide is written for those **good intended individuals** who might not be knowledgeable enough to consider the big picture of online anonymity and privacy.

This guide is not intended for:

- Creating machine accounts of any kind (bots).
- Creating impersonation accounts of existing people (such as identity theft).
- Helping malicious individuals conduct unlawful or unethical activities (such as trolling, disinformation or any criminal activity).
- Use by minors.

Feel free to report issues, recommend improvements or start a discussion on the GitHub repository if you want.

Use at your own risk. Anything in here is not legal advice and you should verify compliance with your local law before use (IANAL[16]). "Trust but verify"[17] all the information in here.

Requirements:

- **Be a permanent Adult resident in Germany where the courts have upheld up the legality of not using real names on online platforms (§13 VI of the German Telemedia Act of 2007[18]). Alternatively be an Adult resident of any other country where you can validate and verify the legality of this guide yourself.**
- This guide will assume you already have access to some personal (Windows/Linux/MacOS) laptop computer (ideally not a work/shared device).
- Don't be evil (for real this time)[19].
- Have patience as this process could take several weeks to finalize if you want to go through all the content.
- Have a little budget to dedicate to this process (you'll need at least budget for an USB key).
- Have some free time on your hands to dedicate to this process (or a lot depending on the route you pick).
- Be prepared to read a lot of references (do read them), guides (don't skip them) and follow a lot of how-to tutorials thoroughly (don't skip them either).

Understanding some basics of how some information can lead back to you and how to mitigate those:

There are many ways you can be tracked besides browser cookies and ads, your e-mail and your phone number. And if you think only the Mossad or the NSA/FSB can find you, you would be terribly wrong.

Here is a non-exhaustive list of some of the many ways you could be tracked and de-anonymized:

Your IP address:

Your IP address[20] is the most known and obvious way you can be tracked. That IP is the IP you're using at the source. This is where you connect to the internet. That IP is usually provided by your ISP (Internet Service Provider) (xDSL, Mobile, Cable, Fiber, Cafe, Bar, Friend, Neighbor). Most countries have data retention regulations[21] which mandates keeping logs of who is using what IP at a certain time/date for up to several years or indefinitely. Your ISP can tell a third party that you were using a specific IP at a specific date and time, years after the fact. If that IP (the origin one) leaks at any point for any reason, it can be used to track down you directly. In many countries, you won't be able to have internet access without providing some form of identification to the provider (address, ID, real name, e-mail …). Useless to say that most platforms (such as social networks) will also keep (sometimes indefinitely) the IP addresses you used to sign-up and sign-in to their services.

For those reasons, we'll need to not use that origin IP (the one tied to your identification) or hide it as much as we can through a combination of various means:

- Using a public WIFI service (free).
- Using the Tor Anonymity Network[22] (free).
- Using a VPN[23] service anonymously (anonymously paid).

All those will be explained later in this guide.

Your DNS requests:

DNS stands for "Domain Name System"[24] and is a service used by your browser (and other apps) to find the IP addresses of a service. It's pretty much a huge "contact list" (phone book for older people) that works like asking it a name and it returns the number to call. Except it returns an IP instead.

Every time your browser wants to access a certain service such as Google through https://www.google.com. Your Browser (Chrome or Firefox) will query a DNS service to find the IP addresses of the Google web servers.

Usually the DNS service is provided by your ISP and automatically configured by the network you're connecting to. This DNS service could also be subject to data retention regulations or will just keep logs for other reasons (data collection for advertising purposes for instance). Therefore this ISP will be capable of telling everything you did online just by looking at those logs which can in turn be provided to an adversary. Conveniently this also the easiest way for many adversaries to apply censoring or parental control by using DNS blocking[25]. The provided DNS servers

will give you a different address (than their real one) for some websites (like redirecting thepiratebay to some government website). Such blocking is widely applied worldwide for certain sites[26].

Using a private DNS service or your own DNS service would mitigate these issues but the other problem is that most of those DNS requests are by default still sent in clear text (unencrypted) over the network. Even if you browse PornHub in an incognito Window, using HTTPS and using a private DNS service, chances are very high that your browser will send a clear text unencrypted DNS request to some DNS servers asking basically "So what's the IP address of www.pornhub.com?".

Because it's not encrypted, your ISP and/or any other adversary could still intercept (using a Man-in-the-middle attack[57]) your request will know and possibly log what your IP was looking for. The same ISP can also tamper with the DNS responses even if you're using a private DNS. Rendering the use of a private DNS service useless.

As a bonus, many devices and apps will use hardcoded DNS servers bypassing any system setting you could set. This is for example the case with most (70%) Smart TVs and a large part (46%) of Game Consoles[27]. For these devices, you'll have to force them[28] to stop using their hardcoded DNS service which could make them stop working properly.

A solution to this is to use encrypted DNS using DoH (DNS over HTTPS[29]), DoT (DNS over TLS[30]) with a private DNS server (this can be self-hosted locally with a solution like pi-hole[31], remotely hosted with a solution like nextdns.io or using the solutions provider by your VPN provider or the Tor network). This should prevent your ISP or some middle-man from snooping on your requests … except it might not.

Unfortunately the TLS protocol used in most HTTPS connections in most Browsers (Chrome/Brave among them) will leak the DNS again through SNI[32] handshakes (this can be checked here at Cloudflare: https://www.cloudflare.com/ssl/encrypted-sni/). **As of the writing of this guide, only Firefox based browsers supports eSNI (encrypted SNI[33] soon to be renamed ECH[34] and soon to be supported by Firefox too[35]) which will encrypt everything end to end (in addition to using a secure private DNS over TLS/HTTPS) and will allow you to actually hide your DNS requests from a third party[36].** And this option is not enabled by default either so you'll have to enable it yourself:

In addition to limited browser support, only Web Services and CDNs[37] behind Cloudflare CDN support eSNI at this stage (and will support ECH in the future[38]). This means that eSNI (or soon ECH) is not supported (as of January 2021) by many mainstream platforms such as:

- Amazon (including AWS, Twitch…)
- Microsoft (including Azure, OneDrive, Outlook, Office 365…)
- Google (including Gmail, Google Cloud…)
- Apple (including iCloud, iMessage…)
- Reddit
- YouTube
- Facebook
- Instagram
- Twitter
- GitHub

Some countries like Russia[39] and China[40] will block eSNI handshakes at network level to allow snooping and prevent bypassing censorship. Meaning you won't be able to establish an HTTPS connection with a service if you don't allow them to see what it was.

Finally, even if you use a custom encrypted DNS server (DoS or DoT) with eSNI support, it might still not be enough as traffic analysis studies[41] have shown it's still possible to reliably fingerprint and block unwanted requests. Only DNS over Tor was able to demonstrate efficient DNS Privacy in recent studies but even that can still be defeated by other means (see Your "Anonymized" Tor/VPN traffic:).

One could also decide to use a Tor Hidden DNS Service or ODoH (Oblivious DNS over HTTPS[42]) to further increase privacy/anonymity but unfortunately, as far as I know, these services are only provided by Cloudflare as of this writing (https://blog.cloudflare.com/welcome-hidden-resolver/, https://blog.cloudflare.com/oblivious-dns/). I **personally** think these are viable and reasonably secure technical options but there is also a moral choice if you want to use them (despite the risk posed by some researchers[43]).

Lastly, there is also this new option called DoHoT which stands for DNS over HTTPS over Tor which could also further increase your privacy/anonymity and which you could consider if you are more skilled with Linux. See https://github.com/alecmuffett/dohot. This guide won't help you with this one at this stage but it might be coming soon.

Here is an illustration showing the current state of DNS privacy based on my current knowledge.

State of DNS and Web privacy

Disclaimer: This illustration only lists the major Browsers, Operating Systems and CDN networks. Browser support means native support without use of additional software/extension. OS Native Support means native support without additional software/packages.

Default DNS
Browser Support:
- Firefox
- Chrome
- Safari
- Edge
- Brave

OS Native Support:
- Windows
- MacOS
- Linux (all)
- iOS
- Android

Unencrypted DNS requests to ISP DNS
Unencrypted SNI request to Service
HTTPS connections to services
DNS Traffic Fingerprinting
Web Traffic Fingerprinting

Custom DNS
Browser Support:
- Firefox
- Chrome
- Safari
- Edge
- Brave

OS Native Support:
- Windows
- MacOS
- Linux (all)
- iOS
- Android

Unencrypted DNS Requests to custom DNS
Unencrypted SNI request to Service
HTTPS connections to services
DNS Traffic Fingerprinting
Web Traffic Fingerprinting

Custom Encrypted DNS (DoH, DoT)
Browser Support:
- Firefox
- Chrome
- Safari
- Edge
- Brave

OS Native Support:
- Linux (all)
- iOS
- Android

Encrypted DNS request to Custom DNS
Unencrypted SNI request to Service
HTTPS connections to services
DNS Traffic Fingerprinting
Web Traffic Fingerprinting

Custom Encrypted DNS (DoH, DoT) with eSNI**
Browser Support:
- Firefox

OS Native Support:
- None

eSNI CDN Support:
- Cloudflare

Encrypted DNS request to Custom DNS
Encrypted SNI requests
HTTPS connections to services
DNS Traffic Fingerprinting
Web Traffic Fingerprinting

Custom Encrypted DNS (ODoH*) with eSNI****
Browser Support:
- None

OS Native Support:
- None

eSNI CDN Support:
- Cloudflare

ODoH DNS Support:
- Cloudflare

Encrypted DNS request to Custom DNS
Encrypted SNI requests
HTTPS connections to services
DNS Traffic Fingerprinting
Web Traffic Fingerprinting

DNS Over Tor
Browser Support:
- Tor Browser

OS Native Support:
- TAILS
- Whonix
- QubesOS

DNS Over Tor DNS Support:
- Cloudflare (hidden resolver)

Encrypted DNS request to ISP DNS*
Encrypted SNI request to Service*
HTTPS connections to services
DNS Traffic Fingerprinting
Web Traffic Fingerprinting at source

DNS Over Tor Over HTTPS (DoHoT)**
- Any Linux

Encrypted DNS request to ISP DNS
Encrypted SNI request to Service*
HTTPS connections to services
DNS Traffic Fingerprinting
Web Traffic Fingerprinting at source

*This an indirect encryption because of the overlaying Tor Onion Router Protocol. The encryption is not done within the DNS/TLS protocol which is done by the Exit Node (encrypted or unencrypted) but the Tor protocol is hiding the origin of the requests which has the same result for an adversary.

**eSNI is soon to be deprecated and replaced by ECH and should be adopted by Firefox and Cloudflare. eSNI is currently only suported on Firefox and services behind Cloudflare CDN.

***ODoH at this time is only provided by CloudFlare.

****DoHoT is quite a new approach and not natively implemented anywhere AFAIK and could IMHO be the actual best option for DNS privacy.

Therefore to mitigate all these issues (as much as possible), this guide we will later recommend a virtualized multi-layered solution of VPN over Tor which should mitigate most of the issues reasonably. Other options will also be explained (Tor over VPN, Tor only, VPN only, No Tor/VPN) but are less recommended if possible.

As for your normal daily use (non-sensitive), remember that only Firefox based browsers support eSNI (soon ECH) so far and that it's only useful with websites hosted behind Cloudflare CDN at this stage. If you prefer a Chrome based version (which is understandable for some due to some better integrated features like on-the-fly Translation), then I

would recommend the use of Brave instead which supports all Chrome extensions and offers much better privacy than Chrome.

Your IMEI and IMSI (and by extension, your phone number):

The IMEI (International Mobile Equipment Identity[44]) and the IMSI (International Mobile Subscriber Identity[45]) are unique numbers created by mobile phone manufacturers and mobile phone operators.

The IMEI is tied directly to the phone you're using. This number is known and tracked by the mobile phone operators and known by the manufacturers. Every time your phone connects to the mobile network, it will register the IMEI on the network along the IMSI (if a SIM card is inserted but that's not even needed). It's also used by many applications (Banking apps abusing the phone permission on Android for instance[46]) and smartphone Operating Systems (Android/IOS) for identification of the device[47]. It is possible but difficult (and not illegal in many jurisdiction[48]) to change the IMEI on a phone but it's probably easier and cheaper to just find and buy some old (working) Burner phone for a few Euros (this guide is for Germany remember) at a flea market or at some random small shop.

The IMSI is tied directly to the mobile subscription or pre-paid plan you're using and is basically tied to your phone number by your mobile provider. The IMSI is hardcoded directly on the SIM card and cannot be changed. Remember that every time your phone connects to the mobile network, it will also register the IMSI on the network along the IMEI. Like the IMEI, the IMSI is also being used by some applications and smartphone Operating systems for identification and are being tracked. Some countries in the EU for instance maintain a database of IMEI/IMSI associations for easy querying by Law Enforcement.

Today, giving away your (real) phone number is basically the same or better than giving away your Social Security number/Passport ID/National ID.

The IMEI and IMSI can be traced back to you by at least 6 ways:

- The mobile operator subscriber logs which will usually store the IMEI along the IMSI and their subscriber information database. If you use a prepaid anonymous SIM (anonymous IMSI but with a known IMEI), they can see this cell belongs to you if you used that cell phone before with a different SIM card (different anonymous IMSI but same known IMEI).
- The mobile operator antenna logs which will conveniently keep a log of which IMEI and IMSI also keep some connection data. They know and log for instance that a phone with this IMEI/IMSI combination connected to a set of Mobile antennas and how powerful the signal to each of those antennas was allowing easy triangulation/geolocation of the signal. They also know which other phones (your real one for instance) connected at the same time to the same antennas with the same signal which would make it possible to know precisely that this "burner phone" was always connected at the same place/time than this other "known phone" which shows up every time the burner phone is being used. This information can be used by various third parties to geolocate/track you quite precisely[49,50].
- The manufacturer of the Phone can trace back the sale of the phone using the IMEI if that phone was bought in a non-anonymous way. Indeed they will have logs of each phone sale (including serial number and IMEI), to which shop/person it was sold to. And if you're using a phone that you bought online (or from someone that knows you). It can be traced to you using that information. Even if they don't find you on CCTV[51] and you bought the phone cash, they can still find what other phone (your real one in your pocket) was there (in that shop) at that time/date by using the antenna logs.
- The IMSI alone can be used to find you as well because most countries now require customers to provide an ID when buying a SIM card (subscription or pre-paid). The IMSI is then tied to the identity of the buyer of the card. In the countries where the SIM can still be bought with cash (like the UK), they still know where (which shop) it was bought and when. This information can then be used to retrieve information from the shop itself (such as CCTV footage as for the IMEI case). Or again the antenna logs can also be used to figure out which other phone was there at the moment of the sale.
- The smartphone OS makers (Google/Apple for Android/IOs) also keep logs of IMEI/IMSI identifications tied to Google/Apple accounts and which user has been using them. They too can trace back the history of the phone and to which accounts it was tied in the past[52].

- Government agencies around the world interested in your phone number can and do use[53] special devices called "IMSI catchers"[54] like the Stingray[55] or more recently the Nyxcell[56]. These devices are able to impersonate (to spoof) a cell phone Antenna and force a specific IMSI (your phone) to connect to it to access the cell network. Once they do, they will be able to use various MITM[57] (Man-In-The-Middle Attacks) that will allow them to:
    - Tap your phone (voice calls and SMS).
    - Sniff and examine your data traffic.
    - Impersonate your phone number without controlling your phone.
    - …

Here is also a good YouTube video on this topic: DEFCON Safe Mode - Cooper Quintin - Detecting Fake 4G Base Stations in Real Time https://www.youtube.com/watch?v=siCk4pGGcqA

For these reasons, it's crucial to get dedicated an anonymous phone number and/or an anonymous burner phone with an anonymous pre-paid sim card that are not tied to you in any way (past or present) for conducting sensitive activities (See more practical guidance in Get an anonymous Phone number: section).

While there are some smartphones manufacturers like Purism with their Librem series[58] who have your privacy in mind, they still do not allow IMEI randomization which I believe is a key anti-tracking feature that should be provided by such manufacturers. While this measure won't prevent IMSI tracking within the SIM card, it would at least allow you to keep the same "burner phone" and only switch SIM cards instead of having to switch both for privacy.

See Appendix N: Warning about smartphones and smart devices

Your Wi-Fi or Ethernet MAC address:

The MAC address[59] is a unique identifier tied to your physical Network Interface (Wired Ethernet or WIFI) and could of course be used to track you if it's not randomized. As it was the case with the IMEI, manufacturers of computers and network cards usually keep logs of their sales (usually including things like: Serial number, IMEI, Mac Addresses, …) and it's possible again for them to track where and when the computer with the MAC address in question was sold and to whom. Even if you bought it with cash in a supermarket, the supermarket might still have CCTV (or a CCTV just outside that shop) and again the time/date of sale could be used to find out who was there using the Mobile Provider antenna logs at that time (IMEI/IMSI).

Operating Systems makers (Google/Microsoft/Apple) will also keep logs of devices and their MAC addresses in their logs for device identification (Find my device type services for example). Apple can tell that the MacBook with this specific MAC address was tied to a specific Apple Account before. Maybe yours before you decided to use the MacBook for sensitive activities. Maybe to a different user who sold it to you but remembers your e-mail/number from when the sale happened.

Your home router/WIFI access point keeps logs of devices that registered on the Wi-Fi and these can be accessed too to find out who's been using your WIFI. Sometimes this can be done remotely (and silently) by the ISP depending if that router/Wi-Fi access point is being "managed" remotely by the ISP (which is often the case when they provide the router to their customers).

So it's important again not to bring your phone along when/where you conduct sensitive activities. If you use your own laptop, then it's crucial to hide that MAC address (and Bluetooth address) anywhere you use it and be extra careful not to leak any information. Thankfully many recent OSes now feature or allow the option to randomize MAC addresses (Android, IOS, Linux and Windows 10) with the notable exception of MacOS which doesn't support this feature even in its latest Big Sur version.

See Appendix N: Warning about smartphones and smart devices

Your Bluetooth MAC address:

Your Bluetooth MAC is like a MAC address except it's for Bluetooth. Again it can be used to track you as manufacturers and operating system makers keep logs of such information. It could be tied to a sale place/time/date or accounts and then could be used to track you with such information, the shop billing information, the CCTV or the mobile antenna logs in correlation.

Operating systems have protections in place to randomize those addresses but are still subject to vulnerabilities[60].

For this reason, and unless you really need those, you should just disable Bluetooth completely in the BIOS/UEFI settings if possible or in the Operating System otherwise.

On Windows, you will need to disable the Bluetooth device in the device manager itself to force a randomization of the address for next use and prevent tracking.

See Appendix N: Warning about smartphones and smart devices

Your Operating Systems and Apps telemetry services:

Whether it's Android, IOS, Windows, MacOS or even Ubuntu. Most popular Operating Systems now collect telemetry information by default. Some like Windows will not even allow disabling telemetry completely without some technical tweaks. This information collection can be extensive and include a staggering amount of details (metadata and data) on your devices and their usage.

Here are good overviews of what is being collected by those 5 popular OSes in their last versions:

- Android/Google:
  - Just have a read at their privacy policy https://policies.google.com/privacy
- IOS/Apple:
  - More information at https://www.apple.com/legal/privacy/en-ww/ and https://support.apple.com/en-us/HT202100
  - Apple does claim[61] that they anonymize this data using differential privacy[62] but you'll have to trust them on that.
- Windows/Microsoft:
  - Full list of required diagnostic data: https://docs.microsoft.com/en-us/windows/privacy/required-windows-diagnostic-data-events-and-fields-2004
  - Full list of optional diagnostic data: https://docs.microsoft.com/en-us/windows/privacy/windows-diagnostic-data
- MacOS:
  - More details on https://support.apple.com/guide/mac-help/share-analytics-information-mac-apple-mh27990/mac
- Ubuntu:
  - Ubuntu despite being a Linux distribution also collects Telemetry Data nowadays. This data however is quite limited compared to the others. More details on https://ubuntu.com/desktop/statistics

Not only are Operating Systems gathering telemetry services but so are Apps themselves like Browsers, Mail Clients, and Social Networking Apps installed on your system.

It's important to understand that this telemetry data can be tied to your device and also help de-anonymizing you and subsequently can be used against you by an adversary that would get access to this data.

Later in this guide, we will use all the means at our disposal to disable and block as much telemetry as possible to mitigate this attack vector in the Operating Systems supported in this guide.

See Appendix N: Warning about smartphones and smart devices

The WIFIs and Bluetooth devices around you:

Geolocation is not only done by using mobile antennas triangulation. It's also done using the WIFIs and Bluetooth devices around you. Operating systems makers like Google (Android[63]) and Apple (IOS[64]) maintain a convenient database of most WIFI access points, Bluetooth devices and their location. When your Android smartphone or IPhone is on (and not in Plane mode), it will scan passively (unless you specifically disable this feature in the settings) WIFI access points and Bluetooth devices around you and will be able to geolocate you with more precision than when using a GPS.

This allows them to provide accurate locations even when GPS is off but it also allows them to keep a convenient record of all Bluetooth devices all over the world. Which can then be accessed by them or third parties for tracking.

Note: If you have an Android smartphone, Google probably knows where it is no matter what you do. You can't really trust the settings. The whole operating system is built by a company that wants your data. Remember that if it's free then you are the product.

But that's not all those WiFis access points can do. Recently developed techs could even allow someone to track your movements accurately just based on radio interferences. What this means is that it's possible to track your movement inside a room/building based on the radio signals passing through. This might seem like a tinfoil hat conspiracy theory claim but here are the references[65] with demonstrations showing this tech in action: http://rfpose.csail.mit.edu/ and the video here: https://www.youtube.com/watch?v=HgDdaMy8KNE

You could therefore imagine many uses cases for such technologies like recording who enters specific buildings/offices (hotels, hospitals or embassies for instance) and then discover who meets who and where by tracking them from outside. Even if they have no smartphone on them.

Again such issue could only be mitigated by being in room/building that would act as a faraday cage.

See Appendix N: Warning about smartphones and smart devices

Your Metadata including your Geo-Location:

Your metadata is all the information about your activities without the actual content of those activities. For instance it's like knowing you had a call from an oncologist before then calling your family and friends successively. You don't know what was said during the conversation but you can guess what it was just from the "metadata"[66].

This metadata will also often include your location that is being harvested by Smartphones, Operating Systems (Android[67]/IOS), Browsers, Apps, Websites. Odds are there are several companies knowing exactly where you are at any time[68] because of your smartphone[69].

This location data has been used in many judicial cases[70] already as part of "geofence warrants" [71] that allows law enforcement to ask companies (such as Google/Apple) a list of all devices present at a certain location at a certain time.

Now let's say you're using a VPN to hide your IP. The social media platform knows you were active on that account on November 4th from 8am to 1pm with that VPN IP. The VPN allegedly keeps no logs and can't trace back that VPN IP to your IP. Your ISP however knows (or at least has the ability to know) you were connected to that same VPN provider on November 4th from 7:30am to 2pm but doesn't know what you were doing with it.

The question is: Is there someone somewhere that would possibly have both pieces of information available[72] for correlation in a convenient database?

Have you heard of Edward Snowden[73]? Now is the time to google him and read his book[74]. I recommend reading about XKEYSCORE[75,76], MUSCULAR[77] and PRISM[78].

"We kill people based on Metadata"[79]

See Appendix N: Warning about smartphones and smart devices

Your Smart devices in general:

You got it, your smartphone is an advanced spying/tracking device that:

- Records everything you say at any time ("Hey Siri", "Hey Google").
- Records your location everywhere you go.
- Records other devices around you at all times (Bluetooth devices, Wi-Fi Access points).
- Records your habits and health data (steps, screen time, exposure to diseases, connected devices data)
- Records all your network locations.
- Records all your pictures and videos (and most likely where they were taken).
- Has most likely access to most of your known accounts including Social Media, Messaging and Financial accounts.

All of this data is very likely being transmitted, processed and stored indefinitely (unencrypted[80]) by various third parties[81].

But that's not all, this section is not called "Smartphones" but "Smart devices" because it's not only your smartphone spying on you. It's also every other smart device you could have.

- Your Smart Watch? (Apple Watch, Android Smartwatch …).
- Your Fitness Devices and Apps? (Strava[82,83], Fitbit[84], Garmin, Polar[85], …)[86]
- Your Smart Speaker? (Amazon Alexa[87], Google Echo, Apple Homepod …).
- Your Smart Transportation? (Car? Scooter?)
- Any other Smart device? There is even a convenient search engine dedicated to finding them: https://www.shodan.io/

See Appendix N: Warning about smartphones and smart devices

Your Devices can be tracked even when offline:

You've seen this in action/spy/Sci-Fi movies and shows, the protagonists always remove the battery of their phones to make sure it can't be used. Well this is now true at least for some devices:

- IPhones and IPads (IOS 13 and above)[88,89].
- Samsung Phones (Android 10 and above)[90].
- MacBooks (MacOS 10.15 and above)[91].

Such devices will continue to broadcast identity information to nearby devices even when offline using Bluetooth Low-Energy[92]. They don't have access to the devices directly (which are not connected to the internet) but instead use BLE to find them through other nearby devices[93].

They can now locate such devices and keep the location in some database that could then be used by third parties or themselves for various purposes.

See Appendix N: Warning about smartphones and smart devices

Your RFID enabled devices:

RFID stands for Radio-frequency identification[94], it's the technology used for instance for contactless payments and various identification systems. Of course your smartphone is among those devices and has RFID contactless payment capabilities through NFC[95]. As with everything else, such capabilities can be used for tracking by various actors.

But unfortunately this is not limited your smartphone and you also probably carry some amount of RFID enabled device with you all the time such as:

- Your contactless enabled credit/debit cards
- Your store loyalty cards
- Your transportation payment cards
- Your work-related access cards
- Your car keys
- Your national ID or driver license
- Your passport
- The price/anti-theft tags on object/clothing
- …

While all these cannot be used to de-anonymize you from a remote online adversary, they can be used to narrow down a search if your approximate location at a certain time is known. For instance, you can't rule out that some stores will effectively scan (and log) all RFID chips passing through the door. They might be looking for their loyalty cards but are also logging others along the way. Such RFID tags could be traced to your identity and allow for de-anonymization.

More information over at Wikipedia: https://en.wikipedia.org/wiki/Radio-frequency_identification#Security_concerns and https://en.wikipedia.org/wiki/Radio-frequency_identification#Privacy

The only way to mitigate this problem is to have no RFID tags on you or to shield them again using a type of faraday cage. You could also use specialized wallets/pouches that specifically block RFID communications. Many of those are now made by well-known brands such as Samsonite[96].

See Appendix N: Warning about smartphones and smart devices

Your Files Properties/Metadata:

This can be obvious to many but not to all. Most files have metadata attached to them. A good example are pictures which store EXIF[97] information which can contain a lot of information such as GPS coordinates, which camera/phone model took it and when it was taken precisely. While this information might not directly give out who you are, it could tell exactly where you were at a certain moment which could allow others to use different sources to find you (CCTV or other footage taken at the same place at the same time during a protest for instance). It's important that you verify any file you would put on those platforms for any properties that might contain any information that might lead back to you.

Here is an example of EXIF data that could be on a picture:

Global Positioning System	
GPS Altitude	31.9 m
GPS Latitude	6deg 14' 7.620"
GPS Longitude	106deg 49' 30.210"
Image Information	
Date and Time	2018:08:24 15:47:27
Manufacturer	Apple
Model	iPhone 6s
Photograph Information	
Aperture	F2.2
Exposure Bias	0 EV
Exposure Mode	Auto
Exposure Program	Auto
Exposure Time	1/874 s
Flash	No, auto
FNumber	F2.2
Focal Length	4.2 mm
ISO Speed Ratings	25
Metering Mode	Multi-segment
Shutter speed	1/874 s
White Balance	Auto

For this reason you'll always have to be very careful when uploading files using your anonymous identities and check the metadata of those files.

Even if you publish a simple text file, you should always double or triple check it for any information leakage before publishing. You'll find some guidance about this in the Some additional measures against forensics: section at the end of the guide.

Malware hidden in your files/documents:

Using steganography or other techniques, it's fairly easy to embed malware into common file formats such as Office Documents, Pictures, Videos, PDF documents…

These could be exploiting a vulnerability in an outdated format or outdated reader. Such exploits could then be used to compromise your system.

See these good videos for more explanations on the matter:

- What is a File Format?: https://www.youtube.com/watch?v=VVdmmN0su6E
- Ange Albertini: Funky File Formats: https://www.youtube.com/watch?v=hdCs6bPM4is

If you want to learn how to remove these, see Appendix T: Checking files for malware

Watermarking:

Pictures/Videos/Audio:

Pictures/Videos often contain visible watermarks indicating who's the owner/creator but there are also invisible watermarks in various products aiming at identifying the viewer itself.

So if you're a whistleblower and thinking about leaking some picture/audio/video file. Think twice. There are chances that those might contain invisible watermarking within them that would include information about you as a

viewer. Such watermarks can be enabled with a simple switch in like Zoom (Video[98] or Audio[99]) or with extensions[100] for popular apps such as Adobe Premiere Pro. These can be inserted by various content management systems.

For a recent example where someone leaking a Zoom meeting recording was caught because it was watermarked: https://theintercept.com/2021/01/18/leak-zoom-meeting/

Such watermarks can be inserted by various products[101,102,103,104] using Steganography[105] and can resist compression[106] and re-encoding[107,108].

These watermarks are not easily detectable and could allow identification of the source despite all efforts.

In addition to watermarks, the camera used for filming (and therefore the device used for filming) a video can also be identified using various techniques such as lens identification[109] which could lead to de-anonymization.

Be extremely careful when publishing videos/pictures/audio files from known commercial platforms as they might contain such invisible watermarks in addition to details in the images themselves.

Printing Watermarking:

Did you know your printer is most likely spying on you too? Even if it's not connected to any network? This is usually a known fact by many people in the IT community but few outside people.

Yes … Your printers can be used to de-anonymize you as well as explained by the EFF here https://www.eff.org/issues/printers

With this (old but still relevant) video explaining how from the EFF as well: https://www.youtube.com/watch?v=izMGMsIZK4U

Basically many printers will print an invisible watermark allowing for identification of the printer on every printed page. There is no real way to mitigate this but to inform yourself on your printer and make sure it doesn't print any invisible watermark. This is obviously important if you intend to print anonymously.

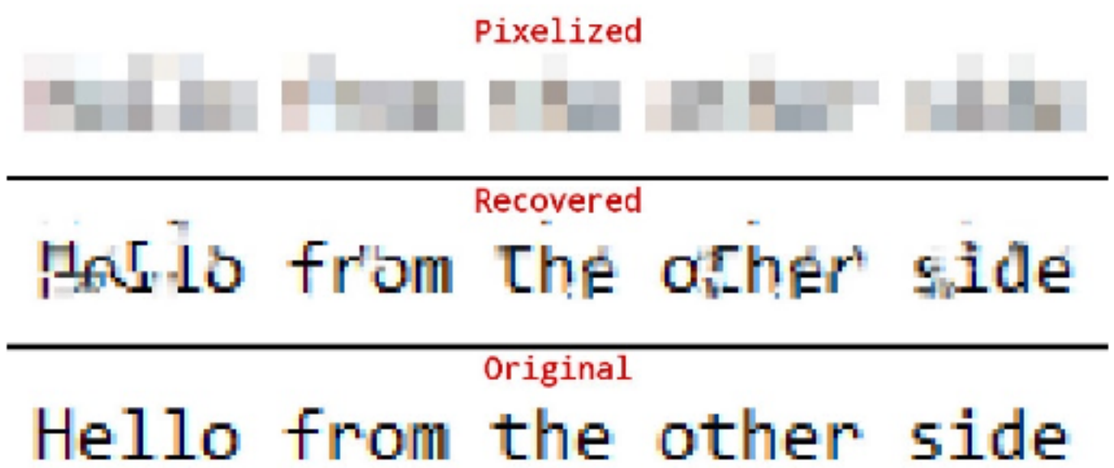
Here is an (old but still relevant) list of printers and brands who do not print such tracking dots provided by the EFF https://www.eff.org/pages/list-printers-which-do-or-do-not-display-tracking-dots

Your Pixelized/Blurred Information:

Did you ever see a document with blurred text? Did you ever make fun of those movies/series where they "enhance" an image to recover seemingly impossible to read information.

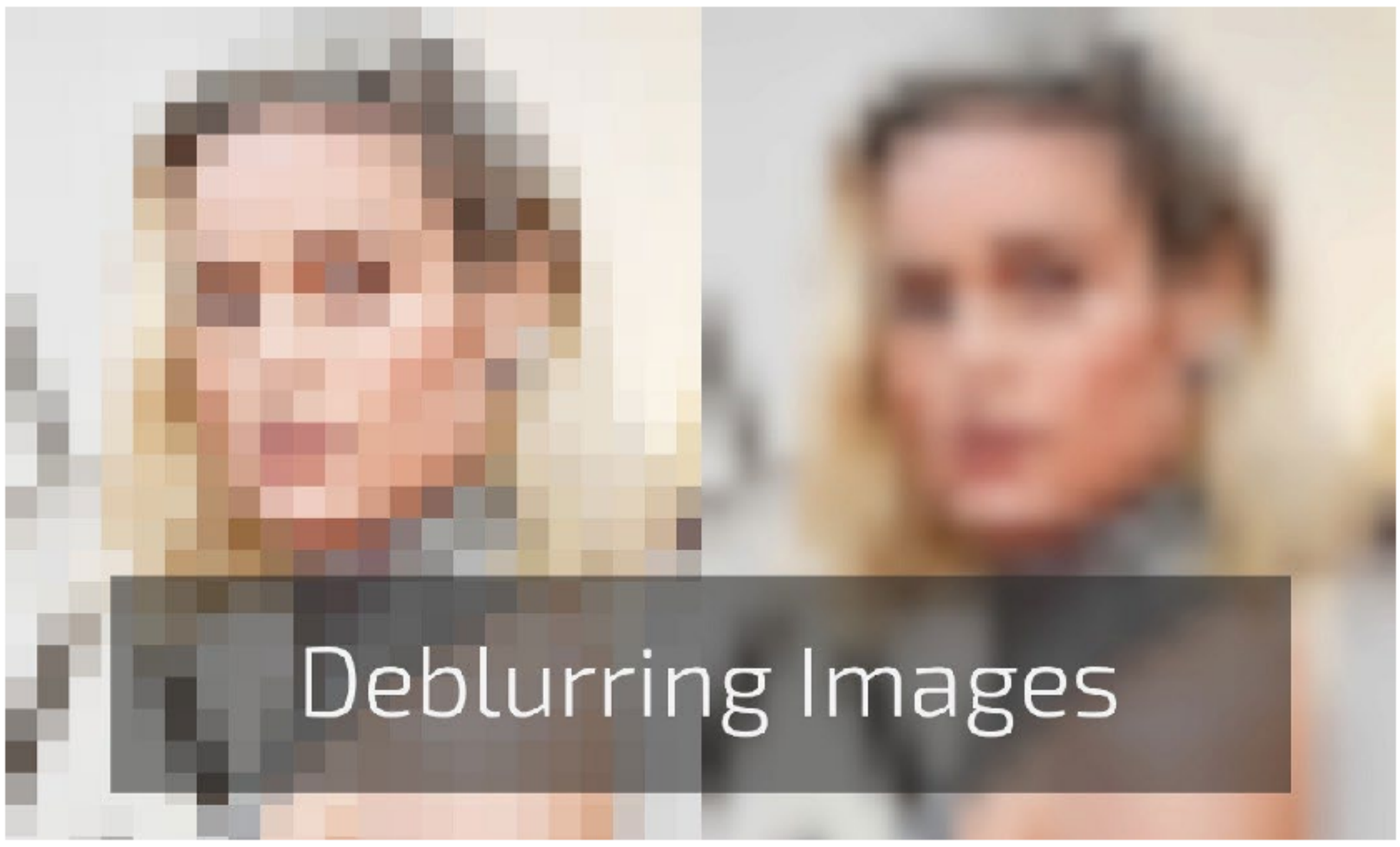
Well there are actually techniques for recovering information from such documents, videos and pictures.

Here is for example an open-source project you could use yourself for recovering text from some blurred images yourself: https://github.com/beurtschipper/Depix



This is of course an open-source project available for all to use. But you can probably imagine that such techniques have probably been used before by other adversaries. These could be used to reveal blurred information from published documents that could then be used to de-anonymize you.

There are also tutorials for using such techniques using Photo Editing tools such as GIMP such as: https://medium.com/@somdevsangwan/unblurring-images-for-osint-and-more-part-1-5ee36db6a70b followed by https://medium.com/@somdevsangwan/deblurring-images-for-osint-part-2-ba564af8eb5d



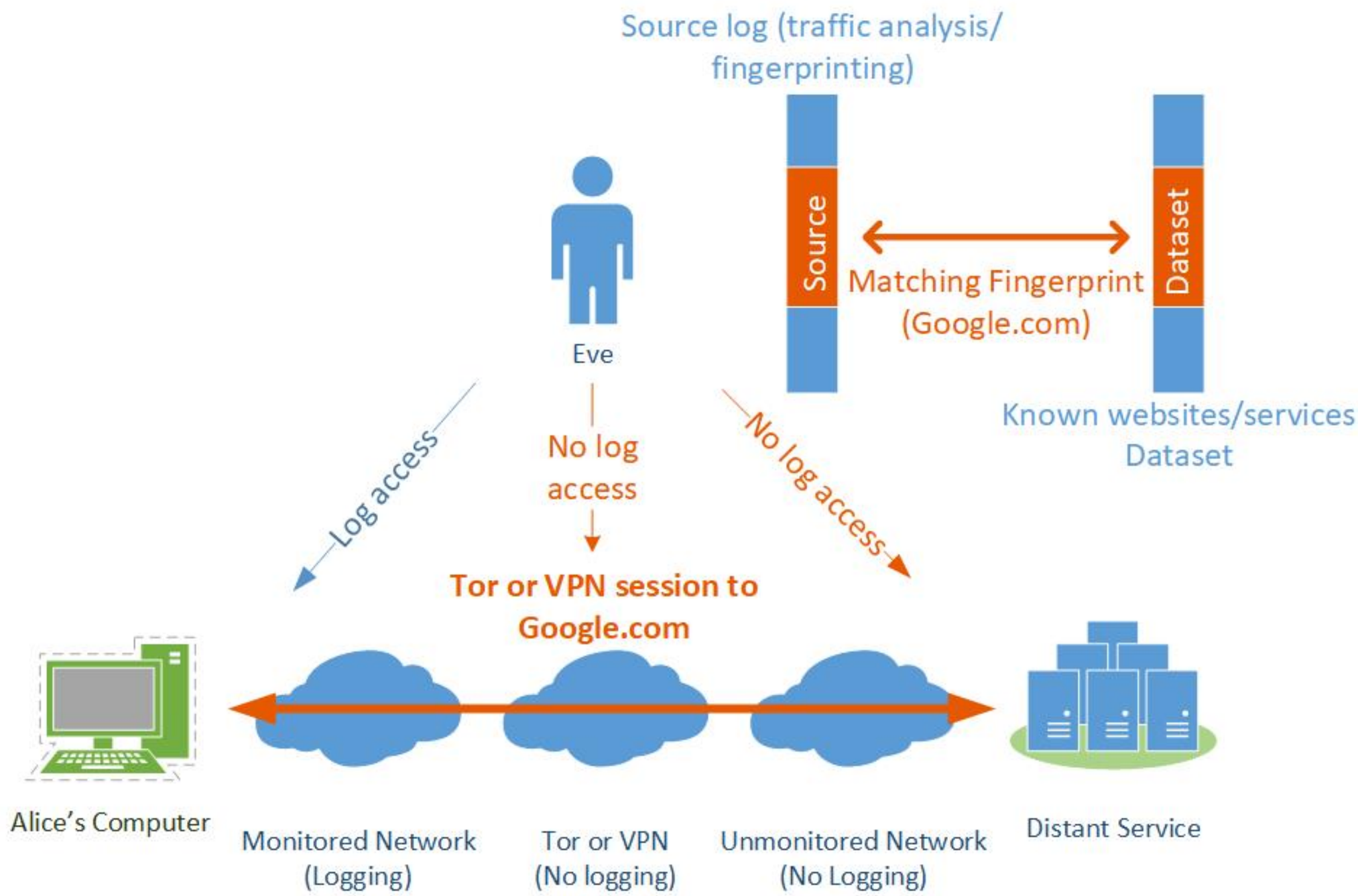
Last but not least you'll find plenty of deblurring resources here: https://github.com/subeeshvasu/Awesome-Deblurring

For this reason it's always extremely important that you correctly redact and curate any document you might want to publish. Blurring is not enough and you should always completely blacken/remove any sensitive data to avoid any attempt at recovering data from any adversary.

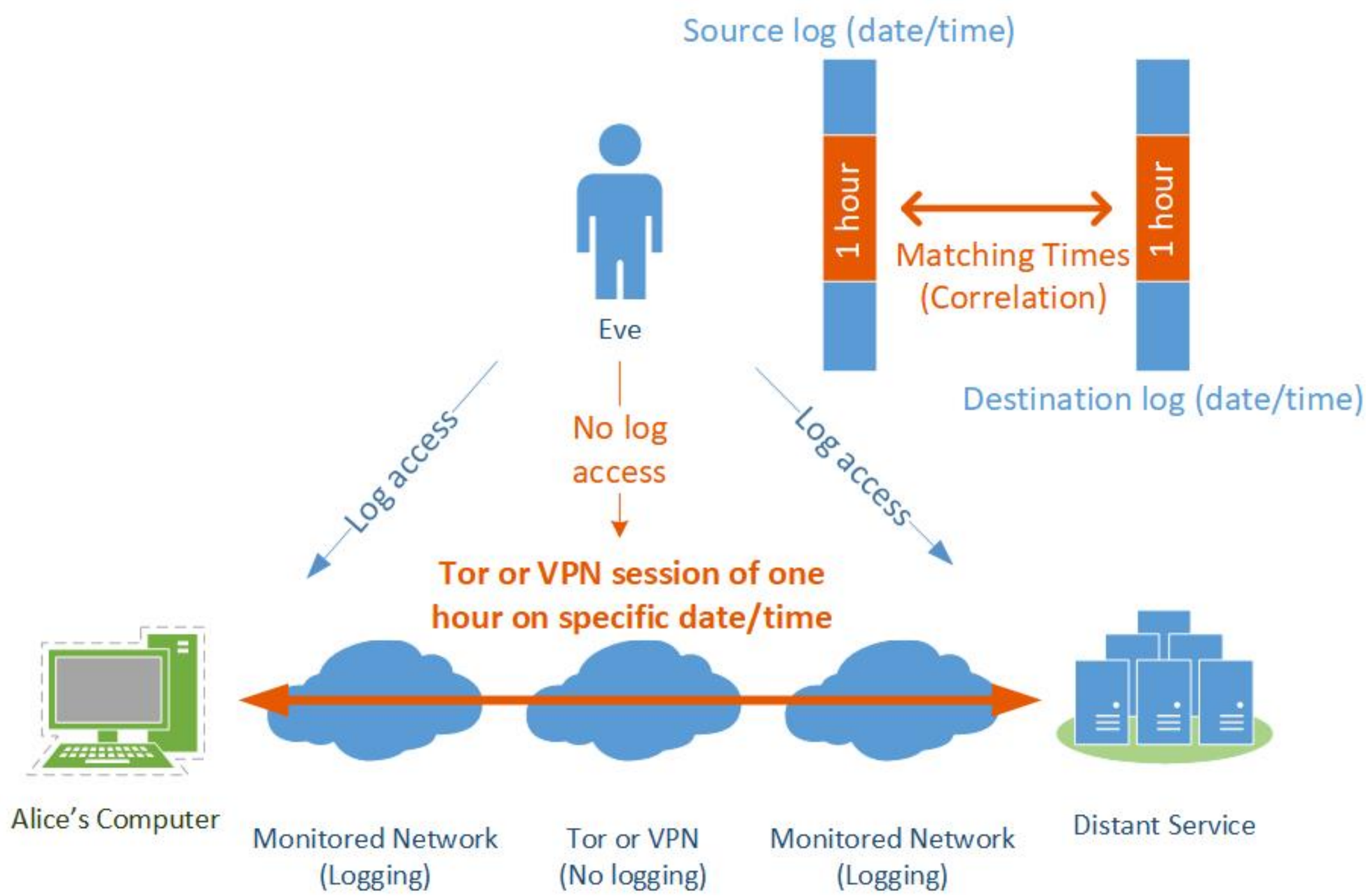
Your "Anonymized" Tor/VPN traffic:

Tor and VPNs are not silver bullets. Many advanced techniques have been developed and studied to de-anonymize encrypted traffic over the years[110]. Most of those techniques are Correlation attacks that will correlate your network traffic in one way or another to logs or datasets. Here are some classic examples:

- Correlation Fingerprinting Attack: As illustrated (simplified) below, this attack will fingerprint[111] your encrypted traffic (like the websites you visited) just based on the analysis of your encrypted traffic (without decrypting it). It's able to do so with a whopping 96% success rate. Such fingerprinting can be used by an adversary that has access to your source network to figure out some of your encrypted activity (such as which websites you visited).

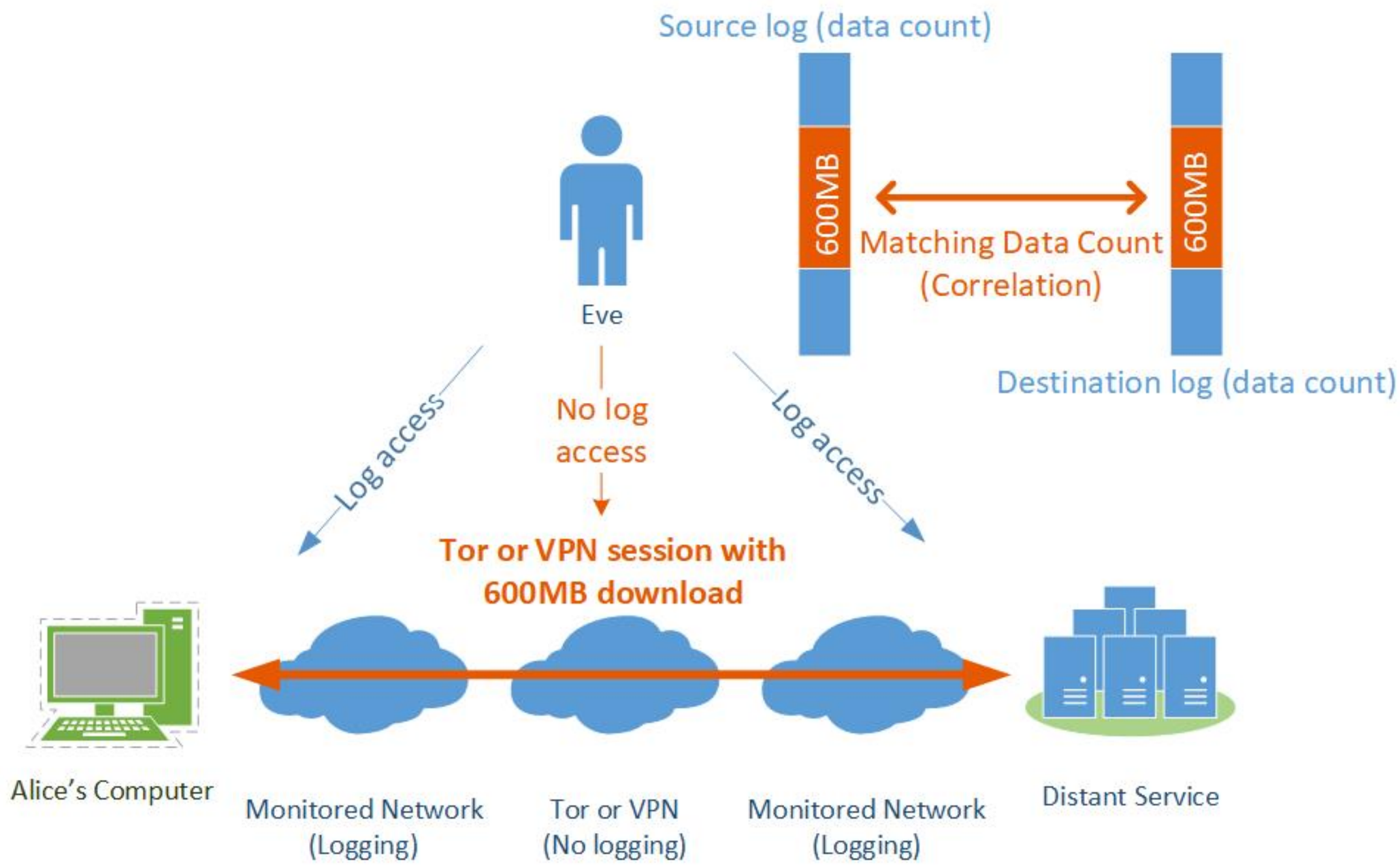


- Correlation Timing Attacks: As illustrated (simplified) below, an adversary that has access to network connection logs (IP or DNS for instance, remember that most VPN servers and most Tor nodes are known and publicly listed) at the source and at the destination could correlate the timings to de-anonymize you without requiring any access to the Tor or VPN network in between. A real use case of this technique was done by the FBI in 2013 to de-anonymize[112] a bomb threat hoax at Harvard University.



- Correlation Counting Attacks: As illustrated (simplified) below, an adversary that has no access to detailed connection logs (can't see that you used Tor or Netflix) but has access to data counting logs could see that

you have downloaded 600MB on a specific time/date that matches the 600MB upload at the destination. This correlation can then be used to de-anonymize you over time.



There are ways to mitigate these such as:

- Do not use Tor/VPNs to access services that are on the same network (ISP) as the destination service. For example do not connect to Tor from your University Network to access a University Service anonymously. Instead use a different source point (such as a public Wi-Fi) that cannot be correlated easily by an adversary.
- Do not use Tor/VPN from an obviously monitored network (such as a corporate/governmental Network) but instead try to find an unmonitored network such as a public Wi-Fi or a residential Wi-Fi.
- Use multiple layers (such as what will be recommended in this guide later: VPN over Tor) so that an adversary might be able to see that someone connected to the service through Tor but won't be able to see that it was you because you were connected to a VPN and not the Tor Network.

Be aware again that this might not be enough against a motivated global adversary[113] with wide access to global mass surveillance (remember XKEYSCORE, MUSCULAR and PRISM). Such adversary might have access to logs no matter where you are and could use those to de-anonymize you.

I also strongly recommend reading this very good, complete and thorough guide on many Attack Vectors on Tor: https://github.com/Attacks-on-Tor/Attacks-on-Tor as well as this recent research publication https://www.researchgate.net/publication/323627387_Shedding_Light_on_the_Dark_Corners_of_the_Internet_A_Survey_of_Tor_Research

(In their defense, it should also be noted that Tor is not designed to protect against a Global adversary. For more information see https://svn-archive.torproject.org/svn/projects/design-paper/tor-design.pdf and specifically, "Part 3. Design goals and assumptions.".)

Lastly, do remember that using Tor in itself can already be considered a suspicious activity[114] and its use could be considered malicious by some[115].

We will later propose a multi-layered approach over a combination of Tor and VPN as well as using public Wi-Fi over a known internet connection.

Your Crypto currencies transactions:

Contrary to popular belief, Crypto transactions (such as Bitcoin and Ethereum) are not anonymous[116]. Most crypto currencies can be tracked accurately through various methods[117].

The main issue is not setting up a random Crypto wallet to receive some currency behind a VPN/Tor address (at this point, the wallet is anonymous). The issue is mainly when you want to convert Fiat money (Euros, Dollars …) to Crypto and then when you want to cash in your Crypto. You'll have few realistic options but to transfer those to an exchange (such as Coinbase/Kraken/Bitstamp/Binance). Those exchanges have known wallet addresses and will keep detailed logs (due to KYC[118] financial regulations) and can then trace back those crypto transactions to you using the financial system.

There are some crypto currencies with privacy in mind like Monero but even those have some and warnings to consider[119,120].

Even if you use Mixers or Tumblers (services that specialize in anonymizing crypto currencies by "mixing them"), keep in mind this is only obfuscation and not actual anonymity[121].

All in all, IMHO, the best option for using Crypto with reasonable anonymity and privacy is still Monero and you should not use any other for sensitive transactions.

Exploits in your apps and services:

So you're using Tor Browser or Brave Browser with a Tor Tab. You could be using those over a VPN for added security. But you should keep in mind that there are exploits[122] (hacks) that could be known by an adversary (but unknown to the App/Browser provider). Such exploits could be used to compromise your system and reveal details to de-anonymize you such as your IP address or other details.

A real use case of this technique was the Freedom Hosting[123] case in 2013 where the FBI inserted malware[124] using a Firefox browser exploit on a Tor website. This exploit allowed them to reveal details of some users. More recently, there was the notable SolarWinds[125] hack that breached several US government institutions by inserting malware into an official software update server.

There are countless examples of malicious browser extensions, smartphone apps and various apps that have been infiltrated with malware over the years.

Here are some steps to mitigate this type of attack:

- You should never have 100% trust in the apps you're using.
- You should always check that you're using the updated version of such apps before use and ideally validate each download using their signature if available (
- You should not use such apps directly from a hardware system but instead use a Virtual Machine for compartmentalization.

To reflect these recommendations, this guide will therefore later guide you in the use of Virtualization so that even if your Browser/Apps get compromised by a skilled adversary, that adversary will find himself stuck in a sandbox[126] without being able to access identifying information or compromise your system.

Your CPU:

All modern CPUs[127] are now integrating hidden management platforms such as the now infamous Intel Management Engine[128] and the AMD Platform Security Processor[129].

Those management platforms are basically small operating systems running directly on your CPU as long as they have power. These systems have full access to your computer's network and could be accessed by an adversary to de-anonymize you in various ways (using direct access or using malware for instance) as shown in this enlightening videos: BlackHat, How to Hack a Turned-Off Computer, or Running Unsigned Code in Intel Management Engine https://www.youtube.com/watch?v=mYsTBPqbya8

These have already been affected by several security vulnerabilities in the past[130] that allowed malware to gain control of target systems. These are also accused by many privacy actors including the EFF and Libreboot of being a backdoor into any system[131].

There are some not so easy ways[132] to disable the Intel IME on some CPUs and you should do so if you can. For some AMD laptops, you can disable it within the BIOS settings by disabling PSP.

If you're feeling a bit more adventurous, you could install your own BIOS using Libreboot[133] or Coreboot[204] if your laptop supports it.

In addition, some CPUs have unfixable flaws (especially Intel CPUs) that could be exploited by various malware. Here is a good current list of such vulnerabilities affecting recent widespread CPUs:

https://en.wikipedia.org/wiki/Transient_execution_CPU_vulnerability

- If you're using Linux you can check the vulnerability status of your CPU to Spectre/Meltdown attacks by using https://github.com/speed47/spectre-meltdown-checker which is available as a package for most Linux distros including Whonix.
- If you're using Windows you can check the vulnerability status of your CPU using inSpectre https://www.grc.com/inspectre.htm

Some of these like can be avoided by the use of Virtualization Software settings that can mitigate such exploits. See this guide for more information https://www.whonix.org/wiki/Spectre_Meltdown (warning: these can severely impact the performance of your VMs).

I will therefore mitigate some of these issues in this guide by recommending the use of virtual machines on a dedicated anonymous laptop for your sensitive activities that will only be used from an anonymous public network.

Malicious/Rogue Wi-Fi Access Points:

These have been used since at least since 2008 using an attack called "Jasager"[134] and can be done by anyone using self-built tools or using commercially available devices such as Wi-Fi Pineapple[135].

Here are some videos explaining more about the topic:

- HOPE 2020, https://archive.org/details/hopeconf2020/20200725_1800_Advanced_Wi-Fi_Hacking_With_%245_Microcontrollers.mp4
- YouTube, Hak5, Wi-Fi Pineapple Mark VII https://www.youtube.com/watch?v=7v3JR4Wlw4Q

These devices can fit in a small bag and can take over the Wi-Fi environment of any place within their range. For instance a Bar/Restaurant/Café/Hotel Lobby. These devices can force Wi-Fi clients to disconnect from their current Wi-Fi (using de-authentication, disassociation attacks[136]) while spoofing the normal Wi-Fi networks at the same location. They will continue to perform this attack until your computer or yourself decides to try to connect to the rogue AP.

These devices can then mimic a captive portal[137] with the exact same layout as the Wi-Fi you're trying to access (for instance an Airport Wi-Fi registration portal). Or they could just give you open access internet that they'll themselves get from the same place.

Once you're connected through the Rogue AP, this AP will be able to execute various man-in-the-middle attacks to perform analysis on your traffic. These could be malicious redirections or just simple traffic sniffing. These can then easily identify any client that would for instance try to connect to a VPN server or to the Tor Network.

This can be useful when you know someone you want to de-anonymize is in a crowded place but you don't know who. This would allow such an adversary to possibly fingerprint any website you visit despite the use of HTTPS, DoT, DoH, ODoH, VPN or Tor using traffic analysis as pointed above in the DNS section.

These can also be used to carefully craft and serve you advanced phishing webpages that would harvest your credentials or try to make you install a malicious certificate allowing them to see your encrypted traffic.

Malicious USB devices:

There are readily available commercial and cheap "badUSB" [138]devices that can take deploy malware, log your typing, geolocate you, listen to you or gain control of your laptop just by plugging them in. Here are some examples that you can already buy yourself.

- Hak5, USB Rubber Ducky https://shop.hak5.org/products/usb-rubber-ducky-deluxe
- Hak5, O.MG Cable https://www.youtube.com/watch?v=V5mBJHotZv0
- Keelog https://www.keelog.com/
- AliExpress https://www.aliexpress.com/i/4000710369016.html

Such devices can be implanted anywhere (charging cable, mouse, keyboard, USB key …) by an adversary and can be used to track you or compromise your computer or smartphone. The most notable example of such attacks is probably Stuxnet[139] in 2005.

While you could inspect an USB key physically, Scan it with various utilities, check the various components to see if they're genuine, you will most likely never be able to discover complex malware embedded in genuine parts of a genuine USB key by a skilled adversary without advanced forensics equipment[140].

To mitigate this, you should never trust such devices and plug them into sensitive equipment. If you use a charging device, you should consider the use of an USB data blocking device that will only allow charging but not any data transfer. Such data blocking devices are now readily available in many online shops. You should also consider disabling USB ports completely within the BIOS of your computer unless you need them (if you can).

Your Cloud backups/sync services:

All companies are advertising their use of end to end encryption (E2EE). This is true for almost every messaging app and website (HTTPS). Apple and Google are advertising their use of encryption on their Android devices and their IPhones.

But what about your backups? Those automated iCloud/google drive backups you have?

Well you should probably know that most of those backups are not fully end to end encrypted and will contain some of your information readily available for a third party. You will see their claims that data is encrypted at rest and safe from anyone … Except they usually do keep a key to access some of the data themselves. These keys are used for them indexing your content, recover your account, collecting various analytics.

There are specialized commercial forensics solutions available (Magnet Axiom[141], Cellebrite Cloud[142]) that will help an adversary analyze your cloud data with ease.

Notable Examples:

- Apple iCloud: https://support.apple.com/en-us/HT202303 : "Messages in iCloud also uses end-to-end encryption. If you have iCloud Backup turned on**, your backup includes a copy of the key protecting your Messages**. This ensures you can recover your Messages if you lose access to iCloud Keychain and your trusted devices. ".
- Google Drive and WhatsApp: https://faq.whatsapp.com/android/chats/about-google-drive-backups/ : "**Media and messages you back up aren't protected by WhatsApp end-to-end encryption while in Google Drive**. ".
- Dropbox: https://www.dropbox.com/privacy#terms "To provide these and other features, **Dropbox accesses, stores, and scans Your Stuff**. You give us permission to do those things, and this permission extends to our affiliates and trusted third parties we work with.".
- Microsoft OneDrive: https://privacy.microsoft.com/en-us/privacystatement : Productivity and communications products, "When you use OneDrive, we collect data about your usage of the service, as well as the content you store, to provide, improve, and protect the services. **Examples include indexing the contents of your OneDrive documents so that you can search for them later and using location information to enable you to search for photos based on where the photo was taken**."

Generally speaking, you should not trust cloud providers with your (not previously and locally encrypted) sensitive data and you should be wary of their privacy claims. In most cases they can access your data and provide it to a third party.

The only way to mitigate this is to encrypt yourself your data on your side and then only upload it to such service.

Your Digital Fingerprint and Footprint:

The digital fingerprint is the way you write[143], the way you behave[144]. The way you click. The way you browse. The fonts you use on your browser[145]. Fingerprinting is being used to guess who someone is by the way that user is behaving. You might be using specific pedantic words or making specific spelling mistakes that could give you away using a simple Google search for similar features because you typed in a similar way on some Reddit post 5 years ago using a not so anonymous Reddit account.

Social Media platforms such as Facebook/Google can go a step further and can register your behavior in the browser itself. For instance they can register everything you type even if you don't send it / save it. Think of when you write an e-mail in Gmail. It's saved automatically as you type. They can register your clicks and cursor movements as well.

This technology is also widely used in CAPTCHAS[259] services to verify that you are "human" and can be used to fingerprint a user.

Analysis algorithms could then be used to match these patterns with other users and match you to a different known user. It's unclear if such data is used or not by Governments and Law Enforcements agencies but it might be in the future. And while this might only be used for advertising/marketing purposes now. It could and probably will be used for investigations in the short or mid-term future.

Your Clues about your Real Life and OSINT:

These are clues you might give over time that could point to your real identity. You might be talking to someone or posting on some board/forum/Reddit. In those posts you might over time leak some information about your real life. These might be memories, experiences or clues you shared that could then allow a motivated adversary to build a profile to narrow their search.

A real use and well-documented case of this was the arrest of the hacker Jeremy Hammond[146] who shared over time several details about his past and was later discovered.

There are also a few cases involving OSINT at Bellingcat[147].Have a look at their very informative toolkit here: https://bit.ly/bcattools

You can also view a very convenient list of available OSINT tools here https://github.com/jivoi/awesome-osint

As well as this interesting Playlist on YouTube: https://t.co/faJEPGT1j0?amp=1

You should never ever share real personal experiences/details that could later lead to you using anonymous identities.

Your Browser and Device Fingerprints:

Your Browser and Device Fingerprints[271] are set of properties/capabilities of your System/Browser. These are used on most websites for invisible user tracking but also to adapt the website user experience depending on their browser. For instance websites will be able to provide a "mobile experience" if you're using a mobile browser or propose a specific language/geographic version depending on your fingerprint. Most of those techniques work with recent Browsers like Chromium[148] based browsers (such as Chrome) or Firefox[149] unless taking special measures.

You can find a lot of detailed information and publications about this on these resources:

- https://amiunique.org/links
- https://brave.com/brave-fingerprinting-and-privacy-budgets/

Most of the time, those fingerprints will unfortunately be unique or nearly unique to your Browser/System. This means that even If you log out from a website and then log back in using a different username, your fingerprint might remain the same if you didn't take precautionary measures.

An adversary could then use such fingerprints to track you across multiple services even if you have no account on any of them and are using ad blocking. These fingerprints could in turn be used to de-anonymize you if you keep the same fingerprint between services.

It should also be noted that while some browsers and extensions will offer fingerprint resistance, this resistance in itself can also be used to fingerprint you as explained here https://palant.info/2020/12/10/how-anti-fingerprinting-extensions-tend-to-make-fingerprinting-easier/

This guide will mitigate these issues by mitigating, obfuscating and randomizing many of those fingerprinting identifiers by using Virtualization and also using by fingerprinting resistant Browsers.

Your Face, Voice, Biometrics and Pictures:

"Hell is other people", even if you evade every methods listed above, you're not out of the woods yet thanks to the widespread use of advanced Face recognition by everyone.

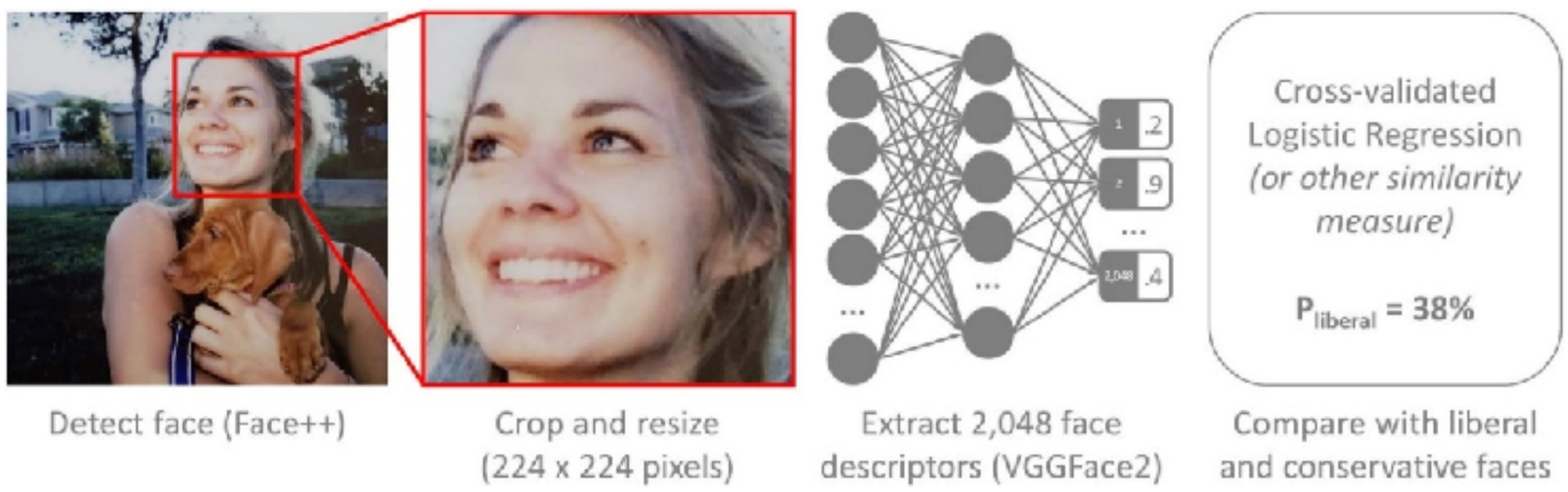
Companies like Facebook have used advanced face recognition for years[150,151] and have been using other means (Satellite imagery) to create maps of "people" around the world[152]. This evolution has been going on for years to the point we can now say "We lost control of our faces"[153].

If you are walking in a touristy place, you'll most likely appear in someone's selfie within minutes without knowing it. That person will then proceed to upload that selfie to various platforms (Twitter, Google Photos, Instagram, Facebook, Snapchat …). Those platforms will then apply face recognition algorithms to those pictures under the pretext of allowing better/easier tagging or to better organize your photo library. In addition to this, the same picture will provide a precise timestamp and in most cases geolocation of where it was taken. Even if the person doesn't provide a timestamp and geolocation, it can still be guessed with other means[154,155].

Here are a few resources for even trying this yourself:

- Bellingcat, Guide To Using Reverse Image Search For Investigations: https://www.bellingcat.com/resources/how-tos/2019/12/26/guide-to-using-reverse-image-search-for-investigations/
- Bellingcat, Using the New Russian Facial Recognition Site SearchFace https://www.bellingcat.com/resources/how-tos/2019/02/19/using-the-new-russian-facial-recognition-site-searchface-ru/
- Bellingcat, Dali, Warhol, Boshirov: Determining the Time of an Alleged Photograph from Skripal Suspect Chepiga https://www.bellingcat.com/resources/how-tos/2018/10/24/dali-warhol-boshirov-determining-time-alleged-photograph-skripal-suspect-chepiga/
- Bellingcat, Advanced Guide on Verifying Video Content https://www.bellingcat.com/resources/how-tos/2017/06/30/advanced-guide-verifying-video-content/
- Bellingcat, Using the Sun and the Shadows for Geolocation https://www.bellingcat.com/resources/2020/12/03/using-the-sun-and-the-shadows-for-geolocation/
- Bellingcat, Navalny Poison Squad Implicated in Murders of Three Russian Activists https://www.bellingcat.com/news/uk-and-europe/2021/01/27/navalny-poison-squad-implicated-in-murders-of-three-russian-activists/

Even if you're not looking at the camera, they can still figure out who you are[156], make out your emotions[157], analyze your gait[158] and probably guess your political affiliation[159].



Those platforms (Google/Facebook) already know who you are for a few reasons:

- Because you have or had a profile with them and you identified yourself.
- Even if you never made a profile on those platforms, you still have one without even knowing it[160,161,162,163,164].
- Because other people have tagged you or identified you in their holidays/party pictures.
- Because other people have put a picture of you in their contact list which they then shared with them.

Governments already know who you are because they have your ID/Passport/Driving License pictures and often added biometrics (Fingerprints) in their database. Those same governments are integrating those technologies (often provided by private companies such as the Israeli AnyVision[165]) in their CCTV networks to look for "persons of interest"[166]. And some heavily surveilled states like China have implemented widespread use of Facial Recognition for various purposes[167] including possibly identifying ethnic minorities[168]. A simple face recognition error by some algorithm can ruin your life[169].

Here are some resources detailing some techniques used by Law Enforcement today:

- CCC video explaining current Law Enforcement surveillance capabilities: https://media.ccc.de/v/rc3-11406-spot_the_surveillance#t=761
- EFF SLS: https://www.eff.org/sls

Apple is making FaceID mainstream and pushing its use it to log you in in various services including the Banking systems.

Same goes with fingerprint authentication being mainstreamed by many smartphone makers to authenticate yourself. A simple picture where your fingers appear can be used to de-anonymize you[170].

We can safely imagine a near future where you won't be able to create accounts or sign-in anywhere without providing unique biometrics (A good time to re-watch Gattaca[171], Person of Interest[172] and Minority Report[173]). And you can safely imagine how useful these large biometrics databases could be to some interested third parties.

In addition, all this information can also be used against you (if you are already de-anonymized) using deepfake[174] by crafting false information (Pictures, Videos, Voice Recordings[175]…) and have already been used for such purposes[176,177]. There are even commercial services for this readily available such as https://www.respeecher.com/ and https://www.descript.com/overdub.

See this demo: https://www.youtube.com/watch?v=t5yw5cR79VA

At this time, there are a few steps[178] you can use to mitigate (and only mitigate) face recognition when conducting sensitive activities where CCTV might be present:

- Wear a facemask as they have been proven to defeat some face recognition technologies[179].
- Wear a baseball cap or hat to mitigate identification from high angle CCTVs (filming from above) from recording your face. Remember this will not help against front-facing cameras.
- Wear sunglasses in addition to the facemask and baseball cap to mitigate identification from your eyes features.

(Note that if you intend to use these where advanced facial recognition systems have been installed, these measures could also flag as you as suspicious by themselves and trigger a human check)

Phishing and Social Engineering:

Phishing[180] is a social engineering[181] type of attack where an adversary could try to extract information from you by pretending to be something/someone else.

A typical case is an adversary using a man-in-the-middle[57] attack or a falsified e-mail/call to ask your credential for a service. This can be your e-mail or your financial services for example.

Such attacks can also be used to de-anonymize someone by tricking them into downloading malware or revealing personal information.

These have been use countless times and the usual one is called the "419 scam" (see https://en.wikipedia.org/wiki/Advance-fee_scam).

Here is a good video if you want to learn a bit more about phishing types: Black Hat, Ichthyology: Phishing as a Science https://www.youtube.com/watch?v=Z20XNp-luNA.

Local Data Leaks and Forensics:

Most of you have probably seen enough Crime dramas on Netflix or TV to know what forensics are. These are technicians (usually working for law enforcement) that will perform various analysis of evidence. This of course could include your smartphone or laptop.

While these might be done by an adversary when you already got "burned", these might also be done randomly during a routine control or a border check. These unrelated checks might reveal secret information to adversaries that had no prior knowledge of such activities.

Forensics techniques are now very advanced and can reveal a staggering amount information from your devices even if they're encrypted[182]. These techniques are widely used by law enforcement all over the world and should be considered.

Here are some recent resources you should read about your smartphone:

- UpTurn, The Widespread Power of U.S. Law Enforcement to Search Mobile Phones https://www.upturn.org/reports/2020/mass-extraction/
- New-York Times, The Police Can Probably Break Into Your Phone https://www.nytimes.com/2020/10/21/technology/iphone-encryption-police.html
- Vice, Cops Around the Country Can Now Unlock iPhones, Records Show https://www.vice.com/en/article/vbxxxd/unlock-iphone-ios11-graykey-grayshift-police

I also highly recommend that you read some documents from a forensics examiner perspective such as:

- EnCase Forensic User Guide, http://encase-docs.opentext.com/documentation/encase/forensic/8.07/Content/Resources/External%20Files/EnCase%20Forensic%20v8.07%20User%20Guide.pdf
- FTK Forensic Toolkit, https://accessdata.com/products-services/forensic-toolkit-ftk
- SANS Digital Forensics and Incident Response Videos, https://www.youtube.com/c/SANSDigitalForensics/videos

And last but not least here is this very instructive detailed paper on the current state of IOS/Android security from the John Hopkins University: https://securephones.io/main.html[183]

When it comes to your laptop, the forensics techniques are many and widespread. Many of those issues can be mitigated by using full disk encryption, virtualization and compartmentalization. This guide will later detail such threats and techniques to mitigate them.

No logging but logging anyway policies:

Many people have the idea that privacy oriented services such as VPN or E-Mail providers are safe due to their no logging policies or their encryption schemes. Unfortunately many of those same people forget that all those providers are legal commercial entities subject to the laws of the countries in which they operate.

Any of those providers can be forced to silently (without your knowing (using for example a court order with a gag order[184] or a national security letter[185]) log your activity to de-anonymize you. There have been several recent examples of those:

- 2020, The Germany based mail provider Tutanota was forced to implement a backdoor to save unencrypted copies of the e-mails of one user[186].
- 2017, PureVPN was forced to disclose information of one user to the FBI[187].
- 2014, EarthVPN user was arrested based on logs provider to the Dutch Police[188].
- 2014, HideMyAss user was de-anonymized and logs were provider to the FBI[189].

- 2013, Secure E-Mail provider Lavabit shuts down after fighting a secret gag order[190].

Some providers have implemented the use of a Warrant Canary[191] that would allow their users to find out if they have been compromised by such orders but this has not been tested yet as far as I know.

Last but not least, it's now well known that some companies might be sponsored front-ends for some state adversaries (see the Crypto AG story[192] and Omnisec story[193]).

For these reasons, it's important that you don't trust such providers for your privacy despite all their claims. In most cases, you will be the last person to know if any of your account was targeted by such orders and you might never know at all.

To mitigate this I will recommend the use of a cash-paid VPN provider over Tor to prevent the VPN service from knowing any identifiable information about you. I will also recommend the creation of anonymous e-mail accounts from this VPN over Tor connection to also prevent them from having any information despite their no logging policies.

Some Advanced targeted techniques:

There are many advanced techniques that can be used by skilled adversaries[194] to bypass your security measures provided they already know where your devices are. Many of those techniques are detailed here https://cyber.bgu.ac.il/advanced-cyber/airgap (Air-Gap Research Page, Cyber-Security Research Center, Ben-Gurion University of the Negev, Israel) and include:

- Attacks that require a malware implanted in some device:
  - Exfiltration of Data through a Malware infected Router: https://www.youtube.com/watch?v=mSNt4h7EDKo
  - Exfiltration of Data through observation of Light variation in a Backlit keyboard with a compromised camera: https://www.youtube.com/watch?v=1kBGDHVr7x0
    - Exfiltration of Data through a compromised Security Camera (that could first use the previous attack) https://www.youtube.com/watch?v=om5fNqKjj2M
    - Communication from outsider to compromised Security Cameras through IR light signals: https://www.youtube.com/watch?v=auoYKSzdOj4
  - Exfiltration of data from a compromised air-gapped computer through acoustic analysis of the FAN noises with a smartphone https://www.youtube.com/watch?v=v2_sZIfZkDQ
  - Exfiltration of data from a malware infected air-gapped computer through HD Leds with a Drone https://www.youtube.com/watch?v=4vIu8ld68fc
  - Exfiltration of data from a USB malware on an air-gapped computer through electromagnetic interferences https://www.youtube.com/watch?v=E28V1t-k8Hk
  - Exfiltration of data from a malware infected HDD drive through covert acoustic noise https://www.youtube.com/watch?v=H7lQXmSLiP8
  - Exfiltration of data through GSM frequencies from a compromised (with malware) air-gapped computer https://www.youtube.com/watch?v=RChj7Mg3rC4
  - Exfiltration of data through electromagnetic emissions from a compromised Display device https://www.youtube.com/watch?v=2OzTWiGl1rM&t=20s
  - Exfiltration of data through magnetic waves from a compromised air-gapped computer to a Smartphone stored inside a Faraday bag https://www.youtube.com/watch?v=yz8E5n1Tzlo
  - Communication between two compromised air-gapped computers using ultrasonic soundwaves https://www.youtube.com/watch?v=yz8E5n1Tzlo
  - Exfiltration of Bitcoin Wallet from a compromised air-gapped computer to a smartphone https://www.youtube.com/watch?v=2WtiHZNeveY
  - Exfiltration of Data from a compromised air-gapped computer using display brightness https://www.youtube.com/watch?v=ZrkZUO2g4DE
  - Exfiltration of Data from a compromised air-gapped computer through vibrations https://www.youtube.com/watch?v=XGD343nq1dg

    - Exfiltration of Data from a compromised air-gapped computer by turning RAM into a Wi-Fi emitter https://www.youtube.com/watch?v=vhNnc0ln63c
    - Exfiltration of Data from a compromised air-gapped computer through power lines https://arxiv.org/abs/1804.04014
- **Attacks that require no malware:**
    - Observing a light bulb from a distance to listen to the sound in the room[195] **without any malware**: Demonstration: https://www.youtube.com/watch?v=t32QvpfOHqw

Here is also a good video from the same authors to explain those topics: Black Hat, The Air-Gap Jumpers https://www.youtube.com/watch?v=YKRtFgunyj4

This guide will be of little help against such adversaries as these malwares could be implanted on the devices by a manufacturer or anyone in the middle or by anyone with physical access to the air-gapped computer but there are still some ways to mitigate such techniques:

- Do not conduct sensitive activity while connected to an untrusted/unsecure power line to prevent power line leaks.
- Do not use your devices in front of a camera that could be compromised.
- Use your devices in a soundproofed room to prevent sound leaks.
- Use your devices in faraday cage to prevent electromagnetic leaks.
- Do not talk sensitive information where lightbulbs could be observed from outside.
- Buy your devices from different/unpredictable/offline places (shops) where the probability of them being infected with such malware is lower.
- Do not let anyone access your air-gapped computers except trusted people.

Notes:

Have a look at https://www.eff.org/issues/privacy for more resources.

If you still don't think such information can be used by various actors to track you, you can see some statistics for yourself for some platforms and keep in mind those are only accounting for the lawful data requests and won't count things like PRISM, MUSCULAR or XKEYSCORE explained earlier:

- Google Transparency Report https://transparencyreport.google.com/user-data/overview
- Facebook Transparency Report https://transparency.facebook.com/
- Apple Transparency Report https://www.apple.com/legal/transparency/
- Cloudflare Transparency Report https://www.cloudflare.com/transparency/
- Snapchat Transparency Report https://www.snap.com/en-US/privacy/transparency
- Telegram Transparency Report https://t.me/transparency (requires telegram installed)
- Microsoft Transparency Report https://www.microsoft.com/en-us/corporate-responsibility/law-enforcement-requests-report
- Amazon Transparency Report https://www.amazon.com/gp/help/customer/display.html?nodeId=GYSDRGWQ2C2CRYEF
- Dropbox Transparency Report https://www.dropbox.com/transparency
- Discord Transparency Report https://blog.discord.com/discord-transparency-report-jan-june-2020-2ef4a3ee346d
- GitHub Transparency Report https://github.blog/2020-02-20-2019-transparency-report/
- Snapchat Transparency Report https://www.snap.com/en-US/privacy/transparency/
- TikTok Transparency Report https://www.tiktok.com/safety/resources/transparency-report?lang=en
- Reddit Transparency Report https://www.reddit.com/wiki/transparency
- Twitter Transparency Report https://transparency.twitter.com/

You might also enjoy taking a look at this service https://tosdr.org/ (Terms of Services, Didn't Read) that will give you a good overview of the various ToS of many services.

General Preparations:

Picking your route:

Timing limitations:

- You have very limited time to learn and need a fast working solution:
    - **Your best option is to go for the TAILS route.**
- You have time to learn:
    - **Go with any other route.**

Budget/Material limitations:

- You only have one laptop available and cannot afford anything else. You use this laptop for either work, family or your personal stuff (or both):
    - **Your best option is to go for the TAILS route.**
- You can afford a spare dedicated laptop for your sensitive activities:
    - But it's old, slow and has bad specs (less than 6GB of RAM, less than 250GB disk space, old/slow CPU)
        - **You should go for the TAILS route.**
    - It's not that old and it has decent specs (at least 6GB of RAM, 250GB of disk space or more, decent CPU)
        - **You could go for TAILS, Whonix routes.**
    - It's new and it has great specs (more than 8GB of RAM, >250GB of disk space, recent fast CPU)
        - **You could go for TAILS, Whonix but I would recommend Qubes.**
    - If it's an ARM based M1 Mac
        - **Not possible at this time for a few reasons:**
            - **Virtualization of x86 images on ARM M1 Macs is still limited to commercial software (Parallels) which is not supported by Whonix yet.**
            - **Virtualbox is not available for ARM architecture yet.**
            - **Whonix is not supported on ARM architecture yet.**
            - **TAILS is not supported on ARM architecture yet.**
            - **Qubes OS is not supported on ARM architecture yet.**

Your only option on M1 Macs is to stick with Tor Browser or Brave Tor Tabs for now. But I would guess that if you can afford an M1 Mac you should probably get a dedicated x86 laptop for more sensitive activities.

Skills:

- You have no IT skills at all the content of this guide looks like an alien language to you?
    - **You should go with the TAILS route.**
- You have some IT skills and mostly understand this guide so far
    - **You should go with TAILS or Whonix routes.**
- You have moderate to high IT skills and you're already familiar with some of the content of this guide
    - **You could go with anything you like but I would strongly recommend Qubes.**
- You are a l33T hacker, there is no spoon, the cake is a lie, you've been using doas for years and all your base are belong to us
    - **This guide is not really meant for you and won't help you with OpenBSD ;-)**

Adversaries (threats):

- If your main concern is forensic examination of your devices:
    - **You should go with the TAILS route.**
- If your main concerns are remote adversaries that might uncover your online identity in various platforms:
    - **You should go with the Whonix or Qubes OS routes.**
- If you absolutely want system wide plausible deniability[215,196] despite the risks[197,217].
    - **You could go with the Whonix Route.**
    - **You could go with a combination of TAILS and Hidden VMs (requires a bit more setup)**

- If you are in an hostile environment where Tor/VPN usage alone is impossible/dangerous/suspicious:
    - **(preferred) You could go with the TAILS route (without using Tor).**
    - **You could go with the Qubes OS route if you want data persistence (without using Whonix).**
    - **You could go with the Whonix route too (without actually using Whonix).**

In all cases, you should read these two pages from the Whonix documentation that will give you in depth insight about your choices:

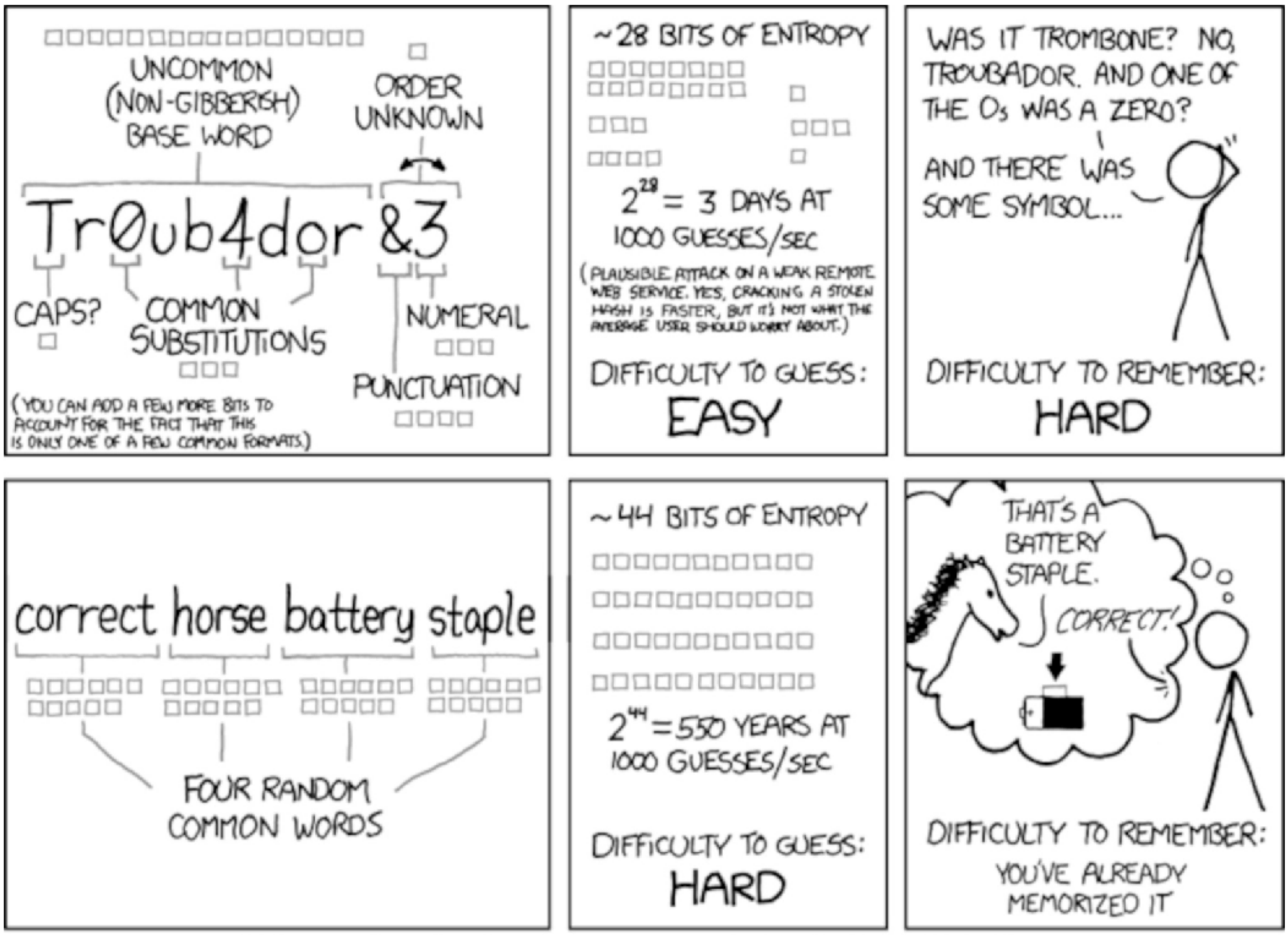
- https://www.whonix.org/wiki/Warning
- https://www.whonix.org/wiki/Dev/Threat_Model
- https://www.whonix.org/wiki/Comparison_with_Others

You might be asking yourself: "How do I know if I'm in a hostile online environment where activities are actively monitored and blocked?"

- First read more about it at the EFF here: https://ssd.eff.org/en/module/understanding-and-circumventing-network-censorship
- Check some data yourself here on the Tor Project OONI[198] (Open Observatory of Network Interference) website: https://explorer.ooni.org/
- Have a look at https://censoredplanet.org/ and see if they have data about your country.
- Test for yourself using OONI (this can be risky in a hostile environment).

Steps for all routes:

Always use passphrases instead of passwords and use a different one for each service. Do not make it easy for an adversary to access all your information because you used the same password everywhere[206].



(Illustration by xkcd.com, licensed under CC BY-NC 2.5)

Get an anonymous Phone number:

Skip this step if you have no intention of creating anonymous accounts on most mainstream platforms but just want anonymous browsing or if the platforms you will use allow registration without a phone number.

Physical Burner Phone and prepaid SIM card:

Get a burner phone:

This is rather easy. Leave your smartphone off or power it off before leaving. Have some cash and go to some random flea market or small shop (ideally one without CCTV inside or outside and while avoiding being photographed/filmed) and just buy the cheapest phone you can find with cash and without providing any personal information. It only needs to be in working order.

Personally I would recommend getting an old "dumbphone" with a removable battery (old Nokia if your mobile networks still allows those to connect as some countries phased out 1G-2G completely). This is to avoid the automatic sending/gathering of any telemetry/diagnostic data on the phone itself. You should never connect that phone to any WIFI.

It will also be crucial not to power on that burner phone ever (not even without the SIM card) in any geographical location that could lead to you (at your home/work for instance) and never ever at the same location as your other known smartphone (because that one has an IMEI/IMSI that will easily lead to you). This might seem like a big burden but it's not really as these phones are only being used during the setup/sign-up process and for verification from time to time.

See Appendix N: Warning about smartphones and smart devices

You should test that the phone is in working order before going to the next step. But I will repeat myself and state again that it's really important that you leave your smartphone at home when going (or turn it off before leaving if you have to keep it) and that you test the phone at a random location that cannot be tracked back to you (and again, don't do that in front of a CCTV, avoid cameras, be aware of your surroundings). No need for WIFI at this place either.

When you are certain the phone is in working order, disable Bluetooth then power it off (remove the battery if you can) and go back home and resume your normal activities. Go to the next step.

Get an anonymous pre-paid SIM card:

This is the hardest part of the whole guide. It's a SPOF (Single Point of Failure). The places where you can still buy prepaid SIM cards without ID registration are getting more and more limited due to various KYC type regulations[199].

So here is a list of places where you can still get them at the moment: https://prepaid-data-sim-card.fandom.com/wiki/Registration_Policies_Per_Country

You should be able to find a place that is "not too far" and just go there physically to buy some pre-paid cards and top-up vouchers with cash. Do verify that no law was passed before going that would make registration mandatory (in case the above wiki was not updated). Try to avoid CCTV and cameras and don't forget to buy a Top Up voucher with the SIM card (if it's not a package) as most pre-paid cards will require a top-up before use.

See Appendix N: Warning about smartphones and smart devices

Double-check that the mobile operators selling the pre-paid SIM cards will accept the SIM activation and top-up without any ID registration of any kind before going there. Ideally they should accept SIM activation and top-up from the country you reside in.

Personally, I would recommend GiffGaff in the UK as they're "affordable", do not require identification for activation and top-pup and will even allow you to change your number up to 2 times from their website. One GiffGaff prepaid SIM card will therefore grant you 3 numbers to use for your needs.

Power off the phone after activation/top-up and before going home. Do not ever power it on again unless you're not at a place that can be used to reveal your identity and unless your smartphone is powered off before going to that "not your home" place.

Online Phone Number (less recommended):

DISCLAIMER: Do not attempt this until you are done setting up a secure environment according to one of the selected routes. This step will require online access and should only be done from an anonymous network. Do not do this from any known/unsecure environment. Skip this until you have finished one of the routes.

There are many commercial services offering numbers to receive SMS messages online but most of those have basically no anonymity/privacy and can be of no help as most Social Media platforms place a limit on how many times a phone number can be used for registration.

There are some forums and subreddits (like r/phoneverification/) where users will offer the service of receiving such SMS messages for you for a small fee (using PayPal or some crypto payment). Unfortunately these are full of scammer and very risky in terms of anonymity. **You should not use those under any circumstance.**

To this date, I do not know any reputable service that would offer this service and accept cash payments (by post for instance) like some VPN providers. But there are a few services I'm that provides online phone numbers and do accept Monero which could be reasonably anonymous (yet less recommended than that physical way in the previous chapter) that you could consider:

- **Recommended**: Do not require any identification (even e-mail):
    - (UK based) https://dtmf.io/ **preferred** because they even provide an onion hidden service address for direct access through the Tor Network at http://dtmfiovjh42uviqez6qn75igbagtiyo724hy3rdxm77dy2m5tt7lbaqd.onion/
    - (Ukraine based) https://virtualsim.net/

- Do require some kind of identification (valid e-mail):
    - (Germany based) https://www.sms77.io/
    - (Russia based) https://onlinesim.ru/

There are some other possibilities listed here https://cryptwerk.com/companies/sms/xmr/. **Use at your own risk.**

DISCLAIMER: I cannot vouch for any of these providers and therefore I will still recommend doing it yourself physically. In this case you will have to rely on the anonymity of Monero and you should not use any service that requires any kind of identification using your real identity.

Therefore IMHO, it's probably just more convenient, cheaper and less risky to just get a pre-paid SIM card from one of the physical places who still sell them for cash without requiring ID registration. But at least there is an alternative if you have no other option.

Get an USB key:

Get at least one decent size generic USB key (at least 16GB).

Please do not buy or use gimmicky self-encrypting devices such as these:
https://syscall.eu/blog/2018/03/12/aigo_part1/

Some might be very efficient[200] but many are gimmicky gadgets that offer no real protection[201].

Find some safe places with decent public WIFI:

You need to find safe places where you'll be able to do your sensitive activities using some publicly accessible WIFI (without any account/ID registration, avoid CCTVs).

This can be anywhere that won't be tied to you directly (your home/work) and where you can use the WIFI for a while without being bothered. But also a place where you can do this without being "noticed" by anyone.

If you think Starbucks is a good idea, you may reconsider:

- They probably have CCTVs in all their shops and keep those recordings for an unknown amount of time.
- You'll need to buy a coffee to get the WIFI access code in most. If you pay this coffee with an electronic method, they will be able to tie your WIFI access with your identity.

Situational awareness is key and you should be constantly aware of your surroundings and avoid touristy places like it was plagued by Ebola. You want to avoid appearing on any picture/video of anyone while someone is taking a selfie, making a TikTok video or posting some travel picture on their Instagram. If you do, remember chances are high that those pictures will end up online (publicly or privately) with full metadata attached to them (time/date/geolocation) and your face. Remember these can and will be indexed by Facebook/Google/Yandex/Apple and probably all 3 letters agencies.

While this won't be available yet to your local cops, it could be in the near future.

You will ideally need a set of 3-5 different places such as this to avoid using the same place twice. Several trips will be required over the weeks for the various steps in this guide.

You could also consider connect to these places from a safe distance for added security. See Appendix Q: Using long range Antenna to connect to Public Wi-Fis from a safe distance:

The TAILS route:

This part of the guide will help you in setting up TAILS if one of the following is true:

- You can't afford a dedicated laptop
- Your dedicated laptop is just too old and too slow
- You have very low IT skills
- You decide to go with TAILS anyway

TAILS[202] stands for **The Amnesic Incognito Live System.** . It's a bootable Live Operating System running from a USB key that is designed for leaving no traces and forcing all connections through the Tor network.

You pretty much insert the Tails USB key into your laptop, boot from it and you have a full operating system running with privacy and anonymity in mind. As soon as you shut down the computer, everything will be gone unless you saved it somewhere.

Tails is a very easy way to get going in no time with what you have and without much learning. It has extensive documentation and tutorials.

It does however have some drawbacks:

- Tails uses Tor and therefore you'll be using Tor to access any resource on the internet. This alone will make you suspicious to most platforms where you want to create anonymous accounts (this will be explained in more details later).
- Your ISP (whether it's yours or some public Wi-Fi) will also see that you're using Tor and this could make you suspicious in itself.
- Tails doesn't include (natively) some of the software you might want to use later which will complicate things quite a bit if you want to run some specific things (Android Emulators for instance).
- Tails uses Tor Browser which while it's very secure will be detected as well by most platforms and will hinder you in creating anonymous identities on many platforms.
- Tails won't protect you more from the 5$ wrench[7].
- Tor in itself might not be enough to protect you from an adversary with enough resources as explained earlier.

You should also read Tails Documentation, Warnings and limitations, before going further https://tails.boum.org/doc/about/warning/index.en.html

Taking all this into account and the fact that their documentation is great, I'll just redirect you towards their well-made and well-maintained tutorial:

https://tails.boum.org/install/index.en.html , pick your flavor and proceed.

When you're done and have a working Tails on your laptop, go to the "Creating your anonymous online identities" Step much further in this guide.

If you're having issue accessing Tor due to censorship or other issues, you can try using Tor Bridges by following this TAILS tutorial: https://tails.boum.org/doc/first_steps/welcome_screen/bridge_mode/index.en.html and find more information about these on Tor Documentation https://2019.www.torproject.org/docs/bridges.

If you think using Tor alone is dangerous/suspicious, see Appendix P: Accessing the internet as safely as possible when Tor/VPN is not an option

Plausible Deniability with the TAILS route:

Consider checking the https://github.com/aforensics/HiddenVM project for TAILS.

This project is a clever idea of a one click self-contained VM solution that you could store on an encrypted disk using plausible deniability[215] (see The Whonix route: first chapters and also for some explanations about Plausible deniability, as well as the How to securely delete specific files/folders/data on your HDD/SDD and Thumb drives: section at the end of this guide for more understanding).

This would allow the creation of a hybrid system mixing TAILS with the virtualization options of the Whonix route in this guide.

In short:

- You could run non-persistent TAILS from one USB key (following their recommendations)
- You could store persistent VMs within a secondary contained that could be encrypted normally or using Veracrypt plausible deniability feature (these could be Whonix VMs for instance or any other).

In that case as the project outlines it, there should be no traces of any of your activities on your computer and the sensitive work could be done from VMs stored into a Hidden container that should not be discoverable by an adversary.

In the future, this should also be supported by the Whonix project themselves as explained here: https://www.whonix.org/wiki/Whonix-Host

Remember that encryption with or without plausible deniability is not a silver bullet and will be of little use in case of torture[7]. As a matter a fact, depending on who your adversary would be (your threat model), it might be wise not to use Veracrypt (formerly TrueCrypt) at all as shown in this demonstration: https://defuse.ca/truecrypt-plausible-deniability-useless-by-game-theory.htm

Plausible deniability is only effective against soft lawful adversaries that won't resort to physical means.

See https://en.wikipedia.org/wiki/Rubber-hose_cryptanalysis

CAUTION: Please see Appendix K: Considerations for using external SSD drives and Understanding HDD vs SSD: sections if you consider storing such hidden VMs on an external SSD drive:

- **Do not use hidden volumes on SSD drives as this is not supported/recommended by Veracrypt[203].**
- **Use instead file containers instead of encrypted volumes.**
- **Make sure you do know how to clean data from an external SSD drive properly.**

Steps for all other routes:

Get a dedicated laptop for your sensitive activities:

Ideally, you should get a dedicated laptop that won't be tied to you in any easy way (ideally paid with cash anonymously and using the same precautions as previously mentioned for the phone and the SIM card). It's recommended but not mandatory because this guide will help you harden your laptop as much as possible to prevent data leaks through various means. There will be several lines of defense standing between your online identities and yourself that should prevent most adversaries from de-anonymizing you besides state/global actors with considerable resources.

This laptop should ideally be a clean freshly installed Laptop (Running Windows, Linux or MacOS), clean of your normal day to day activities and offline (never connected to the network yet). In the case of a Windows laptop, and if you used it before such a clean install, it should also not be activated (re-installed without a product key).

This is to mitigate some future issues in case of online leaks (including telemetry from your OS or Apps) that could compromise any unique identifiers of the laptop while using it (MAC Address, Bluetooth Address, and Product key …). But also to avoid being tracked back if you need to dispose of the laptop.

If you used this laptop before for different purposes (like your day to day activities), all its hardware identifiers are probably known and registered by Microsoft or Apple. If later any of those identifiers is compromised (by malware, telemetry, exploits, human errors …) they could lead back to you.

The laptop should have at least 250GB of Disk Space **at least 6GB** of RAM and should be able to run a couple of Virtual Machines at the same time. It should have a working battery that lasts a few hours.

This laptop could have an HDD (7200rpm) or an SSD. Both possibilities have their perks and issues that will be detailed later.

All future online steps performed with this laptop should ideally be done from a safe network such as a Public Wi-Fi in a safe place (see Find some safe places with decent public WIFI:). But several steps will have to be taken offline first.

Some laptop recommendations:

If you can afford it, you might consider getting a Purism Librem laptop (https://puri.sm) or System76 laptops (https://system76.com/) while using Coreboot[204] (where Intel IME is disabled from factory).

In other cases, I would strongly recommend getting Business Laptops (meaning not consumer/gaming laptops) if you can. For instance some ThinkPad from Lenovo (my personal favorites). Here are lists of laptops currently supporting Libreboot and others where you can flash Coreboot yourself (that will allow you to disable Intel IME or AMD PSP):

- https://freundschafter.com/research/system-alternatives-without-intel-me-iamt-and-amd-psp-secure-technology/
- https://libreboot.org/docs/hardware/
- https://coreboot.org/status/board-status.html

This is because those business laptops usually offer better and more customizable security features with longer support than most consumer laptops (Asus, MSI, Gigabyte, Acer…). The interesting features to look for are IMHO:

- Better custom Secure Boot **settings (where you can selectively manage all the keys and not just use the Standard ones)**
- HDD/SDD passwords in addition to just BIOS/UEFI passwords.
- AMD laptops could be more interesting as some provide the ability to disable AMD PSP (the AMD equivalent of Intel IME) from the BIOS/UEFI settings by default. However if you're going for the Qubes Route consider Intel as they do not support AMD with their anti-evil-maid system[205].
- Secure Wipe tools from the BIOS (especially useful for SDD drives, see Appendix M: BIOS/UEFI options to wipe disks in various Brands).
- Better control over the disabling/enabling of select peripherals (USB ports, WiFis, Bluetooth, Camera, Microphone …).
- Better security features with Virtualization.
- Native anti-tampering protections.
- Longer support with BIOS/UEFI updates (and security updates).
- Some are supported by Libreboot

Bios/UEFI/Firmware Settings of your laptop:

PC:

These settings can be accessed through the boot menu of your laptop. Here is a good tutorial from HP explaining all the ways to access the BIOS on various computers: https://store.hp.com/us/en/tech-takes/how-to-enter-bios-setup-windows-pcs

Usually how to access it is pressing a specific key (F1, F2 or Del) at boot (before your OS).

Once you're in there you'll need to apply a few recommended settings:

- Disable Bluetooth completely if you can.
- Disable Biometrics (fingerprint scanners) if you have any if you can. However you could add a biometric additional check for booting only (pre-boot) but not for accessing the BIOS/UEFI settings.
- Disable the Webcam and Microphone if you can.
- Enable BIOS/UEFI password and use a long passphrase[206] instead of a password if you can and make sure this password is required for:
    - Accessing the BIOS/UEFI settings themselves
    - Changing the Boot order
    - Startup/Power-on of the device
- Enable HDD/SDD password if the feature is available. This feature will add another password on the HDD/SDD itself (not in the BIOS/UEFI firmware) that will prevent this HDD/SDD from being used in a different computer without the password. Note that this feature is also specific to some manufacturers and could require specific software to unlock this disk from a completely different computer.
- Prevent accessing the boot options (the boot order) without providing the BIOS/UEFI password if you can.
- Disable USB/HDMI or any other port (Ethernet, Firewire, SD card …) if you can.
- Disable Intel ME if you can.
- Disable AMD PSP if you can (AMD's equivalent to IME, see Your CPU:)

- Disable Secure Boot if you intend to use QubesOS as they do not support it out of the box[207]. Keep it on if you intend to use Linux/Windows.
- Check if your laptop BIOS has a secure erase option for your HDD/SDD that could be convenient in case of need.

Only enable those on a "need to use" basis and disable then again after use. This can help mitigate some attacks in case your laptop is seized while locked but still on OR if you had to shut it down rather quickly and someone took possession of it (this topic will be explained later in this guide).

About Secure boot:

So what is Secure Boot[208]? In short, it's a UEFI security feature designed to prevent your computer from booting an operating system from which the bootloader was not signed by specific keys stored in the UEFI firmware of your laptop.

Basically when the Operating Systems (or the Bootloader[209]) supports it, you can store the keys of your bootloader in your UEFI firmware and this will prevent booting up any unauthorized Operating System (such as a live OS USB or anything similar).

Secure Boot settings are protected be the password you setup to access the BIOS/UEFI settings. If you have that password, you can disable Secure Boot and allow unsigned OSes to boot on your system. This can help mitigate some Evil-Maid attacks (explained later in this guide).

In most cases Secure Boot is disabled by default or is enabled but in "setup" mode which will allow any system to boot. In order for Secure Boot to work, your Operating System will have support it and then sign its bootloader and push those signing keys to your UEFI firmware. After that you'll have to go to your BIOS/UEFI settings and save those pushed keys from your OS and change the Secure Boot from setup to user mode (or custom mode in some cases).

After doing that step, only the Operating Systems from which your UEFI firmware can verify the integrity of the bootloader will be able to boot.

Most laptops will have some default keys already stored in the secure boot settings. Usually those from the manufacturer itself or from some companies such as Microsoft. So this means that by default, it will always be possible to boot some USB disks even with secure boot. These includes Windows, Fedora, Ubuntu, Mint, Debian, CentOS, OpenSUSE, TAILS, Clonezilla and many others. Secure Boot is however not supported at all by QubesOS at this point.

In some laptops, you can manage those keys and remove the ones you don't want with a "custom mode" to only authorize your own bootloader that you could sign yourself if you really want to.

So what is Secure Boot protecting you from? It will protect your laptop from booting unsigned bootloaders (by the OS provider) with for instance injected malware.

What is Secure Boot not protecting you from?

- Secure Boot is not encrypting your disk and an adversary can still just remove the disk from your laptop and extract data from it using a different machine. Secure Boot is therefore useless without full disk encryption.
- Secure Boot is not protecting you from a signed bootloader that would be compromised and signed by the manufacturer itself (Microsoft for example in the case of Windows).
- Secure Boot will probably by default not protect your computer from running any OS that supports it (like Clonezilla) and had their bootloader signed by Microsoft or the Laptop manufacturer.
- Secure Boot can have flaws and exploits like any other system. If you're running an old laptop that doesn't benefit from new BIOS/UEFI updates, these can be left unfixed.

Additionally, there are number of attacks that could be possible against Secure Boot as explained (in depth) in these technical videos:

- Defcon 22, https://www.youtube.com/watch?v=QDSlWa9xQuA
- BlackHat 2016, https://www.youtube.com/watch?v=0fZdL3ufVOI

So it can be useful as an added measure against some adversaries but not all. Secure Boot in itself is not encrypting your hard drive. It's an added layer but that's it.

I recommend you keep it on if you can.

Mac:
Take a moment to set a firmware password according to the tutorial here: https://support.apple.com/en-au/HT204455

You should also enable firmware password reset protection (available from Catalina) according to the documentation here: https://support.apple.com/en-gb/guide/security/sec28382c9ca/web

This feature will mitigate the possibility for some adversaries to use hardware hacks to disable/bypass your firmware password. Note that this will also prevent Apple themselves from accessing the firmware in case of repair.

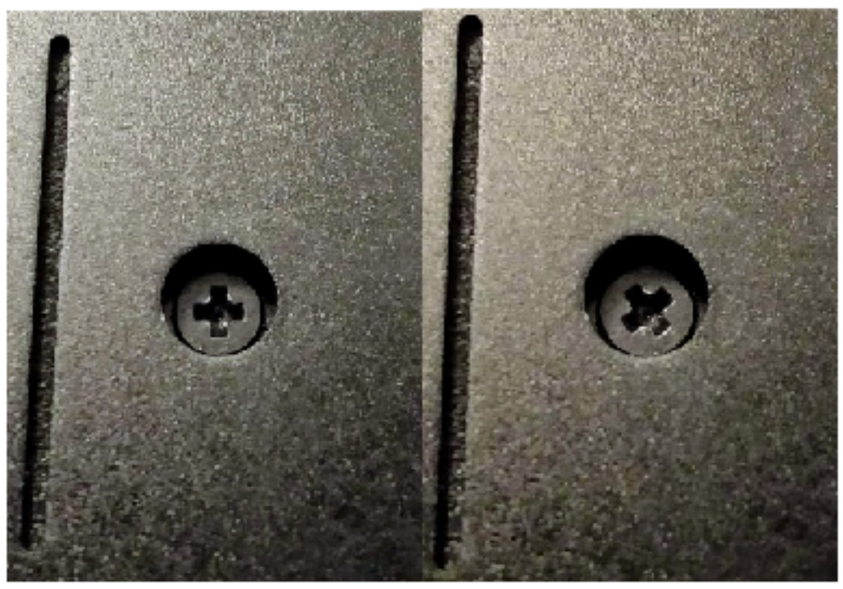
Physically Tamper protect your laptop:

At some point you'll inevitable leave this laptop alone somewhere. You won't sleep with it and take it everywhere every single day. You should make it has hard as possible for anyone to tamper with it without you noticing it. This is mostly useful against some limited adversaries that won't use a 5$ wrench against you[7].

It's important to know that it's trivially easy for some specialists to install a key logger in your laptop, or to just make a clone copy of your hard drive that could later allow them to detect the presence of encrypted data in it using forensic techniques (more on that later).

Here is a good cheap method to make your laptop tamper proof using Nail Polish (with glitter) https://mullvad.net/en/help/how-tamper-protect-laptop/ [210] (with pictures).

While this is a good cheap method, it could also raise suspicions as it's quite "noticeable" and might just reveal that you "have something to hide". So there are more subtle ways of achieving the same result. You could also for instance make a close macro photography of the back screws of your laptop or just use a very small amount of candle wax within one of the screws that could just look like usual dirt. You could then check for tampering by comparing the photographs of the screws with new ones. Their orientation might have changed a bit if your adversary wasn't careful when tightening them exactly the same way they were before. Or the wax within the bottom of a screw head might have been damaged compared to before.



Same techniques can be used with USB ports where you could just put a tiny amount of candle wax within the plug that would be damaged by inserting an USB key in it.

Check the laptop for tampering before using on a regular basis.

The Whonix route:

Picking your Host OS (the OS installed on your laptop):

This route will make extensive use of Virtual Machines[211], they'll require a host OS to run the Virtualization software. You have 3 recommended choices in this part of the guide:

- Your Linux distribution of choice (excluding Qubes OS)
- Windows 10 (preferably Home edition due to the absence of Bitlocker)
- MacOS

This guide will not really recommend using MacOS Big Sur nor M1 Macs as the host OS due to the “unblockable” telemetry[212, 213] issues. Big Sur on M1 maintains a persistent, hardware-serial-number linked TLS connection to Apple (for APNS[214], just like on all iOS devices) at all times when you are logged in, even if you don't use iCloud, App Store, iMessage, or FaceTime, and have all analytics turned off. There's no UI to disable this at the moment. This means that Apple possibly has the coarse location track log (due to geolocation of the client IP) for every M1 serial number. When you open the App Store app, that serial number is also sent, and associated with your Apple ID (email/phone) if you log in. Apple knows when you leave home, or arrive at the office, or travel to a different city, all with no Apple ID, no iCloud, and no location services.

In addition, changes are high that your Mac is or has been tied to an Apple account (at the time or purchase or after signing-in) and therefore its unique hardware identifiers could lead back to you in case of hardware identifiers leak.

Linux is also not necessarily the best choice for anonymity depending on your threat model. This is because using Windows will allow us to use Plausible Deniability[215] easily at the OS level. Windows is also unfortunately at the same time a privacy nightmare[216] but is the only (convenient) option for using OS wide plausible deniability. Windows telemetry and telemetry blocking is also widely documented which should mitigate many issues.

So what is Plausible Deniability? It’s the ability for you to cooperate with an adversary requesting access to your device/data without revealing your true secret. All this using Deniable Encryption[217].

A soft lawful adversary could ask for your encrypted laptop password. At first you could refuse to give out any password (using your “right to remain silent”, “right not to incriminate yourself”) but some countries are implementing laws[218, 219] to exempt this from such rights (because terrorists and “think of the children”). In that case you might have to reveal the password or maybe face jail time in contempt of court. This is where plausible deniability will comes into play.

You could then reveal a password but that password will only give access to “plausible data” (a decoy OS). The forensics will be well aware that it’s possible for you to have hidden data but should not be able to prove this **(if you do this right)**. You will have cooperated and the investigators will have access to something but not what you actually want to hide. Since the burden of proof should lie on their side, they will have no options but to believe you unless they have a proof that you have hidden data.

This feature can be used at the OS level (a plausible OS and a hidden OS) or at the files level where you will have an encrypted file container (similar to a zip file) where different files will be shown depending on the encryption password you use.

This also means you could set-up your own advanced “plausible deniability” setup using any Host OS by storing for instance Virtual Machines on a Veracrypt hidden volume container (be careful for traces in the Host OS tho that would need to be cleaned if the host OS is persistent, see Some additional measures against forensics: section later). There is a project for achieving this within TAILS (https://github.com/aforensics/HiddenVM) which would make your Host OS non persistent and use plausible deniability within TAILS.

In the case of Windows, plausible deniability is also the reason you should ideally have Windows 10 Home (and not Pro). This is because Windows 10 Pro natively offers a full-disk encryption system (Bitlocker[220]) where Windows 10 Home offers no full-disk encryption at all. We will later use a third-party open-source software for encryption that will allow full-disk encryption on Windows 10 Home. This will give you a good (plausible) excuse to use this software. While using this software on Windows 10 Pro would be suspicious.

Unfortunately, encryption is not magic and there are some risks involved:

Threats with encryption:

The 5$ Wrench:

Remember that encryption with or without plausible deniability is not a silver bullet and will be of little use in case of torture[7]. As a matter a fact, depending on who your adversary would be (your threat model), it might be wise not to use Veracrypt (formerly TrueCrypt) at all as shown in this demonstration: https://defuse.ca/truecrypt-plausible-deniability-useless-by-game-theory.htm

Plausible deniability is only effective against soft lawful adversaries that won't resort to physical means.

See https://en.wikipedia.org/wiki/Rubber-hose_cryptanalysis

Evil-Maid Attack:

Evil Maid Attacks[221] are conducted when someone tampers with your laptop while you're away. For install to clone your hard drive, install malware or a key logger. If they are able to clone your hard drive, they can compare one image of your hard drive at the time they took it while you were away with the hard drive when they seize it from you. If you used the laptop again in between, forensics examiners might be able to prove the existence of the hidden data by looking at the variations between the two images in what should be an empty/unused space. This could lead to strong evidence of the existence of a hidden data. If they install a key logger or malware within your laptop (software or hardware), they will be able to simply get the password from you for later use when they seize it. Such attacks can be done at your home, your hotel, a border crossing or anywhere you leave your devices unattended.

You can mitigate this attack by doing the following (as recommended earlier):

- Have a basic tamper protection (as explained previously) to prevent physical access to the internals of the laptop without your knowing. This will prevent them from cloning your disks and installing a physical key logger without your knowledge.
- Disable all the USB ports (as explained previously) within a password protected BIOS/UEFI. Again they won't be able to turn them on (without physically accessing the motherboard to reset the BIOS) to boot a USB device that could clone your hard drive or install a software based malware that could act as a key logger.
- Set-up BIOS/UEFI/Firmware passwords to prevent any unauthorized boot of an unauthorized device.
- Some OSes and Encryption software have anti-EvilMaid protection that can be enabled. This is the case with Windows/Veracrypt and QubeOS.

Cold-Boot Attack:

Cold Boot attacks[222] are trickier than the Evil Maid Attack but can be part of an Evil Maid attack as it requires an adversary to come into possession of your laptop while you're actively using your device or shortly afterward.

The idea is rather simple, as shown in this video[223], an adversary could theoretically quickly boot your device on a special USB key that would copy the content of the RAM (the memory) of the device after you shut it down. If the USB ports are disabled or if they feel like they need more time, they could open it and "cool down" the memory using a spray or other chemicals (liquid nitrogen for instance) preventing the memory decaying. They could then be able to copy its content for analysis. This memory dump could contain the key to decrypt your device. We will later apply a few principles to mitigate these.

In the case of Plausible Deniability, there have been some forensics studies[224] about technically proving the presence of the hidden data with a simple forensic examination (without a Cold Boot/Evil Maid Attack) but these have been contested by other studies[225] and by the maintainer of Veracrypt[226] so I wouldn't worry too much about those yet.

The same measures used to mitigate Evil Maid attacks should be in place for Cold Boot attacks with some added ones:

- If your OS or Encryption software allows it, you should consider encrypting the keys within RAM too (this is possible with Windows/Veracrypt and will be explained later)
- You should limit the use of Sleep stand-by and instead use Shutdown or Hibernate to prevent the encryption keys from staying in RAM when your computer goes to sleep. This is because sleep will maintain power to your memory for resuming your activity faster. Only hibernation and shutdown will actually clear the key from the memory[227].

See also https://www.whonix.org/wiki/Cold_Boot_Attack_Defense and https://www.whonix.org/wiki/Protection_Against_Physical_Attacks

Here are also some interesting tools to consider for Linux users to defend against these:

- https://github.com/0xPoly/Centry (unfortunately unmaintained it seems so I made a fork and pull request updating for Veracrypt https://github.com/AnonymousPlanet/Centry which should still work)

- https://github.com/hephaest0s/usbkill (unfortunately unmaintained as well it seems)
- https://github.com/Lvl4Sword/Killer
- https://askubuntu.com/questions/153245/how-to-wipe-ram-on-shutdown-prevent-cold-boot-attacks
- (Qubes OS, Intel CPU only) https://github.com/QubesOS/qubes-antievilmaid

About Sleep, Hibernation and Shutdown:

If you want the better security, you should shut down your laptop completely every time you leave it unattended or close the lid. This should clean and/or release the RAM and provide mitigations against cold boot attacks. However this can be a bit inconvenient as you'll have to reboot completely and type in a ton of passwords into various apps. Restart various VMs and other apps. So instead you could also use hibernation instead (not supported on Qubes OS). Since the whole disk is encrypted, hibernation in itself should not pose a large security risk but will still shutdown your laptop and clear the memory while allowing you to conveniently resume your work afterward. **What you should never do it use the standard sleep feature which will keep your computer on and the memory powered. This is an attack vector against evil-maid and cold-boot attacks discussed earlier. This is because your powered on memory holds the encryption keys to your disk (encrypted or not) and could then be accessed by a skilled adversary.**

This guide will provide guidance later on how to enable hibernation on various host OSes (except Qubes OS) if you don't want to shut down every time.

Local Data Leaks (traces) and forensics examination:

As mentioned briefly earlier, these are data leaks and traces from your operating system and apps when you perform any activity on your computer. These mostly apply to encrypted file containers (with or without plausible deniability) than OS wide encryption. Such leaks are less "important" if your whole OS is encrypted (if you're not compelled to reveal the password).

Let's say for example you have a Veracrypt encrypted USB key with plausible deniability enabled. Depending on the password you use when mounting the USB key, it will open a decoy folder or the sensitive folder. Within those folders, you'll have decoy documents/data within the decoy folder and sensitive documents/data within the sensitive folder.

In all cases, you'll (most likely) open these folders with Windows Explorer, MacOS Finder or any other utility and do whatever you planned to do. Maybe you'll edit a document within the sensitive folder. Maybe you'll search a document within the folder. Maybe you'll delete one or watch a sensitive video using VLC.

Well, all those Apps and your Operating System might keep logs and traces of that usage. This might include the full path of the folder/files/drives, the time those were accessed, temporary caches of those files, the "recent" lists in each apps, the file indexing system that could index the drive and even thumbnails that could be generated

Here are some examples of such leaks:

Windows:

- Windows ShellBags that are stored within the Windows Registry silently storing various histories of accessed volumes/files/folders[228].
- Windows Indexing keeping traces of the files present in your User folder by default[229].
- Recent lists (aka Jump Lists) in Windows and various apps keeping traces of recently accessed documents[230].
- Many more traces in various logs, please see this convenient interesting poster for more insight: https://www.sans.org/security-resources/posters/windows-forensic-analysis/170/download

MacOS:

- Gatekeeper[231] and XProtect keeping track of your download history in a local database and file attributes.
- Spotlight Indexing
- Recent lists in various apps keeping traces of recently accessed documents.
- Temporary folders keeping various traces of App usage and Document usage.
- MacOS Logs
- …

Linux:

- Tracker Indexing
- Bash History
- USB logs
- Recent lists in various apps keeping traces of recently accessed documents.
- Linux Logs
- …

Forensics could[225,228] use all those leaks[235] to prove the existence of hidden data and defeat your attempts at using plausible deniability and to find out about your various sensitive activities.

It will be therefore important to apply various steps to prevent forensics from doing this by preventing and cleaning these leaks/traces and more importantly by using whole disk encryption, virtualization and compartmentalization.

Forensics cannot extract local data leaks from an OS they can't access. And you will be able to clean most of those traces by wiping the drive or by securely erasing your virtual machines (which is not as easy as you think on SSD drives).

Some cleaning techniques will nevertheless be covered in the "Cover your Tracks" part of this guide at the very end.

Online Data Leaks:

Whether you're using simple encryption or plausible deniability encryption. Even if you covered your tracks on the computer itself. There is still a risk of online data leaks that could reveal the presence of hidden data.

Telemetry is your enemy. As explained earlier in this guide, the telemetry of Operating Systems but also from Apps can send staggering amounts of private information online.

In the case of Windows, this data could for instance be used to prove the existence of a hidden OS / Volume on a computer and would be readily available at Microsoft. Therefore it's critically important that you disable and block telemetry with all the means at your disposal. No matter what OS you're using.

Conclusion:

You should never conduct sensitive activities from a non-encrypted system. And even if it's encrypted, you should probably never conduct sensitive activities from the Host OS itself. Instead you should use a VM to be able to efficiently isolate and compartmentalize your activities and prevent local data leaks.

If you have little to no knowledge of Linux or if you want to use OS wide plausible deniability, I would recommend going for Windows (or back to the TAILS route) for convenience. This guide will help you hardening it as much as possible to prevent leaks. This guide will also help you hardening MacOS and Linux as much as possible to prevent similar leaks.

If you have no interest for OS wide plausible deniability and feel like learning to use Linux, I would strongly recommend going for Linux or the Qubes route if your hardware allows it.

In all cases, the host OS should never be used to conduct sensitive activities directly. The host OS will only be used to connect to a public Wi-Fi Access Point. It will be left unused while you conduct sensitive activities and should ideally not be used for any of your day to day activities.

Consider also reading **https://www.whonix.org/wiki/Full_Disk_Encryption#Encrypting_Whonix_VMs**

Linux Host OS:

As mentioned earlier, I do not recommend using your daily laptop for very sensitive activities. Or at least I do not recommending using your in-place OS for these. Doing that might result in unwanted data leaks that could be used to de-anonymize you. If you have a dedicated laptop for this, you should reinstall a fresh clean OS. If you don't want to wipe your laptop and start over, you should consider the TAILS route or proceed at your own risks.

I also recommend that you do the initial installation completely offline to avoid any data leak.

You should always remember that despite the reputation, Linux mainstream distributions (Ubuntu for instance) are not necessarily better at security than other systems such as MacOS and Windows. See this reference to understand why https://madaidans-insecurities.github.io/linux.html.

Full disk encryption:

There are two possibilities here with Ubuntu:

- (Recommended and easy) Encrypt as part of the installation process: https://ubuntu.com/tutorials/install-ubuntu-desktop
    - This process requires the full erasure of your entire drive (clean install).
    - Just check the "Encrypt the new Ubuntu installation for security)
- (Tedious but possible) Encrypt after installation: https://help.ubuntu.com/community/ManualFullSystemEncryption

For other distros, you'll have to document yourself but it will likely be similar. Encryption during install is just much easier in the context of this guide.

Reject/Disable any telemetry:

- During the install, just make sure you do not allow any data collection if prompted.
- If you're not sure, just make sure you didn't enable any telemetry and follow this tutorial if needed https://vitux.com/how-to-force-ubuntu-to-stop-collecting-your-data-from-your-pc/
- Any other distro: You'll need to document yourself and find out yourself how to disable telemetry if there is any.

Disable anything unnecessary:

- Disable Bluetooth if enabled by following this guide https://www.addictivetips.com/ubuntu-linux-tips/disable-bluetooth-in-ubuntu/ or issuing the following command:
    - sudo systemctl disable bluetooth.service --force
- Disable Indexing if enabled by default (Ubuntu >19.04) by following this guide https://www.linuxuprising.com/2019/07/how-to-completely-disable-tracker.html or issuing the following commands:
    - sudo systemctl --user mask tracker-store.service tracker-miner-fs.service tracker-miner-rss.service tracker-extract.service tracker-miner-apps.service tracker-writeback.service
        - You can safely ignore any error if it says some service does not exist
    - sudo tracker reset -hard

Hibernation:

As explained previously, you should not use the sleep features but shutdown or hibernate your laptop to mitigate some evil-maid and cold-boot attacks. Unfortunately this feature is disabled by default on many Linux distros including Ubuntu. It's possible to enable it but it might not work as expected. Follow this information at your own risk. If you don't want to do this, you should never use the sleep function and power off instead (and probably set the lid closing behavior to power off instead of sleep).

Follow this tutorial to enable Hibernate: https://help.ubuntu.com/16.04/ubuntu-help/power-hibernate.html

After Hibernate is enabled, change the behavior so that your laptop will hibernate when you close the lid by following this tutorial for Ubuntu 20.04 http://ubuntuhandbook.org/index.php/2020/05/lid-close-behavior-ubuntu-20-04/ and this tutorial for Ubuntu 18.04 https://tipsonubuntu.com/2018/04/28/change-lid-close-action-ubuntu-18-04-lts/

Unfortunately this will not clean the key from memory directly from memory when hibernating. To avoid this at the cost of some performance, you might consider encrypting the swap file by following this tutorial: https://help.ubuntu.com/community/EnableHibernateWithEncryptedSwap

These settings should mitigate cold boot attacks if you can hibernate fast enough.

Enable MAC address randomization:

- Ubuntu, follow these steps https://help.ubuntu.com/community/AnonymizingNetworkMACAddresses.
- Any other distro: you will have to find the documentation yourself but it should be quite similar to the Ubuntu tutorial.
- Consider this tutorial which should still work: https://josh.works/shell-script-basics-change-mac-address

Hardening Linux:

Please refer to this guide https://madaidans-insecurities.github.io/guides/linux-hardening.html

Setting up a safe Browser:

See Appendix G: Safe Browser

MacOS Host OS:

Note: At the moment, this guide will not work yet with new ARM M1 MacBooks. Due to Virtualbox not supporting this architecture yet. It could however be possible if you use commercial tools like VMWare or Parallels but those are not covered in this guide.

As mentioned earlier, I do not recommend using your daily laptop for very sensitive activities. Or at least I do not recommending using your in-place OS for these. Doing that might result in unwanted data leaks that could be used to de-anonymize you. If you have a dedicated laptop for this, you should reinstall a fresh clean OS. If you don't want to wipe your laptop and start over, you should consider the TAILS route or proceed at your own risks.

I also recommend that you do the initial installation completely offline to avoid any data leak.

Do not ever sign in with your Apple account using this Mac.

Last but not least, due to the issues mentioned earlier about Big Sur and telemetry, I do not recommend using Big Sur for now and recommend instead staying on MacOS Catalina until the situation is "clarified".

During the install:

- Stay Offline
- Disable all data sharing requests when prompted including location services
- Do not sign-in with Apple
- Do not enable Siri

Hardening MacOS:

For securing and hardening your MacOS, I recommend reading this GitHub guide which should cover many of the issues: https://github.com/drduh/macOS-Security-and-Privacy-Guide

Here are the basic steps you should take after your offline installation:

Enable Firmware password with "disable-reset-capability" option:

First you should set-up a firmware password following this guide from Apple: https://support.apple.com/en-us/HT204455

Unfortunately, some attacks are still possible and an adversary could disable this password so you should also follow this guide to prevent disabling the firmware password from anyone including Apple: https://support.apple.com/en-gb/guide/security/sec28382c9ca/web

Enable Hibernation instead of sleep:

Again this is to prevent some cold-boot and evil-maid attacks by powering down your RAM and cleaning the encryption key when you close the lid. You should always either hibernate or shutdown. On MacOS, the hibernate feature even has a special option to specifically clear the encryption key from memory when hibernating (while you might have to wait for the memory to decay on other Operating Systems). Once again there are no easy options to do this within the Settings so instead we'll have to do this by running a few commands to enable hibernation:

- Open a Terminal
- Run: "sudo pmset -a destroyfvkeyonstandby 1"
    - This command will instruct MacOS to destroy the Filevault key on Standby (sleep)

- Run: “sudo pmset -a hibernatemode 25”
    - This command will instruct MacOS to power off the memory during sleep instead of doing a hybrid hibernate that still keeps the memory powered on. It will result in slower wakes but will increase battery life.

Now when you close the lid of your MacBook, it should hibernate instead of sleep and mitigate attempts at performing cold-boot attacks.

In addition, you should also setup an automatic sleep (Settings > Energy) to that your MacBook will hibernate automatically if left unattended.

Disable unnecessary services:

Disable some unnecessary settings within the settings:

- Disable Bluetooth
- Disable the Camera and Microphone
- Disable Location Services
- Disable Airdrop
- Disable Indexing

Prevent Apple OCSP calls:

These are the infamous “unblockable telemetry” calls from MacOS Big Sur disclosed here: https://sneak.berlin/20201112/your-computer-isnt-yours/

You could block OCSP reporting by issuing the following command in Terminal:

- sudo sh -c 'echo "127.0.0.1 ocsp.apple.com" >> /etc/hosts'

But you should probably document yourself on the actual issue before acting. This page is a good place to start: https://blog.jacopo.io/en/post/apple-ocsp/

Up to you really. I would block it because I do not want any telemetry at all from my OS to the mothership without my specific consent. None.

Enable Full Disk encryption (Filevault):

You should enable full disk encryption on your Mac using Filevault according to this part of the guide: https://github.com/drduh/macOS-Security-and-Privacy-Guide#full-disk-encryption

Be careful when enabling. Do not store the recovery key at Apple if prompted (shouldn’t be an issue since you should be offline at this stage). You don’t want a third party to have your recovery key obviously.

MAC Address Randomization:

Unfortunately MacOS does not offer a native convenient way of randomizing your MAC Address and so you’ll have to do this manually. This will be reset at each reboot and you’ll have to re-do it each time to ensure you do not use your actual MAC Address when connecting to various Wi-Fis

You can do by issuing the following commands in terminal (without the parentheses):

- (Turn the Wi-Fi off) “networksetup -setairportpower en0 off”
- (Change the MAC Address) “sudo ifconfig en0 ether 88:63:11:11:11:11”
- (Turn the Wi-Fi back on) “networksetup -setairportpower en0 on”

Setting up a safe Browser:

See Appendix G: Safe Browser

Windows Host OS:

As mentioned earlier, I do not recommend using your daily laptop for very sensitive activities. Or at least I do not recommending using your in-place OS for these. Doing that might result in unwanted data leaks that could be used

to de-anonymize you. If you have a dedicated laptop for this, you should reinstall a fresh clean OS. If you don't want to wipe your laptop and start over, you should consider the TAILS route or proceed at your own risks.

I also recommend that you do the initial installation completely offline to avoid any data leak.

Installation:

You should follow Appendix A: Windows Installation

Enable MAC address randomization:

You should randomize your MAC address as explained earlier in this guide:

Go into Settings > Network & Internet > Wi-Fi > Enable Random hardware addresses

Alternatively, you could use this free piece of software: https://technitium.com/tmac/

Setting up a safe Browser:

See Appendix G: Safe Browser

Enable some additional privacy settings on your Host OS:

See Appendix B: Windows Additional Privacy Settings

Windows Host OS encryption:

Veracrypt[232] is the software I will recommend for full disk encryption, file encryption and plausible deniability. It is a fork of the well-known but deprecated and unmaintained TrueCrypt. It can be used for

- Full Disk simple encryption (your hard drive is encrypted with one passphrase).
- Full Disk encryption with plausible deniability (this means that depending on the passphrase entered at boot, you will either boot a decoy OS or a hidden OS).
- File container simple encryption (it's a large file that you will be able to mount within Veracrypt as if it was an external drive to store encrypted files within).
- File container with plausible deniability (it's the same large file but depending on the passphrase you use when mounting it, you will either mount a "hidden volume" or the "decoy volume").

It is to my knowledge the only (convenient and usable by anyone) free, open-source and openly audited[233] encryption software that also provides plausible deniability for general use and it works with Windows Home Edition.

You might be wondering why not BitLocker? Well here are a few reasons I prefer Veracrypt:

- Bitlocker is only available on Windows Pro and above (not on Windows Home) where Veracrypt works on all.
- Veracrypt supports more and stronger encryption algorithms.
- Veracrypt can be used on multiple platforms if needed where Bitlocker is limited to Windows
- Bitlocker is Microsoft proprietary closed-source when Veracrypt is audited open-source.
- Bitlocker does not support plausible deniability.

If you decide to use Bitlocker anyway, feel free to do so.

Go ahead and download and install Veracrypt from: https://www.veracrypt.fr/en/Downloads.html

After installation, please take a moment to review the following options that will help mitigate some attacks:

- Encrypt the memory with a Veracrypt option[234] (settings > performance/driver options > encrypt RAM) at a cost of 5-15% performance. This will also disable hibernation (which doesn't clear the key when hibernating) and instead encrypt the memory altogether to mitigate some cold-boot attacks.
- Enable the Veracrypt option to wipe the keys from memory if a new device is inserted (system > settings > security > clear keys from memory if a new device is inserted). This could help in case your system is seized while still on (but locked).

- Enable the Veracrypt option to mount volumes as removable volumes (Settings > Preferences > Mount volume as removable media). This will prevent Windows from writing some logs about your mounts in the Event logs[235] and prevent some local data leaks.
- Be careful and have a good situational awareness, if you sense something weird. Shut your laptop down as fast as possible.
- While Veracrypt newer versions do support Secure Boot, I would recommend disabling it from the BIOS as I prefer Veracrypt Anti-Evil Maid system over Secure Boot.

If you do not want to use encrypted memory (because performance might be an issue), you should at least enable hibernation instead of sleep. This will not clear the keys from memory (you are still vulnerable to cold boot attacks) but at least should mitigate them somewhat if your memory has enough time to decay.

Enable Hibernation (optional):

Again as explained earlier. You should never use the sleep feature to mitigate some cold-boot and evil-maid attacks. Instead you should Shut down or hibernate. You should therefore switch your laptop for sleeping to hibernating when closing the lid or when your laptop goes to sleep.

The reason is that Hibernation will actually shut down your laptop completely and clean the memory. Sleep on the other hand will leave the memory powered on (including your decryption key) and could leave your laptop vulnerable to cold-boot attacks.

By default, Windows 10 might not offer you this possibility so you should enable it by following this Microsoft tutorial: https://docs.microsoft.com/en-us/troubleshoot/windows-client/deployment/disable-and-re-enable-hibernation

- Open an administrator command prompt (right click on Command Prompt and "Run as Administrator")
- Run: powercfg.exe /hibernate on
- Now run the additional command: **powercfg /h /type full**
    - **This command will make sure your hibernate mode is full and will fully clean the memory (not securely tho).**

After that you should go into your power settings:

- Open the Control Panel
- Open System & Security
- Open Power Options
- Open "Choose what the power button does"
- Change everything from sleep to hibernate or shutdown
- Go back to the Power Options
- Select Change Plan Settings
- Select Advanced Power Settings
- Change all the Sleep Values for each Power Plan to 0 (Never)
- Make sure Hybrid Sleep is Off for each Power Plan
- Enable Hibernate After the time you'd like
- Disable all the Wake timers

Deciding which sub-route you'll take:

Now you'll have to pick your next step between two options:

- Route A: Simple encryption of your current OS
    - Pros:
        - Doesn't require you to wipe your laptop
        - No issue with local data leaks
        - Works fine with an SSD drive
        - Works with any OS
        - Simple

    - Cons:
        - You could be compelled by adversary to reveal your password and all your secrets and will have no plausible deniability.
        - Danger of Online data leaks
- Route B: Simple encryption of your current OS with later use of plausible deniability on files themselves:
    - Pros:
        - Doesn't require you to wipe your laptop
        - Works fine with an SSD drive
        - Works with any OS
        - Plausible deniability possible with "soft" adversaries
    - Cons:
        - Danger of Online Data leaks
        - Danger of Local Data leaks (that will lead to more work to clean up those leaks)
- Route C: Plausible Deniability Encryption of your Operating system (you'll have a "hidden OS" and a "decoy OS" running on the laptop):
    - Pros:
        - No issues with local Data leaks
        - Plausible deniability possible with "soft" adversaries
    - Cons:
        - Requires Windows (this feature is not supported on Linux).
        - Danger of online Data leaks
        - Requires full wipe of your laptop
        - No use with an SSD drive due to requirement of disabling Trim[236] Operations[237]. This will severely degrade the performance/health of your SSD drive over time.

As you can see, Route C really only offers two privacy advantages over the others and it will only be of use against a soft lawful adversary. Remember https://en.wikipedia.org/wiki/Rubber-hose_cryptanalysis.

Deciding which route you'll take is up to you. Route A is a minimum.

Always be sure to check for new versions of Veracrypt frequently to ensure you benefit from the latest patches. Especially check this before applying large Windows updates that might break the Veracrypt bootloader and send you into a boot loop.

NOTE THAT BY DEFAULT VERACRYPT WILL ALWAYS PROPOSE A SYSTEM PASSWORD IN QWERTY (display the password as a test). This can cause issues if your boot input is actually using your laptop's keyboard (AZERTY for example) as you'll have setup your password in QWERTY and will input it at boot time in AZERTY. So make sure you check when doing the test boot what keyboard layout your BIOS is using. You could fail to log-in just because the QWERTY/AZERTY mix-up. If your BIOS boots using AZERTY, you will need to type the password in QWERTY within Veracrypt.

Route A and B: Simple Encryption (Windows tutorial)

You don't have to have an HDD for this method and you don't need to disable Trim on this route. Trim leaks will only be of use to forensics in detecting the presence of a Hidden Volume but won't be of much use otherwise.

This route is rather straightforward and will just encrypt your current Operating System in place without losing any data. Be sure to read all the texts Veracrypt is showing you so you have a full understanding of what is going on.

- Launch VeraCrypt
- Go into Settings:
    - Settings > Performance/driver options > Encrypt RAM
    - System > Settings > Security > Clear keys from memory if a new device is inserted
    - System > Settings > Windows > Enable Secure Desktop
- Select System
- Select Encrypt System Partition/Drive
- Select Normal (Simple)

- Select Single-Boot
- Select AES as encryption Algorithm (click the test button if you want to compare the speeds)
- Select SHA-512 as hash Algorithm (because why not)
- Enter a strong passphrase (longer the better)[206]
- Collect some entropy by randomly moving your cursor around until the bar is full
- Click Next as the Generated Keys screen
- To rescue disk[238] or not rescue disk, well that's up to you. I recommend making one (just in case), just make sure to store it outside your encrypted drive (USB key for instance, or wait and see the end of this guide for guidance on safe backups). This rescue disk will not store your passphrase and you'll still need it to use it.
- Wipe mode:
    - If you have no sensitive data yet on this laptop, select None
    - If you have sensitive data on an SSD, Trim alone should take care of it[239] but I would recommend 1 pass (random data) just to be sure.
    - If you have sensitive data on an HDD, There is no Trim and I would recommend at least 1-pass.
- Test your setup. Veracrypt will now reboot your system to test the bootloader before encryption. This test has to pass in order for encryption to go forward.
- After your computer rebooted and the test is passed. You will be prompted by Veracrypt to start the encryption process.
- Start the encryption and wait for it to complete.
- You're done, skip Route B and go the next steps.

There will be another section on creating encrypted file containers with Plausible Deniability on Windows.

Route B: Plausible Deniability Encryption with a Hidden OS (Windows only)

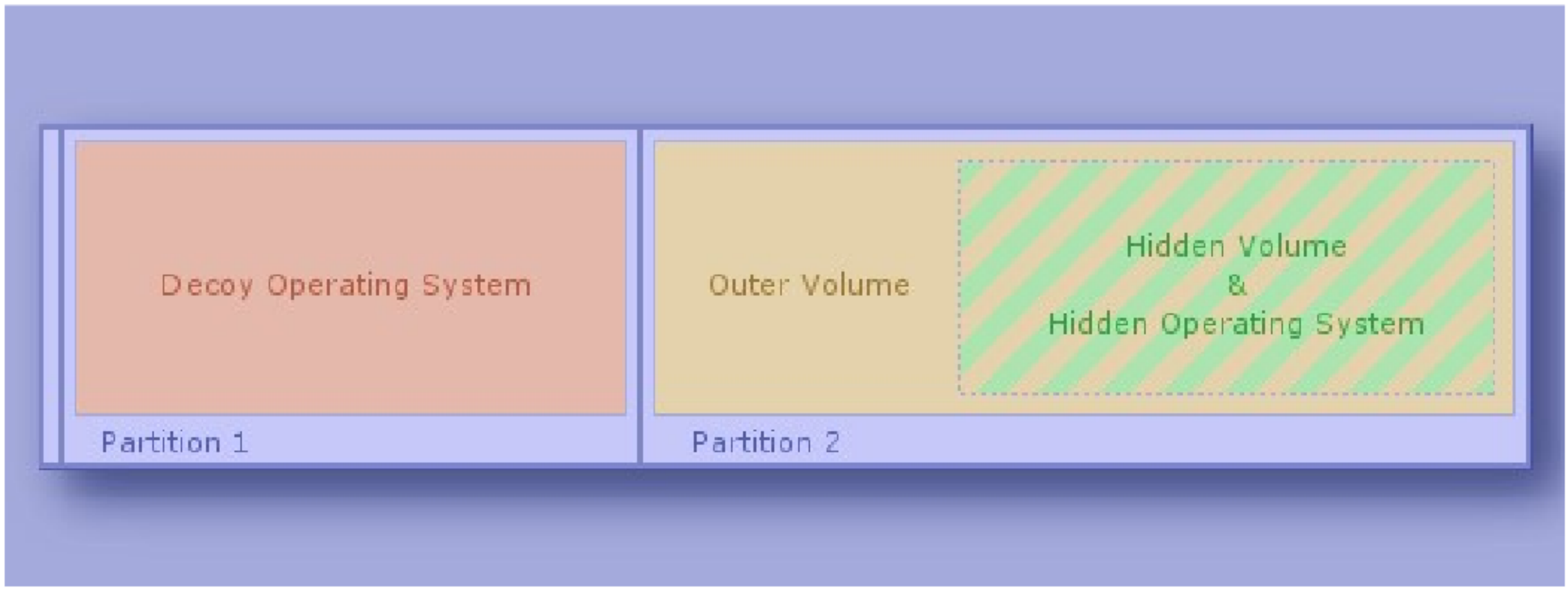
This is only supported on Windows.

This is only recommended on an HDD drive. This is not recommended on an SSD drive.

Your Hidden OS should not be activated (with a MS product key). Therefore this route will recommend and guide you through a full clean installation that will wipe everything on your laptop.

Read the Veracrypt Documentation https://www.veracrypt.fr/en/VeraCrypt%20Hidden%20Operating%20System.html (Process of Creation of Hidden Operating System part) and https://www.veracrypt.fr/en/Security%20Requirements%20for%20Hidden%20Volumes.html (Security Requirements and Precautions Pertaining to Hidden Volumes).

This is how your system will look after this process is done:



As you can see this process requires you to have two partitions on your hard drive from the start.

This process will do the following:

- Encrypt your second partition (the outer volume) that will look like an empty unformatted disk from the decoy OS.
- Prompt you with the opportunity to copy some decoy content within the outer volume.
    - This is where you will copy your decoy Anime/Porn collection from some external hard drive to the outer volume.
- Create a hidden volume within the outer volume of that second partition. This is where the hidden OS will reside.
- Clone your currently running Windows 10 installation onto the hidden volume.
- Wipe your currently running Windows 10.
- This means that your current Windows 10 will become the hidden Windows 10 and that you will need to reinstall a fresh decoy Windows 10 OS.

Mandatory if you have an SSD drive and you still want to do this against the recommendation: Disable SSD Trim in Windows[240] (again this is NOT recommended at all as disabling Trim in itself is highly suspicious).Also as mentioned earlier, disabling Trim will reduce the lifetime of your SSD drive and will significantly impact its performance over time (your laptop will become slower and slower over several months of use until it becomes almost unusable, you will then have to clean the drive and re-install everything). But you have to do it to prevent data leaks[241] that could allow forensics to defeat your plausible deniability[242][243]. The only way around this at the moment is to have a laptop with a classic HDD drive instead.

Step 1: Create a Windows 10 install USB key

See Appendix C: Windows Installation Media Creation and go with the USB key route.

Step 2: Boot the USB key and start the Windows 10 install process (Hidden OS)

- Insert the USB key into your laptop
- See Appendix A: Windows Installation and proceed with installing Windows 10 Home.

Step 3: Privacy Settings (Hidden OS)

See Appendix B: Windows Additional Privacy Settings

Step 4: Veracrypt installation and encryption process start (Hidden OS)

Remember to read https://www.veracrypt.fr/en/VeraCrypt%20Hidden%20Operating%20System.html

Do not connect this OS to your known Wi-Fi. You should download Veracrypt installer from a different computer and copy the installer here using an USB key.

- Install Veracrypt
- Start Veracrypt
- Go into Settings:
    - Settings > Performance/driver options > Encrypt RAM
    - System > Settings > Security > Clear keys from memory if a new device is inserted
    - System > Settings > Windows > Enable Secure Desktop
- Go into System and select Create Hidden Operating System
- Read all the prompts with thoroughly
- Select Single-Boot if prompted
- Create the Outer Volume using AES and SHA-512.
- Use all the space available on the second partition for the Outer Volume
- Use a strong passphrase[206]
- Select yes to Large Files
- Create some Entropy by moving the mouse around until the bar is full and select NTFS (do not select exFAT as we want this outer volume to look “normal” and NTFS is normal).
- Format the Outer Volume
- Open Outer Volume:

    - At this stage, you should copy decoy data onto the outer volume. So you should have some sensitive but not so sensitive files/folders to copy there. In case you need to reveal a password to this Volume. This is a good place for your Anime/Mp3/Movies/Porn collection.
    - I recommend you don't fill the outer volume too much or too little (about 40%). Remember you have to leave enough space for the Hidden OS (which will be same size as the first partition you created during installation).
- Use a strong passphrase[206] for the Hidden Volume (obviously a different one than the one for the Outer Volume).
- Now you will create the Hidden Volume, select AES and SHA-512
- Fill the entropy bar until the end with random mouse movements
- Format the hidden Volume
- Proceed with the Cloning
- Veracrypt will now restart and Clone the Windows where you started this process into the Hidden Volume. This Windows will become your Hidden OS.
- When the cloning is complete, Veracrypt will restart within the Hidden System
- Veracrypt will inform you that the Hidden System is now installed and then prompt you to wipe the Original OS (the one you installed previously with the USB key).
- Use 1-Pass Wipe and proceed.
- Now your Hidden OS will be installed, proceed to next step

Step 5: Reboot and boot the USB key and start the Windows 10 install process again (Decoy OS)

Now that the Hidden OS is fully installed, you will need to install a Decoy OS.

- Insert the USB key into your laptop
- See Appendix A: Windows Installation and proceed with installing Windows 10 Home again (do not Install a different version and stick with Home).

Step 6: Privacy settings (Decoy OS)

See Appendix B: Windows Additional Privacy Settings

Step 7: Veracrypt installation and encryption process start (Decoy OS)

Now we will encrypt the Decoy OS:

- Install Veracrypt
- Launch VeraCrypt
- Select System
- Select Encrypt System Partition/Drive
- Select Normal (Simple)
- Select Single-Boot
- Select AES as encryption Algorithm (click the test button if you want to compare the speeds)
- Select SHA-512 as hash Algorithm (because why not)
- Enter a short weak password (yes this is serious, do it, it will be explained later).
- Collect some entropy by randomly moving your cursor around until the bar is full
- Click Next as the Generated Keys screen
- To rescue disk[244] or not rescue disk, well that's up to you. I recommend making one (just in case), just make sure to store it outside your encrypted drive (USB key for instance, or wait and see the end of this guide for guidance on safe backups). This rescue disk will not store your passphrase and you'll still need it to use it.
- Wipe mode: Select 1-Pass just to be safe
- Pre-Test your setup. Veracrypt will now reboot your system to test the bootloader before encryption. This test has to pass in order for encryption to go forward.
- After your computer rebooted and the test is passed. You will be prompted by Veracrypt to start the encryption process.
- Start the encryption and wait for it to complete.
- Your Decoy OS is now ready for use.

Step 8: Test your setup (Boot in Both)

Time to test your setup.

- Reboot and input your Hidden OS passphrase, you should boot within the Hidden OS.
- Reboot and input your Decoy OS passphrase, you should boot within the Decoy OS.
- Launch Veracrypt on the Decoy OS and mount the second partition using the Outer Volume Passphrase (mount it as read-only, by going into Mount Options and Selecting Read-Only) and it should mount the second partition as a read-only displaying your decoy data (your Anime/Porn collection). You are mounting it as read-only now because if you were to write data on it, you could override content from your Hidden OS.

Step 9: Changing the decoy data on your Outer Volume safely

Before going to next step, you should learn the way to mount your Outer Volume safely for writing content on it. This is also explained in this official Veracrypt Documentation
https://www.veracrypt.fr/en/Protection%20of%20Hidden%20Volumes.html

You should do this from a safe trusted place.

Basically you're going to mount your Outer Volume while also providing the Hidden Volume passphrase within the Mount Options to protect the Hidden Volume from being overwritten. Veracrypt will then allow you write data to the Outer volume without risking overwriting any data on the Hidden Volume.

This operation will not actually mount the Hidden Volume and should prevent the creation of any forensic evidence that could lead to the discovery of the Hidden OS. However while you are performing this operation, both passwords will be stored in your RAM and therefore you could still be susceptible to a Cold-Boot Attack. To mitigate this, be sure to have the option to encrypt your RAM too.

- Open Veracrypt
- Select your Second Partition
- Click Mount
- Click Mount Options
- Check the "Protect the Hidden volume…" Option
- Enter the Hidden OS passphrase
- Click OK
- Enter your Outer Volume passphrase
- Click OK
- You should now be able to open and write to your Outer volume to change the content (copy/move/delete/edit…)

Step 10: Leave some forensics evidence of your outer Volume (with the decoy Data) within your Decoy OS

We have to make the Decoy OS as plausible as possible. We also want your adversary to think you're not that smart.

Therefore it's important to voluntarily leave some forensic evidence of your Decoy Content within your Decoy OS. This evidence will let forensic examiners see that you mounted your Outer Volume frequently to access its content.

Here are good tips to leave some forensics evidence:

- Play the content from the Outer Volume from your Decoy OS (using VLC for instance). Be sure to keep a history of those.
- Edit Documents and work in them.
- Enable File Indexing again on the Decoy OS and include the Mounted Outer Volume.
- Unmount it and Mount it frequently to watch some Content.
- Copy some Content from your Outer Volume to your Decoy OS and then delete it unsafely (just put it in the recycle Bin).
- Have a Torrent Client installed on the Decoy OS use it from time to time to Download some similar stuff that you will leave on the Decoy OS.
- You could have a VPN client installed on the Decoy OS with a known VPN of yours (non-cash paid).

Do not put anything suspicious on the Decoy OS such as:

- This guide
- Any links to this guide
- Any suspicious anonymity software such as Tor Browser

Notes:

Remember that you will need valid excuses for this plausible deniability scenario to work:

Take some time to read again the "Possible Explanations for Existence of Two Veracrypt Partitions on Single Drive" of the Veracrypt documentation here
https://www.veracrypt.fr/en/VeraCrypt%20Hidden%20Operating%20System.html

- **You are using Veracrypt because you are using Windows 10 Home which doesn't feature Bitlocker but still wanted Privacy.**
- **You have two Partitions because you wanted to separate the System and the Data for easy organization and because some Geek friend told you this was better for performance.**
- **You have used a weak password for easy convenient booting on the System and a Strong long passphrase on the Outer Volume because you were too lazy to type a strong passphrase at each boot.**
- **You encrypted the second Partition with a different password than the System because you don't want anyone in your entourage to see your stuff. And so you didn't want that data available to anyone.**

Be careful:

- **You should never mount the Hidden Volume from the Decoy OS (NEVER EVER). If you did this, it will create forensics evidence of the Hidden Volume within the Decoy OS that could jeopardize your attempt at plausible deniability**. If you did this anyway (intentionally or by mistake) from the Decoy OS, there are ways to erase forensics evidence that will be explained later at the end of this guide.
- **Never ever Use the Decoy OS from the same network (public Wi-Fi) as the Hidden OS.**
- **When you do mount the Outer Volume from the Decoy OS, do not write any Data within the Outer Volume as this could override what looks like Empty Space but is in fact your Hidden OS. You should always mount it as read-only.**
- **If you want to change the Decoy content of the Outer Volume, you should use a Live OS USB Key that will run Veracrypt.**
- **Note that you will not use the Hidden OS to perform sensitive activities in itself, this will be done later from a VM within the Hidden OS. The Hidden OS is only meant to protect you from a soft adversary that could gain access to your laptop and compel you to reveal your password.**
- **Be careful of any tampering with your laptop. Evil-Maid Attacks can reveal your hidden OS.**

Virtualbox on your Host OS:

This step and the following steps should be done from within the Host OS. This can either be your Host OS with simple encryption (Windows/Linux/MacOS) or your Hidden OS with plausible deniability (Windows only).

In this route, we will make extensive use of the free Oracle Virtualbox[245] software. This is a virtualization software in which you can create Virtual Machines that emulate a computer running a specific OS (if you want to use something else like Xen, Qemu, KVM or VMWARE, feel free to do so but this part of the guide covers Virtualbox only for convenience).

You should be aware that Virtualbox is not the best virtualization software in terms of security and some of the reported issues[246] have not be completely fixed to this date[247] and if you're using Linux with a bit more technical skills, you should consider using KVM instead by following the guide available at Whonix here
https://www.whonix.org/wiki/KVM

Some steps should be taken

All your sensitive activities will be done from within a client Virtual Machine running Windows 10 Pro (not Home this time), Linux or MacOS.

This has a few advantages that will greatly help you remain anonymous:

- It should prevent the guest VM OS (Windows/Linux/MacOS), Apps and any telemetry within the VMs from accessing your hardware directly. Even if your VM is compromised by malware, this malware should not be able to the VM and compromise your actual laptop.
- It will allow us to force all the network traffic from your client VM to run through another Gateway VM that will direct (torify) all the traffic towards the Tor Network. This is a network "kill switch". Your VM will lose its network connectivity completely and go offline if the other VM loses its connection to the Tor Network.
- The VM itself that only has internet connectivity through a Tor Network Gateway will connect to your cash-paid VPN service through Tor.
- DNS Leaks will be impossible because the VM is on an isolated network that has to go through Tor no matter what.

Pick your connectivity method:

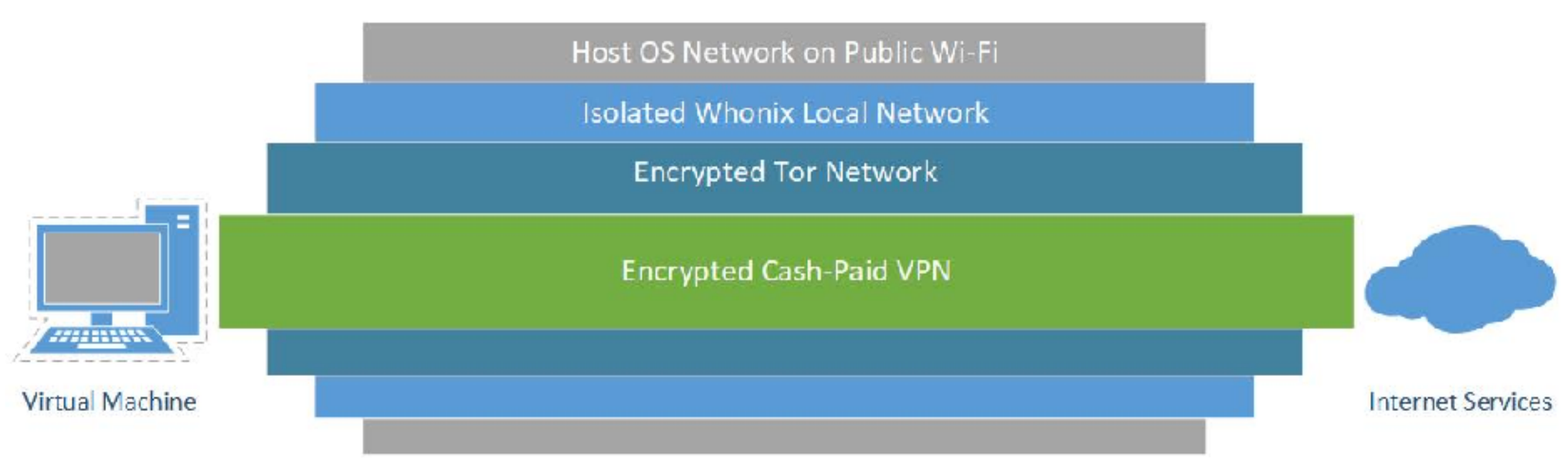
There are 7 possibilities within this route:

- Recommended and preferred:
  - Use VPN over Tor (User > Tor > VPN > Internet)
- Recommended
  - Use Tor alone (User > Tor > Internet)
  - Use VPN over Tor over VPN (User > VPN > Tor > VPN > Internet)
- Less recommended:
  - Use Tor over VPN (User > VPN > Tor > Internet)
- Not recommended and risky:
  - Use VPN alone (User > VPN > Internet)
  - Use VPN over VPN (User > VPN > VPN > Internet)
- **Not recommended and highly risky (but possible)**
  - No VPN and no Tor (User > Internet)

VPN over Tor:

This is the recommended and preferred method.

As you can see in this illustration, if your cash (preferred)/Monero paid VPN is compromised by an adversary (despite their privacy statement and no-logging policies), they will only find an anonymous cash/Monero paid VPN account connecting to their services from a Tor Exit node.



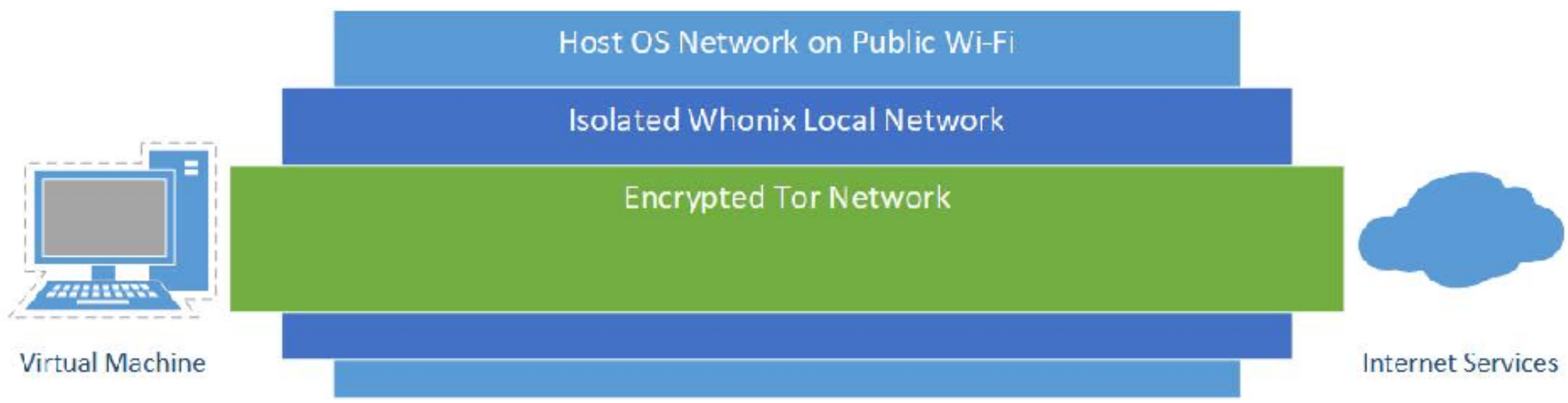
If an adversary somehow manages to compromise the Tor network too, they will only reveal the IP of a random public Wi-Fi that is not tied to your identity.

If an adversary somehow compromises your VM OS (with a malware or exploit for instance), they will be trapped within the internal Network of Whonix and should be unable to reveal the IP of the public Wi-Fi.

This solution brings the best of both worlds: the anonymity of the Tor Network combined with the convenience of VPN usage (less Captchas and issues accessing various services).

Tor only:

This other illustration shows the other possibility without a VPN. It's very similar but has one less defense layer. If the Tor network is compromised by a Global/State actor, then you will be likely de-anonymized faster.



IMHO while this option can be sufficient, you should still aim for a VPN over Tor if you can afford it.

Tor over VPN:

You might be wondering: Well what about using Tor over VPN instead of VPN over Tor? Well I would not recommend it because:

- Your VPN provider is just another ISP that will then know your origin IP and will be able to de-anonymize you if required. We don't trust them. I prefer a situation where your VPN provider doesn't know who you are.
- This would result in you connecting to various services using the IP of a Tor Exit Node which are banned/flagged in many places.
- **The only advantage really is that if you are in a hostile environment where Tor access is impossible/dangerous/suspicious but VPN is okay.**

In addition, if you're having issues accessing the Tor Network due to blocking/censorship, you could try using Tor Bridges (see Tor Documentation https://2019.www.torproject.org/docs/bridges and Whonix Documentation https://www.whonix.org/wiki/Bridges).

So if you have to use a VPN to access Tor, I would then recommend using another scheme instead:

VPN over Tor over VPN (User > VPN > Tor > VPN > Internet) using two cash/Monero paid VPNs instead. This means that you'll connect the Host OS to a first VPN from your Public Wi-Fi, then Whonix will connect to Tor and finally your VM will connect to a second VPN over Tor over VPN (see https://www.whonix.org/wiki/Tunnels/Connecting_to_a_VPN_before_Tor).

This will of course have a significant performance impact and might be quite slow but I think Tor is necessary somewhere for achieving reasonable anonymity.

Achieving this technically is easy within this route, you need two separate anonymous VPN accounts and have to connect to the first VPN from the Host OS and follow the route.

VPN only:

This route won't be explained nor recommended.

If you have the ability to use VPNs then you should be able to add a Tor layer over it. And if you can use Tor, then you can add an anonymous VPN over Tor to get the preferred solution.

Just using a VPN or even a VPN over VPN makes no sense as those can be traced back to you over time. One of the VPN providers will know your real origin IP (even if it's in a safe public space) and even if you add one over it, the second one will still know you were using that other first VPN service. This will only slightly delay your de-anonymization. Yes it's an added layer … but it's a persistent centralized added layer and you can be de-anonymized over time. This is just chaining 3 ISPs that are all subject to lawful requests.

For more info, please see the following references:

- https://www.whonix.org/wiki/Comparison_Of_Tor_with_CGI_Proxies,_Proxy_Chains,_and_VPN_Services#Tor_and_VPN_Services_Comparison

- https://www.whonix.org/wiki/Why_does_Whonix_use_Tor
- https://www.researchgate.net/publication/324251041_Anonymity_communication_VPN_and_Tor_a_comparative_study
- https://gist.github.com/joepie91/5a9909939e6ce7d09e29#file-vpn-md
- https://schub.wtf/blog/2019/04/08/very-precarious-narrative.html

In the context of this guide, Tor is required somewhere to achieve reasonable and safe anonymity and you should use it if you can.

No VPN/Tor:

If you can't use VPN nor Tor where you are, you probably are in a very hostile environment where surveillance and control is very high.

Just don't, it's not worth it and too risky IMHO. You can be de-anonymized almost instantly by any motivated adversary that could get to your physical location in a matter of minutes.

Don't forget to check back on Adversaries (threats): and Appendix S: Check your network for surveillance/censorship using OONI.

If you have absolutely no other option and absolutely still want to do something, see Appendix P: Accessing the internet as safely as possible when Tor/VPN is not an option **(at your own risk).**

Conclusion:

Unfortunately, using Tor alone will raise the suspicion of many platforms. You will face many hurdles (captchas, errors, difficulties signing-up) if you only use Tor.

For these reasons, I strongly recommend the first option using a cash (preferred)/Monero paid VPN over Tor solution.

This multi-layered approach should IMHO reduce the odds of your adversaries being able to de-anonymize you easily while providing better usability on most platforms.

For more information, you can also see the discussions here that could help make a decision yourself:

- Tor Project: https://gitlab.torproject.org/legacy/trac/-/wikis/doc/TorPlusVPN
- Tails Documentation:
  - https://gitlab.tails.boum.org/tails/blueprints/-/wikis/vpn_support/
  - https://tails.boum.org/support/faq/index.en.html#index20h2
- Whonix Documentation (in this order):
  - https://www.whonix.org/wiki/Tunnels/Introduction
  - https://www.whonix.org/wiki/Tunnels/Connecting_to_Tor_before_a_VPN
  - https://www.whonix.org/wiki/Tunnels/Connecting_to_a_VPN_before_Tor
- Some papers on the matter :
  - https://www.researchgate.net/publication/324251041_Anonymity_communication_VPN_and_Tor_a_comparative_study

Get an anonymous VPN:

Skip this step if you want to use Tor only or if VPN cannot be used in your environment.

See Appendix O: Get an anonymous VPN

Whonix:

Skip this step if you cannot use Tor (natively or through a VPN).

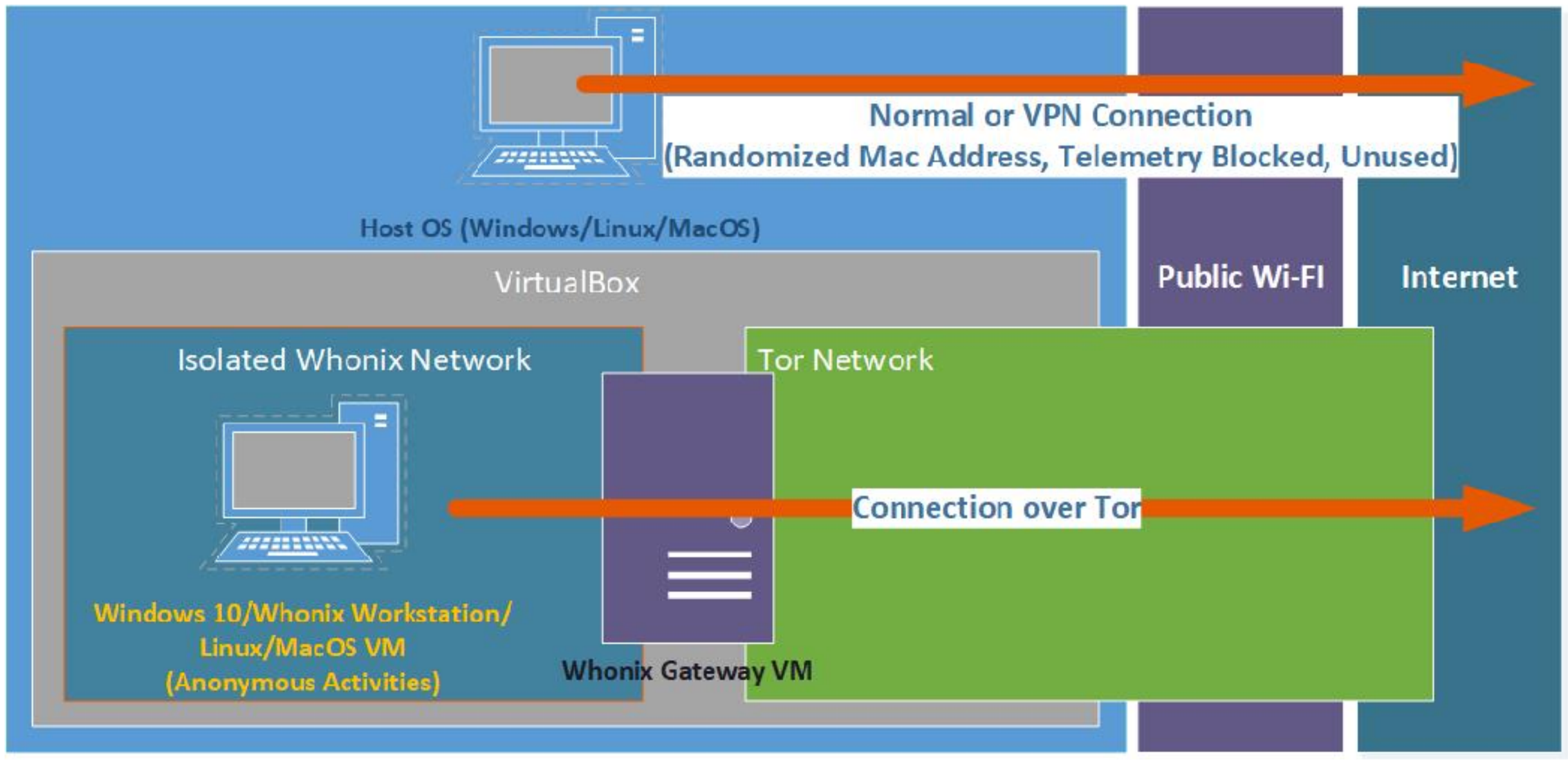
This route will use Virtualization and Whonix[248] as part of the anonymization process. Whonix is a Linux distribution composed of two Virtual Machines:

- The Whonix Workstation (this is a VM where you can conduct sensitive activities)

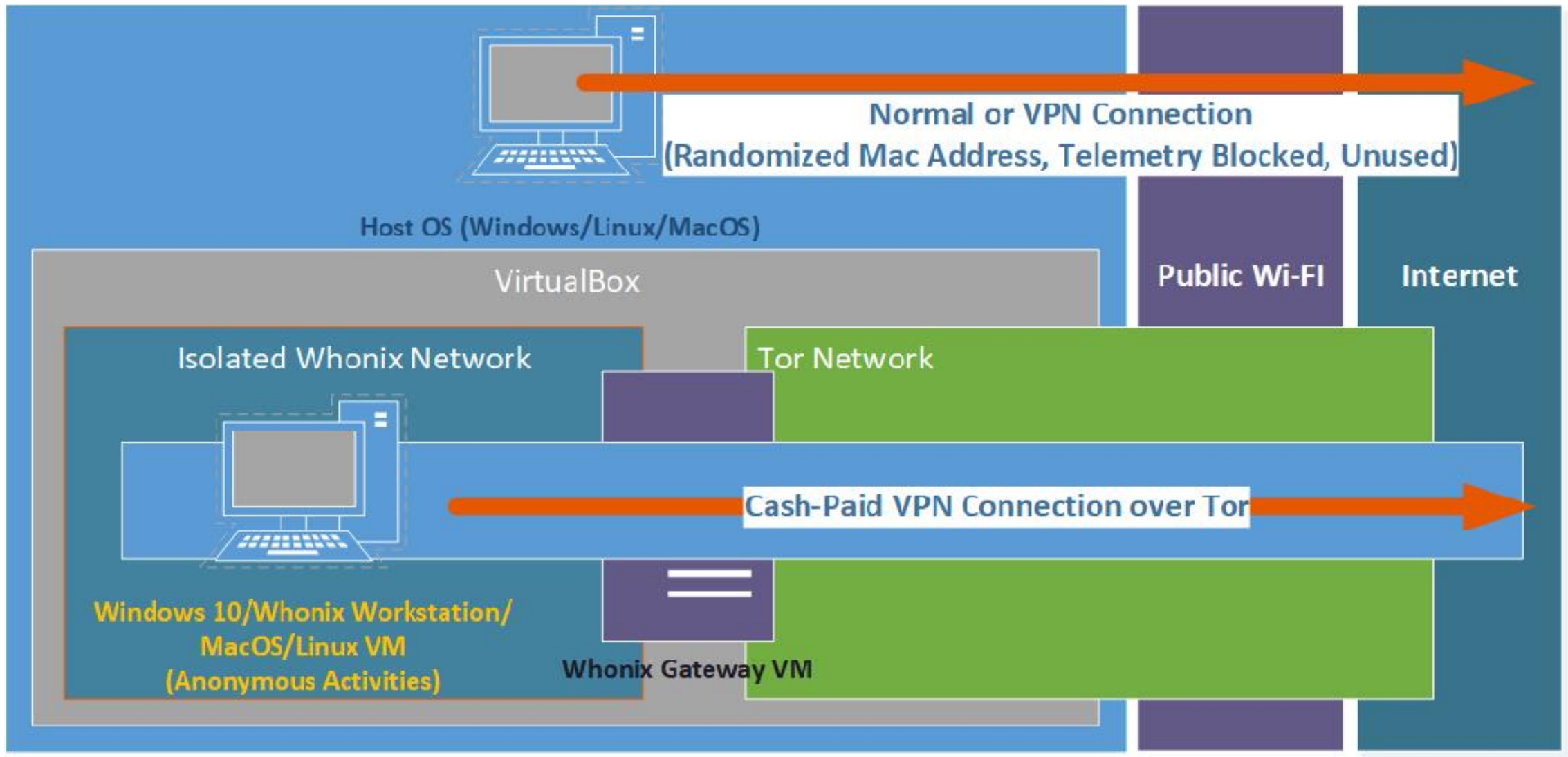
- The Whonix Gateway (this VM will establish a connection to the Tor network and route all the network traffic from the Workstation through the Tor network).

This guide will therefore propose 2 flavors of this route:

- The Whonix only route where all traffic is routed through the Tor Network (Tor Only or Tor over VPN).



- A Whonix hybrid route where all traffic is routed through a cash (preferred)/Monero paid VPN over the Tor Network (VPN over Tor or VPN over Tor over VPN).



You will be able to decide which flavor to use based on my recommendations. I recommend the second one as explained before.

Whonix is well maintained and has extensive and incredibly detailed documentation.

A note on Virtualbox Snapshots:

Later on you will create and run several Virtual Machines within Virtualbox for your sensitive activities. Virtualbox provides a feature called "Snapshots"[249] that allow for saving the state of a VM at any point in time. If for any reason later you want to go back to that state, you can restore that snapshot at any moment.

I strongly recommend that you do make use of this feature by creating a snapshot after the initial installation / update of each VM. This snapshot should be done before their use for any sensitive/anonymous activity.

This will allow you to turn your VMs into a kind of a disposable "Live Operating Systems" (like TAILS discussed earlier). Meaning that you will be able to erase all the traces of your activities within a VM by restoring a Snapshot to an earlier state. Of course this won't be "as good" as TAILS (where everything is stored in memory) as there might be traces of this activity left on your hard disk. Forensics studies have shown the ability to recover data from a reverted VM[250]. Fortunately there will be ways to remove those traces after deletion or reverting to a previous snapshot. Such techniques will be discussed in the Some additional measures against forensics: section of this guide.

Download Virtualbox and Whonix utilities:

You should download a few things within the host OS.

- The latest version of the Virtualbox installer according to your Host OS https://www.virtualbox.org/wiki/Downloads
- (Skip this if you cannot use Tor natively of through a VPN) The latest Whonix OVA file from https://www.whonix.org/wiki/Download according to your preference (Linux/Windows, with a Desktop interface XFCE for simplicity or only with the text-client for advanced users)

This will conclude the preparations and you should now be ready to start setting up the final environment that will protect your anonymity online.

Virtualbox Hardening recommendations:

For ideal security, you should follow the recommendations provided here for each Virtualbox Virtual Machine https://www.whonix.org/wiki/Virtualization_Platform_Security#VirtualBox_Hardening :

- Disable Audio.
- Do not enable Shared Folders.
- Do not enable video acceleration.
- Do not enable 3D acceleration.
- Do not enable the Serial Port.
- Remove the Floppy drive.
- Remove the CD/DVD drive.
- Do not enable the Remote Display server.
- Enable PAE/NX (NX is a security feature).
- Disable Advanced Configuration and Power Interface (ACPI).
- Do not attach USB devices.
- Disable the USB controller which is enabled by default. Set the Pointing Device to "PS/2 Mouse" or changes will revert.

Finally also follow this recommendation to desync the clock you're your VM compared to your host OS https://www.whonix.org/wiki/Network_Time_Synchronization#Spoof_the_Initial_Virtual_Hardware_Clock_Offset

This offset should be within a 60000 milliseconds range and should be different for each VM and here are some examples (which can be later applied to any VM):

- VBoxManage modifyvm "Whonix-Gateway-XFCE" --biossystemtimeoffset -35017
- VBoxManage modifyvm "Whonix-Gateway-XFCE" --biossystemtimeoffset +27931
- VBoxManage modifyvm "Whonix-Workstation-XFCE" --biossystemtimeoffset -35017
- VBoxManage modifyvm "Whonix-Workstation-XFCE" --biossystemtimeoffset +27931

Also consider applying these mitigations from VirtualBox to mitigate Spectre[251]/Meltdown[252] vulnerabilities by running this command from the VirtualBox Program Directory. All of these are described here: https://www.whonix.org/wiki/Spectre_Meltdown (be aware these can impact severely the performance of your VMs but should be done for best security).

Last but not least consider the security advice from Virtualbox themselves here https://www.virtualbox.org/manual/ch13.html

Tor over VPN:

Skip this step if you don't intend to use Tor over VPN.

If you intend to use Tor over VPN for any reason. You first have to configure a VPN service on your host OS.

Remember that in this case, I recommend having two VPN accounts. Both paid with cash/Monero (see Appendix O: Get an anonymous VPN). One will be used in the Host OS for the first VPN connection. The other could be used in the VM to achieve VPN over Tor over VPN (User > VPN > Tor > VPN).

If you intend to only use Tor over VPN, you only need one VPN account.

See Appendix R: Installing a VPN on your VM or Host OS. for instructions.

Whonix Virtual Machines:

Skip this step if you cannot use Tor because it's too dangerous/suspicious.

- Start Virtualbox on your Host OS.
- Import Whonix file Into Virtualbox following the instructions on https://www.whonix.org/wiki/VirtualBox/XFCE
- Start the Whonix VMs

Remember at this stage that if you're having issues connecting to Tor due to censorship or blocking, you should consider connecting using Bridges as explained in this tutorial https://www.whonix.org/wiki/Bridges.

- Update the Whonix VMs by following the instructions on https://www.whonix.org/wiki/Operating_System_Software_and_Updates#Updates
- Shutdown the Whonix VMs
- Take a Snapshot of the updated Whonix VMs within Virtualbox (select a VM and click the Take Snapshot button). More on that later.
- Go to next step

Important Note: You should also read these very good recommendations over there https://www.whonix.org/wiki/DoNot as most of those principles will also apply to this guide. You should also read their general documentation here https://www.whonix.org/wiki/Documentation which will also provide tons of advice similar to this guide.

Pick your guest workstation Virtual Machine:

Using Whonix/Linux will require more skills on your side as these are Linux distributions. You will also encounter more difficulties if you intend to use specific software that might be harder to use on Whonix/Linux. Setting up a VPN over Tor on Whonix will also be more complicated than on Windows as well.

If you can use Tor:

You can decide if you prefer to conduct your sensitive activities from the Whonix Workstation provided in the previous section **(highly recommended)** or from a Custom VM that will use the Whonix Gateway like the Whonix Workstation (less secure but might be required depending on what you intend to do).

If you cannot use Tor:

If you cannot use Tor, you can use a Custom VM of your choice that will ideally use the anonymous VPN if possible.

Linux Virtual Machine (Whonix or Linux):

*Whonix Workstation **(recommended and preferred)**:*

Skip this step if you cannot use Tor.

Just use the provided Whonix Workstation VM. **It's the safest and most secure way to go in this route.**

If you want additional software on the Workstation (such as another Browser), follow their guide here https://www.whonix.org/wiki/Install_Software

Consider running Whonix in Live Mode if for extra malware protection, See https://www.whonix.org/wiki/Anti-Forensics_Precautions

Don't forget to apply the VM hardening recommendations here: Virtualbox Hardening recommendations:

Linux (any distro):

Be careful, any customization you make to the Non-Whonix guest VMs (keyboard layout, language, time zone, screen resolution or other) could be used to fingerprint your VMs later. See https://www.whonix.org/wiki/VM_Fingerprinting

If you can use Tor (natively or over a VPN):

Use the Linux Distro of your choice. Personally I would recommend Ubuntu or Fedora for convenience but any other would work too. Be sure to not enable any telemetry.

Refer to this tutorial https://www.whonix.org/wiki/Other_Operating_Systems for detailed instructions.

If you cannot use Tor:

Use the Linux Distro of your choice. Personally I would recommend Ubuntu or Fedora for convenience but any other would work too. Be sure to not enable any telemetry.

Windows 10 Virtual Machine:

Be careful, any customization you make to the Non-Whonix guest VMs (keyboard layout, language, time zone, screen resolution or other) could be used to fingerprint your VMs later. See https://www.whonix.org/wiki/VM_Fingerprinting

Windows 10 ISO download:

See Appendix C: Windows Installation Media Creation and go with the ISO route.

If you can use Tor (natively or over a VPN):

Refer to this tutorial https://www.whonix.org/wiki/Other_Operating_Systems for detailed instructions.

Install:

- Shutdown the Whonix Gateway VM (this will prevent Windows from sending out telemetry and allow you to create a local account).
- Open Virtualbox
- Select Machine > New > Select Windows 10 64bit
- Allocate a minimum amount of 2048MB but ideally 4096MB if your Ram allows it
- Create a Virtual Disk using the VDI format and select Dynamically Allocated
- Keep the disk size at 50GB (this is a maximum, it won't reach that much)
- Select the VM and click Settings, Go into the Network Tab
- Select "Internal Network" in the "Attached to" Field and select Whonix.
- Go into the Storage Tab, Select the Empty CD and click the icon next to SATA Port 1
- Click on "Choose a disk file" and select the Windows ISO you previously downloaded
- Click ok and Start the VM
- Virtualbox will prompt you to select a Starting disk (the ISO file) , select it and click Start
- Follow the Steps in Appendix A: Windows Installation
- Start the Whonix Gateway VM

Network Settings:

- Go back into Settings then Network & Internet
- Click Properties (Below Ethernet)
- Edit IP settings:
- Enable IPv4 and set the following:
    - IP address 10.152.152.50 (increase this IP by 1 for any other VM)

    - Subnet prefix length 18 (255.255.192.0)
    - Gateway 10.152.152.10 (this is the Whonix Gateway)
    - DNS 10.152.152.10 (this is again the Whonix Gateway)
    - Save
- Windows will prompt you if you want to be "discoverable" on this network. Click NO.

Every time you will power on this VM in the future, make sure you change its Ethernet Mac Address before each boot. You can do this in Virtualbox > Settings > Network > Advanced > Click the refresh button next to the MAC address. You can only do this while the VM is powered off.

Choose a browser within the VM:

This time, I will recommend Brave browser instead of Tor Browser. The reasons are:

- You will encounter far less issues later with account creations (captchas …).
- You will enjoy ad-blocking where none is available in Tor Browser.
- The whole traffic will be routed over a VPN over Tor anyway.
- Performance is far better than Tor Browser

But again, if you are extra paranoid and want to use Tor Browser and have "Tor over VPN over Tor", you should skip this step and go with Tor Browser within the VM as well.

If you want to use Brave:

- Download and install Brave browser from https://brave.com/download/
- Open Brave Browser
- Go into Settings
- Go into Shields
- Set Trackers and Ads blocking to Aggressive
- Set Upgrade to HTTPS to enabled
- Set Fingerprinting blocking to Standard
- Go into Clear Browsing Data
- Select On Exit
- Check all options
- Do Not Enable Brave Rewards

If you cannot use Tor:

Install:

- Open Virtualbox
- Select Machine > New > Select Windows 10 64bit
- Allocate a minimum amount of 2048MB but ideally 4096MB if your Ram allows it
- Create a Virtual Disk using the VDI format and select Dynamically Allocated
- Keep the disk size at 50GB (this is a maximum, it won't reach that much)
- Go into the Storage Tab, Select the Empty CD and click the icon next to SATA Port 1
- Click on "Choose a disk file" and select the Windows ISO you previously downloaded
- Click ok and Start the VM
- Virtualbox will prompt you to select a Starting disk (the ISO file) , select it and click Start
- Follow the Steps in Appendix A: Windows Installation

Network Settings:

- Windows will prompt you if you want to be "discoverable" on this network. Click NO.

Every time you will power on this VM in the future, make sure you change its Ethernet Mac Address before each boot. You can do this in Virtualbox > Settings > Network > Advanced > Click the refresh button next to the MAC address. You can only do this while the VM is powered off.

Choose a browser within the VM:

This time, I will recommend Firefox or Brave browser. The reasons for preferring Brave are:

- You will encounter far less issues later with account creations (captchas …).
- You will enjoy native ad-blocking without added extensions

If you want to use Brave:

- Download and install Brave browser from https://brave.com/download/
- Open Brave Browser
- Go into Settings
- Go into Shields
- Set Trackers and Ads blocking to Aggressive
- Set Upgrade to HTTPS to enabled
- Set Fingerprinting blocking to Standard
- Go into Clear Browsing Data
- Select On Exit
- Check all options
- Do Not Enable Brave Rewards

Only use Private Windows.

Additional Privacy settings in Windows 10:

See Appendix B: Windows Additional Privacy Settings

MacOS Virtual Machine:

Yes you can actually run MacOS within Virtualbox (on Windows/Linux/MacOS host systems) if you really want to use MacOS. You can run any version of MacOS you want but I wouldn't recommend Big Sur due to their issues with telemetry (as stated earlier) and because it's significantly slower than Catalina on a VM in my experience.

Therefore I would personally recommend using MacOS Catalina instead which will just run faster and until this telemetry issue gets resolved.

If you can use Tor (natively or over a VPN):

During the following tutorials, before starting the MacOS VM, make sure you do put the MacOS VMs on the Whonix Network.

- Select the VM and click Settings, Go into the Network Tab
- Select "Internal Network" in the "Attached to" Field and select Whonix

Afterward, and during the install, you'll need to input an IP address manually to connect through the Whonix Gateway.

Use these settings when prompted in the MacOS installation process:

  - IP address 10.152.152.50 (increase this IP by 1 for any other VM)
  - Subnet prefix length 18 (255.255.192.0)
  - Gateway 10.152.152.10 (this is the Whonix Gateway)
  - DNS 10.152.152.10 (this is again the Whonix Gateway)

If you cannot use Tor:

Just use the tutorials as is.

Installation:

- Windows Host OS:
  - Virtualbox Catalina Tutorial: https://www.wikigain.com/install-macos-catalina-on-virtualbox-on-windows/

    - Virtualbox Big Sur Tutorial: https://www.wikigain.com/how-to-install-macos-big-sur-on-virtualbox-on-windows-pc/
- MacOS Host OS:
    - Just use the same tutorials as above but execute the various commands in terminal. It should work without issue.
- Linux Host OS:
    - Just use the same tutorials as above but execute the various commands in terminal. It should work without issue.

There are some drawbacks with running MacOS on Virtual Machines. The main one is that they don't actually have a serial number (0 by default) and you will be unable to log-in into any Apple provided service (iCloud, iMessage…) without a genuine ID. You can set such IDs using this script: https://github.com/myspaghetti/macos-virtualbox but keep in mind randomly generated IDs will not work and using the ID of someone else will definitely break their Terms of Services and could count as impersonation (and therefore could be illegal).

Note: I also ran in multiple issues with running these on AMD processors. This can be fixed so here is the configuration I used which worked fine with Catalina and Big Sur which will tell Virtualbox to emulate an Intel Processor instead:

- VBoxManage modifyvm "MacOSCatalina" –-cpuidset 00000001 000106e5 00100800 0098e3fd bfebfbff
- VBoxManage setextradata "MacOSCatalina" "VBoxInternal/Devices/efi/0/Config/DmiSystemProduct" "MacBookPro15,1"
- VBoxManage setextradata "MacOSCatalina" "VBoxInternal/Devices/efi/0/Config/DmiBoardProduct" "Mac-551B86E5744E2388"
- VBoxManage setextradata "MacOSCatalina" "VBoxInternal/Devices/smc/0/Config/DeviceKey" "ourhardworkbythesewordsguardedpleasedontsteal(c)AppleComputerInc"
- VBoxManage setextradata "MacOSCatalina" "VBoxInternal/Devices/smc/0/Config/GetKeyFromRealSMC" 1
- VBoxManage modifyvm "MacOSCatalina" --cpu-profile "Intel Core i7-6700K"
- VBoxManage setextradata "MacOSCatalina" VBoxInternal2/EfiGraphicsResolution 1920x1080

KeePassXC:

You will need something to store your data (logins/passwords, identities and TOTP[253] information).

For this purpose I strongly recommend KeePassXC because of their TOTP feature. This is the ability to create entries for 2FA[254] authentication with the authenticator feature.

Remember this should ideally be installed on your Guest VM and not on your Host OS. You should never do any sensitive activities from your Host OS.

Here are the tutorials:

- TAILS: KeePassXC is integrated by default
- Whonix: https://www.whonix.org/wiki/KeePassXC
- Linux:
    - Download from https://keepassxc.org/download/
    - Follow the tutorial here https://keepassxc.org/docs/KeePassXC_GettingStarted.html#_linux
- Windows:
    - Download from https://keepassxc.org/download/
    - Follow the tutorial here https://KeePassXC.org/docs/KeePassXC_GettingStarted.html#_microsoft_windows
- MacOS:
    - Download from https://keepassxc.org/download/
    - Follow the tutorial here https://keepassxc.org/docs/KeePassXC_GettingStarted.html#_macos

Test that KeePassXC is working before going to next step.

VPN client installation (cash/Monero paid):

If you decided to not use a cash-paid VPN and just want to use Tor, skip this step.

If you cannot use a VPN at all in a hostile environment, skip this step.

Otherwise, see Appendix R: Installing a VPN on your VM or Host OS. to install a VPN client on your client VM.

This should conclude the Route and you should now be ready.

Final steps:

Take a post-install VirtualBox snapshot of your VM.

You are done and can now skip the rest to go to the Getting Online: part.

The Qubes Route:

As they say on their own website, Qubes OS is a reasonably secure, free, open-source and security-oriented operating system for single-user desktop computing. Qubes OS leverages and extensively uses Xen-based virtualization to allow for the creation and management of isolated compartments called qubes.

Qubes OS is different from other Linux distributions because it will make extensive use of Virtualization and Compartmentalization so that any app will run in a different VM (qube). As a bonus, Qubes OS integrates Whonix by default and allows for increased privacy and anonymity. It's highly recommended that you document yourself over Qubes OS principles prior to going this route. Here are some recommended resources:

- Qube OS Introduction, https://www.qubes-os.org/intro/
- Qube OS Video Tours, https://www.qubes-os.org/video-tours/
- Qube OS Getting Started, https://www.qubes-os.org/doc/getting-started/
- YouTube, Life Behind the Tinfoil: A Look at Qubes and Copperhead - Konstantin Ryabitsev, The Linux Foundation https://www.youtube.com/watch?v=8cU4hQg6GvU
- YouTube, I used the reasonably-secure Qubes OS for 6 months and survived - Matty McFatty [@themattymcfatty] https://www.youtube.com/watch?v=sbN5Bz3v-uA
- YouTube, Qubes OS: How it works, and a demo of this VM-centric OS https://www.youtube.com/watch?v=YPAvoFsvSbg

This OS is recommended by prominent figures such as Edward Snowden and Privacytools.io.

In itself Qubes is the best option in this guide for people who are more comfortable with Linux and tech in general. But it has some downsides such as the lack of OS wide plausible deniability and its hardware requirements and hardware compatibilities. While you can run this on 4GB of RAM as per their requirements[255], the recommended RAM is 16GB. I would advise against using Qubes OS if you have less than 8GB of RAM.

The reason for this RAM requirement is that each app will run in a different VM and each of those VM will require and allocate a certain amount of memory that won't be available for other apps. If you're running native Windows apps within Qubes OS qubes, the ram overhead will be significant.

You should also check their hardware compatibility here https://www.qubes-os.org/hcl/ before proceeding. Your mileage might vary and you might experience several issues with regards to hardware compatibility that you will have to troubleshoot and solve yourself.

I think that if you can afford it and are comfortable with the idea of using Linux, you should go with this route as it is probably the best one in terms of security and privacy. The only disadvantage of this route is that it doesn't provide a way to enable OS wide plausible deniability[215] unlike the Whonix route.

Pick your connectivity method:

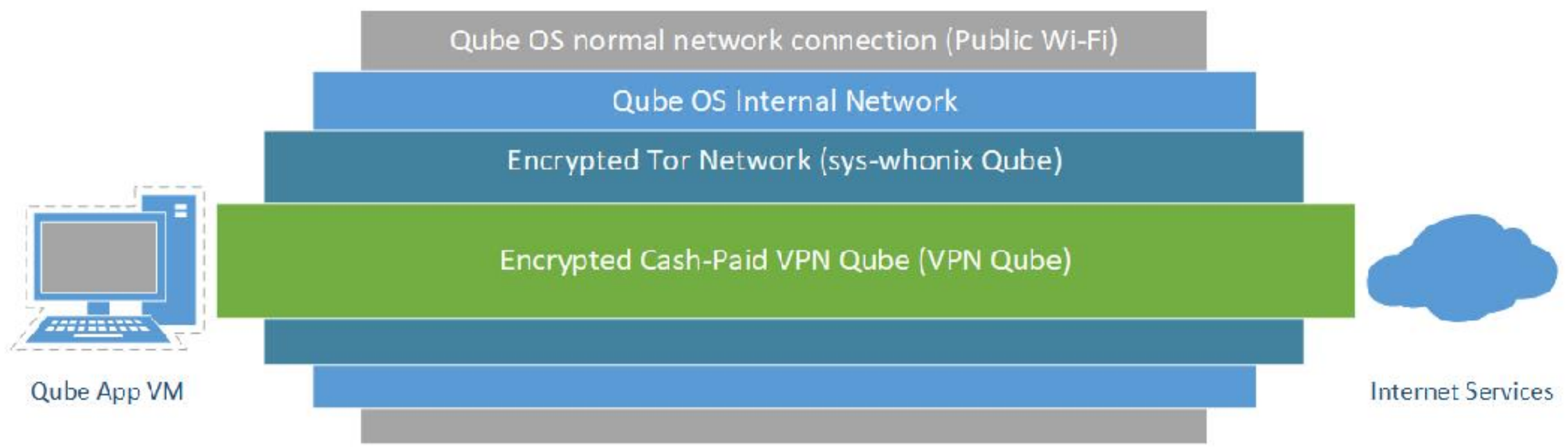
There are 7 possibilities within this route:

- Recommended and preferred:
  - Use VPN over Tor (User > Tor > VPN > Internet)
- Recommended

    - Use Tor alone (User > Tor > Internet)
    - Use VPN over Tor over VPN (User > VPN > Tor > VPN > Internet)
- Less recommended:
    - Use Tor over VPN (User > VPN > Tor > Internet)
- Not recommended and risky:
    - Use VPN alone (User > VPN > Internet)
    - Use VPN over VPN (User > VPN > VPN > Internet)
- **Not recommended and highly risky (but possible)**
    - No VPN and no Tor (User > Internet)

VPN over Tor:

As you can see in this illustration, if your cash (preferred)/Monero paid VPN is compromised by an adversary (despite their privacy statement and no-logging policies), an adversary will only find an anonymous cash (preferred)/Monero paid account connecting to their services from a Tor Exit node.



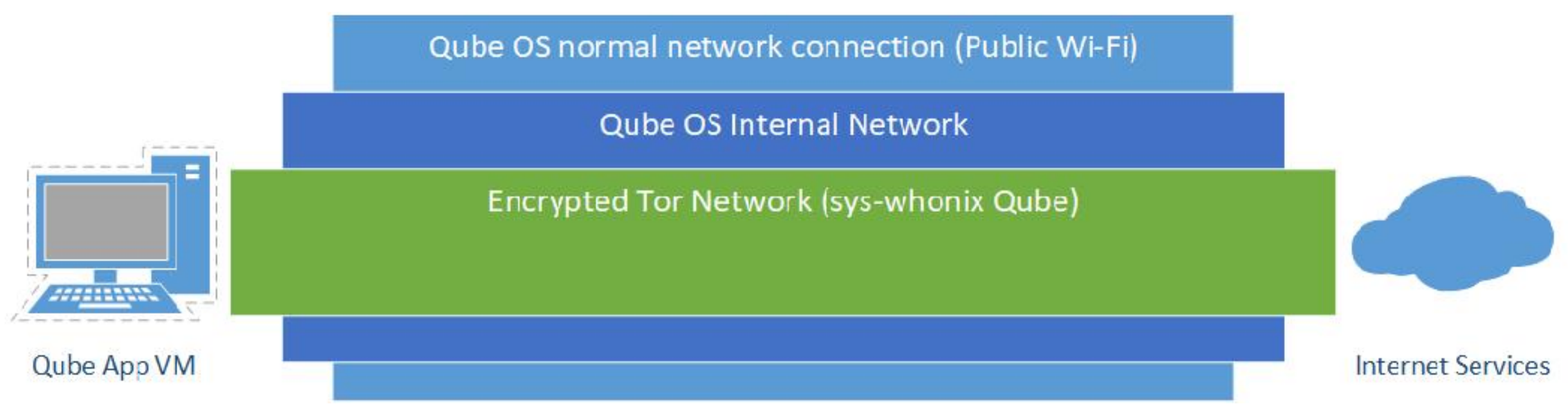
If they somehow also manage to compromise the Tor network, they will still only reveal the IP or a random public Wi-Fi that is not tied to your identity.

If they somehow compromise your Qube App VM (with a malware/exploit), they will be faced with an internal IP of Qube OS and shouldn't be able to figure out the Public Wi-Fi IP.

This solution brings the best of both worlds: the anonymity of the Tor Network combined with the convenience of VPN usage (less Captchas and issues accessing various services).

Tor only:

This other illustration shows the other possibility without a cash (preferred)/Monero paid VPN. It's very similar but has one less defense layer. If the Tor network is compromised by a Global/State actor, then you will be likely de-anonymized faster.



IMHO while this option can be sufficient, you should still aim for a VPN over Tor if you can afford it.

Tor over VPN:

You might be wondering: Well what about using Tor over VPN instead of VPN over Tor? Well I would not recommend it because:

- Your VPN provider is just another ISP that will then know your origin IP and will be able to de-anonymize you if required. We don't trust them.
- This would result in you connecting to various services using the IP of a Tor Exit Node which are banned/flagged in many places.
- **The only advantage really is that if you are in a hostile environment where Tor access is impossible/dangerous/suspicious but VPN is and you might have no other choice.**

In addition, if you're having issues accessing the Tor Network due to blocking/censorship, you could try using Tor Bridges (see Tor Documentation https://2019.www.torproject.org/docs/bridges and Whonix Documentation https://www.whonix.org/wiki/Bridges).

VPN only:

This route won't be explained nor recommended.

If you have the ability to use VPNs then you should be able to add a Tor layer over it. And if you can use Tor, then you can add an anonymous VPN over Tor to get the preferred solution.

Just using a VPN or even a VPN over VPN makes no sense as those can be traced back to you over time. One of the VPN providers will know your real origin IP (even if it's in a safe public space) and even if you add one over it, the second one will still know you were using that other first VPN service. This will only slightly delay your de-anonymization. Yes it's an added layer … but it's a persistent centralized added layer and you can be de-anonymized over time. This is just chaining 3 ISPs that are all subject to lawful requests.

For more info, please see the following references:

- https://www.whonix.org/wiki/Comparison_Of_Tor_with_CGI_Proxies,_Proxy_Chains,_and_VPN_Services#Tor_and_VPN_Services_Comparison
- https://www.whonix.org/wiki/Why_does_Whonix_use_Tor
- https://www.researchgate.net/publication/324251041_Anonymity_communication_VPN_and_Tor_a_comparative_study
- https://gist.github.com/joepie91/5a9909939e6ce7d09e29#file-vpn-md
- https://schub.wtf/blog/2019/04/08/very-precarious-narrative.html

In the context of this guide, Tor is required somewhere to achieve reasonable and safe anonymity and you should use it if you can.

No VPN/Tor:

If you can't use VPN nor Tor where you are, you probably are in a very hostile environment where surveillance and control is very high.

Just don't, it's not worth it and too risky IMHO. You can be de-anonymized almost instantly by any motivated adversary that could get to your physical location in a matter of minutes.

Don't forget to check back on Adversaries (threats): and Appendix S: Check your network for surveillance/censorship using OONI.

If you have absolutely no other option and absolutely still want to do something, see Appendix P: Accessing the internet as safely as possible when Tor/VPN is not an option **(at your own risk).**

Conclusion:

Unfortunately, using Tor alone will raise the suspicion of many platforms. You will face many hurdles (captchas, errors, difficulties signing-up) if you use Tor directly.

For these reasons, I recommend the first option using a cash (preferred)/Monero paid VPN over Tor solution.

This multi-layered approach should IMHO reduce the odds of your adversaries being able to de-anonymize you easily.

For more information, you can also see the discussions here that could help make a decision yourself:

- Tor Project: https://gitlab.torproject.org/legacy/trac/-/wikis/doc/TorPlusVPN
- Tails Documentation:
  - https://gitlab.tails.boum.org/tails/blueprints/-/wikis/vpn_support/
  - https://tails.boum.org/support/faq/index.en.html#index20h2
- Whonix Documentation (in this order):
  - https://www.whonix.org/wiki/Tunnels/Introduction
  - https://www.whonix.org/wiki/Tunnels/Connecting_to_Tor_before_a_VPN
- Some papers on the matter :
  - https://www.researchgate.net/publication/324251041_Anonymity_communication_VPN_and_Tor_a_comparative_study

Get an anonymous VPN:

Skip this step if you want to use Tor only or VPN is not an option.

See Appendix O: Get an anonymous VPN

Installation:

We will follow the instructions from their own guide https://www.qubes-os.org/doc/installation-guide/:

Secure Boot is not supported as per their FAQ: https://www.qubes-os.org/faq/#is-secure-boot-supported so it should be disabled in the BIOS/UEFI settings.

- Download the latest Qubes OS installation ISO according to their hardware compatibility list.
- Prepare an USB key with the Qubes OS ISO file
- Install Qubes OS according to the installation guide:
  - **If you want to use Tor or VPN over Tor: Check the "Enabling system and template updates over the Tor anonymity network using Whonix" during the last step. This will force all Qubes OS updates to go through Tor. While this will significantly reduce your update speed, it will increase your anonymity from the start.** (If you're having issues connecting to Tor due to censorship or blocking, consider using Tor Bridges as recommended earlier. Just follow the tutorial provided here: https://www.whonix.org/wiki/Bridges)
  - If you want to use Tor over VPN or can't use any of those, leave it unchecked.
- If you can't use Tor at all, there is also no point in installing Whonix. So you should disable Whonix installation within the Software Selection Menu.

Lid Closure Behavior:

Unfortunately Qubes OS does not support hibernation[256] which is IMHO an issue regarding cold-boot attacks. To mitigate those I highly recommend that you configure Qubes OS to shut down on any power action (power button, lid closure). You can do set this from the XFCE Power Manager. Do not use the sleep features.

Connect to a Public Wi-Fi:

Remember this should be done from a safe place (see Find some safe places with decent public WIFI: and Appendix Q: Using long range Antenna to connect to Public Wi-Fis from a safe distance:).

- In the upper right corner, Left click the network icon and note the Wi-Fi SSID you want to connect to
- Now right click the network icon and select Edit Connections
- Add one using the + sign
- Select Wi-Fi
- Enter the SSID of the desired network you noted before (if required)
- Select Cloned Mac Address
- Select Random to randomize your Mac Address
- Save
- Now again Left click the connection account and connect to the desired Wi-Fi
- If this is an Open Wi-Fi requiring registration: You will have to start a browser to register
  - After you are connected, Start a Disposable Fedora Firefox Browser

    - Go into the upper left Menu
    - Select Disposable, Fedora, Firefox
    - Open Firefox and register (anonymously) into the Wi-Fi

Update Qube OS:

After you are connected to a Wi-Fi you need to update Qube OS and Whonix. It's important that you keep Qube OS updated at all times before conducting any sensitive activities. Especially your Browser VMs. Normally, Qube OS will warn you about updates in the upper right corner with a gear icon. As this might take a while in this case due to using Tor, you can force the process by doing the following:

- Click the upper left Applications icon
- Select System Tools
- Select Qubes Update and Launch it
- Check the "Enable updates for qubes without known available updates"
- Select all the Qubes
- Click Next and update
- If you checked the Tor option during install, wait patiently as this might take a while over Tor

Setup the VPN ProxyVM:

Skip this step if you don't want to use a VPN and just use Tor only or if VPN is not an option either.

This tutorial should also work with any OpenVPN provider (Mullvad, IVPN or ProtonVPN for instance).

This is based on the tutorial provided by Qube OS themselves (https://github.com/Qubes-Community/Contents/blob/master/docs/configuration/vpn.md). If you are familiar with this process, you can follow their tutorial. Here is mine:

Create the ProxyVM:

- Click the Applications icon (upper left corner)
- Click Create Qubes VM
- Name and label as you wish: I suggest "VPNGatewayVM"
- Select Type: Standalone Qube copied from a template
- Select Template: debian-10
- Select Networking:
    - Select sys-whonix if you want to do VPN over Tor (recommended)
    - Select sys-firewall if you want to do Tor over VPN
- Advanced: Check provides network
- Check "Start qube automatically on boot"
- Create the VM
- Test your Connectivity:
    - If you're going for VPN over Tor, Test the VM connectivity to Tor by launching a Browser within the ProxyVM and going to https://check.torproject.org (It should say you're connected to Tor)
    - If you're going for Tor over VPN, Test the VM connectivity to the internet by launching a Browser within the ProxyVM and access any website.

Download the VPN configuration from your cash/Monero paid VPN provider:

If you can use Tor:

Using a Tor browser (be careful not to use any Clearnet Browser for this, use Tor Browser or Brave with a Tor Tab, double or triple-check), download the necessary OpenVPN configuration files for Linux from your VPN provider.

This can be done by using the Qubes OS integrated Tor Browser by accessing the Applications icon (upper left corner) and selecting the Disposable Tor Browser application.

If you cannot use Tor:

Launch a browser from a DisposableVM and download the necessary OpenVPN configuration files for Linux from your VPN provider.

When you are done downloading the configuration files within the Disposable Browser (usually a zip file), copy them to your ProxyVM VPN Gateway machine (using right click on the file and send to another AppVM).

Configure the ProxyVM:

- Click the upper left corner
- Select the VPN VM you just created
- Open the Files of the VPN VM
- Go into "Qubesincoming" > dispXXXX (This was your Disposable Browser VM)
- Double Click your downloaded zip file containing your OpenVPN configuration files to unzip it
- Now select the VPN VM again and start a terminal
- Install OpenVPN with the following command "sudo apt-get install openvpn"
- Copy all the OpenVPN configuration files provided by your VPN provider in /etc/openvpn/
- For all the OpenVPN configuration files (for each location):
    - Edit each file using "sudo nano configfile" (don't forget sudo to edit file within /etc)
    - Change the protocol from "udp" to "tcp" (Tor does not support UDP)
    - Change the port to a supported (by your VPN provider) TCP port (like 80 or 443)
    - Save and Exit each file
- Edit the OpenVPN config file (/etc/default/openvpn) by typing "sudo nano /etc/default/openvpn" (because I don't like vi editor)
    - Change #AUTOSTART="all" to AUTOSTART="all" (in other words, remove the "#")
    - Save and Exit
- Edit the Qubes firewall rules file (/rw/config/qubes-firewall-user-script) by typing "sudo nano /rw/config/qubes-firewall-user-script"
    - Add the following lines (without the remarks in parentheses)
        - virtualif=10.137.0.17

          (This is the IP of the ProxyVM, this is not dynamic and you might need to change it at reboot)

        - vpndns1=10.8.0.1

          (This is the first DNS server of your VPN provider, it shouldn't change)

        - vpndns2=10.14.0.1

          (This is the second DNS server of your VPN provider, it shouldn't change)

        - iptables -F OUTPUT
        - iptables -I FORWARD -o eth0 -j DROP
        - iptables -I FORWARD -i eth0 -j DROP
        - ip6tables -I FORWARD -o eth0 -j DROP
        - ip6tables -I FORWARD -i eth0 -j DROP

          (These will block outbound traffic when the VPN is down, it's a kill switch, more information here https://linuxconfig.org/how-to-create-a-vpn-killswitch-using-iptables-on-linux)

        - iptables -A OUTPUT -d 10.8.0.1 -j ACCEPT
        - iptables -A OUTPUT -d 10.14.0.1 -j ACCEPT

          (These will allow DNS request to your VPN provider DNS to resolve the name of the VPN servers in the OpenVPN configuration files)

        - iptables -F PR-QBS -t nat
        - iptables -A PR-QBS -t nat -d $virtualif -p udp --dport 53 -j DNAT --to $vpndns1
        - iptables -A PR-QBS -t nat -d $virtualif -p tcp --dport 53 -j DNAT --to $vpndns1
        - iptables -A PR-QBS -t nat -d $virtualif -p udp --dport 53 -j DNAT --to $vpndns2
        - iptables -A PR-QBS -t nat -d $virtualif -p tcp --dport 53 -j DNAT --to $vpndns2

          (These will redirect all DNS requests from the ProxyVM to the VPN provider DNS servers)

- Restart the ProxyVM by typing “sudo reboot”
- Test the ProxyVM VPN connectivity by starting a Browser within it and going to your VPN provider test page. It should now say you’re connected to a VPN:
    - Mullvad: https://mullvad.net/en/check/
    - IVPN: https://www.ivpn.net/ (check the top banner)
    - ProtonVPN: Follow their instructions here https://protonvpn.com/support/vpn-ip-change/

VPN over Tor:

Setup a disposable Browser Qube for VPN over Tor use:

- Within the Applications Menu (upper left corner), Select the Disposable Fedora VM
- Go into Qube Settings
- Click Clone qube and name it (like “VPNoverTor”)
- Again within the Application Menu, Select the Clone you just created
- Go into Qube Settings
- Change the Networking to your ProxyVPN created earlier
- Click OK
- Start a Browser within the Whonix Workstation
- Check that you have VPN connectivity and it should work

You should now have a Disposable Browser VM that works with your cash/Monero paid VPN over Tor.

Tor Over VPN:

Reconfigure your Whonix Gateway VM to use your ProxyVM as NetVM instead of sys-firewall.

- Within the Applications Menu (upper left corner), Select the sys-whonix VM.
- Go into Qube Settings
- Change the Networking NetVM to your ProxyVPN created earlier instead of sys-firewall
- Click OK
- Create a Whonix Workstation Disposable VM (follow this tutorial https://www.whonix.org/wiki/Qubes/DisposableVM)
- Launch a browser from the VM and Check that you have VPN connectivity and it should work.

Alternatively you can also create any other type of disposable VM (but probably less secure than the Whonix one):

- Within the Applications Menu (upper left corner), Select the Disposable Fedora VM
- Go into Qube Settings
- Click Clone qube and name it (like “TorOverVPN”)
- Again within the Application Menu, Select the Clone you just created
- Go into Qube Settings
- Change the Networking to your sys-whonix created earlier
- Click OK
- Start a Browser within the VM
- Check that you have VPN connectivity and it should work

You should now have a Disposable Browser VM that works with Tor over a cash/Monero paid VPN.

Any other combination? (VPN over Tor over VPN for instance)

By now you should understand how easy it is to route traffic from one VM to the other with Qubes.

You can create several ProxyVMs for VPN accesses and keep the Whonix one for Tor. You just need to change the NetVM settings of the various VMs to change the layout.

You could have:

- One VPN ProxyVM for the base Qube OS connection
- Use the sys-whonix VM (Whonix Gateway) getting its network from the first ProxyVM

- A second VPN ProxyVM getting network from sys-whonix
- Disposable VMs getting their NetVM from the second ProxyVM

This would result in User > VPN > Tor > VPN > Internet (VPN over Tor over VPN). Experiment for yourself. Qubes OS is great for these things.

Setup Brave Browser within Qube OS (optional but recommended):

Fedora Disposable VM:

This guide recommends using Brave Browser as an option because it will significantly reduce your usability issues when creating anonymous online identities. This will be explained later.

- Within the Applications Menu (upper left), Select the Fedora-30 template
- Go into Qube Settings
- Clone the VM and name it fedora-30-brave (this VM template will have Brave)
- Again go into the Applications Menu and select the clone you just created
- Go into Qube Settings
- Change its network to the ProxyVPN and Apply
- Launch a terminal from the VM

Apply the instructions from https://brave.com/linux/ (Fedora 28+ section) and run the following commands:

- sudo dnf install dnf-plugins-core
- sudo dnf config-manager --add-repo https://brave-browser-rpm-release.s3.brave.com/x86_64/
- sudo rpm --import https://brave-browser-rpm-release.s3.brave.com/brave-core.asc
- sudo dnf install brave-browser

Whonix Disposable VM:

Edit the Whonix Disposable VM template and follow instructions here
https://www.whonix.org/wiki/Install_Software

KeePassXC:

You will need something to store your data (logins/passwords, identities and TOTP[257] information).

For this purpose I strongly recommend KeePassXC because of their TOTP feature. This is the ability to create entries for 2FA[258] authentication with the authenticator feature.

In the context of Qube OS you should probably store your sensitive information within the Domain-vault qube.

- First click the Applications icon (upper left) and select the Domain: Vault qube.
- Click Qubes Settings
- Temporarily enable network by changing the network to your VPN ProxyVM you created earlier
- Open a terminal within the Domain: Vault qube
- Type: “sudo dnf install keepassxc” and wait for it to install
- Close the terminal and disable network by changing back the network to (none)
- Go back into the Domain: Vault Qube Settings and into the Applications tab
- Click Refresh
- Add KeePassXC to the Selected tab
- Launch KeePassXC within the Domain: Vault qube

You are done and can now skip the rest to go to the “Creating your anonymous online identities” part.

Creating your anonymous online identities:

Understanding the methods used to prevent anonymity and verify identity:

Captchas:

(Illustration by xkcd.com, licensed under CC BY-NC 2.5)

Captcha[259] stands for "Completely Automated Public Turing test to tell Computers and Humans Apart" are Turing tests[260] puzzles you need to complete before accessing a form/website. You'll mostly encounter those provided by Google (reCaptcha service[261]) and Cloudflare (hCaptcha[262]). hCaptcha is used on 15% of the internet by their own metrics[263].

They're designed to separate bots from humans but in reality are also used to deter anonymous and private users.

If you frequently use VPNs or Tor, you'll quickly encounter many captchas everywhere[264]. Quite often when using Tor, even if you succeed in solving all the puzzles, you'll still be denied after solving the puzzles.

While most people think those puzzles are only about solving a little puzzle, it's important to understand that it's much more complex and that modern Captchas uses advanced machine learning and risk analysis algorithms to check if you're human[265]:

- They check your browser, cookies and browsing history using Browser fingerprinting[266].
- They track your cursor movements (speed, accuracy) and use algorithms to determine if it's "human".
- They track your behavior before/during/after the tests to ensure you're "human"[267].

It's also very likely that those platforms could already reliably identify you based on the unique way you interact with those puzzles. This could work despite obfuscation of your IP address / Browser and clearing all cookies.

You will often experience several in a row (sometimes endless[268]) and sometimes very difficult ones involving reading undecipherable characters or identifying various objects on endless pictures set. You'll also have more captchas if you use ad blocking system or if your account was flagged for any reason for using VPNs or Tor previously.

You'll also have (in my experience) more Captchas (reCaptcha) in Google if you don't use Chrome. But this can be mitigated by using Chromium based browsers such as Brave. There is also a Browser extension called Buster that could help you those https://github.com/dessant/buster.

As for Cloudflare (hCaptcha), you could also use their Accessibility solution here (https://www.hcaptcha.com/accessibility) which would allow you to sign-up (with your anonymous identity created later) and set a cookie within your Browser that would allow you to bypass their captchas. Another solution to mitigate hCaptcha would be to use their own solution called "Privacy Pass"[269] https://privacypass.github.io/ in the form of a Browser extension you could install in your VM Browser.

You should therefore deal with those carefully and force yourself to alter the way you're solving them (speed/movement/accuracy/…) as to prevent "Captcha Fingerprinting".

Fortunately as far as I'm aware, these are not yet officially/publicly used to de-anonymize users for third parties.

Phone verification:

Phone verification is advertised by most platforms as a way to verify you're human. But don't be fooled, the main reason for phone verification is not only to check if you're human but also to be able to de-anonymize you if needed.

Most platforms (including the privacy oriented ones such as Signal/Telegram/ProtonMail will require a phone number to register and most countries now make it mandatory to submit a proof of ID to register[270].

E-Mail verification:

E-Mail verification is what used to be enough but is not anymore in most cases. What is important to know is that open e-mail providers (disposable e-mail providers for instance) are flagged as much as open proxies (like Tor).

Most platforms will not allow you to register using an "anonymous" or disposable e-mail. As they won't allow you to register using an IP address from the Tor network.

The key thing to this is that it is becoming increasingly difficult to sign-up for a free e-mail account anywhere without providing (you guessed it) … a mobile phone number. That same mobile phone number that can be used conveniently to track you down in most places.

User details checking:

Obviously, Reddit doesn't do this (yet) but Facebook most likely does and will look for "suspicious" things in your details (which could include face recognition).

Some examples:

- IP address from a country different than your profile country?
- Age in the profile not matching the picture age?
- Ethnicity in the profile not matching the picture ethnicity?
- Language not matching the country language?
- Details unknowns in their contact discovery service? (Meaning nobody else knows you?)
- Locking down privacy settings after signing-up?
- Name that doesn't match the correct ethnicity/language/country?

Proof of ID verification:

The deal-breaker in most cases. As far as I know, only Facebook and LinkedIn (outside of financial services) have requested such verifications which involves sending pictures of some form of identification (passport, national ID card, driver license …). The only way to do this would involve creating fake official documents (forgery) using some decent Photoshop skills and this might be illegal in most places.

Therefore, this is a line I'm not going to help you cross within this guide. Some services are offering such services online but I think they most likely are *bad actors* and are most likely overstepping their boundaries.

In many countries, only law enforcement, some very specific processes (such as GDPR request) and some well regulated financial services are authorized to request a proof of identification. So the legality of asking such documents is debatable and I think such platforms should not be allowed to require those.

In few countries (like Germany), this practice is illegal and online platforms such as Facebook or LinkedIn are legally bound to allow you use a pseudonym and remain anonymous.

IP Filters:

As stated before in this guide, many platforms will apply filters on the IPs of the users. Tor exit nodes are publicly listed and VPN exit servers are "well known". There are many commercial and free services providing the ability to block those IPs with ease (hi Cloudflare).

Many platforms operators and administrators do not want traffic from these IPs as they often drive a lot of unlawful/malicious/unprofitable traffic to their platforms. Usually using the same excuses:

- Unlawful because "Think of the children" or "Terrorists".
- Malicious because "Russian trolls".
- Unprofitable because "Well it's noise in the data we sell to advertisers" (AdSense, Facebook Ads …). Yet we still pay traffic for them so let's just deny them all instead.

Fortunately, those systems are not "perfect' and you will (still) be able to get around those restrictions by switching identities (in the case of Tor) and looking trying to access the website each time until you find an Exit Node that is not blacklisted (yet).

Sometimes some platforms will allow you to log-in with a Tor IP but not sign-up. Obviously those platforms will keep a convenient permanent log of the IP you used during sign-up. And some will keep such logs indefinitely including all the IPs you used to logging in (hi Facebook).

The tolerance is much higher with VPNs as they're not considered "open proxies" but that won't stop many platforms from making them hard to use by forcing increasingly difficult captchas on most VPN users.

For this reason, this guide recommends the use of VPN over Tor (and not Tor over VPN).

Browser and Device Fingerprinting:

Browser and Device[271] Fingerprinting are usually integrated into the Captcha services but also in other various services.

Many platforms (like Google[272]) will check your browser for various capabilities and settings and block Browsers they don't like. This is one of the reasons I recommend using Brave Browser over Tor Browser within your VM.

Here are some of the things they check within recent browsers:

- User Agent: This is your Browser name and Version.
- HTTP_ACCEPT Headers: This is the type of content your Browser is able to handle.
- Time Zone and Time Zone Offset: Your time zone.
- Screen Size and Color Depth: The resolution of your screen.
- System Fonts: The typing fonts installed on your system.
- Cookies support: If your Browser supports cookies or not.
- Hash of Canvas fingerprint and Hash of WebGL fingerprint: These are generated unique IDs based on your graphic rendering capabilities.
- WebGL Vendor & Renderer: Name of your Video card
- Do-Not-Track enabled or not: Well yes they can use your DNT information to track you
- Language: The language of your Browser
- Platform: The Operating System you're using
- Touch Support: If your system supports touch (such as a phone/tablet or touchscreen enabled laptop)
- Ad Blocking use: If your browser block ads

- AudioContext fingerprint: Similar to the Canvas and WebGL fingerprints these will fingerprint your audio capabilities.
- CPU: What kind of CPU you're using and how many of them
- Memory: How much memory you have in your System
- Browser Permissions: Is your browser allowing some things like geolocation or microphone/webcam access.
- …

Here are two services you can use to check your browser Fingerprinting:

- https://coveryourtracks.eff.org/
- https://amiunique.org

Chances are you'll find your browser fingerprint unique no matter what you do.

Human interaction:

Some platforms will add this as a bonus step and require you to have an actual human interaction with a customer care representative. Usually by e-mail but sometimes by chat/phone. They'll want to verify that you exist by asking you to reply to an e-mail/chat/phone call.

It's annoying but very easy to deal with in our case. We're not making bots. This guide is for humans making human accounts.

User Moderation:

Many platforms will delegate and rely on their own users to moderate the others and their content. These are the "report" features that you will find on most platforms.

Getting reported thousands of times doesn't matter when you're Donald Trump or Kim Kardashian but if you as a sole "friendless" anonymous user gets reported even once, you might get suspended/flagged/banned instantly.

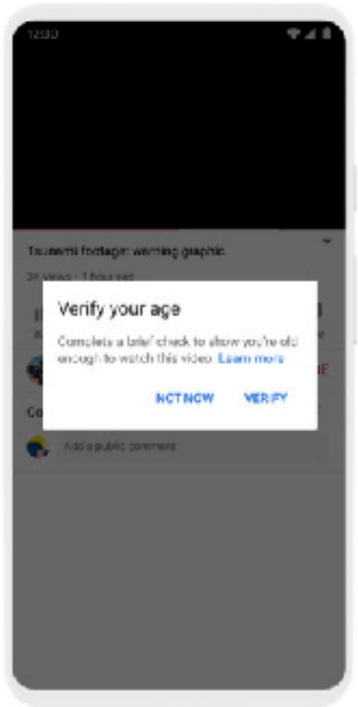
Behavioral Analysis:

This is the part where you should watch the documentary "The Social Dilemma"[273] on Netflix as they cover this topic much better than anyone else IMHO.

Financial transactions:

Simple and efficient, some platforms will require than you perform some kind of financial transaction to verify your account sometimes under the pretext of verifying your age. This could be a credit card verification or a very small amount bank wire. Some will accept a donation in a main crypto like Bitcoin or Ethereum.

This is for instance now the case on YouTube for some European Users[274].



Again this is just a way for them to delegate the user verification and identification on the financial system.

Sign-in with some platform:

Why do this user verification ourselves when we can just ask others to deal with it?

You'll notice this and you probably already encountered this. Some apps/platforms will ask/require you to sign-in with a well-known and well-used reputable platform instead of their own system (Sign-in with Google/Facebook/Apple/Twitter).

This option is often presented as the "default one", hiding away the "Sign-in with e-mail and password" with clever Dark Patterns[275] and unfortunately sometimes required.

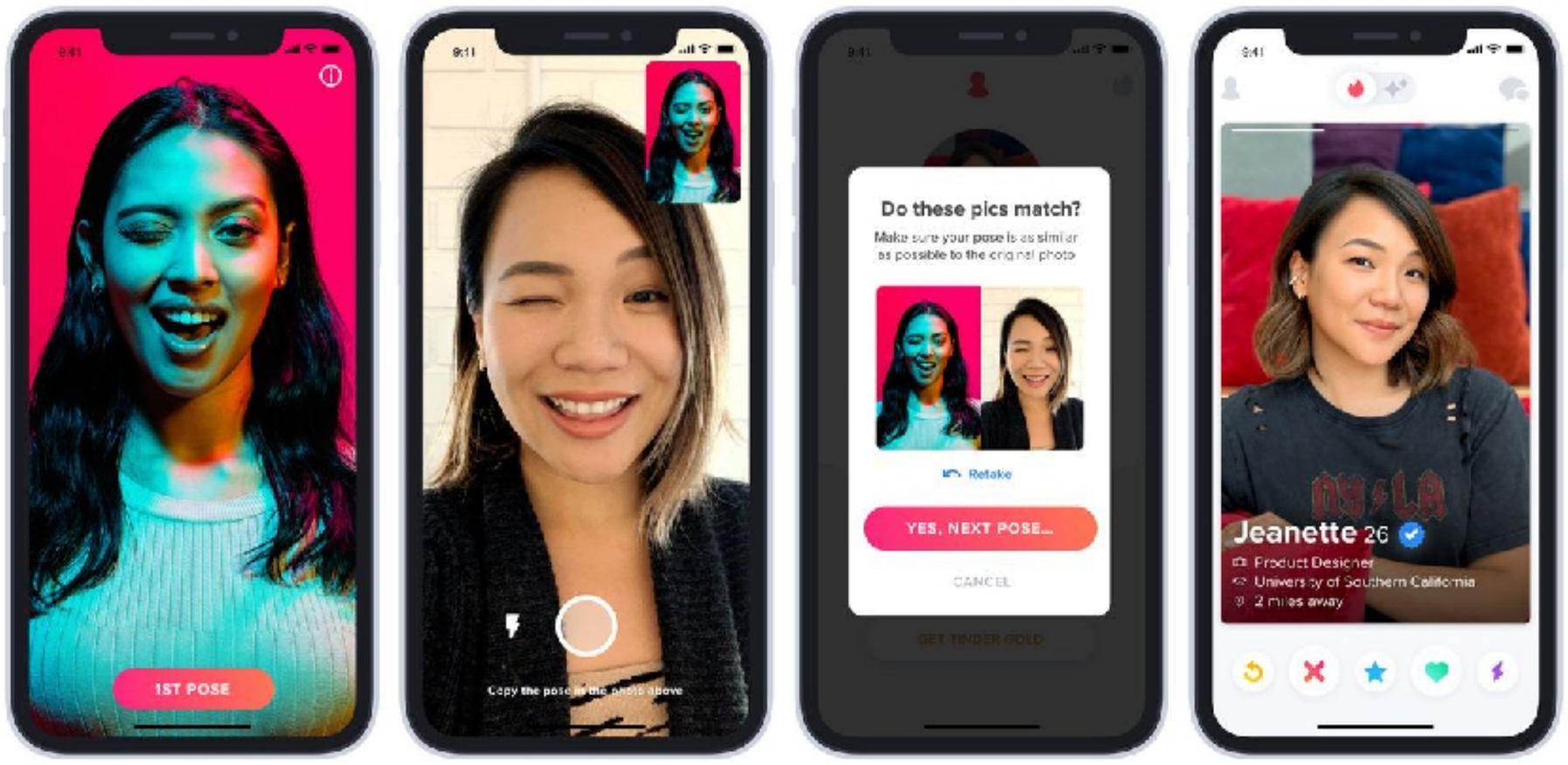
This method will delegate the verification process on those platforms instead assuming that you won't be able to create an anonymous Google/Facebook/Apple/Twitter account with ease.

Fortunately it is still possible to this day do create those.

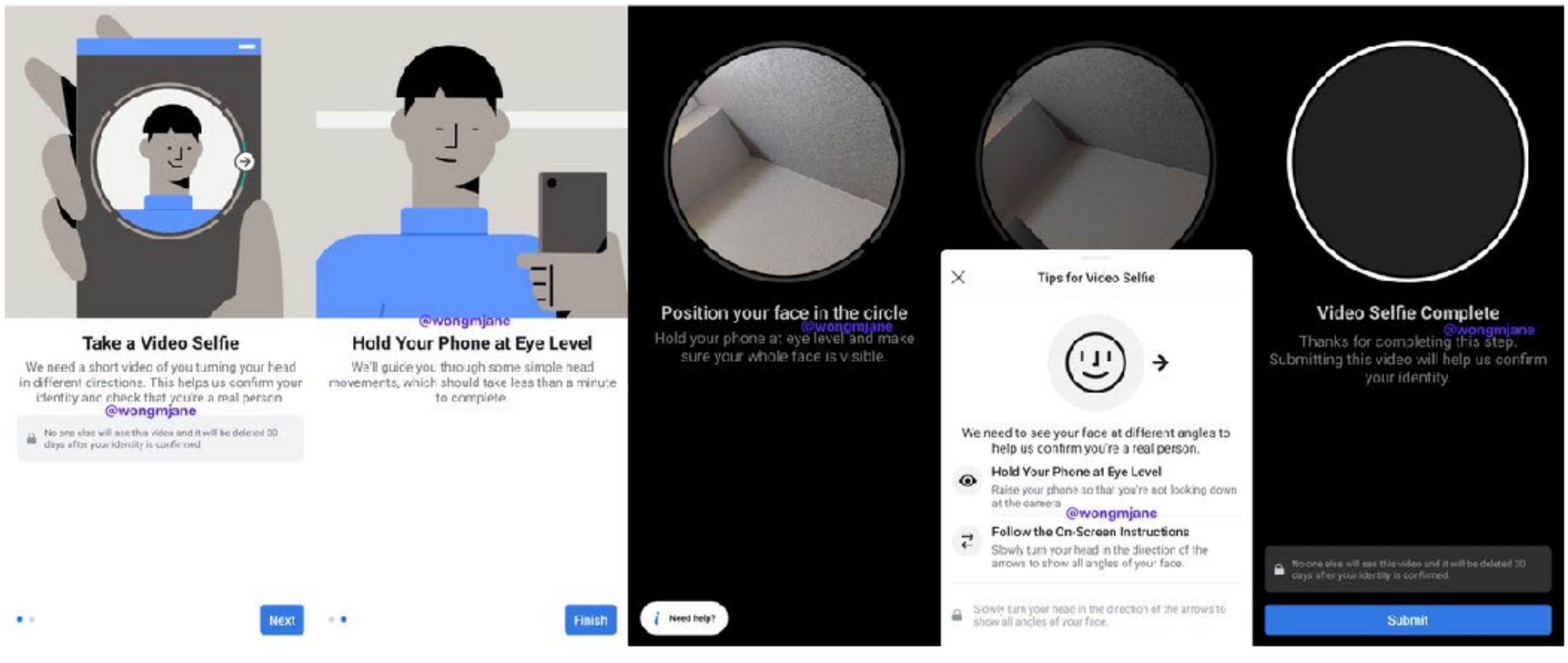
Live Face recognition and biometrics (again):

This is a common method used on some Crypto trading platforms and some dating Apps.

Some platforms/apps will require you to take a live picture of yourself either doing something (a wink, holding an arm up …) or showing a custom piece of information (a hand written text, a passport or ID) within the picture. Sometimes the platform/app will require several pictures to increase their certainty.



This guide won't cover this one (yet) as it's mainly used on financial platforms (that will be able to identify you with other means anyway) and some dating apps like Tinder[276]. Unfortunately this method is now also sometimes being used on Facebook[277] and Instagram as part of their verification methods (tho I didn't face it yet so far).



In some cases these verifications have to be done from your Smartphone and with an "in-app" camera to prevent you from sending a previously saved (edited) image.

Recently even platforms such as PornHub decided to implement similar measures in the future[278].

This verification is very hard to defeat but not impossible. A method to possibly defeat those would be to use "deep fake" technology software such as the open-source FaceSwap https://github.com/deepfakes/faceswap to generate the required verification pictures using a randomly computer generated face that would be swapped over the picture of a complicit model (or a stock photo).

Unfortunately some apps require direct access to a smartphone camera to process the verification. In that case we'll need to find a way to do such "face swaps" on the fly using a filter and another way to feed this into the camera used by the app.

Manual reviews:

These can be triggered by any of the above and just means someone (usually specialized employees) will review your profile manually and decide if it's real or not based on their opinion.

Pros: Usually that verdict is "final" and you'll probably avoid further issues if you're good.

Cons: Usually that verdict is "final" and you'll probably be banned without any appeal possibility if you're not good.

Getting Online:

Now that you have an understanding of all the ways you can be de-anonymized, tracked and verified. Let's get started at evading these while remaining anonymous. Remember:

- You can't trust ISPs
- You can't trust VPS providers
- You can't trust public Wi-Fi providers
- You can't trust Mobile Network providers
- You can't trust VPN providers
- You can't trust any Online Platform
- You can't trust Tor
- You can't trust your Operating systems (especially Android and Windows).
- You can't trust your Laptop
- You can't trust your Smartphone (especially Android).
- You can't trust your Smart devices
- Above all, you can't trust people.

So what? Well instead of not trusting anyone or anything, I would advise to **"Trust but verify"**[279] instead.

Do not start this process unless:

- **You consulted your local law for compliance and the legality of your actions.**
- **You are well aware of your threat model.**
- **You are in a safe place with a public Wi-Fi without your smartphone or any other smart device on you. And preferably in a place without CCTV filming you (remember Find some safe places with decent public WIFI: and Appendix Q: Using long range Antenna to connect to Public Wi-Fis from a safe distance:)**
- **You are fully done and preparing one of the routes.**
- **Again, it is crucially important to understand that you will be unable to create most accounts without a valid phone number. Therefore, most of your anonymity on mainstream platforms depends on the anonymity of your online phone number and/or the burner phone with its pre-paid SIM card (if you use one). If your phone number is not anonymous or your burner phone can be traced back to you then you can be de-anonymized. If you can't get this anonymous phone number and/or a physical SIM with a Burner phone, then you'll have to restrict yourself to platforms not asking for phone number verification.**

Remember see Appendix N: Warning about smartphones and smart devices

Creating new identities:

This is the fun part where you will now create your identities from thin air. These identities do not exist but should be plausible and look "organic". They should ideally have a story … a "legend" (yes this is actually the official term for this[280]).

Now is also the moment where you could finally consider getting an online phone number as explained in the Online Phone Number (less recommended): section.

I will help you bit by listing a few tips I learned while doing research over the years **(disclaimer: this is based on my personal experiences alone)**:

- "Some animals are more equal than others".
    - Ethnicity is important and you will have less issues and attract less attention to verification algorithms if you're Caucasian/East-Asian than if you're Arabic/Black.
    - Age is important and you will have less issues if you're young (18-22) than if you're middle-aged or older.
    - Sex/Gender is important and you'll have fewer issues if you're a female than if you're a male.
    - Country of origin is important and you'll have fewer issues if your identity is Norwegian than if it's Ukrainian or Mexican.
    - Country of residence is important and you'll have fewer issues if your identity has its residence in Oslo or Paris than if you decide to reside in Kiev or Cairo.
    - Language is important and you'll have fewer issues if you speak proper English or the language of your Identity than if you use a non-related language. Don't make a Norwegian born Caucasian 20 year old female that speaks Ukrainian or Arabic.
- Identities that are "EU residents" with an "EU IP" (VPN/Tor) will benefit from GDPR protections on many platforms. Others will not. GDPR is your friend in most cases.
- Similarly, origin IP geolocation (your IP/location when you go to "whatsmyipaddress.com") should match your identity location as much as possible (You can pick this in the VPN client if you use the 3 layers approach or just create a new identity in Tor Browser or Brave Tor Tab until you get the appropriate Exit node, or alternatively configure Tor to restrict your Exit Nodes). You should exclude any exit IP that is not located in Western Europe/US/Canada/Japan/South Korea/Australia/New Zealand as you will have less issues. Ideally, you should get a European Union IP to get additional GDPR protection and if possible a German exit IP due to their legal stance on using anonymous accounts on online platforms.
- Brave Browser (Chromium based) with a Private Tor Tab has (IMHO) a better acceptance level than Tor Browser (Firefox based). You will have less issues with captchas and online platforms[272] if you use Brave than if you use Tor Browser (feel free to try this yourself).
- Every identity you should have a matching profile picture associated to it. For this purpose I recommend you just go to https://thispersondoesnotexist.com/ and generate a computer generated profile picture. You can also generate such pictures yourself from your computer if you prefer by using the open-source StyleGan project here https://github.com/NVlabs/stylegan2. Just refresh the page until you find a picture that matches your identity in all aspects (age, sex, and ethnicity) and save that picture. It would be even better to have several pictures associated to that identity but I don't have an "easy way" of doing that yet.
- Create in advance and store in KeePassXC each identity details that should include some crafted details:
    - Date of Birth
    - Country of Birth
    - Nationality
    - Country of Residence
    - Address of Residence
    - Languages spoken
    - Occupation (Job Title, University…)
    - Various Interests (Art, Politics, Tech…)
    - Phone number (this is your pre-paid SIM card phone number on your Burner phone or your online number paid with Monero)

- Do not pick an occupation at a well-known private corporations/company as they have people in their HR departments monitoring activities in platforms such as LinkedIn and will report your profile as being fake if it doesn't match their database. Instead pick an occupation as a freelancer or at a large public institution where you'll face less scrutiny.
- Keep track (write down) of the background stories of your Identities. You should always use the same dates and answers everywhere. Everything should always match up. Even the stories you tell about your imaginary life should always match. If you say you work as an intern at the Department of Health one day and later on another platform say you work as an intern at the Department of Transportation, people might question your identity. Be consistent.
- Use a different phone number each identity. Online platforms do keep track of phone number usage and if one identity/number gets flagged for violating Community Guidelines or Terms of Services, it might also get the other identities using the same number flagged/banned as well.
- Adapt your language/writing to the identity to not raise suspicions and lower your chances of being fingerprinted by online platforms. Be especially careful with using pedantic words and figures of speech/quotes that could allow some people to guess your writing is very similar to that person with this Twitter handle or this Reddit user.
- Always use TOTP 2FA (not SMS to prevent Sim Swapping attacks[281]) using KeePassXC when available to secure your logins to various platforms.
- Remember, **always use passphrases instead of passwords and use a different one for each service. Do not make it easy for an adversary to access all your information because you used the same password everywhere[206].**
- **Consider using the recommended tools on https://privacytools.io/ for your various purposes instead of the usual mainstream tools.**

Note: If you're having trouble finding an Exit node in the country of your choice you can force using specific countries for Exit Nodes (and therefore exit countries) on Tor by editing the torrc file on the Whonix Gateway or even the Tor Browser:

- Whonix/Tails: Create/Edit a file /usr/local/etc/torrc.d/50_user.conf[282].
- On Tor Browser: Edit the torrc file located at Browser/TorBrowser/Data/Tor[283].

Once you're in the file, you can do the following:

- Specify the Exit Nodes by adding those two lines (which will require an Exit Node in China/Russia/Ukraine:
  - ExitNodes {CH},{RU},{UA}
  - StrictNodes 1
- Exclude specific Exit Nodes by adding this line (which will exclude all Exit Nodes from France/Germany/USA/UK):
  - ExcludeNodes {FR},{DE},{US},{UK}

Always use uppercase letter for any setting.

Please note that this is restricting Onion Routing could limit your Anonymity if you're too restrictive. You can see a visualized list of available Exit Nodes here: https://www.bigdatacloud.com/insights/tor-exit-nodes

Here is the list of possibilities (this is a general list and many of those countries might not have Exit nodes at all): https://b3rn3d.herokuapp.com/blog/2014/03/05/tor-country-codes/

Overview:

Service	Against ToS	Require Phone	Require E-Mail	VPN Sign-up	Tor Sign-up	Captchas	ID Verification	Facial verification	Overall difficulty
Discord	No	No	Yes	Yes	Yes	No	No	No	Medium
Facebook	Yes	Yes	Yes	Maybe	Maybe	Yes	Maybe	Maybe	Hard
GitHub	No	No	Yes	Yes	Yes	Yes	No	No	Easy

Service	Against ToS	Require Phone	Require E-Mail	VPN Sign-up	Tor Sign-up	Captchas	ID Verification	Facial verification	Overall difficulty
Google	No	Maybe	Maybe	Yes	Yes	Yes	No	No	Medium
Instagram	Yes	Maybe	Yes	Yes	Yes	Yes	No	Maybe	Medium
LinkedIn	Yes	Yes	Yes	Yes	Yes	Yes	Maybe	Maybe	Hard
Microsoft	Yes	Maybe	Maybe	Yes	Yes	Yes	No	No	Medium
ProtonMail	No	Maybe	Maybe	Yes	Yes	Yes	No	No	Medium
Reddit	No	No	No	Yes	Yes	No	No	No	Easy
Telegram	No	Yes	No	Yes	Yes	No	No	No	Easy
Twitter	No	Maybe	Yes	Yes	Yes	Yes	No	No	Medium
4chan	No	No	No	No	No	Yes	No	No	Hard

Discord:

- Is this against their TOS? No https://discord.com/terms
- Will they require a phone number? No they require an e-mail
- Can you create accounts through Tor? I had no issues with that so far using the Desktop Client

You might encounter more issues using the Web Client (Captchas). Especially with Tor Browser.

I suggest using the Discord Client app on a VM through Tor or ideally through VPN over Tor to mitigate such issues.

Steps after creating: Enable 2FA authentication with KeePassXC TOTP

Facebook:

- Is this against their TOS? Yes https://www.facebook.com/terms.php

"1. Who can use Facebook

When people stand behind their opinions and actions, our community is safer and more accountable. For this reason, you must:

- Use the same name that you use in everyday life.
- Provide accurate information about yourself.
- Will they require a phone number? Yes and probably more later
- Can you create accounts through Tor? Yes but it's very difficult and their onion address[284] won't help. In most cases you'll just have a random error at sign-up and your account suspended after sign-in.

Facebook is one of the most aggressive platforms in identity verification and is pushing hard their "real name policy". It is why this guide is only advised to German residents.

Over my tests tho I was able to pinpoint a few tips:

- It will be easier if you have an Instagram account first.
- Signing-up through Tor is almost impossible and will only succeed if you're "lucky" (I assume if you're using an exit Node that is not yet known by Facebook verification systems). It will not allow registration at all and will just fail with "An error has occurred during registration".
- Signing-up through VPNs is more likely to succeed but might still result in the same error. So you have to be ready for a lot of trial and errors here.
- My previous entry in the guide about the Orwellian quote from Animal Farm is in full effect on Facebook. You will experience huge variation in acceptance depending on age/sex/ethnicity/nationality/… This is where you will have far less issues if you're making an account of a Young European Caucasian Female. You will almost certainly fail if you try making a Middle-Aged Male where my other accounts are still unsuspended/unbanned to this day.
- Logging-in (after you sign-up) however works fine with VPN and Tor but might still get your account suspended for violating Community Guidelines or Terms of Services (despite you not using the account at all for anything else than signing-up/logging-in).

I also suspect strongly based on my test that the following points have an impact on your likelihood of being suspended over time:

- Not having friends
- Not having interests and an "organic activity"
- Not being in the contacts of any other user
- Not being on other platforms (such as Instagram/WhatsApp)
- Restricting your profile privacy settings too soon after signing-up

If your account gets suspended, you will need to appeal the decision through a very simple form that will require you to submit a "proof of ID". However that proof of ID verification system is more lenient than LinkedIn and will allow you send various documents which require far less Photoshop skills.

It's also possible that they ask you to take a selfie video or picture making certain gestures to prove your identity. If that's the case, I'm afraid it's a dead end for now.

If you do file an appeal, you will have to wait for Facebook to review it (I don't know if this is automatic or human) and you will have to wait and hope for them to unsuspend your account.

GitHub:

- Is this against their TOS? No https://docs.github.com/en/free-pro-team@latest/github/site-policy/github-terms-of-service
- Will they require a phone number? Nope, all good
- Can you create accounts through Tor? Yes but expect some captchas

GitHub is straightforward and requires no phone number.

Just Sign-up with e-mail and password and enable two-factor authentication (TOTP in KeePassXC). By default your e-mail will be private.

Be sure to go into Settings > E-Mail and make your e-mail private as well as block any push that would reveal your e-mail.

Google:

- Is this against their TOS? No https://policies.google.com/terms
- Will they require a phone number? Yes they will. There is no escape here.
- Can you create accounts through Tor? Yes but expect some captchas and your phone number will definitely be required

ProtonMail is good … but to appear less suspicious, it's just better to also have a Google Mail account.

As ProtonMail, Google will also most likely require a phone number during sign-up as part of their verification process. However contrary to ProtonMail, Google will store that phone number during the sign-up process and will also limit the amount of accounts that can be created during the sign-up[285, 286].

From my experience during my research, this count is limited to 3 accounts / phone number. If you are unlucky with your number (if it was previously used by another mobile user), it might be less.

You should therefore use again your online phone number OR your burner phone and pre-paid SIM card to create the account. Don't forget to use the identity details you made up earlier (birthdate). When the account is created, please do take some time to do the following:

- Log into Google Mail and Go into the Gmail Settings > Go into the mail Forwarding options > Set up a mail forwarding to your ProtonMail Address > Verify (using ProtonMail) > Go back to Gmail and set the forwarding to forward and delete Google copy > Save. This step will allow you to check your Google Mail using ProtonMail instead and will allow you to avoid triggering Google Security checks by Logging in from various VPN/Tor exit IP addresses in the future while storing your sensitive e-mail at ProtonMail instead.

- Enable 2FA within the Google account settings. First you'll have to enable 2FA using the phone number. Then you'll see the option appear to enable 2FA using an Authenticator app. Use that option and set it up with a new KeePassXC TOTP entry. When it's done, remove the phone 2FA from the Google account. This will prevent someone from using that phone number in the future (when you don't have it anymore) to recover/gain access to that account.
- Add ProtonMail as a recovery e-mail address for the account.
- Remove the phone number from the account details as a recovery option.
- Upload a Google profile picture you made earlier during the identity creation step.
- Review the Google Privacy settings to disable as much as you can:
    - Activity logging
    - YouTube
- Log out and don't touch it unless needed (as mentioned, you will use ProtonMail to check your Gmail).

Keep in mind that there are different algorithms in place to check for weird activity. If you receive any mail (on ProtonMail) prompting about a Google Security Warning. Click it and Click the button to say "Yes it was me". It helps.

Do not use that account for "sign-up with Google" anywhere unless necessary.

Be extremely careful if you decide to use the account for Google activities (such as Google Maps reviews or YouTube Comments) as those can easily trigger some checks (Negative reviews, Comments breaking Community Guidelines on YouTube).

If your account gets suspended [287] (this can happen on sign-up, after signing-up or after using it in some Google services), you can still get it unsuspended by submitting[288] an appeal/verification (which will again require your Phone number and possibly an e-mail contact with Google support with the reason). Suspension of the account does not disable the e-mail forwarding but suspended account will be deleted after a while.

After suspension, if your Google account is restored, you should be fine.

If your account gets banned, you will have no appeal and the forwarding will be disabled. Your phone number will be flagged and you won't be able to use it to sign-up on a different account. Be careful when using those to avoid losing them. They're precious.

Instagram:

- Is this against their TOS? Yes https://help.instagram.com/581066165581870?ref=dp

"You can't impersonate others or provide inaccurate information. You don't have to disclose your identity on Instagram, but you must provide us with accurate and up to date information (including registration information). Also, you may not impersonate someone you aren't, and you can't create an account for someone else unless you have their express permission." But this clause of their TOS is illegal in Germany (see Requirements:).

- Will they require a phone number? Maybe but less likely over VPN and very likely over Tor
- Can you create accounts through Tor? Yes but expect some captchas and your phone number will definitely be required

It's also possible that they ask you to take a selfie video or picture making certain gestures to prove your identity (within the app or through an e-mail request). If that's the case, I'm afraid it's a dead end for now.

It's no secret that Instagram is part of Facebook however it's more lenient than Facebook when it comes to user verification. It's quite unlikely you'll get suspended or banned after signing-up. But it could help.

For instance I noticed that you'll have less issues creating a Facebook account if you already have a valid Instagram account. You should always create an Instagram account before attempting Facebook.

Unfortunately there are some limitations when using the web version of Instagram. For instance you won't be able to enable Authenticator 2FA from the web for a reason I don't understand.

After sign-up, do the following:

- Upload a picture of your generated identity if you want.
- Go into your Settings
- Make the account private (initially at least)
- Do not show activity status
- Do not allow sharing

LinkedIn:

- Is this against their TOS? Yes https://www.linkedin.com/legal/user-agreement

" To use the Services, you agree that: (1) you must be the "*Minimum Age*" (described below) or older; (2) **you will only have one LinkedIn account, which must be in your real name**; and (3) you are not already restricted by LinkedIn from using the Services. **Creating an account with false information is a violation of our terms**, including accounts registered on behalf of others or persons under the age of 16. "

But this clause of their TOS is illegal in Germany (see Requirements:).

- Will they require a phone number? Yes they will.
- Can you create accounts through Tor? Yes but expect some captchas and your phone number will definitely be required

LinkedIn is far less aggressive than twitter but will nonetheless require a valid e-mail (preferably again your Gmail) and a phone number in most cases (tho not always).

LinkedIn however is relying a lot on reports and user/customer moderation. You should not create a profile with an occupation inside a private corporations or a small startup company. The company employees are monitoring LinkedIn activity and receive notifications when new people join. They can then report your profile as fake and your profile will then be suspended or banned pending appeal.

LinkedIn will then require you go through a verification process that will unfortunately require you to send an ID proof (identity card, passport, driver license). This ID verification is processed by a company called Jumio[289] that specializes in ID proofing. This is most likely a dead end as this would force you to develop some strong Photoshop skills.

Instead you are far less likely to be reported if you just stay vague (say you're a student/intern/freelance) or pretend you work for a large public institution that is too large for anyone to care of check.

As with Twitter and google, you should do the following after signing-up:

- Disable ads
- Disable notifications
- Disable lookup by phone/e-mail
- Upload a picture of your identity

Microsoft:

- Is this against their TOS? Yes https://www.microsoft.com/en/servicesagreement/

"i. Creating an Account. You can create a Microsoft account by signing up online. You agree not to use any false, inaccurate or misleading information when signing up for your Microsoft account." But this clause of their TOS is illegal in Germany (see Requirements:).

- Will they require a phone number? Maybe but not always. Even on Tor depending on your luck/exit node, it's possible that they will only require e-mail verification.
- Can you create accounts through Tor? Yes you can but expect captchas, e-mail verification at least **and maybe phone verification.**

So yes it's still possible to create an MS account without a phone number and using Tor or VPN but you might have cycle through a few exit nodes to achieve this.

After signing-up you should setup 2FA authentication within security and using KeePassXC TOTP.

ProtonMail:

- Is this against their TOS? No https://ProtonMail.com/terms-and-conditions
- Will they require a phone number? Maybe. This depends on the IP you're coming from. If you come from Tor, it's likely. From a VPN, it's less likely.
- Can you create accounts through Tor? Yes but very likely that a phone number will be required when only an e-mail will be over a VPN.

You obviously need an e-mail for your online identity and disposable e-mails are pretty much banned everywhere.

ProtonMail is a free e-mail provider based in Switzerland that advocates security and privacy.

They're recommended by privacytools.io[290]. Their only apparent issue is that they do require (in most cases) a phone number or another e-mail address for registration (when you try to register from a VPN or Tor at least).

They claim they do not store/link the phone/e-mail associated with the registration but only store a hash that is not linked to the account[291]. As long as their claim is true and the hash is not linked to your account, and that you followed my guide regarding the phone number, you should be reasonably safe from tracking.

Create this e-mail account first using the phone as verification if necessary.

When you're done creating the account, please go into the settings and enable 2FA (Two Factor Authentication). You will use KeePassXC TOTP feature (create a new entry "Identity ProtonMail TOTP" and just use the TOTP menu to set it up). Save the rescue codes within your KeePassXC entry.

This e-mail account will be used in the next step for creating a Google/Gmail account.

Reddit:

- Is this against their TOS? No https://www.redditinc.com/policies
- Will they require a phone number? No they won't.
- Can you create accounts through Tor? Yes

Reddit is simple. All you need to register is a valid username and a password. Normally they don't even require an e-mail (you can skip the e-mail when registering leaving it blank).

You should still enable 2FA in the settings after signing-up. I had no issues whatsoever signing-up over Tor or VPN besides the occasional Captchas.

Telegram:

- Is this against their TOS? No https://telegram.org/tos
- Will they require a phone number? Yes unfortunately
- Can you create accounts through Tor? Yes but sometimes you randomly get banned without any reason

Telegram is quite straightforward and you can download their portable Windows app to sign-up and login.

It will require a phone number (that can only be used once) and nothing else.

In most cases I had no issues whether it was over Tor or VPN but I had a few cases where my telegram account was just banned for violating terms of services (not sure which one?). This again despite not using them for anything.

They provide an appeal process through e-mail but I had no success with getting any answer.

Their appeal process is just sending an e-mail to recover@telegram.org stating your phone number and issue and hope they answer.

After signing-up you should do the following:

- Go into Edit profile
- Set a Username
- Go into Settings (Desktop App)
- Set the Phone Number visibility to Nobody

- Set Last Seen & Online to Nobody
- Set Forwarded Messages to Nobody
- Set Profile photos to Contacts
- Set Calls to Contacts
- Set Group & Channels to Contacts

Twitter:

- Is this against their TOS? No https://twitter.com/en/tos
- Will they require a phone number? They might not at sign-up but they will just after sign-up or later.
- Can you create accounts through Tor? Yes but expect some captchas and your phone number will definitely be required after a while.

Twitter is extremely aggressive in preventing anonymity on their network. You should sign-up using e-mail and password (not phone) and not using "Sign-in with Google". Use your Gmail as the e-mail address.

More than likely, your account will be suspended immediately during the sign-up process and will require you to complete a series of automated tests to unlock. This will include a series of captchas, confirmation of your e-mail and twitter handle or other information. In some cases, it will also require your phone number.

In some cases, despite you selecting a text verification, Twitter verification system will call the phone no matter what. In that case you'll have to pick up and actually hear the verification code. I suspect this is another method of preventing automated systems and malicious users from selling text receiving services over the internet.

Twitter will store all this information and link it to your account including your IP, e-mail and phone number. You will not be able that phone number to create a different account.

Once the account is restored, you should take some time to do the following:

- Upload the identity profile picture.
- Enable 2FA from the security settings using a new KeePassXC TOTP entry, save the security codes in KeePassXC as well.
- Disable Photo tagging
- Disable E-mail lookup
- Disable Phone lookup
- Disable all personalized advertising settings
- Disable geolocation of tweets
- Remove the phone number from the account
- Follow some people based
- Log out and leave it be.

After about a week, you should check the twitter again and the chances are quite high that it will be suspended again for "suspicious activity" or "violating community guidelines" despite you not using it at all (not even a single tweet/follow/like/retweet or DM) but this time by another system. I call this the "Double tap".

This time you will need to submit an appeal using a form[292], provide a good reason and wait for the appeal to be processed by Twitter. During that process, it's possible that you will receive an e-mail (on ProtonMail) asking you to reply to a customer service ticket to prove that you do have access to your e-mail and that it's you. This will be directed toward your Gmail address but will arrive on your ProtonMail.

Obviously do not reply from ProtonMail as this will raise suspicions, you have to sign-in into Gmail (unfortunately) and compose a new mail from there copy pasting the E-Mail, Subject and Content from ProtonMail. As well as a reply confirming you have access to that e-mail.

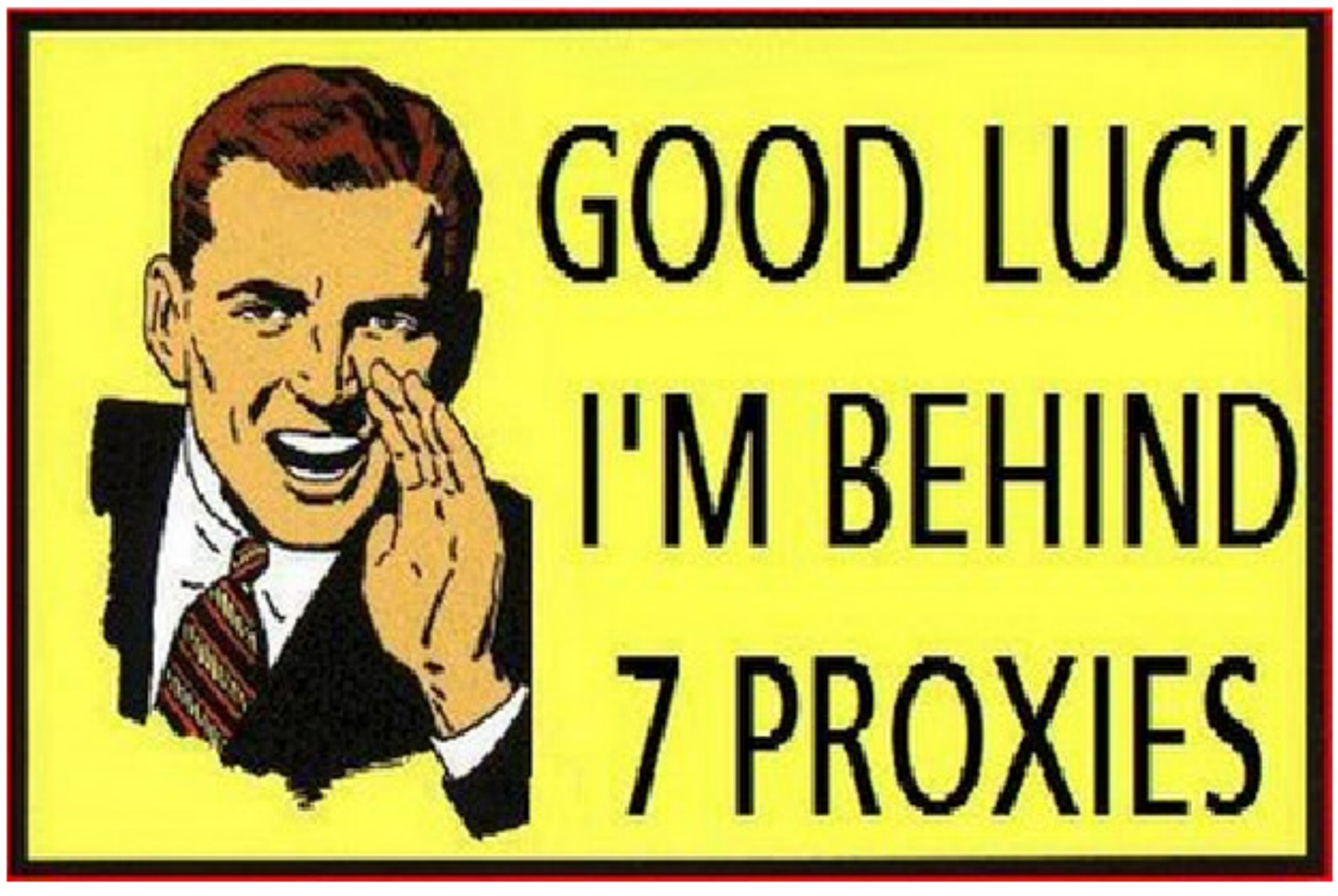
After a few days, your account should get unsuspended "for good". I had no issues after that but keep in mind they can still ban your account for any reason if you violate the community guidelines. The phone number and e-mail will then be flagged and you will have no other option but to get a new identity with a new number to sign-up again. Don't use this account for trolling.

4chan:

- Is this against their TOS? No
- Will they require a phone number? No they won't.
- Can you post there with Tor or VPN? Not likely

4chan is 4chan … This guide won't explain 4chan to you. They block Tor exit nodes and VPN IP ranges.

You're going to have to find a different way to post there using at least seven proxies[293] that are not known by 4chan blocking system (hint: Anonymous VPS using Monero is probably your best option).



Crypto Wallets:

Use any crypto wallet app within the Windows Virtual Machine. But be careful not to transfer anything toward an Exchange or a known Wallet. Crypto is in most case NOT anonymous and can be traced back to you when you buy/sell any (remember the Your Crypto currencies transactions: section).

If you really want to use Crypto, use Monero which is the only one with reasonable privacy/anonymity.

Ideally, you should find a way to buy/sell crypto with cash from an unknown person.

What about those mobile only apps (WhatsApp/Signal):

There are only three ways of securely using those anonymously (that I would recommend). Using a VPN on your phone is not among those ways. All of those are unfortunately "tedious" to say the least.

- Use an Android Emulator (Youwave[294], Bluestacks[295] or if you have a powerful PC, Android Studio) within the Windows VM and run the App through your multi-layer of Tor/VPN. Drawback is that such emulators are usually quite resource hungry and will slow down your VM and use more battery. Here is also an (outdated) guide on this matter: https://www.bellingcat.com/resources/how-tos/2018/08/23/creating-android-open-source-research-device-pc/
- Use a non-official app (such as Wassapp for WhatsApp) to connect from the Windows VM to the app. But at your own risk as you could get banned for violating the terms of services by using a non-official App.
- (Least recommended and most complicated) Have a burner Smartphone that you will connect to the VM layered network through Tethering/Sharing of the connection through Wi-Fi. I won't detail this here but it's an option if you really want to.

There is no way to reliably set this multi-layered connectivity approach easily on an Android phone (it's not even possible on IOS as far as I know). By reliable I mean being sure that the smartphone won't leak anything such as geolocation or anything else from booting up to shutting down.

Anything else:

You should use the same logic and security for any other platform that with these mentioned in this guide.

It should work in most cases with most platforms. **The hardest platform to use with full anonymity is Facebook.**

This will obviously not work with Banks and most Financial platforms (such as PayPal or Crypto Exchanges) requiring actual real official and existing identification. This guide won't help you there as this would actually be illegal in most places.

Maintenance tasks:

You should sign-up carefully into your accounts from time to time to keep them alive.

Check your e-mail regularly for security checks and any other account notification.

Backing-up your work securely:

Do not ever upload encrypted file containers with plausible deniability (hidden containers within them) to most cloud services (ICloud, Google Drive, OneDrive, Dropbox) without safety precautions. This is because most cloud services keep backups/versioning of your files and such backups/versioning of your encrypted containers can be used for differential analysis to prove the existence of a hidden container.

Instead this guide will recommend other methods of backing up your stuff safely.

Offline Backups:

These backups can be done on an external hard drive or an USB key. Here are the various possibilities.

Selected Files Backups:

Requirements:

For these back-ups, you'll need an USB key or an external hard drive with enough capacity to store the files you want to back-up.

Veracrypt:

For this purpose I will recommend the use of Veracrypt on all platforms (Linux/Windows/MacOS) for convenience/security and portability.

Normal File containers:

The process is fairly simple and all you'll need is to follow Veracrypt tutorial here:
https://www.veracrypt.fr/en/Beginner%27s%20Tutorial.html

In this container, you can then store sensitive data manually and or use any backup utility you want to backup files from the OS to that container.

You can then store this container anywhere safe.

Hidden File containers with plausible deniability:

The process is also fairly simple and similar to the previous tutorial except this time you'll use the Veracrypt wizard to create a Hidden Veracrypt Volume instead of a Standard Veracrypt Volume.

You can create a Hidden volume within an existing Standard Volume or just use the wizard to create a new one.

Let's say you want a container of 8GB, the Wizard will first create an "outer volume" where you will be able to store decoy information when prompted. Some decoy files (somewhat sensible, plausible but what you really want to hide) should be stored in the decoy volume.

Then Veracrypt will ask you to create a smaller hidden container (for instance 2GB or 4GB) within the outer volume where you can store your actual hidden files.

When you select the file for mounting in Veracrypt, depending on which password you provide, it will mount the Outer decoy volume or the Hidden volume.

You can then mount your hidden volume and use it to store sensitive files normally.

Be careful when mounting the Outer decoy volume to update its content. You should protect the hidden volume from being overwritten when doing this as working in the decoy volume could overwrite data in the hidden volume.

To do this, when mounting the Decoy Volume, select Mount Options and Check the "Protect hidden volume" option and also provide the hidden volume password on the same screen. Then mount the decoy volume. This will protect the hidden volume from being overwritten when changing the decoy files. This is also explained here in Veracrypt documentation: https://www.veracrypt.fr/en/Protection%20of%20Hidden%20Volumes.html

Be extremely cautious with these file containers:

- **Do not store multiple versions of them or store them anywhere where some versioning is being done (by the file system or the storage system). These file containers should be identical everywhere you store them. If you have a backup of such containers somewhere, it needs to be absolutely identical to the one you're using. If you don't take this precaution, an adversary could compare two different versions of this container and prove the existence of hidden data. Follow carefully the recommendations here https://www.veracrypt.fr/en/Security%20Requirements%20for%20Hidden%20Volumes.html. Remember the Local Data Leaks and Forensics: section.**
- I strongly recommend storing such containers on external USB keys that you will only mount from your guest VMs and never from your Host OS. **After each modification to the files, you should clean the free space on the USB disk and make sure that any backup of such containers are absolutely identical on each key and your computer. See the How to securely delete specific files/folders/data on your HDD/SDD and Thumb drives: section of this guide for help on doing this.**
- If you have time, **I would even recommend you delete wipe the keys completely before making any modification on such containers on your computer (if you don't work from the USB key directly).** This is to prevent an adversary that would seize your assets before you could update the keys from having multiple versions of the containers that could lead to proving the existence of hidden data using forensics techniques.
- **Do not ever store such containers on cloud storage platforms that have backups and where you have no direct control over permanent deletion. They might keep "old versions" of your files which can then also be used by forensics to prove the existence of hidden data.**
- If you're mounting the hidden volume from your Host OS (**not recommended**), you should erase all traces of this hidden volume everywhere after use. There could be traces in various places (system logs, file systems journaling, recent documents in your applications, indexing, registry entries…). Refer to the Some additional measures against forensics: section of this guide to remove such artifacts. Especially on Windows. Instead you should mount them on your Guest VMs. With Virtualbox for instance, you could take a snapshot of the VM before opening/working the hidden volume and then restore the snapshot prior to opening/working on it after use. This should erase the traces of its presence and mitigate the issue. Your Host OS might keep logs of the USB key being inserted but not of the hidden volume usage. This is why I do not recommend using these from your host OS.
- Do not store these on external SSD drives if you're not sure you have the ability to use Trim on them (see the Understanding HDD vs SSD: section).

Full Disk/System Backups:

TLDR version: Just use Clonezilla as it worked reliably and consistently with all my tests on all operating systems except for Macs where you should probably use native utilities (Time Machine/Disk utility instead) to avoid compatibility issues and since you're using Native MacOS encryption. When using Windows, do not backup a partition containing a hidden OS in case you use Plausible Deniability (as explained before, this backup could allow an adversary to prove the existence of the hidden OS by comparing the last backup to the current system where data will have changed and defeat plausible deniability, use file containers instead).

You will have two options here:

- (Not recommended) Doing your backup from the live operating system using a back-up utility (commercial utilities such as EaseUS Todo Free, Macrium Reflect…) or native utilities like MacOS Time Machine, QubesOS Backup, Ubuntu Déjà Dup or Windows Backup…).
    - This backup can be done while the Operating System is running.
    - This backup will not be encrypted using the disk encryption but using the Backup utility encryption algorithm (which you'll have to trust and can't really control for most). Alternatively, you could

encrypt the backup media yourself separately (for instance with Veracrypt). I'm not aware of any free or non-free utility that natively supports Veracrypt.
    - Some utilities will allow for differential/incremental backups instead of full backups.
    - These backup utilities will not be able to restore your encrypted drive as-is as they do not support those encrypted file systems natively. And so these restore will require more work to restore your system in an encrypted state (re-encryption after restore).
- (Recommended) Doing it offline from a boot drive (such as with the free open-source Clonezilla).
    - This backup can only be done while the Operating System is not running.
    - This backup will back up the encrypted disk as-is and therefore will be encrypted by default with the same mechanism (it's more like a fire and forget solution). The restore will also restore the encryption as-is and your system will immediately be ready to use after a restore.
    - This method will not allow incremental/differential back-ups (meaning you'll have to re-do a full back-up every time).
    - This method is clearly the easiest to manage.

I made extensive testing using live backups utilities (Macrium Reflect, EaseUS Todo Reflect, Déjà Dup…) and personally I don't think it's worth it. Instead I would recommend that you periodically back-up your system with a simple Clonezilla image. It's much easier to perform, much easier to restore and usually works reliably without issues in all cases. And contrary to many beliefs, it's not that slow with most backups taking about an hour depending on the speed of your destination media.

For backing up single files while you work, I recommend using file containers or encrypted media directly and manually as explained in the previous section.

Requirements:

You will need a separate external drive with at least the same or more free space available than your source disk. If your Laptop has a 250GB disk. You will need at least 250GB of free disk space for the full image backup. Sometimes this will be reduced significantly with compression by the backup utility but as a safety rule you should have at least the same or more space on your backup drive.

Some general warnings and considerations:

- If you use Secure Boot, you will need a backup utility that supports Secure Boot which includes Clonezilla AMD64 versions.
- Consider the use of exFAT as file system for your backup drives as those will provide better compatibility between various OSes (MacOS, Linux, and Windows) vs NTFS/HFS/ext4...

Linux:

Ubuntu (or any other distro of choice):

I will recommend the use of the open-source Clonezilla utility for convenience and reliability but there are many other native Linux utilities and methods you could use for this purpose.

So you should follow the steps in Appendix E: Clonezilla

QubesOS:

Qubes OS recommends using their own utility for backups as documented here https://www.qubes-os.org/doc/backup-restore/ . But I think it's just a hassle and provides limited added value unless you just want to back-up a single Qube. So instead I'm also recommending just making a full image with Clonezilla which will remove all the hassle and bring you back a working system in a few easy steps.

So you should follow the steps in Appendix E: Clonezilla

Windows:

I will only recommend the use of the open-source and free Clonezilla utility for this purpose. There are commercial utilities that offer the same functionality but I don't see any advantage in using any of them vs Clonezilla.

Some warnings:

- If you use Bitlocker for encryption with TPM[296] enabled, you might need to save your Bitlocker Key (safely) somewhere as well as this might be needed to restore your drive if your HDD/SDD or other hardware parts changed. Another option would be to use Bitlocker without the use of TPM which would not require this option. But again I do not recommend using Bitlocker at all.
- You should always have a backup of your Veracrypt rescue disk at hand somewhere to able to resolve some issues that might still appear after a restore. Remember this rescue disk does not contain your passphrase or any sensitive information. You can store it as is.
- If you changed the HDD/SDD after a failure, it's possible that Windows 10 will refuse to boot if your hard drive ID changed. You should also save this ID prior to backing up as you might need to change the ID of the new drive as Windows 10 might require a matching ID before booting. See Appendix F: Diskpart
- **In case you're using Plausible Deniability on Windows. DO NOT back-up the hidden OS partition as this image could be used by Forensics to prove the existence of the hidden volume as explained earlier. It's okay to back-up the Decoy OS partition without issues but you should absolutely not backup the partition containing the Hidden OS.**

Follow the steps in Appendix E: Clonezilla

MacOS:

I would recommend just using the native Time Machine backup with encryption (and a strong passphrase that could be the same as your OS) as per the guides provided at Apple: https://support.apple.com/en-ie/guide/mac-help/mh21241/mac and https://support.apple.com/en-ie/guide/mac-help/mh11421/11.0/mac/11.0.

So plug in an external drive and it should prompt you to use it as a Time Machine backup.

You should however consider formatting this drive as exFAT to that it's also usable by other OSes conveniently (Windows/Linux) without added software using this guide: https://support.apple.com/en-ie/guide/disk-utility/dskutl1010/mac

It's just simpler and will work online while you work. You'll be able to recover your data on any other Mac from the recovery options and you'll be also able to use this disk for backing up other devices.

It's not impossible to also use Clonezilla to clone your Mac Hard Drive but it could bring hardware compatibility issues and probably won't add much in terms of security. So for MacOS I'm not specifically recommending Clonezilla.

Online Backups:

This is a tricky one. The problem is that it depends on your threat model.

- If you use containers with plausible deniability, you should never ever store them on any platform where you don't have full control over the deletion process as the platform will most likely have backups of previous versions for some time and again these previous versions could allow forensics to prove the existence of hidden data and defeat plausible deniability. This includes platforms like DropBox, Google Drive, OneDrive or others. The only acceptable online storage of those could be "cold storage" (meaning you will never change those files again and just keep them away untouched compared to any local version).
- If you use normal encrypted backups without plausible deniability, you could store them pretty much anywhere as long as they're properly encrypted locally before uploading (with strong passphrases and encryption). **Do not ever trust encryption of any online provider. Only trust your own local encryption (using Veracrypt for instance).** For these cases, you could store your backups pretty much anywhere in the accounts of your online identities (iCloud, Google Drive, DropBox…) as long as they're strongly encrypted locally before uploading.

Obviously do not ever do/access those backups from unsecure/unsafe devices but only from the secure environments you picked before.

Self-hosting (using Nextcloud for instance) is also a possibility provided you do have an anonymous hosting. You could use https://evolution-host.com/vps-hosting.php (who supports Whonix) which does accept Monero payments. Or pick one from the list here that do accept Monero: https://www.getmonero.org/community/merchants/#hosting

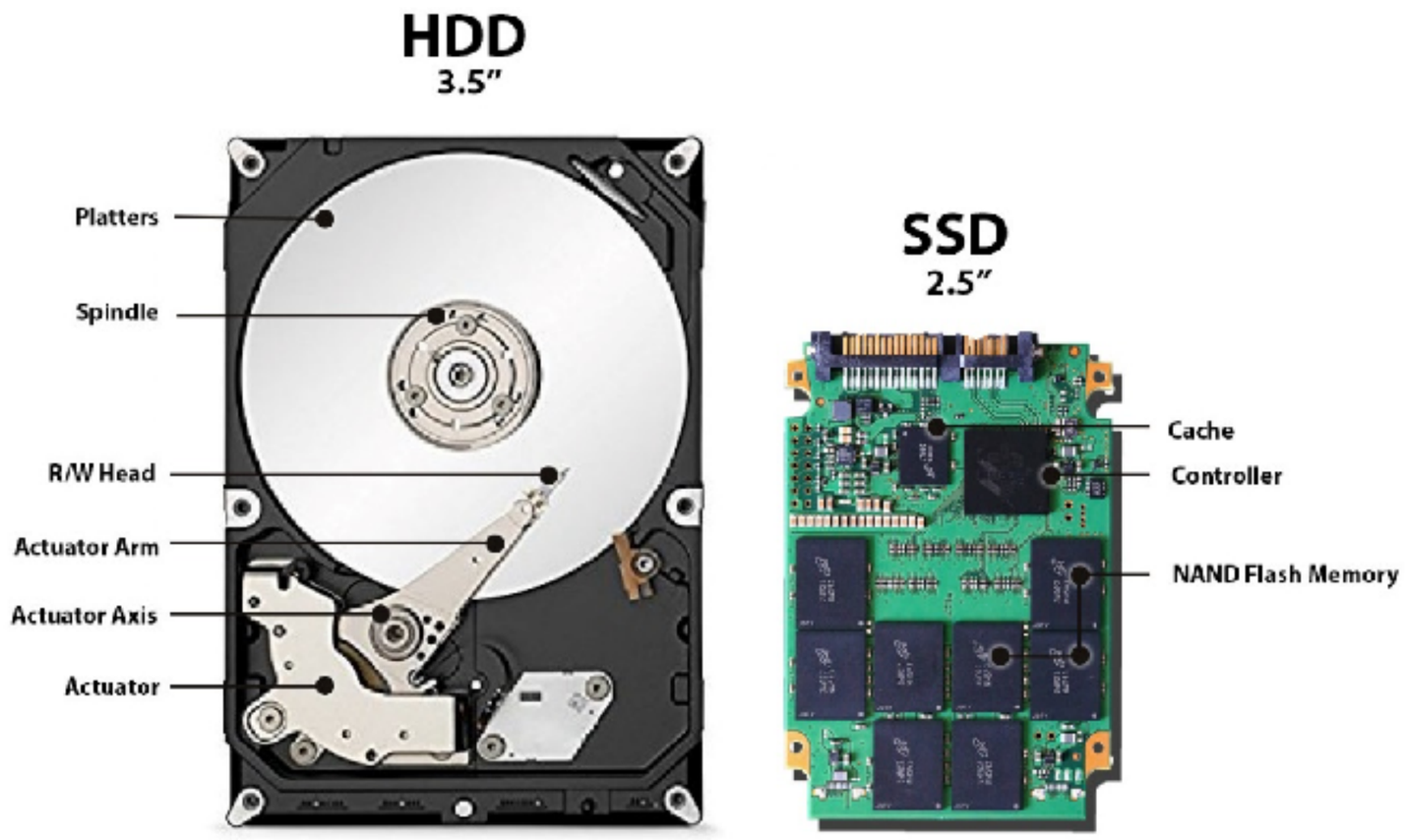
Consider tho that only one of those accepting Monero is actually recommended by Privacytools.io (https://privacytools.io/providers/hosting/) and it's Njalla https://njal.la/

I am currently not aware of any online storage/hosting platform accepting cash payments unlike some VPN providers mentioned before.

My personal recommendation is just to keep offline backups on external HDD (not SSD) drives (where wiping data securely and selectively is just easier as explained in the next section) or on smaller USB keys where you can also fully wipe them over and over in a relatively short amount of time.

Covering your tracks:

Understanding HDD vs SSD:



If you intent to wipe your whole HDD laptop, the process is rather simple and straightforward. The data is written at a precise location on a magnetic (hard) platter (why it's called a hard drive) and your OS knows precisely where it is on the platter, where to delete it and where to overwrite it for secure deletion using simple processes (like just overwriting that location over and over until no traces are left).

On the other hand, if you're using an SSD drive, the process is not as simple as the drive uses several internal mechanisms to extent its lifespan and performance. Three of those processes are of particular interest when it comes to us in this guide. SSD drives are divided themselves into 2 main categories:

- ATA Drives (usually SATA and usually 2.5" format as the image above).
- NVMe Drives (usually M.2 format as the illustration below).

Here are examples of the most common formats:

All of these are sold as internal and external drives within enclosures.

The methods and utilities to manage/wipe them will vary depending on the type of drive you're using. So it's important you know which one you have within your laptop.

On most recent laptops, chances are high that it will be one of the middle options (M.2 SATA or M.2 NVMe).

Wear-Leveling.

These drives use a technique called wear leveling[297]. At a high level, wear leveling works as follows. The space on every disk is divided into blocks that are themselves divided into pages, kind of like the chapters in a book are made of pages. When a file is written to disk, it's assigned to a certain set of pages and blocks. If you wanted to overwrite the file in an HDD, then all you would have to do is tell the disk to overwrite those blocks. But in SSDs and USB drives, erasing and re-writing the same block can wear it out. Each block can only be erased and rewritten a limited number of times before that block just won't work anymore (the same way if you keep writing and erasing with a pencil and paper, eventually the paper might rip and be useless). To counteract this, SSDs and USB drives will try to make sure that the amount of times each block has been erased and rewritten is about the same, so that the drive will last as long as possible (thus the term wear leveling). As a side effect, sometimes instead of erasing and writing the block a file was originally stored on, the drive will instead leave that block alone, mark it as invalid, and just write the modified file to a different block. This is kind of like leaving the chapter in the book unchanged, writing the modified file on a different page, and then just updating the book's table of contents to point to the new location. All of this occurs at a very low level in the electronics of the disk, so the operating system doesn't even realize it's happened. This means, however, that even if you try to overwrite a file, there's no guarantee the drive will actually overwrite it, and that's why secure deletion with SSDs is so much harder.

Wear-leveling alone can therefore be seen as a disadvantage for security and an advantage for adversaries such as forensics examiners. This feature makes classic "secure deletion" counter-productive and useless and is why this feature was removed on some Operating Systems like MacOS (a as from version 10.11 El Capitan) where you could enable it before on the Recycle Bin.

Most of those old secure deletion utilities were written with HDD in mind and have no control over wear-leveling and are completely pointless when using an SDD. Avoid them on an SSD drive.

Trim Operations:

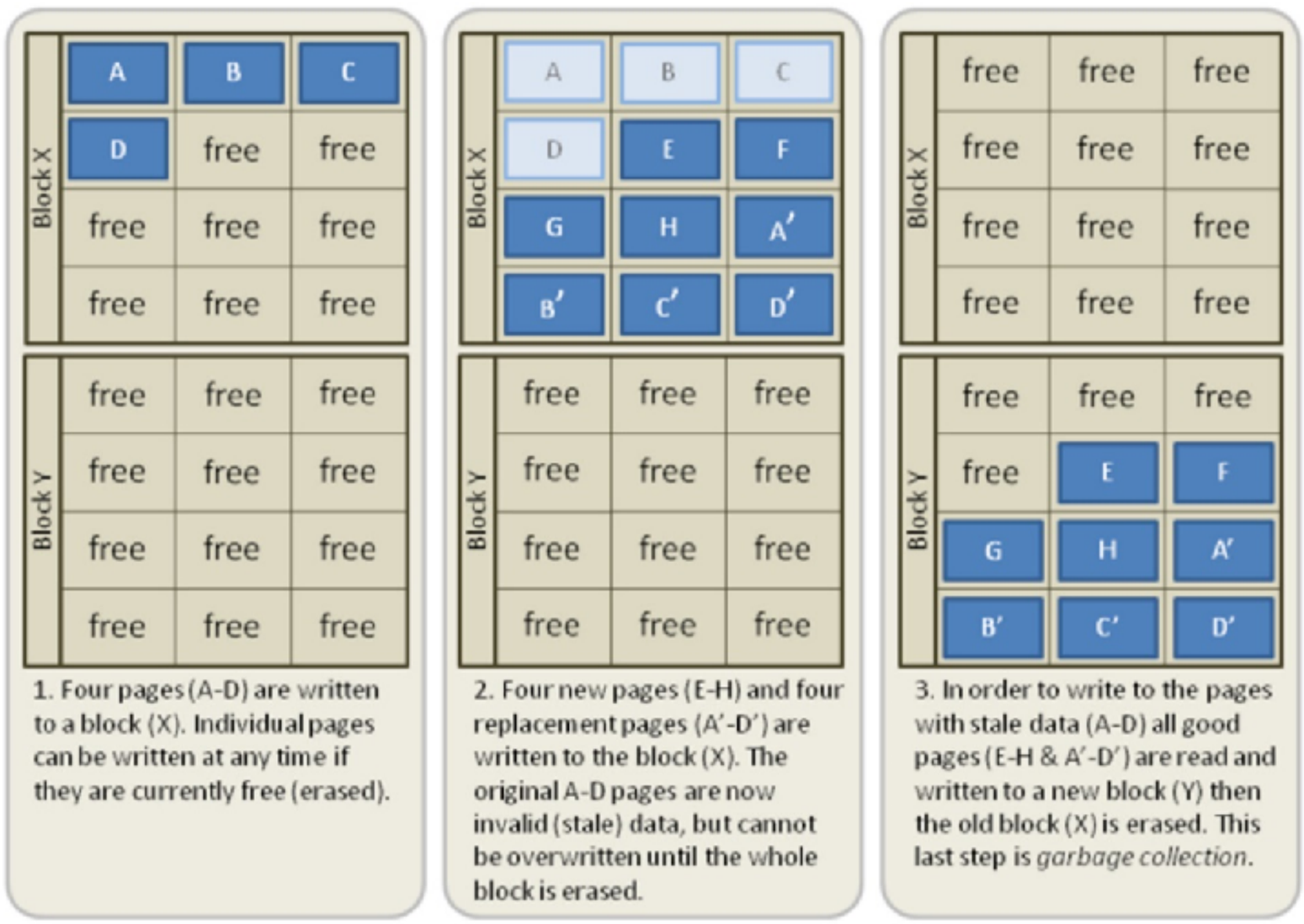
So what now? Well here come the Trim[236, 298] operation. When you delete data on your SDD, your OS should support what is called a Trim operation command and **could (should)** issue this Trim command to the SSD drive periodically (daily, weekly, monthly…). This Trim command will then let know the SSD drive controller that there are pages within blocks containing data which are now free to be actually really deleted without deleting anything itself.

Trim should be enabled by default on all modern Operating Systems detecting an SSD drive covered in this guide (MacOS, Windows 10, Ubuntu, Qubes OS…).

If Trim operations are not done regularly (or at all), then the data is never deleted pro-actively and at some point all the blocks and pages will be occupied by data. Your OS won't see this and will just see free space as you delete files but your SSD controller will not (this is called Write Amplification[299]). This will then force the SSD controller to erase those pages and blocks on the fly which will reduce the write performance. This is because while your OS/SSD can write data to any free page in any bock, erasure is only possible on entire blocks therefore forcing your SSD to perform many operations to write new data. Overwriting is just not possible. This will defeat the wear-leveling system and cause performance degradation off SSD over time. Every time you delete a file on an SSD, your OS should issue a Trim command along with the deletion to let the SSD controller know the pages containing the file data are now free for deletion.

So Trim itself does not delete any data but just marks it for deletion. Data deleted without using Trim (if Trim has been disabled/blocked/delayed for instance) will still be deleted at some point by the SSD garbage collection or if you want to overwrite what the OS sees at free space. But it might stick around for a bit longer than if you use Trim.

Here is an illustration from Wikipedia showing how it works on an SSD drive:



As you can see in the above illustration, data (from a file) will be written to the 4 first pages of Block X. Later new data will be written to the remaining pages and the data from the first files will be marked as invalid (for instance by a Trim operation when deleting a file). As explained on https://en.wikipedia.org/wiki/Trim_(computing); the erase operation can only be done on entire blocks (and not on single pages).

In addition to marking files for deletion (on reputable SSD drives) Trim usually makes those unreadable using a method called "Deterministic Read After Trim" or "Deterministic Zeroes After Trim". This means that if an adversary tries to read data from a trimmed page/block and somehow manages to disable garbage collection, the controller will not return any meaningful data.

Trim is your ally and should always be enabled when using an SSD drive and should offer sufficient reasonable protection. And this is also the reason you should not use Veracrypt Plausible deniability on a Trim enabled SSD as this feature is incompatible with Trim[300].

Garbage Collection:

Garbage collection[301] is an internal process running within your SSD drive that looks for data marked for erasure. This process is done by the SSD controller and you have no control over it. If you go back to the illustration above, you will see that Garbage collection is the last step and will notice that some pages are marked for deletion in a specific block, then copy the valid pages (not marked for deletion) to a different free destination block and then will be able to erase the source block entirely.

Garbage collection in itself does NOT require Trim to function but it will much faster and more efficient if Trim is performed. Garbage collection is one of the processes that will actually erase data from your SSD drive permanently.

Conclusion:

So the fact is that it's very unlikely[302,303] and difficult for a forensic examiner to be able to recover data from a Trimmed SSD but it's not completely impossible either[304,305,306] if they're fast enough and have access to extensive equipment, skills and motivation.

Within the context of this guide which also uses full disk encryption. Deletion and Trim should be reasonably enough on any SSD drive and will be recommended as the standard method of deletion.

How to securely wipe your whole Laptop/Drives if you want to erase everything:

So you want to be sure. In order to achieve 100% secure deletion on an SSD drive, we will need to use specific SDD techniques (If you're using an HDD drive, skip this part and go to your OS of choice):

- Easy options for less experienced users:
  - If available, just use the Secure Erase option available from your BIOS/UEFI (ATA/NVME Secure Erase or Sanitize).
  - Just re-install a fresh operating system (delete/quick format the drive) and re-encrypt it. The full disk encryption process should erase all previous data from the disk.
  - Buy PartedMagic[307] for 11$ and use it to erase any disk.
- Technical options for more advanced users:
  - ATA/NVMe Secure Erase: This method will remove the mapping table that keeps track of allocated data on the storage Blocks but does not destroy the actual data.
  - ATA/NVMe Sanitize Crypto Scramble (aka Instant Secure Erase, Crypto Erase), which applies to self-encrypting SSD drives: This method will change the encryption key of the self-encrypting SDD drive and render all the data stored in it unreadable.
  - ATA/NVMe Sanitize Block Erase: This method performs an actual block erase on every storage block and will actually destroy the data and change the encryption key if present.
  - ATA/NVMe Sanitize Overwrite **(very slow, could be dangerous and not recommended)**: This method performs a block erase and then overwrite every storage block (it's the same as Block Erase but will overwrite data in addition). This method is overkill and not necessary IMHO.

For maximum overkill paranoia security, Sanitize Block Erase option should be preferred but Secure Erase is probably more than enough when taking into account your drive is already encrypted. Unfortunately are no **free** easy (bootable with a graphical menu) all-in-one tools available and you'll be left with either going with drive manufacturers provided tools, the free manual hdparm[308] and nvme-cli[309] utilities or going with a commercial tool such as PartedMagic.

This guide will therefore recommend the use of the free utilities hdparm and nvme-cli using a Live System Rescue system.

If you can afford it, just buy Parted Magic for 11$ which provides an easy to use graphical tool for wiping SSD drives using the option of your choice[310,311].

Note: Again before proceeding, you should check your BIOS as some will offer a built-in tool to securely erase your drive (ATA/NVMe Secure Erase or ATA/NVMe Sanitize). If this is available, you should use that and the following steps won't be necessary. Check this before proceeding to avoid the hassle, see Appendix M: BIOS/UEFI options to wipe disks in various Brands).

Linux (all versions including Qubes OS):

System/Internal SSD:

- Option A: Check if your BIOS/UEFI has a built-in option to do so and if it does, use the correct option ("ATA/NVMe Secure Erase" or "ATA/NVMe Sanitize"). Do not use wipe with passes on an SSD drive.
- Option B: See Appendix D: Using System Rescue to securely wipe an SSD drive.
- Option C: Wipe your disk and re-install Linux with a new full disk encryption to overwrite all sectors with new encrypted data. **This method will be very slow compared to Option A and B as it will slowly overwrite your whole SDD.**

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

External SSD:

First please see Appendix K: Considerations for using external SSD drives

Trim should be sufficient in most cases and you could just use the blkdiscard command to force an entire device trim as explained here: https://wiki.archlinux.org/index.php/Solid_state_drive#Trim_an_entire_device

If your USB controller and USB SSD disk supports Trim and ATA/NVMe secure erase, you could wipe them cautiously using hdparm using the same method as the System Disk above except you won't install Linux on it obviously. Keep in mind tho that this is not recommended (see Considerations above).

If it doesn't support Trim and/or ATA secure erase, you could (not securely) wipe the drive normally (without passes like an HDD) and re-encrypt it completely using your utility of choice (LUKS or Veracrypt for instance). The full disk decryption and re-encryption process will overwrite the entirety of the SSD disk and should ensure a secure wipe.

Alternatively, you could also (not securely) wipe the disk normally and then fill it completely with pseudorandom data which should also ensure secure deletion (this can be done with BleachBit https://www.bleachbit.org/download/linux or from the command line using secure-delete using this tutorial https://superuser.com/questions/19326/how-to-wipe-free-disk-space-in-linux).

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

Internal/System HDD:

- Option A: Check if your BIOS/UEFI has a built-in options and use them and if it does, use the correct option (Wipe + Passes in the case of an HDD).
- Option B: See Appendix I: Using ShredOS to securely wipe an HDD drive:
- Option C: Wipe your disk and re-install Linux with a new full disk encryption to overwrite all sectors with new encrypted data. **This method will be very slow compared to Option A and B as it will slowly overwrite your whole HDD.**

External/Secondary HDD and Thumb Drives:

- Option A: Follow one of these tutorials:
  - https://linuxhint.com/completely_wipe_hard_drive_ubuntu/
  - https://linoxide.com/linux-command/commands-wipe-disk-linux/
  - https://wiki.archlinux.org/index.php/Securely_wipe_disk

  I recommend using dd or shred for this purpose.

- Option B: Install and use BleachBit https://www.bleachbit.org/download/linux or follow this EFF tutorial https://ssd.eff.org/en/module/how-delete-your-data-securely-linux
- Option C: See Appendix I: Using ShredOS to securely wipe an HDD drive:

Windows:

Unfortunately you won't be able to wipe your Host OS using the Microsoft built-in tools within the settings. This is because your bootloader was modified with Veracrypt and will make the operation fail. In addition, this method would not be effective with an SSD drive.

System/Internal SSD:

- Option A: Check if your BIOS/UEFI has a built-in option to do so and if it does, use the correct option ("ATA/NVMe Secure Erase" or "ATA/NVMe Sanitize"). Do not use wipe with passes on an SSD drive.
- Option B: Check Appendix J: Manufacturer tools for Wiping HDD and SSD drives:
- Option C: See Appendix D: Using System Rescue to securely wipe an SSD drive.
- Option D: Wipe your disk and re-install Windows before performing a new full disk encryption (using Veracrypt or Bitlocker) to overwrite all sectors with new encrypted data. **This method will be slower compared to Option A and B as it will overwrite your whole SSD.**

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

External SSD:

First please see Appendix K: Considerations for using external SSD drives

Use the manufacturer provided tools if possible. Those tools should provide support for safe secure erase or sanitize over USB and are available for most brands: See Appendix J: Manufacturer tools for Wiping HDD and SSD drives:

If you're not sure about the Trim support on your USB disk, (not securely) wipe it normally (simple quick format will do) and then encrypt the disk again using Veracrypt or alternatively Bitlocker. The full disk decryption and re-encryption process will overwrite the entirety of the SSD disk and should ensure a secure wipe.

Alternatively, you could also (not securely) wipe the disk normally and then fill it completely with pseudorandom data which should also ensure secure deletion (this can be done with BleachBit or PrivaZer free space erase options). See Extra Tools Cleaning:.

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

Internal/System HDD:

- Option A: Check if your BIOS/UEFI has a built-in option to do so and if it does, use the correct option (Wipe + Passes).
- Option B: Check Appendix J: Manufacturer tools for Wiping HDD and SSD drives:
- Option C: See Appendix I: Using ShredOS to securely wipe an HDD drive:

External/Secondary HDD and Thumb Drives:

- Option A: Check Appendix J: Manufacturer tools for Wiping HDD and SSD drives:
- Option B: Use external tools such as:
  - Eraser (open-source): https://eraser.heidi.ie/download/
  - KillDisk Free: http://killdisk.com/killdisk-freeware.htm
- Option C: See Appendix I: Using ShredOS to securely wipe an HDD drive:

MacOS:

System/Internal SSD:

Unfortunately the MacOS Recovery disk utility will not be able to perform a secure erase of your SSD drive as stated in Apple documentation https://support.apple.com/en-gb/guide/disk-utility/dskutl14079/mac.

In most cases, if your disk was encrypted with Filevault and you just perform a normal erase, it should be "enough" according to them. It's not according to me so you have no option besides re-installing MacOS again and re-encrypt it with Filevault again after re-installing. This should perform a "crypto erase" by overwriting your previous install and encryption. This method will be quite slow unfortunately.

If you want to do a faster secure erase (or have no time to perform a re-install and re-encryption), you can try using the method described in Appendix D: Using System Rescue to securely wipe an SSD drive.**(This will not work on M1 Macs). Be careful tho as this will also erase your recovery partition which is needed to reinstall MacOS.**

External SSD:

First please see Appendix K: Considerations for using external SSD drives

If your USB controller and USB SSD disk supports Trim and ATA secure erase, and if Trim is enabled on the disk by MacOS, you can just wipe the whole disk normally and data should not be recoverable on recent disks.

If you're not sure about Trim support or want more certainty, you can (not securely) wipe it using MacOS disk utility before fully re-encrypting them again using these two tutorials from Apple:

- https://support.apple.com/guide/disk-utility/erase-and-reformat-a-storage-device-dskutl14079/mac
- https://support.apple.com/guide/disk-utility/encrypt-protect-a-storage-device-password-dskutl35612/mac or using Veracrypt full disk encryption.

The full disk re-encryption process will overwrite the entirety of the SSD disk and should ensure a secure wipe.

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

External HDD and Thumb Drives:

Follow this tutorial: https://support.apple.com/guide/disk-utility/erase-and-reformat-a-storage-device-dskutl14079/mac and use the secure erase option from Disk Utility which should work fine on HDD and Thumb drives.

How to securely delete specific files/folders/data on your HDD/SDD and Thumb drives:

The same principles from the previous chapters apply to this one. The same issues arise too.

With an HDD drive, you can securely delete files by just deleting it and then apply one of more "passes" to overwrite the data in question. This can be done with many utilities on all OSes.

With an SSD drive however, again everything becomes a bit complicated because you're never sure anything is really deleted due to wear leveling, reliance on the Trim operation and garbage collection of the drive. An adversary that has the decryption key of your SDD (whether it's LUKS, Filevault 2, Veracrypt or Bitlocker) could unlock your drive and then attempt recovery using classic recovery utilities[312] and could succeed if the data wasn't trimmed properly. But this is again highly unlikely.

Since the Trim operation is not continuous on most recent hard drive but scheduled, simply forcing a Trim operation should be enough. But again the only way to be 100% sure a file is securely deleted from your unlocked encrypted SDD is to again overwrite all the free space after deletion of the files in question or to decrypt/re-encrypt the drive. But I think this is overkill and not necessary. A simple disk wide Trim should be sufficient.

Remember tho that no matter the deletion method you use for any file on any medium (HDD drive, SDD, USB Thumb drive). It will probably leave other traces (logs, indexing, shellbags …) within your system and those traces will also need to be cleaned. Also remember that your drives should be fully encrypted and so this is most likely an extra measure. More on that later in the Some additional measures against forensics: section.

Windows:

Remember you cannot use Trim at all if you're using Plausible Deniability on an SSD drive against all recommendations.

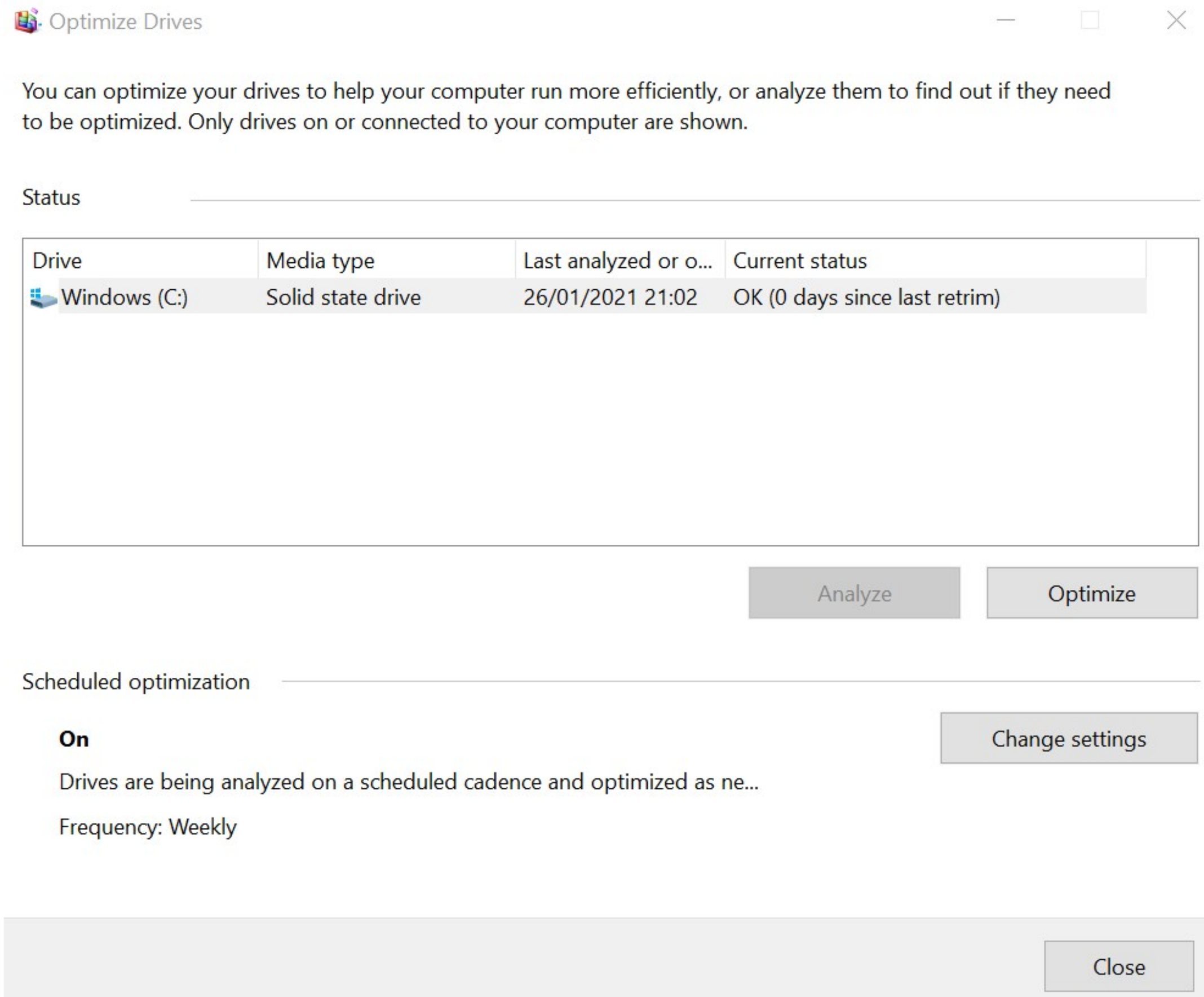
System/Internal SSD drive:

At this stage, and just delete the file permanently (empty the recycle bin) and trim/garbage collection will do the rest. This should be sufficient.

If you don't want to wait for the periodic Trim (set to Weekly by default in Windows 10), you could also force a disk wide Trim using the Windows native Optimize tool (see Appendix H: Windows Cleaning Tools).

If data was deleted by some utility (for instance by Virtualbox when reverting a snapshot), you could also issue a disk wide Trim to clean anything remaining using the same Optimize tool.

Just open Windows Explorer, Right Click on your System Drive and click Properties. Select Tools. Click Optimize and then Optimize again to force a Trim. You're done. I think that's probably enough in my opinion.



If you want more security and don't trust the Trim operation then you'll have no option but to either:

- Decrypt and re-encrypt (using Veracrypt or Bitlocker) the whole drive to overwrite all free space after data deletion. This will ensure overwriting of all the free space.
- Trim and then fill up the entire free space of the disk using a utility such as BleachBit or PrivaZer.

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

Internal/External HDD or a USB Thumb Drive:

Please refer to Appendix H: Windows Cleaning Tools and pick a utility before proceeding.

The process is very simple depending on the tool you picked from the Appendix:

- Right Click a file/folder:
  - PrivaZer: Delete without a trace
  - BleachBit: Shred with BleachBit (or see this tutorial from the EFF https://ssd.eff.org/en/module/how-delete-your-data-securely-windows)

In the case of USB thumb drives, consider wiping free space using one of the above utilities after file deletion or wiping them completely using Eraser / KillDisk as instructed previously.

External SSD drive:

First please see Appendix K: Considerations for using external SSD drives

If Trim is supported and enabled by Windows for your external SSD drive. There should be no issue in securely deleting data normally just with normal delete commands. Additionally, you could also force a Trim using the Windows native Optimize tool (see Appendix H: Windows Cleaning Tools):

Just open Windows Explorer, Right Click on your System Drive and click Properties. Select Tools. Click Optimize and then Optimize again to force a Trim. You're done. I think that's probably enough in my opinion.

If Trim is not supported or you're not sure, you might have to ensure secure data deletion by:

- Filling up all the free space after any deletion (using BleachBit or PrivaZer for instance).
- Decrypt and Re-encrypt the disk with a different key after each deletion (using Veracrypt or Bitlocker).

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

Linux (non Qubes OS):

System/Internal SSD drive:

Just permanently delete the file (and empty recycle bin) and it should be unrecoverable due to Trim operations and garbage collection.

If you don't want to wait for the periodic Trim (set to Weekly by default in Ubuntu), you could also force a disk wide Trim by running "fstrim --all" from a terminal. This will issue an immediate trim and should ensure sufficient security. This utility is part of the "util-linux" package on Debian/Ubuntu and should be installed by default on Fedora.

If you want more security and don't trust the Trim operation then you'll have no option but to either:

- Decrypt and re-encrypt (using LUKS for instance following this tutorial https://wiki.archlinux.org/index.php/dm-crypt/Device_encryption#Re-encrypting_devices) the whole drive to overwrite all free space after data deletion. This will ensure overwriting of all the free space.
- Trim using "fstrim --all" and then fill up the entire free space of the disk using an utility such as:
  - BleachBit https://www.bleachbit.org/download/linux
  - Install secure-delete package and use sfill on the root of the drive:
    - "sudo sfill -l -l /" for instance should do the trick (this will take a substantial amount of time)
  - Use the old school dd method (taken from this answer https://superuser.com/questions/19326/how-to-wipe-free-disk-space-in-linux) run these commands on the drive you want to fill:
    - dd if=/dev/zero of=zero.small.file bs=1024 count=102400
    - dd if=/dev/zero of=zero.file bs=1024
    - sync ; sleep 60 ; sync
    - rm zero.small.file
    - rm zero.file

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

Internal/External HDD drive or a Thumb Drive:

- You can do this the graphical way with BleachBit following this tutorial from the EFF: https://ssd.eff.org/en/module/how-delete-your-data-securely-linux
- Or you can do this from the command line following this tutorial: https://linuxhint.com/completely_wipe_hard_drive_ubuntu/ (For this purpose I recommend wipe and shred).

External SSD drive:

First please see Appendix K: Considerations for using external SSD drives

If Trim is supported and enabled by your Linux Distribution for your external SSD drive. There should be no issue in securely deleting data normally and just issue an “fstrim --all” from terminal to trim the drive. This utility is part of the “util-linux” package on Debian/Ubuntu and should be installed by default on Fedora.

If Trim is not supported or you want to be sure, you might have to ensure secure data deletion by filling up the entire free space of the disk using a utility such as:

- Decrypt and re-encrypt (using LUKS using this tutorial https://wiki.archlinux.org/index.php/dm-crypt/Device_encryption#Re-encrypting_devices or Veracrypt from the graphical interface for instance) the whole drive to overwrite all free space after data deletion. This will ensure overwriting of all the free space.
- Fill the free space using one of those methods:
    - BleachBit https://www.bleachbit.org/download/linux
    - Install secure-delete package and use sfill on the root of the drive:
        - “sudo sfill -l -l /” for instance should do the trick (this will take a substantial amount of time)
    - Use the old school dd method (taken from this answer https://superuser.com/questions/19326/how-to-wipe-free-disk-space-in-linux) run these commands:
        - dd if=/dev/zero of=zero.small.file bs=1024 count=102400
        - dd if=/dev/zero of=zero.file bs=1024
        - sync ; sleep 60 ; sync
        - rm zero.small.file
        - rm zero.file

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don’t, wear-leveling mechanisms might prevent this from working properly.

Linux (Qubes OS):

System/Internal SSD drive:

As with other Linux distros, normal deletion and trim should be sufficient on most SSD drives. So just permanently delete the file (and empty any recycle bin) and it should be unrecoverable due to periodic Trim operations and garbage collection.

Please follow this documentation to Trim within Qubes OS: https://github.com/Qubes-Community/Contents/blob/master/docs/configuration/disk-trim.md

As with other Linux Systems, if you want more security and don’t trust the Trim operation then you’ll have no option but to either:

- Decrypt and re-encrypt the whole drive to overwrite all free space after data deletion. This will ensure overwriting of all the free space. I didn’t find a reliable tutorial on how to do this safely on Qubes OS but it’s possible this Tutorial could work as well https://wiki.archlinux.org/index.php/dm-crypt/Device_encryption#Re-encrypting_devices (at your own risk, this has not been tested yet).
- Refer to this Documentation (https://github.com/Qubes-Community/Contents/blob/master/docs/configuration/disk-trim.md) and then trim using “fstrim --all” and then fill up the entire free space of the disk using an utility such as:
    - BleachBit https://www.bleachbit.org/download/linux
    - Install secure-delete package and use sfill on the root of the drive:
        - “sudo sfill -l -l /” for instance should do the trick (this will take a substantial amount of time)
    - Use the old school dd method (taken from this answer https://superuser.com/questions/19326/how-to-wipe-free-disk-space-in-linux) run these commands on the drive you want to fill:
        - dd if=/dev/zero of=zero.small.file bs=1024 count=102400
        - dd if=/dev/zero of=zero.file bs=1024
        - sync ; sleep 60 ; sync
        - rm zero.small.file

        - rm zero.file

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

Internal/External HDD drive or a Thumb Drive:

Use the same method as Linux from a Qubes connected to that specific USB device

- You can do this the graphical way with BleachBit following this tutorial from the EFF: https://ssd.eff.org/en/module/how-delete-your-data-securely-linux
- Or you can do this from the command line following this tutorial: https://linuxhint.com/completely_wipe_hard_drive_ubuntu/ (For this purpose I recommend wipe and shred).

External SSD drive:

First please see Appendix K: Considerations for using external SSD drives

If Trim is supported and enabled by your Linux Distribution for your external SSD drive. There should be no issue in securely deleting data normally and just issue an "fstrim --all" from terminal to trim the drive. Refer to this Documentation (https://github.com/Qubes-Community/Contents/blob/master/docs/configuration/disk-trim.md) to enable trim on a drive.

If Trim is not supported or you want to be sure, you might have to ensure secure data deletion by filling up the entire free space of the disk using a utility from a Qubes connected to the USB device in question:

- Decrypt and re-encrypt (using LUKS using this tutorial https://wiki.archlinux.org/index.php/dm-crypt/Device_encryption#Re-encrypting_devices or Veracrypt from the graphical interface for instance) the whole drive to overwrite all free space after data deletion. This will ensure overwriting of all the free space.
- Fill the free space using one of those methods:
    - BleachBit https://www.bleachbit.org/download/linux
    - Install secure-delete package and use sfill on the root of the drive:
        - "sudo sfill -l -l /" for instance should do the trick (this will take a substantial amount of time)
    - Use the old school dd method (taken from this answer https://superuser.com/questions/19326/how-to-wipe-free-disk-space-in-linux) run these commands:
        - dd if=/dev/zero of=zero.small.file bs=1024 count=102400
        - dd if=/dev/zero of=zero.file bs=1024

Repeat these steps on any other partition if there separate partitions on the same SSD drive before deleting the files.

        - sync ; sleep 60 ; sync
        - rm zero.small.file
        - rm zero.file

Repeat these steps on any other partition if there separate partitions on the same SSD drive.

Keep in mind all of these options need to be applied on the entire physical drive and not on a specific partition/volume. If you don't, wear-leveling mechanisms might prevent this from working properly.

MacOS:

System/Internal SSD drive:

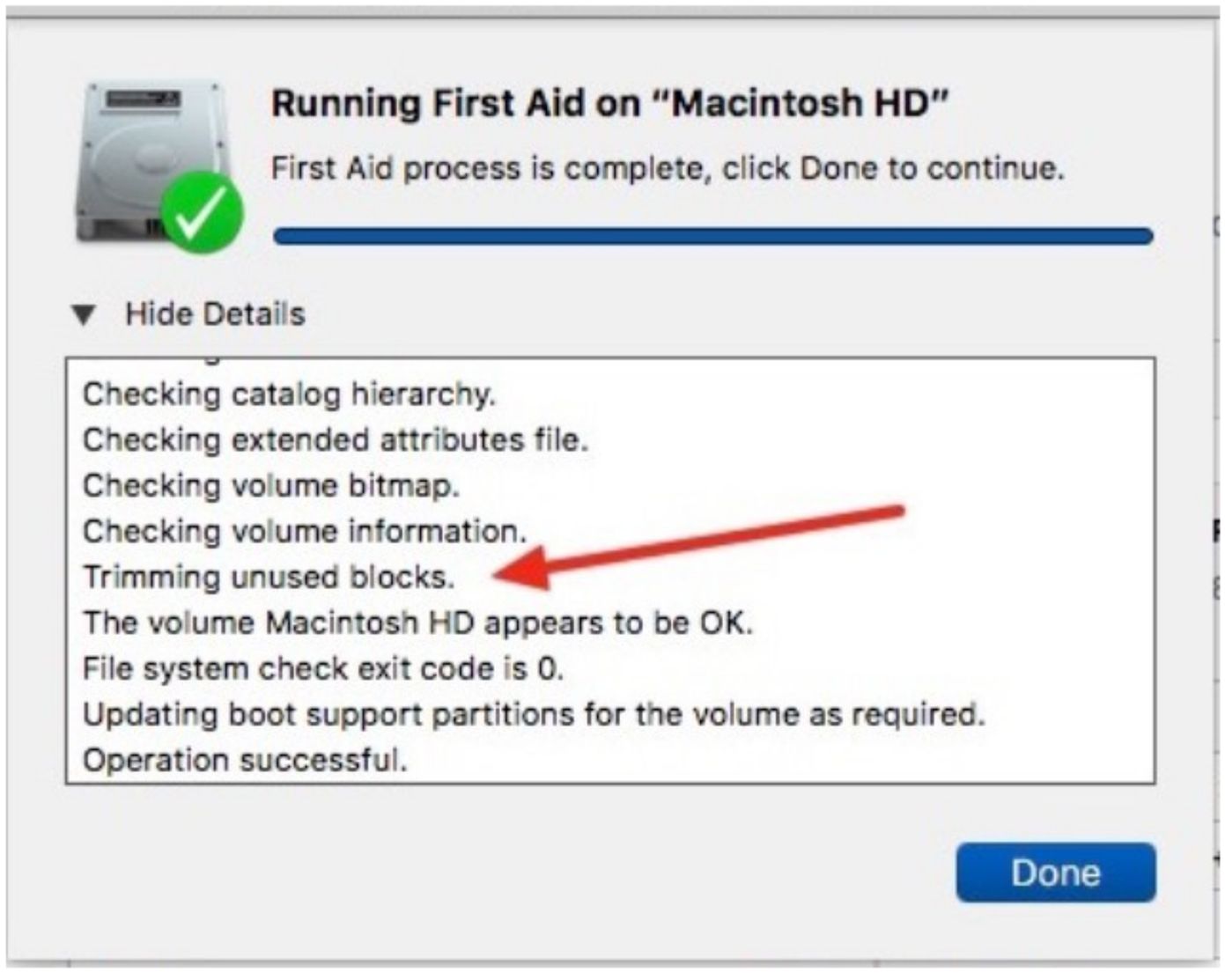
Just permanently delete the file (and empty recycle bin) and it should be unrecoverable due to trim operations and garbage collection.

- If your file system is APFS, you don't need to worry about Trim, it apparently happens asynchronously as the OS writes data[313] according to their own documentation.

"Does Apple File System support TRIM operations?

Yes. TRIM operations are issued asynchronously from when files are deleted or free space is reclaimed, which ensures that these operations are performed only after metadata changes are persisted to stable storage."

- If your file system is HFS+, you could run First Aid on your System Drive from the Disk Utility which should perform a Trim operation in the details (https://support.apple.com/en-us/HT210898)



System/Internal, External HDD drive or a Thumb Drive:

Unfortunately Apple has removed the secure erase options from the trash bin even for HDD drives[314]. So you're left with using other tools:

- Permanent Eraser http://www.edenwaith.com/products/permanent%20eraser/
- From the terminal you can use the "rm –P filename" command which should erase the file and overwrite it as explained in this EFF tutorial https://ssd.eff.org/en/module/how-delete-your-data-securely-macos.

In the case of USB thumb drives, consider wiping them completely using Disk Utility as instructed previously.

External SSD drive:

First please see Appendix K: Considerations for using external SSD drives

If Trim is supported and enabled by MacOS for your external SSD drive. There should be no issue in securely deleting data.

If Trim is not supported, you might have to ensure secure data deletion by:

- Filling up all the free space after any deletion using the Linux Method above (dd).
- Decrypt and Re-encrypt the disk with a different key after each deletion (using Disk Utility or Veracrypt).

Some additional measures against forensics:

Note that the same SSD issue discussed in the previous section will arise here. You can never really be absolutely 100% sure your SSD data is deleted when you ask it to do so unless you wipe the whole drive using specific methods above.

I'm not aware of any 100% reliable method to selectively securely delete single files on SSD drives unless overwriting ALL the free space (which might reduce the lifespan of your SSD) after Deletion + Trim of these files. Without doing that, you'll have to trust the SSD Trim operation **which in my opinion is enough**. **It's reasonable and again very unlikely that forensics will be able to restore your files after a Deletion with Trim.**

In addition, most of these measures here should not be needed since your whole drive should be encrypted and therefore your data should not be accessible for forensic analysis through SSD/HDD examination anyway. So these are just "bonus measures" for weak/unskilled adversaries.

Consider also reading this documentation if you're going with Whonix https://www.whonix.org/wiki/Anti-Forensics_Precautions as well as their general hardening tutorial for all platforms here https://www.whonix.org/wiki/System_Hardening_Checklist

Removing Metadata from Files/Documents/Pictures:

Pictures:

On Windows I would recommend this NirSoft utility that allows you to view the EXIF data of pictures https://www.nirsoft.net/utils/exif_data_view.html. Alternatively on Windows and Linux I would recommend ExifTool that allows viewing AND editing those properties https://exiftool.org/.

Here is a tutorial to remove metadata from a Picture using OS provided tools: https://www.purevpn.com/internet-privacy/how-to-remove-metadata-from-photos

MS Office Documents:

Here is a tutorial to remove metadata from Office documents: https://support.microsoft.com/en-us/office/remove-hidden-data-and-personal-information-by-inspecting-documents-presentations-or-workbooks-356b7b5d-77af-44fe-a07f-9aa4d085966f. Make sure however that you do use the latest version of Office with the latest security updates.

LibreOffice Documents:

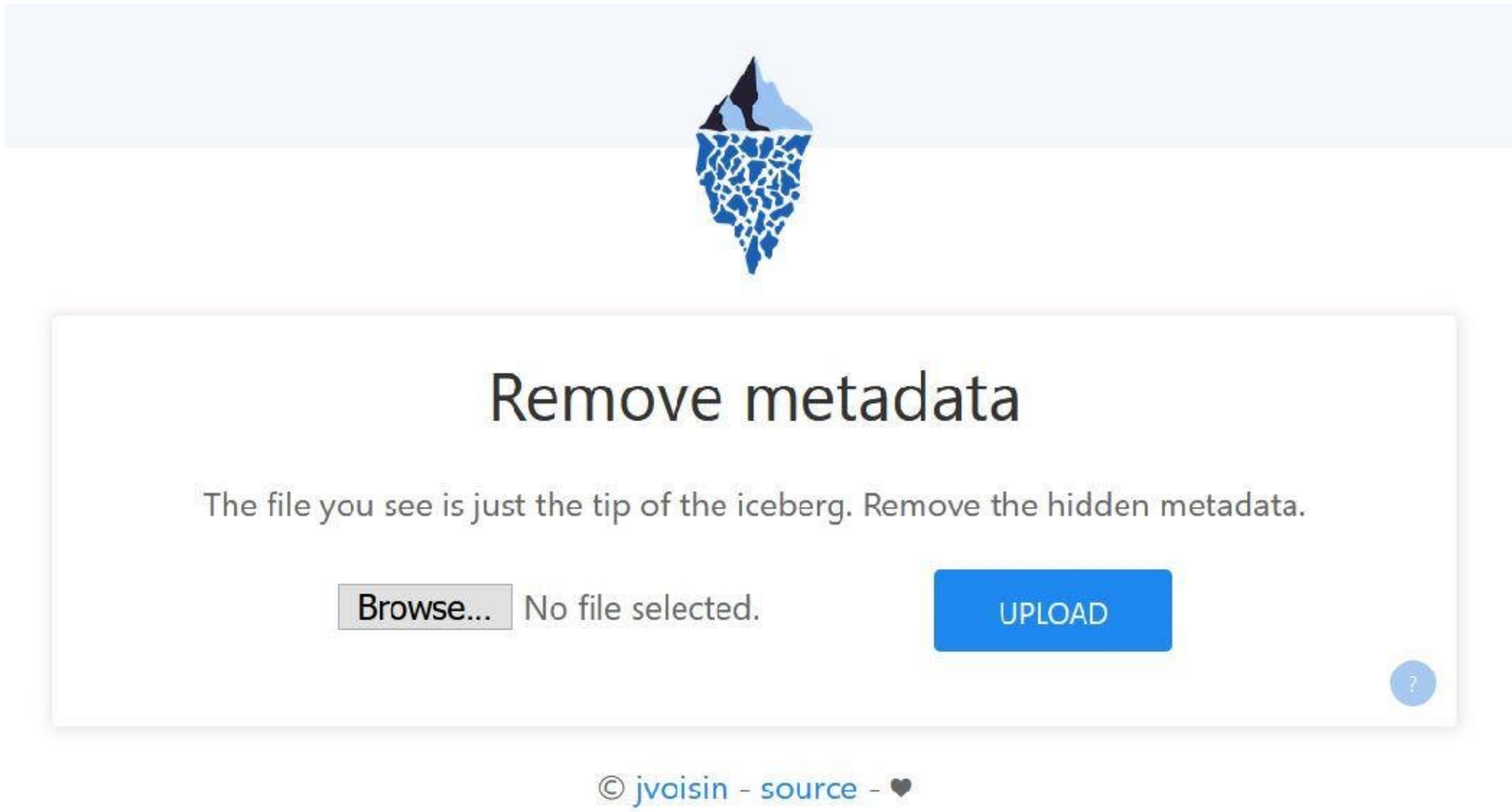
Go to Tools > Options > Security and Check:

- All the warnings
- Remove Personal information on saving

All-in-one Tool:

The best tool IMHO to remove metadata from various documents is the open-source mat2 recommended by privacytools.io[315] (https://0xacab.org/jvoisin/mat2) which you can use on Linux quite easily. I never managed to make it work properly within Windows due various dependencies issues despite the provided instructions. It's however very straightforward to install and use on Linux.

So I would suggest creating a small Debian VM within Virtualbox (behind your Whonix Gateway) which you can then use from your other VMs to analyze various files from a convenient web interface. For this see Appendix L: Creating a mat2-web guest VM for removing metadata from files



Mat2 is also pre-installed on the Whonix Workstation VM[316] and available on TAILS by default[317].

TAILS:

TAILS is great for this, you have nothing to worry about even if you use an SSD drive. Shut it down and it's all gone as soon as the memory decays.

Whonix:

Note that it's possible to run Whonix in Live mode leaving no traces when you shut down the VMs, consider reading their documentation here https://www.whonix.org/wiki/VM_Live_Mode and here https://www.whonix.org/wiki/Warning#Whonix_.E2.84.A2_Persistence_vs_Live_vs_Amnesic.

MacOS:

Guest OS:

Revert to a previous snapshot on Virtualbox (or any other VM software you're using) and perform a Trim command on your Mac using Disk Utility by executing a first-aid on the Host OS again as explained at the end of the next section.

Host OS:

Most of the info from this section can also be found at this nice guide https://github.com/drduh/macOS-Security-and-Privacy-Guide

Quarantine Database (used by Gatekeeper and XProtect):

MacOS (up to and included Big Sur) keeps a Quarantine SQL Database of all the files you ever downloaded from a Browser. This database is located at "~/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2".

You can query it yourself by running the following command from terminal: "echo 'SELECT datetime(LSQuarantineTimeStamp + 978307200, "unixepoch") as LSQuarantineTimeStamp, ' \
'LSQuarantineAgentName, LSQuarantineOriginURLString, LSQuarantineDataURLString from LSQuarantineEvent;' | \
sqlite3 /Users/$USER/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2"

Obviously this is a goldmine for forensics and you should disable this:

- Run the following command to clear the database completely: ":>~/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2"

- Run the following command to lock the file and prevent further download history from being written there: "sudo chflags schg ~/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2"

Refer to this section of this guide for further information https://github.com/drduh/macOS-Security-and-Privacy-Guide#gatekeeper-and-xprotect

In addition to this convenient database, each saved file will also carry detailed file system HFS+/APFS attributes showing for instance when it was downloaded, with what and from where.

You can view these just by opening a terminal and typing "mdls filename" and "xattr -l filename" on any downloaded file from any browser.

To remove such attributes, you'll have to do it manually from the terminal:

- Run "xattr -d com.apple.metadata:kMDItemWhereFroms filename" to remove the origin
    - You can also just use -dr to do it recursively on a whole folder/disk
- Run "xattr -d com.apple.quarantine filename" to remove the quarantine reference
    - You can also just use -dr to do it recursively on a whole folder/disk
- Verify by running "xattr –l filename" and there should be no output

(Note that Apple has removed the convenient xattr –c option that would just remove all attributes at once so you'll have to do this for each attribute on each file)

These attributes and entries will stick even if you clear your Browser history and this is obviously bad for privacy (right?) and I'm not aware of any convenient tool that will deal with those at the moment.

Fortunately there are some mitigations for avoiding this issue in the first place as these attributes and entries are set by the browsers. So I tested various browsers and here are the results:

Browser	Quarantine DB Entry	Quarantine Attribute	Origin Attribute
Safari (Normal)	Yes	Yes	Yes
Safari (Private Window)	No	No	No
Firefox (Normal)	Yes	Yes	Yes
Firefox (Private Window)	No	No	No
Chrome (Normal)	Yes	Yes	Yes
Chrome (Private Window)	Partial (timestamp only)	No	No
Brave (Normal)	Partial (timestamp only)	No	No
Brave (Private Window)	Partial (timestamp only)	No	No
Brave (Tor Window)	Partial (timestamp only)	No	No
Tor Browser	No	No	No

As you can see for yourself the easiest mitigation is to just use Private Windows. These do not write those origin/quarantine attributes and do not store the entries in the QuarantineEventsV2 database.

Clearing the QuarantineEventsV2 is easy as explained above. Removing the attributes takes some work. **Brave is the only tested browser that will not store those attributes by default in normal operations.**

Various Artifacts:

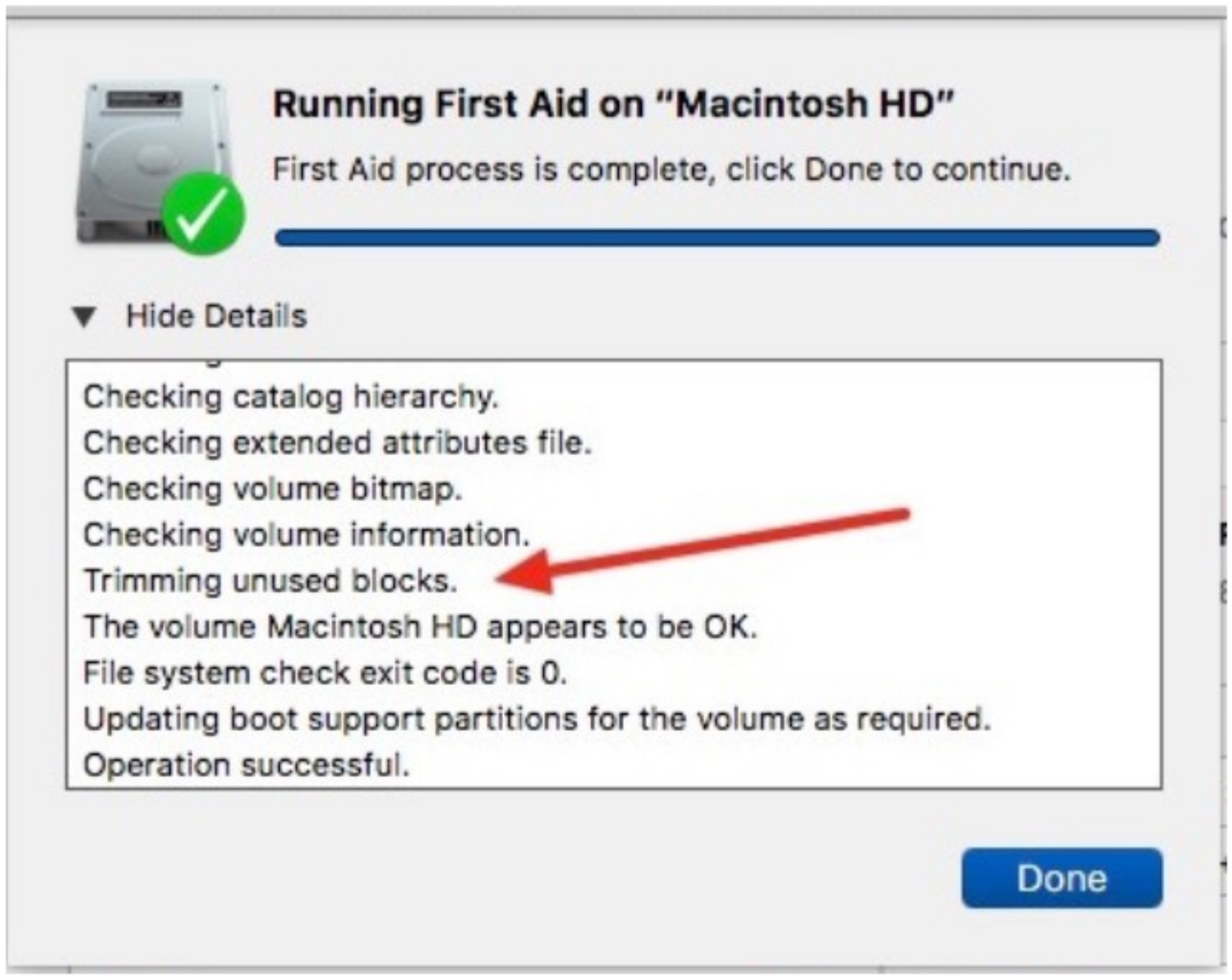
In addition, MacOS keeps various logs of mounted devices, connected devices, known networks, analytics, documents revisions…

See this section of this guide for guidance on where to find and how to delete such artifacts: https://github.com/drduh/macOS-Security-and-Privacy-Guide#metadata-and-artifacts

Many of those can be deleted using some various commercial third party tools but I would personally recommend using the free and well known Onyx which you can find here: https://www.titanium-software.fr/en/onyx.html. Unfortunately it's closed-source but it's notarized, signed and has been trusted for many years.

Force a Trim operation after cleaning:

- If your file system is APFS, you don't need to worry about Trim, it happens asynchronously as the OS writes data.
- If your file system is HFS+ (or any other than APFS), you could run First Aid on your System Drive from the Disk Utility which should perform a Trim operation in the details (https://support.apple.com/en-us/HT210898).



Linux (Qubes OS):

Please consider their guidelines https://github.com/Qubes-Community/Contents/blob/master/docs/security/security-guidelines.md

If you're using Whonix on Qubes OS, please consider following some of their guides:

- Whonix System Hardening guide https://www.whonix.org/wiki/System_Hardening_Checklist
- Enabling App Armor on Qubes https://www.whonix.org/wiki/Qubes/AppArmor
- Also consider the use of Linux Kernel Guard https://www.whonix.org/wiki/Linux_Kernel_Runtime_Guard_LKRG

Linux (non-Qubes):

Guest OS:

Revert to a previous snapshot of the Guest VM on Virtualbox (or any other VM software you're using) and perform a trim command on your laptop using "fstrim --all". This utility is part of the "util-linux" package on Debian/Ubuntu and should be installed by default on Fedora. Then switch to the next section.

Host OS:

Normally you shouldn't have traces to clean within the Host OS since you're doing everything from a VM if you follow this guide.

Nevertheless you might want to clean some logs. Just use this convenient tool: https://github.com/sundowndev/go-covermyass (instructions on the page)

After cleaning up, make sure you have the fstrim utility installed (should be by default on Fedora) and part of the "util-linux" package on Debian/Ubuntu. Then just run "fstrim --all" on the Host OS. This should be sufficient on SSD drives as explained earlier.

Consider the use of Linux Kernel Guard as an added measure https://www.whonix.org/wiki/Linux_Kernel_Runtime_Guard_LKRG

Windows:

Guest OS:

Revert to a previous snapshot on Virtualbox (or any other VM software you're using) and perform a trim command on your Windows using the Optimize as explained in the end of the next section

Host OS:

Now that you had a bunch of activities with your VMs or Host OS, you should take a moment to cover your tracks. **Most of these steps should not be undertaken on the Decoy OS in case of use of plausible deniability. This is because you want to keep decoy/plausible traces of sensible but not secret activities available for your adversary. If everything is clean then you might raise suspicion.**

Diagnostic Data and Telemetry:

First let's get rid of any diagnostic data that could still be there:

- After each use of your Windows devices, go into Settings, Privacy, Diagnostic & Feedback, and Click Delete.

Then let's re-randomize the MAC addresses of your Virtual Machines and the Bluetooth Address of your Host OS.

- After each shutdown of your Windows VM, change its MAC address for next time by going into Virtualbox > Select the VM > Settings > Network > Advanced > Refresh the MAC address.
- After each use of your Host OS Windows (your VM shouldn't have Bluetooth at all), Go into the Device Manager, Select Bluetooth, Disable Device and Re-Enable device (this will force a randomization of the Bluetooth Address).

Event logs:

Windows Event logs will keep many various pieces of information that could contain traces of your activities such as the devices that were mounted (including Veracrypt NTFS volumes for instance[235]), your network connections, app crash information and various errors. It's always best to clean those up regularly. Do not do this on the Decoy OS.

- Start , search for Event Viewer and launch Event Viewer:
    - Go into Windows logs.
    - Select and clear all 5 logs using right click.

Veracrypt History:

By default, Veracrypt saves a history of recently mounted volumes and files. You should make sure Veracrypt never saves History. Again do not do this on the Decoy OS if you're using plausible deniability for the OS. We need to keep the history of mounting the decoy Volume as part of the plausible deniability.

- Launch Veracrypt
- Make sure the "Never saves history" checkbox is checked (this should not be checked on the Decoy OS)

Now you should clean the history within any app that you used including Browser history, Cookies, Saved Passwords, Sessions, and Form History.

- Brave (in case you didn't enable cleaning on exit)
    - Go into Settings
    - Go into Shields
    - Go into Clear Browsing Data
    - Select Advanced
    - Select "All Time"
    - Check all the options
    - Clear Data
- Tor Browser
    - Just close the Browser and everything is cleaned

Wi-Fi History:

Now it's turn to clear the history of the Wi-Fi you connect to. Unfortunately Windows keeps storing a list of past Networks in the registry even if you "forgot" those in the Wi-Fi settings. As far as I know, no utilities clean those yet (BleachBit or PrivaZer for instance) so you'll have to do it the manual way:

- Launch Regedit using this tutorial: https://support.microsoft.com/en-us/windows/how-to-open-registry-editor-in-windows-10-deab38e6-91d6-e0aa-4b7c-8878d9e07b11
- Within Regedit, enter this to the address bar: " Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles "
- There you'll see a bunch of folders to the right. Each of those folders is a "Key". Each of those keys will contain information about your current known Wi-Fi or past networks you used. You can explore them one by one and see the description on the right side.
- Delete all those keys.

Shellbags:

As explained earlier, Shellbags are basically histories of accessed volumes/files on your computer. Remember that shellbags are very good sources of information for forensics[228] and you need to clean those. Especially if you mounted any "hidden volume" anywhere. Again you shouldn't do this on the Decoy OS.

- Download Shellbag Analyzer & Cleaner from https://privazer.com/en/download-shellbag-analyzer-shellbag-cleaner.php
  - Launch it
  - Analyze
  - Click Clean and select:
    - Deleted Folders
    - Folders on Network / External devices
    - Search Results
  - Select advanced
    - Check all except the two backup options (don't backup)
    - Select SDD cleanup (if you have an SSD)
    - Select 1 pass (All zero)
    - Clean

Extra Tools Cleaning:

After cleaning those previous traces, you should also use third party utilities than can be used to clean various traces. These include the traces of the files/folders you deleted.

Please refer to Appendix H: Windows Cleaning Tools before continuing.

PrivaZer:

Here are the steps for PrivaZer:

- Download and install PrivaZer from https://privazer.com/en/download.php
- Run PrivaZer after install
- Don't use their Wizard
- Select Advanced User
- Select Scan In Depth and pick your Target
- Select Everything you want to Scan and push Scan
- Select What you want cleaned (skip the shell bag part since you used the other utility for that)
  - **You should just skip the free space cleaning part if using an SSD and instead just use the native Windows Optimize function (see below) which should be more than enough. I would only use this on an HDD drive.**
- (If you did select Free Space cleaning) Select Clean Options and make sure your type of Storage if well detected (HDD vs SSD).

- (if you did select Free Space cleaning) Within Clean Options **(Be careful with this option as it will erase all the free space on the selected partition, especially if you are running the decoy OS. Do not erase the free space or anything else on the second partition as you risk destroying your Hidden OS)**
    - If you have an SSD drive:
        - Secure Overwriting Tab: Personally I would just pick Normal Deletion + Trim (Trim itself should be enough[239]). Secure Deletion with Trim[236] (1 pass) might be redundant and overkill here if you intend to overwrite the free space anyway.
        - Free Space Tab: Personally and again "just to be sure", I would select Normal Cleanup which will fill the entire free space with Data. I don't really trust Smart Cleanup as it doesn't actually fill all the free space of the SSD with Data. But again I think this is probably not needed and overkill in most cases.
    - If you have an HDD drive:
        - Secure Overwriting Tab: I would just pick Secure Deletion (1 pass).
        - Free Space: I would just pick Smart Cleanup as there is no reason to overwrite sectors without data on an HDD drive.
- Select Clean and Pick your flavor:
    - Turbo Cleanup will only do normal deletion (on HDD/SDD) and won't clean free space. It's not secure on an HDD nor an SSD.
    - Quick Cleanup will do secure deletion (on HDD) and normal deletion + trim (on SDD) but won't clean free space. I think this is secure enough for SSD but not for HDD.
    - Normal Cleanup will do secure deletion (on HDD) and normal deletion + trim (on SDD) and will then clean the whole free space (Smart Cleanup on HDD and Full Cleanup on SDD) and should be secure. I think this option is the best for HDD but completely overkill for SDD.
- Click Clean and wait for cleaning to finish. Could take a while and will fill your whole free space with data.

BleachBit:

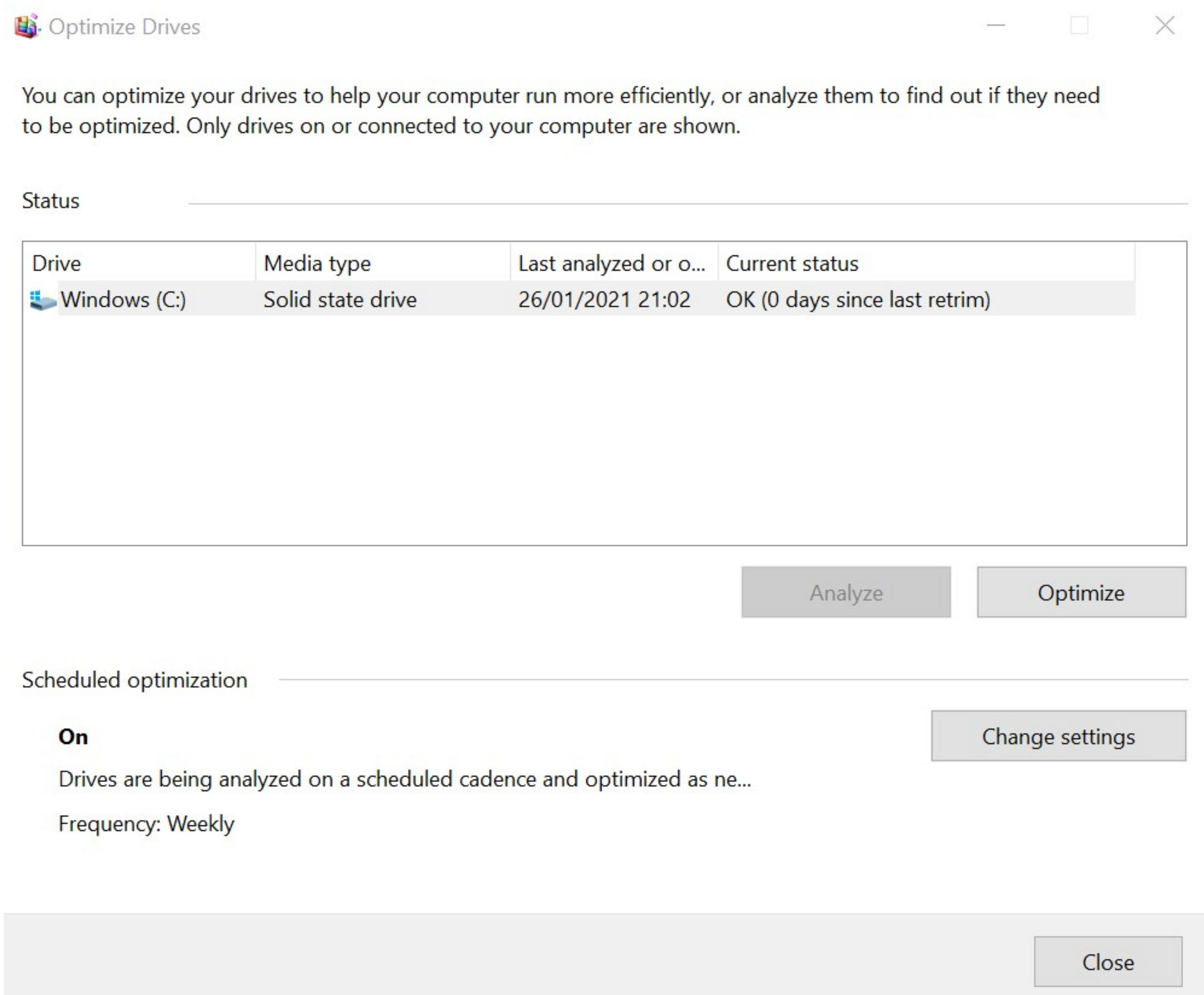
Here are the steps for BleachBit:

- Get and install the latest version from BleachBit here https://www.bleachbit.org/download
- Run BleachBit
- Clean at least everything within those sections:
    - Deep Scan
    - Windows Defender
    - Windows Explorer (including Shellbags)
    - System
    - Select any other traces you want to remove from their list
        - Again as with the previous utility, I would not clean the free space on an SDD drive because I think the Windows native "optimize" utility is enough (see Below) and that filling up the free space on a trim enabled SSD is just completely overkill and unnecessary.
    - Click Clean and wait. This will take a while and will fill your whole free space with data on both HDD and SDD drives.

Force a Trim with Windows Optimize (for SSD drives):

With this Native Windows 10 utility, you can just trigger a Trim on your SSD which should be more than enough to securely clean all deleted files that somehow would have escaped Trim when deleting them.

Just open Windows Explorer, Right Click on your System Drive and click Properties. Select Tools. Click Optimize and then Optimize again. You're done. I think that's probably enough in my opinion.



Communicating sensitive information to various organizations:

You might be interested in communicating information to some organization such as the press anonymously.

If you have to do so, you should take some steps because you can't really trust any organization to protect your anonymity[318]:

- Check the files for any metadata: see Removing Metadata from Files/Documents/Pictures:
- Check the files for anything malicious: see Appendix T: Checking files for malware
- Check the files for any watermarking: see Watermarking:
- Assess carefully the potential consequences and risks of communicating any sensitive information for you and others (legally, ethically and morally). Remember … Don't be evil. Legal is not necessarily Good.

After curating the files for anything you want to leave out. Double check and even Triple check them. Then you could consider sending them to an organization such as a press organization or others.

For this, I strongly recommend the use of SecureDrop[319] (https://securedrop.org/) which is an open-source project from the Freedom of the Press foundation.

Ideally you should use SecureDrop over Tor and you'll find a curated list of those here https://github.com/alecmuffett/real-world-onion-sites

If not SecureDrop is not available, you could consider any other mean of communication but you should privilege those that are encrypted end to end. **Do not ever do this from your real identity but only from a secure environment using an anonymous identity.**

Without SecureDrop you could consider:

- Using e-mail with GPG encryption provided your destination has published a GPG key somewhere. You can look this up here:
    - On their verified Social Media accounts if they provided it.
    - On keybase.io (Tor address http://keybase5wmilwokqirssclfnsqrjdsi7jdir5wy7y7iu3tanwmtp6oid.onion)
    - On open PGP directories such as: **(be careful as those are public directories and anyone can upload any key for any e-mail address, you'll have to cross-check the signature with other platforms to be sure it's actually theirs).**
        - http://keys.gnupg.net/
        - https://pgp.mit.edu/
        - https://keyserver.ubuntu.com/
- Using any other platform (even Twitter DMs) but again using GPG to encrypt the message for the destination.

What you should avoid IMHO:

- Do not send physical materials using the post due to the risk of leaving DNA/Fingerprints or other traceable information (see Cash-Paid VPN (preferred):).
- Do not use methods linked to a phone number (even a burner one) such as Signal/WhatsApp/Telegram.
- Do not use any kind of voice/video communication.
- Do not leak any clues about your real identity when exchanging messages.
- Do not meet people in real life unless you have absolutely no other option (this is a last resort).

If you intend to break your anonymity to protect your safety:

- Assess the risks very carefully first.
- Inform yourself carefully on the legality/safety of your intent and the consequences for you and others. Think about it carefully.
- Possibly reach out to a **trusted** lawyer before doing so.

Removing some traces of your identities on search engines and various platforms:

Chances are your actions (such as posts on various platforms, your profiles) will be indexed (and cached) by many search engines.

Contrary to popular belief, it's possible to have some but not all of this information removed by following some steps. While this might not remove the information on the websites themselves, it will make it harder for people to find it using search engines.

- First you'll have to delete your identities from the platform themselves if you can. Most will allow this but not all. For some you might have to contact their support/moderators and for others there will be readily available forms to do so.
- If they do not allow removal/deletion of profiles, there might be a possibility for you to rename your identity. Change the username if you can and all account information with bogus information including the e-mail.
- If allowed, you can also sometimes edit past posts to remove the information within those.

You can check some useful information about how to and get delete various accounts on these websites:

- https://justdeleteme.xyz/
- https://justgetmydata.com/

When you're done with this part, you should now handle search engines and while you may not be able to have the information deleted, you can ask them to update/remove outdated information which could then remove some cached information.

Google:

Unfortunately this will require you to have a Google account to request the update/removal (however this can be done with any Google account from anyone). There is no way around this except waiting.

Go to their "Remove outdated content from Google Search" page here: https://search.google.com/search-console/remove-outdated-content and submit a request accordingly.

If your profile/username was deleted/changed, they should re-index the content and update accordingly and remove these traces.

These requests might take several days to process. Be patient.

Bing:

Unfortunately this will require you to have a Microsoft account to request the update/removal (however this can be done with any Microsoft account from any identity). There is no way around this except waiting.

Go to their "Content Removal" page here: https://www.bing.com/webmasters/tools/contentremoval and submit a request accordingly.

If your profile/username was deleted/changed, they should re-index the content and update accordingly and remove these traces.

This might take several days to process. Be patient.

DuckDuckGo:

DuckDuckGo does not store cached version of pages[320] and will instead forward you to a Google/Bing cached version if available.

In addition, DuckDuckGo source most of their searches from Bing (and not google)[321] and therefore removing the content from Bing should in time have it removed it from DuckDuckGo too.

Yandex:

Unfortunately this will require you to have a Yandex account to request removals (however this can be done with any Yandex account from any identity). There is no way around this except waiting.

Once have your Yandex account, head to the Yandex Webmaster tools https://webmaster.yandex.com and then select Tools and Delete URL https://webmaster.yandex.com/tools/del-url/

There you can input the URL that do not exist anymore if you had them deleted.

This will only work with pages that have been deleted and therefore will not work with removing cache of existing records. For that unfortunately there is no tool available to force a cache update but you can still try their feedback tool:

Search for the page that was changed (where your profile was deleted/changed) and click the arrow next to the result. Select Complain. And submit a complaint about the page not matching the search result. Hopefully this will force Yandex to re-crawl the page and re-index it after some time. This could take days or weeks.

Qwant:

As far as I know, there is no readily available tool to force this and you'll have to wait for the results to get updated if there is any. If you know a way, please report this to me through the GitHub issues.

Yahoo Search:

Yes Yahoo Search still exists but as per their help page https://help.yahoo.com/kb/SLN4530.html , there is no way to remove information or refresh information besides waiting. This could take 6 to 8 weeks.

Baidu:

As far as I know, there is no readily available tool to force this unless you control the website (and do it through their webmaster tools). Therefore you'll have to wait for the results to get updated if there is any. If you know a way, please report this to me through the GitHub issues.

Wikipedia:

As far as I know, there is no way to remove information from Wikipedia articles themselves but if you just want to remove traces of your username from it (as a user that contributed), you can do so by following these steps: https://en.wikipedia.org/wiki/Wikipedia:Courtesy_vanishing

This will not remove any information about your online identities that could appear in other articles but only your own identity on Wikipedia as a user.

Internet Archive:

You can remove pages from internet archives but only if you actually own the website in question and contact them about it. Most likely you will not be able to remove archives from say "Reddit posts" or anything alike. But you could still ask and see what they answer.

As per their help page https://help.archive.org/hc/en-us/articles/360004651732-Using-The-Wayback-Machine

"How can I exclude or remove my site's pages from the Wayback Machine?

You can send an email request for us to review to info@archive.org with the URL (web address) in the text of your message."

Some low tech old school tricks:

Hidden communications in plain sight:

You have to keep in mind that using all those security measures (encryption, plausible deniability, VPN, tor, secure operating systems …) can make you suspicious just by using them. Using could be the equivalent of stating openly "I have something to hide" to an observer which could then motivate some adversaries to investigate/survey you further.

So there are other ways you could exchange or send messages online to others in case of need without disclosing your identity or establishing direct communication with them. These have been in use by various organizations for decades and can be of help if you do not want to attract attention by using secure tech while still communicating some sensitive information without attracting attention.

A commonly used technique which combines the idea of a Dead Drop[322] and Secure Communication Obfuscation[323] through Steganography[324] and/or Kleptography[325] and has many names such as Koalang[326] or "Talking Around" or even "Social Steganography". This technique is very old and still widely used nowadays by teenagers to bypass parental control. It's hiding in plain sight.

Here is one example if you want to let someone know something is wrong and they should go dark? That they should immediately wipe all their data, get rid of their burner phones and sensitive information?

What if you want to let someone you trust (friends, family, lawyers, journalists …) that you're in trouble and they should look out for you?

All this without revealing the identity of the person you're sending the message to nor disclosing the content of that message to any third party and without raising suspicions and without using any of the secure methods mentioned above.

Well you could just use any online public platform for this (Instagram, Twitter, Reddit, Any forum, YouTube …) by using in-context (of the chosen platform/media) agreed upon (between you and your contact) coded messages that only your contact would understand.

This could be a set of specific Emoji's or a specifically worded mundane comment. Or even just a like on a specific post from a known influencer you usually watch and like. While this would look completely normal to anyone, this could in fact mean a lot to a knowledgeable reader who could then take appropriate agreed upon actions.

You don't even have to go that far. A simple "Last seen" time on a specific account could be enough to trigger a message agreed upon. If your interlocutor sees that such account was online. It could mean there is an issue.

How to spot if someone has been searching your stuff:

There are some old tricks that you can use to spot if people have been messing with your stuff while you were away.

One trick for instance is very simple and just requires a wire/cable. Simply dispose objects on your desk/night table or in your drawers following a straight line. You can use a simple USB cable or charger to do this.

Draw a line with your cable and place objects along the line. When you're back, just check those places and check if the objects are still placed along the line. This allows you not to remember precisely where your things were.

Fortunately modern technology has made this even simpler. If you suspect someone might be looking through your stuff while you're away, you can just take a picture of the area with your phone before leaving. When you're back, just compare the areas with your pictures and everything should be exactly where you left it. If anything moved then someone was there.

It will be very hard and time consuming for an adversary to search through your stuff and then replace it exactly as you left it with complete precision.

What if it's a printed document or book and you want to know if someone read it? Even simpler. Just carefully make a note within the document with a pencil. And then erase it with any pencil eraser as if you wanted to correct it. The trick is to carefully leave the eraser traces/residues on the area you erased/pencil written areas and close the document. You could also take a picture of the residues before closing the document.

Most likely if someone went through your document to read it and re-placed it carefully, this residue will fall off or be moved significantly. It's a simple old school trick that could tell you someone searched a document you had.

Some last OPSEC thoughts:

- **Remember to use passphrases instead of passwords and use a different one for each service.**
- Make sure you don't keep a copy of this guide anywhere unsafe after. The sole presence of this guide will most likely defeat all your plausible deniability possibilities.
- Consider the use of Haven https://guardianproject.github.io/haven/ on some old android phone to keep watch on your home/room while you're away.
- Doxx "yourself" and your identities from time to time by looking for them yourself online using various search engines to monitor your online identities. You can even automate the process somewhat using various tools such as Google Alerts https://www.google.com/alerts.
- Remember Appendix N: Warning about smartphones and smart devices. Don't forget your smart devices can compromise your anonymity.
- Do not ever use biometrics alone to safeguard your secrets. Biometrics can be used without your consent.
- Do not ever travel with those devices if you have to pass strong border checks and where they could be illegal or raise suspicion.
- Do not plug any equipment in that laptop unless you trust it. Use an USB data blocker for charging.
- Do check the signatures and hashes of Software you download before installing them.
- Remember the first rule of fight club and don't talk to anyone about your sensitive activities using your real identity.
- Keep a normal life and don't be weird. If you spend all your online time using Tor to access the internet and have no social network accounts at all … You're already suspicious and attracting unnecessary attention.
- Encrypt everything but don't take it as granted. Remember the 5$ wrench[7] .
- Keep plausible deniability as an option but remember it won't help against the 5$ wrench either[7].
- Never ever leave your laptop unattended/on/unlocked anywhere when conducting sensitive activities. Remember the story of Ross Ulbricht and his arrest https://en.wikipedia.org/wiki/Ross_Ulbricht#Silk_Road,_arrest_and_trial.
- Check for tampering regularly (not only your devices but also your home/room).
- If you can, don't talk to the police (at least if you're in the US) https://www.youtube.com/watch?v=d-7o9xYp7eE without a lawyer.

- Know and always have at your disposal the details of a lawyer that could help you as a last resort in case things go wrong.

If you think you got burned:

If you have some time:

- Don't Panic.
- Delete everything you can from the internet related to that specific identity (accounts, comments …).
- Delete everything offline you have related to that identity including the backups.
- (If using a physical SIM) Destroy the SIM card and trash it in a random trash can somewhere.
- (If using a physical Burner Phone) Erase then destroy the Burner phone and trash it in a random trashcan somewhere.
- Securely erase the laptop hard drive and then ideally proceed to physically destroy the HDD/SDD/Laptop and trash it somewhere.
- Do the same with your backups.
- Keep the details of your lawyer nearby or if needed, call him/her in advance to prepare your case if needed.
- Return to your normal activities and hope for the best.

If you have no time:

- Don't Panic.
- Try to shut down/hibernate the laptop as soon as possible and hope for the best. If you're fast enough, your memory should decay or be cleaned and your data should be mostly safe for the time being.
- Contact a lawyer if possible and hope for the best and if you can't contact one (yet), try to remain silent (if your country allows it) until you have a lawyer to help you and if your law allows you to remain silent.

Keep in mind that many countries have specific laws to compel you to reveal your passwords that could override your "right to remain silent". See this Wikipedia article: https://en.wikipedia.org/wiki/Key_disclosure_law and this other visual resource with law references https://www.gp-digital.org/world-map-of-encryption/.

A final editorial note:

After reading this whole guide, I hope you will have gained some additional beneficial insight about privacy and anonymity. It is clear now, in my humble opinion, that the world we live in has only few safe harbors remaining where one could have a reasonable expectation of privacy and even less so anonymity. Many will often say that 1984 by George Orwell was not meant to be an instruction book. Yet today this guide and its many references should, I hope, reveal to you how far down we are in the rabbit hole.

You should also know that most of the digital information described in lengths in this guide can be forged or tampered by a motivated adversary for any purpose. Even if you do manage to keep secrets from prying eyes, it is not impossible for anyone to fabricate anything to fit their narrative.

- IP logs, DNS logs, Geolocation logs and Connection logs can be forged or tampered with by anyone using a simple text editor without leaving traces.
- Files and their properties can be created, altered and timestamped by anyone using simple utilities without leaving traces.
- EXIF information of pictures and videos can be altered by anyone using simple utilities without leaving traces.
- Digital Evidence (Pictures, Videos, Voice Recordings, E-Mails, Documents…) be crafted, placed, removed or destroyed with ease without leaving traces.

You should not hesitate to question this type of information from any source in this age of disinformation.

Please keep thinking for yourself and be open to critical thinking. Please keep an open mind.

If you want to give a hand to users facing censorship and oppression accessing the Tor Network, consider using https://addons.mozilla.org/en-US/firefox/addon/torproject-snowflake/ [327]

“In the end the Party would announce that two and two made five, and you would have to believe it.”, George Orwell, 1984.

Appendix A: Windows Installation

This is the Windows 10 installation process that should be valid for any Windows 10 install within this guide.

Installation:

DO NOT CONNECT WINDOWS TO ANY NETWORK DURING THE INSTALLATION PROCESS (This will allow us to create a Local Account and not use a Microsoft account and it will also prevent any telemetry from being sent out during the install process).

- Click “Install Now”
- Select “I don’t have a product key”
- Select the flavor you want:
    - Host OS: Use Windows Home or Windows Home N
    - VM OS: Use Windows Pro or Windows Pro N
- Select Custom
- Storage:
    - If this is a simple OS installation (Host OS with Simple Encryption) or VM without encryption, **select the whole disk** and proceed with installation (skip next step).
    - If this is part of a plausible deniability encryption setup on the Host OS:
        - If you are installing Windows for the first time (Hidden OS):
            - Delete the current partitions
            - Create a First partition with at least 50GB of disk space (about a third of the total disk space).
            - Create a Second partition with the remaining two thirds of the total disk space.
        - If you are installing Windows for the second time (Decoy OS):
            - Do not Delete the current partitions
            - Install Windows on the first partition you created during the first install.
        - Proceed with the install in the First partition
- Start the install process
- Select the Region “United Kingdom”
- Skip the additional Keyboard Layout
- Select “I don’t have internet”
- Select “Continue with limited setup”
- Create a username of your choice.
- Use a password of your choice.
- Select all 3 security questions and answer whatever you want (not real data).
- Don’t use Online Speech Recognition
- Don’t let app use your location
- Don’t enable “find my device”
- Only send “required diagnostic data”
- Do not improve Inking and Typing
- Do not get any improved tailored experience.
- Do not let apps use Advertising ID
- Select “Now” at the Cortana prompt

Privacy Settings:

- When the install is finished, get into Settings > Privacy and do the following:
    - General: All Off
    - Speech: Off
    - Inking and Typing: Off

    - Diagnostic: Required level at off, options on OFF, **Delete your data**, frequency set to Never
    - Activity History: all Off and Clear the history
    - Location, all Off (change button) and Clear it
    - Camera: Disable it (change button)
    - Microphone: Disable it (change button)
    - Voice Activation: All Off
    - Notification: Disable it (change button)
    - Account info: Disable it (change button)
    - Contact info: Disable it (change button)
    - Calendar access: Disable it (change button)
    - Phone calls: Disable it (change button)
    - Call History: Disable it (change button)
    - E-mail: Disable it (change button)
    - Tasks: Disable it (change button)
    - Messaging: Disable it (change button)
    - Radios: Disable it (change button)
    - Other devices: Set to Off
    - Background Apps: Disable it (change button)
    - App Diagnostics: Disable it (change button)
    - Documents: Disable it (change button)
    - Pictures: Disable it (change button)
    - Videos: Disable it (change button) and set to off
    - File system: Disable it (change button)
- Disable File Indexing by going into the "Indexing Options" (Go into Windows 10 Control Panel, Switch the view to "Large Icons" and select Indexing Options.
    - Modify the list and remove all locations.
    - Go into Advanced and click Rebuild.
- (Host OS only) Disable Bluetooth in the settings:
    - Go into Settings
    - Go into Devices
    - Select Bluetooth and turn it off
- (Host OS Only) Tape the Webcam and Microphone anyway for extra paranoia.
- (Host OS Only) Go into Settings > Network & Internet > Wi-Fi and Enable Random Hardware Address.

Appendix B: Windows Additional Privacy Settings

As written earlier in this guide and as noted by Privacytools.io[328], Windows 10 is a privacy nightmare. And disabling everything during and after the installation using the settings available to you is not enough. The amount of telemetry data collected by Microsoft is staggering and could defeat your attempts at keeping secrets. You will need to download and use a couple utilities to (hopefully) force Windows 10 into not sending data back to Microsoft.

Here are the steps in details:

- **DO NOT EVER USE A MICROSOFT ACCOUNT TO LOG IN: If you are, you should be re-installing this Windows Machine without connecting to a network and use a local account instead.**

Do these steps from a different computer in order to not connect Windows 10 to the internet before those settings are applied. You can download and copy those to the USB key (for transfer onto a Windows 10 fresh installation) or if it's a VM, you can transfer them to the VM within Virtualbox (VM Settings > General > Advanced > Drag n Drop > Enable Host to Guest).

- Download and install W10Privacy from https://www.w10privacy.de/english-home/
    - Open the app as Administrator (right click > more > run as administrator)
    - Check all the recommended (Green) settings and Save.

    - Optional but recommended (but could break things, use at your own risk), also check the orange/red settings and Save.
    - Reboot
- Download and run WindowsSpyBlocker from https://crazymax.dev/WindowsSpyBlocker/download/
    - Type 1 and go into Telemetry
    - Type 1 and go into Firewall
    - Type 2 and add Spy Rules
    - Reboot
- Go back one last time Settings > Privacy > Diagnostic and Delete all Data.

These measures added to the settings during installation should be hopefully sufficient to prevent Microsoft from snooping on your OS.

You will need to update and re-run W10Privacy and WindowsSpyBlocker frequently and after any Windows update as they tend to silently re-enable telemetry using those updates.

Appendix C: Windows Installation Media Creation

These are the steps to create a Windows 10 (20H2) Installation Media using this tool and instructions:

https://www.microsoft.com/en-us/software-download/windows10

- Download the tool and execute it from your Download folder.
- Agree to the terms
- Select the process to Create and installation Media.
- Select Windows 10 64 Bits edition with the language of your choice.
- Pick which process you want:
    - If installing on a physical computer: Select USB Flash Drive
    - If installing on a Virtual Machine: Select ISO file and save it.
- Proceed

Appendix D: Using System Rescue to securely wipe an SSD drive.

These instructions are valid for all Operating Systems:

- System Rescue:
    - Create a System Rescue USB disk following these instructions https://www.system-rescue.org/Installing-SystemRescue-on-a-USB-memory-stick/ (download the ISO and write to an USB stick with Rufus).
    - Disable Secure Boot in your BIOS/UEFI settings and change the boot order to the USB disk (System Rescue bootloader is not signed and won't boot with secure boot enabled).
    - Follow the instructions to change the keyboard layout by typing "stkmap".
    - (optional) Run startx afterward to start a graphical environment.
- SATA SSD:
    - (if you ran startx) Open a terminal
    - ATA Secure Erase:
        - Follow one of these tutorials
            - https://wiki.archlinux.org/index.php/Solid_state_drive/Memory_cell_clearing
            - https://ata.wiki.kernel.org/index.php/ATA_Secure_Erase
            - https://tinyapps.org/docs/wipe_drives_hdparm.html
    - ATA Sanitize:
        - Follow this tutorial https://tinyapps.org/docs/ata_sanitize_hdparm.html
- NVMe SSD:
    - (if you ran startx) Open a terminal
    - Follow one of these tutorials:
        - https://wiki.archlinux.org/index.php/Solid_state_drive/Memory_cell_clearing

- https://tinyapps.org/docs/nvme-secure-erase.html
- https://tinyapps.org/docs/nvme-sanitize.html

Appendix E: Clonezilla

- Get Clonezilla by just following these instructions: https://clonezilla.org/liveusb.php (I recommend the Alternative version AMD64 that should work with most recent laptops)
- Boot from Clonezilla
- Follow these steps to make a backup: https://clonezilla.org/show-live-doc-content.php?topic=clonezilla-live/doc/01_Save_disk_image
    - **If you're backing up a disk with simple Encryption, encryption of the backup is not required since you're backing up an already encrypted disk but you can still encrypt the backup anyway if you want additional security (and slower backup).**
    - **If you intend to back-up a device with plausible deniability encryption, I strongly advise against it as this backup image could be used to prove the existence of the hidden volume using forensics techniques as explained earlier. Do not make an image backup of the partition containing your hidden OS.**
- You're done, if you need to restore, follow these instructions: https://clonezilla.org/show-live-doc-content.php?topic=clonezilla-live/doc/02_Restore_disk_image

Each backup could take a while depending on the speed of your laptop and the speed of your external drive. In my experience, expect about 1 hour per backup depending on the drive size and the write speed of your backup media (my tests were done backing up 256GB SDDs on a USB 3.0 7200rpm HDD).

Appendix F: Diskpart

Diskpart is a Windows utility that can be used to perform various operations on your hard drive. In this case we will use Diskpart to show the Disk ID but also to change it if necessary.

This could be needed if you restore a backup on a new HDD/SDD that has an ID that differs from the one backed up and Windows could refuse to boot.

Diskpart can be run from any Windows environment using a command prompt. This includes recovery disks created by utilities such as Macrium Reflect, any Windows Installation media, EaseUS Todo Free rescue disks.

- **Displaying the disk ID**
    - Run Diskpart to enter the Diskpart utility
    - Issue the "list disk" command to list the disks
    - Issue the "sel disk x" (replace x with your system disk) to select your system disk
    - Issue the "detail disk" to show the details of this disk
    - Take note of the disk ID (this should be done BEFORE backing up your disks).
- **Changing the disk ID**
    - This step should only be done if, after restoring a full disk backup to a new hard drive, Windows refuses to boot
    - Issue the same commands as above on the target new disk
    - Issue in addition the command "uniqueid disk id=02345678" (where you replace the id by the one you noted before)

Appendix G: Safe Browser

If you can use Tor:

This guide will recommend using Tor browser within the host OS because it has the best protections by default. The only other acceptable option in my opinion would be to use Brave Browser (with a Tor tab) **but keep in mind that Brave themselves recommend the use of Tor Browser if you feel your safety depends on being anonymous[329]: " With Tor, Brave works hard to ensure that you're extremely difficult to track online while providing a delightful**

browsing experience. But if your personal safety depends on remaining anonymous you may wish to use the Tor Browser instead. ".

This Browser on the host OS will only be used to download various utilities and will never be used for actual sensitive activities.

- Download and install Tor Browser according to the instructions from https://www.torproject.org/download/
- Open Tor Browser

If you're experiencing issues connecting to Tor due to Censorship or Blocking, you might consider using Tor bridges as explained here: https://bridges.torproject.org/

Use this browser for all the next steps within the host OS unless instructed otherwise.

If you cannot use Tor:

Because it's too dangerous/risky/suspicious in itself. I would recommend as a last resort using Firefox or Brave only using Private Windows for now.

See Appendix P: Accessing the internet as safely as possible when Tor and VPNs are not an option before continuing.

Only do this from a different safe public Wi-Fi every time (See Find some safe places with decent public WIFI:) and using a long range connection (See Appendix Q: Using long range Antenna to connect to Public Wi-Fis from a safe distance:).

Clean all the data from the browser after each use.

Use this method for all the next steps within the host OS unless instructed otherwise.

Appendix H: Windows Cleaning Tools

In this guide I will recommend two third native tools and also two third party tools:

- Native Tools:
  - Windows 10 Disk Cleanup Utility: https://support.microsoft.com/en-us/windows/disk-cleanup-in-windows-10-8a96ff42-5751-39ad-23d6-434b4d5b9a68

    This tool will cleanup a bunch of things natively. It's not enough and I instead recommend using third party tools below to clean more stuff. PrivaZer for instance will use the disk cleanup utility directly itself and BleachBit will use its own mechanisms.

  - Windows 10 Optimize Utility (Defrag on HDD Drives): https://support.microsoft.com/en-us/windows/defragment-your-windows-10-pc-048aefac-7f1f-4632-d48a-9700c4ec702a

    For security, this tool is very useful on SSD drives at this "Optimize" function will in fact force a Disk wide Trim operation to occur. This will most likely be more than enough to make sure any deleted data that was not trimmed before for any reason will be this time. Deleted data with Trim is very unlikely to be recovered as explained before in this guide.

- Third Party Tools:
  - The open-source utility BleachBit https://www.bleachbit.org/
  - The closed-source utility PrivaZer https://privazer.com/

Personally I prefer PrivaZer because it has more customization and smarter features but I would understand if you don't trust them and prefer open-source software in which case I would recommend BleachBit which offers a bit less customization but similar functionalities.

Both these tools can be used for cleaning many things such as:

- The Windows USN journal which stores plenty of information[330].
- The Windows System Resource Usage Monitor (SRUM)[331].

- Various histories of various programs (such as the recent lists).
- Various logs
- The free (unallocated) space of your hard drive[332].
- Secure deletion of files
- Secure wiping of USB drives

Both these utilities can delete files and can overwrite the free space after deletion to improve secure deletion even on SSD drives. Remember this can reduce the lifespan of your SSD drives a bit.

Appendix I: Using ShredOS to securely wipe an HDD drive:

There are several utilities that are recommend (like the old unmaintained DBAN[333]) or System Rescue CD (https://www.system-rescue.org/) for this but personally, I will recommend the use of ShredOS.

Feel free do go with DBAN instead if you want (using this tutorial: https://www.lifewire.com/how-to-erase-a-hard-drive-using-dban-2619148), the process is basically the same but will not work out of the box with UEFI laptops.

If you want to go with System-Rescue, just head to their website and follow the instructions.

Windows:

- Download ShredOS from https://github.com/PartialVolume/shredos.2020.02
- Unzip the ISO file
- Download Rufus from https://rufus.ie/
- Launch Rufus
- Select the ShredOS IMG file
- Write it to an USB key
- When done, reboot and boot the USB key (you might have to go into your BIOS settings to change the boot order for this).
- Follow the instructions on screen

Linux:

- Follow instructions on https://github.com/PartialVolume/shredos.2020.02
- Reboot and boot the USB key
- Follow the instructions on screen

Appendix J: Manufacturer tools for Wiping HDD and SSD drives:

Always check your laptop BIOS/UEFI for native utilities first.

Be sure to use the right wipe mode for the appropriate disk. Wipe and Passes are for HDD drives. There are specific options for SDD drives (such as ATA Secure Erase or Sanitize).

Unfortunately most of these tools are Windows only.

Tools that provide a boot disk for wiping from boot:

- SanDisk DashBoard: https://kb.sandisk.com/app/answers/detail/a_id/15108/~/dashboard-support-information
- Seagate SeaTools: https://www.seagate.com/support/downloads/seatools/
- Samsung Magican: https://www.samsung.com/semiconductor/minisite/ssd/download/tools/
- Kingston SSD Manager: https://www.kingston.com/unitedstates/en/support/technical/ssdmanager
- Lenovo:
  - Most likely native utility available within the BIOS/UEFI, please check
  - Drive Erase Utility: https://support.lenovo.com/us/en/downloads/ds019026-thinkpad-drive-erase-utility-for-resetting-the-cryptographic-key-and-erasing-the-solid-state-drive-thinkpad
- Crucial Storage Executive: https://www.crucial.com/support/storage-executive
- Western Digital Dashboard: https://support.wdc.com/downloads.aspx?p=279

- HP: Follow instructions on https://store.hp.com/us/en/tech-takes/how-to-secure-erase-ssd
- Transcend SSD Scope: https://www.transcend-info.com/Support/Software-10/
- Dell:
    - Most likely native utility available within the BIOS/UEFI, please check https://www.dell.com/support/kbdoc/en-us/000134997/using-the-dell-bios-data-wipe-function-for-optiplex-precision-and-latitude-systems-built-after-november-2015?lwp=rt

Tools that provide only support from running OS (for external drives).

- Toshiba Storage Tools: https://www.toshiba-storage.com/downloads/

Appendix K: Considerations for using external SSD drives

I do not recommend using external SSDs due to the uncertainty about their support for Trim, ATA Secure Erase and Sanitize options through USB controllers. Instead I recommend using external HDD disks which can be cleaned/wiped safely and securely without hassle (albeit much slower than SSD drives).

Please do not buy or use gimmicky self-encrypting devices such as these:
https://syscall.eu/blog/2018/03/12/aigo_part1/

Some might be very efficient[334] but many are gimmicky gadgets.

If you really want to use an external SSD drive for sensitive storage:

- Please consider the support for:
    - Trim operations and ATA/NVMe secure erase operations from your Laptop USB controller.
    - Trim operations and ATA/NVMe secure erase operations from your USB SSD disk itself.
- Always use full disk encryption on those disks
- **Use the manufacturer provided tools to securely erase them if possible (see Appendix K: Considerations for using external SSD drives).**
- Consider manually wiping data on them after use by doing a full decryption/encryption or filling them completely with random data.

So how to check if your external USB SSD supports Trim and other ATA/NVMe operations from your Host OS?

Windows:

Trim Support:

It's possible Windows will detect your external SSD properly and enable Trim by default. Check if Optimize Works using the Windows Native disk utility as explained in the internal SSD section of Windows.

ATA/NVMe Operations (Secure Erase/Sanitize):

Use the manufacturer provided tools to check and perform these operations … It's pretty much the only way to be sure it's not only supported but actually works. Some utilities can tell you if it's supported or not like CrystalDiskInfo[335] but won't actually check if it's working. See Appendix J: Manufacturer tools for Wiping HDD and SSD drives:

If it doesn't work. Just decrypt and re-encrypt the whole drive or fill up the free space as instructed in the guide. There is no other way AFAIK. Besides booting up a System Rescue Linux CD and see the next section.

Linux:

Trim Support:

Follow this good tutorial: https://www.glump.net/howto/desktop/enable-trim-on-an-external-ssd-on-linux

ATA/NVMe Operations (Secure Erase/Sanitize):

It's not "recommended". Please read the disclaimers here https://ata.wiki.kernel.org/index.php/ATA_Secure_Erase and here https://wiki.archlinux.org/index.php/Solid_state_drive/Memory_cell_clearing

But this seems to be based on anecdotal experiences. So if you're sure your external SSD supports Trim (see vendor documentation). You could just **try at your own risk** to use nvme-cli or hdparm to issue secure erases.

See also this tutorial https://code.mendhak.com/securely-wipe-ssd/

Your mileage may vary. Use at your own risk.

MacOS:

Trim Support:

According to Apple Documentation[313], Trim is supported on APFS (asynchronously) and HFS+ (through period trim or first-aid).

So if it's supported (and enabled on your external SSD), you should be able to issue a Trim on a non-APFS drive using Disk Utility and First Aid which should issue a Trim.

If your disk supports it but it's not enabled in MacOS. You could try issuing a "sudo trimforce enable" command from the Terminal and see if it enables Trim on your external SSD. And then again check the first aid command if it's not APFS (see this Tutorial for info https://www.lifewire.com/enable-trim-for-ssd-in-os-x-yosemite-2260789)

If it doesn't work, I'm not aware of any reliable method to enable TRIM besides the commercial utility Trim Enabler here https://cindori.org/trimenabler/ which claims support for external drives.

ATA/NVMe Operations (Secure Erase/Sanitize):

I'm not aware of any method of doing so reliably and safely on MacOS. So you'll have to try one of these options:

- Use a bootable System Rescue USB Linux to do it
- Just decrypt and re-encrypt the drive using Disk Utility or Veracrypt
- Fill up the free space of the disk using the Linux method (dd)

Appendix L: Creating a mat2-web guest VM for removing metadata from files

Download the latest Debian stable amd64 netinst ISO from https://www.debian.org/CD/netinst/

- Create a new machine with any name like mat2
- Select Linux as Type
- Select Debian (64-bit) as Version
- Leave the default options and click create
- Select the VM and click Settings
- Select System and disable the Floppy disk on the Motherboard tab
- Select the Processor tab and enable PAE/NX
- Select Audio and disable Audio
- Select USB and disable the USB controller
- Select Storage and select the CD drive to mount the Debian Netinst ISO
- Select Network and Attach to Internal Network
- Pick Whonix as the network
- Launch the VM
- Select Install (not Graphical install)
- Select Language, Location and Keyboard layout as you wish
- Wait for Network detection to fail and select "Configure Network Manually"
- Put an IP in the Whonix range like for example 10.152.152.55 (don't use the same IP as any other VM you put behind the Whonix Gateway)
- Put 255.255.192.0 as Subnet Mask
- Put 10.152.152.10 as Gateway (the Whonix Gateway IP)
- Put 10.152.152.10 as DNS (the Whonix Gateway IP again)
- Pick a name like "Mat2"
- Leave the domain empty

- Set a Root password as you wish (preferably a good one still)
- Create a new user and password as you wish (preferably a good one still)
- Select the Time Zone of your choice
- Select Guided - Use entire disk
- Select the only ask available
- Select All files in one partition
- Confirm and write changes to disk
- Select NO to scan any other CD or DVD
- Select any region and any mirror of your choice and leave proxy blank
- Select no to participate in any survey
- Select only System Standard Utilities (uncheck everything else)
- Select Yes to install GRUB bootloader
- Select /dev/sda and continue
- Complete the install and reboot
- Login with your user or root (you should never use root directly as a best security practice but in this case I think it's “okay")
- Update your install by running "su apt upgrade" (but it should be upgraded since it's a net install)
- Install the necessary packages for mat2 by running "su apt install ffmpeg uwsgi python3-pip uwsgi-plugin-python3 librsvg2-dev git mat2 apache2 libapache2-mod-proxy-uwsgi"
- Go to the /var/www directory by running "cd /var/www/"
- Clone mat2-web from the mat2-web repository by issuing "git clone https://0xacab.org/jvoisin/mat2-web.git"
- Create a directory for uploads by running "mkdir ./mat2-web/uploads/"
- Give permissions to Apache2 to read the files by running "chown -R www-data:www-data ./mat2-web"
- Enable apache2 uwsgi proxy by running "/usr/sbin/a2enmod proxy_uwsgi"
- Upgrade pip by running "python3 -m pip install pip --upgrade"
- Install some python modules by running "python3 -m pip install flasgger pyyaml flask-restful flask cerberus flask-cors"
- Move to the config directory of mat2 by running "cd /var/www/mat2-web/config/"
- Copy the apache2 config file to etc by running "cp apache2.config /etc/apache2/sites-enabled/apache2.conf"
- Remove the default config file by running "rm /etc/apache2/sites-enabled/000-default.conf"
- Edit the apache2 config file provided by mat2-web by running "nano /etc/apache2/sites-enabled/apache2.conf"
- Remove the first line "Listen 80"
- Change the uwsgi path from "/var/www/mat2-web/mat2-web.sock" to "/run/uwsgi/uwsgi.sock" and save/exit
- Copy the uwsgi config file to etc by running "cp uwsgi.config /etc/uwsgi/apps-enabled/uwsgi.ini"
- Restart uwsgi by running "systemctl restart uwsgi" (there should be no errors)
- Restart apache2 by running "systemctl restart apache2" (there should be no errors)
- From the Whonix Workstation or any other VM on the Whonix Network, Open a Browser and go to the IP of your Debian VM (for example http://10.152.152.55)
- You should now see a Mat2-Web website running smoothly
- Shutdown the Mat2 VM by running "shutdown -h now"
- Take a Snapshot of the VM within Virtualbox
- Restart the Mat2 VM and you are ready to use Mat2-web to remove metadata from most files
- After use, shutdown the VM and revert to the snapshot to remove traces of the uploaded files
- For updates of Debian, start the VM and run "apt update" followed by "apt upgrade"
- For updates of mat2-web, go to /var/www/mat2-web and run "git pull"
- After updates, shutdown, take a new snapshot, remove the previous one keep going

You're done.

Now you can just start this small mat2 VM when needed, browse to it from your Guest VM and use the interface to remove any metadata from most files.

After each use of this VM, you should revert to the Snapshot to erase all traces.

Appendix M: BIOS/UEFI options to wipe disks in various Brands

Here are some links on how to securely wipe your drive (HDD/SDD) from the BIOS for various brands:

- Lenovo ThinkPads: https://support.lenovo.com/be/en/solutions/migr-68369
- HP (all): https://support.hp.com/gb-en/document/c06204100
- Dell (all): https://www.dell.com/support/kbdoc/en-us/000146892/dell-data-wipe
- Acer (Travelmate only): https://us.answers.acer.com/app/answers/detail/a_id/41567/~/how-to-use-disk-sanitizer-on-acer-travelmate-notebooks
- Asus: no option AFAIK except maybe for some ROG models.
- Gigabyte : no option AFAIK
- Honor : no option AFAIK
- Huawei : no option AFAIK

Appendix N: Warning about smartphones and smart devices

When conducting sensitive activities, remember that:

- You should not to bring your smartphone/smart devices with you (even turned off, unless you can remove the battery or are certain it's completely powered off).
- If you really have to take them with you, you could consider the use of a faraday cage[336] bag to store your devices. There are many such faraday "signal blocking" bags available for sale and some of these have been studied[337] for their effectiveness. If you can't afford such bags, you can probably achieve a "decent result" with one or several sheets of aluminum foil (as shown in the previously linked study).
- Consider leaving your smart devices at home online and doing something (watching YouTube/Netflix or something similar) instead of taking them with you powered off. This will mitigate tracking efforts but also create digital traces that could indicate you were at home.

Appendix O: Get an anonymous VPN

If you follow my advice, you will also need a VPN subscription but this time you will need an actually anonymous one that can't be tied to you by the financial system. Meaning you will need to buy a VPN subscription with cash or a reasonably private crypto currency (Monero). You will later use this VPN to connect to the various services anonymously but never directly from your IP.

I only see two possible options for you to get an anonymous VPN:

Cash-Paid VPN (preferred):

There are three VPN companies recommended by privacytools.io (https://privacytools.io/providers/vpn/) that accept cash payments: Mullvad, IVPN and ProtonVPN.

Personally I would recommend Mullvad due to personal experience.

How does this work?

- Access the VPN website with a Safe Browser (see Appendix G: Safe Browser)
- Go to IVPN, ProtonVPN or Mullvad website and create a new Account ID (on the login page).
- This page will give you an account ID, a token ID (for payment reference) and the details where to send the money by post.
- Send the required cash amount for the subscription you want in a sealed postal envelope to their offices, including a paper with the Token ID without a return address.

- Wait for them to receive the payment and enable your account (this can take a while).
- Open Tor Browser.
- Check your account status and proceed when your account is active.

For extra-security consider:

- Wearing gloves while manipulating anything to avoid leaving fingerprints[338] and touch DNA[339].
- Do not use any material/currency that was manipulated by someone that can be related to you in any way.
- Do not use currency you just got from an ATM that could record dispensed bills serial numbers.
- Be careful if you print anything that it's not watermarked by your printer (See Printing Watermarking:).
- Don't lick the envelope or the stamps[340] if you use them to avoid leaving DNA traces.
- Make sure there are no obvious DNA traces in or on the materials (like hairs).
- Consider doing the whole operation outdoor to reduce the risks residual DNA traces from your environment or yourself contaminating the materials.

Do not in any circumstance use this new VPN account unless instructed or connect to that new VPN account using your known connections. This VPN will only be used later in a secure way as we don't trust VPN providers "no logging policies". This VPN provider should ideally never know your real origin IP (your home/work one for instance).

Self-hosted VPN on a VPS paid with Monero (for skilled users familiar with Linux):

The other alternative is setting up your own VPN using a VPS (Virtual Private Server) on a hosting platform that accepts Monero (recommended).

This will offer some advantages as the chances of your IP being blacklisted somewhere are lower than known VPN providers.

This does offer some disadvantage as Monero is not perfect as explained earlier in this guide and some global adversaries could maybe still track you. You will need to get Monero from an Exchange using the normal financial system and then pick a hosting (list here https://www.getmonero.org/community/merchants/#exchanges)

You could use https://evolution-host.com/vps-hosting.php (who supports Whonix) which does accept Monero payments. Or pick one from the list here that do accept Monero: https://www.getmonero.org/community/merchants/#hosting

Consider tho that only one of those accepting Monero is actually recommended by Privacytools.io (https://privacytools.io/providers/hosting/) and it's Njalla https://njal.la/

There are plenty of tutorials on how to do this https://proprivacy.com/vpn/guides/create-your-own-vpn-server

Do not in any circumstance use this new VPS/VPN using your known connections. Only access it through Tor using Whonix Workstation for instance (this is explained later). This VPN will only be used later within a Virtual Machin over the Tor Network in a secure way as we don't trust VPN providers "no logging policies". This VPN provider should never know your real origin IP.

Appendix P: Accessing the internet as safely as possible when Tor and VPNs are not an option

USE EXTREME CAUTION: THIS IS HIGHLY RISKY.

There might be worst case situations where using Tor and VPNs are not possible due to active censorship or blocking even when using Tor Bridges (https://bridges.torproject.org/).

There might be situations where simply using Tor or a VPN alone would be seen as suspicious and could be dangerous for your safety. If this is case, you could be on a very hostile environment where surveillance and control is high. Consider reading this guide https://archive.flossmanuals.net/bypassing-censorship/index.html

But you still want to do something anonymously without disclosing/leaking any information.

In that case my last resort recommendation is to connect safely **from a distance** to a Public Wi-Fi (see Find some safe places with decent public WIFI:) using your laptop and TAILS "unsafe browser". See https://tails.boum.org/contribute/design/Unsafe_Browser/.

In this case, I would strongly recommend the use of a long-range "Yagi" type directional Antenna with an appropriate USB Wi-Fi Adapter. At least this will allow you to connect to public Wi-Fis from a "safe distance" but keep in mind that triangulation by a motivated adversary is still possible with the appropriate equipment. So this option should not be used during long period of times (minutes at best). See Appendix Q: Using long range Antenna to connect to Public Wi-Fis from a safe distance:.

Using TAILS should prevent local data leaks (such as MAC addresses or telemetry) and allow you to use a Browser to get what you want (utilities, VPN account) before leaving that place as fast as possible.

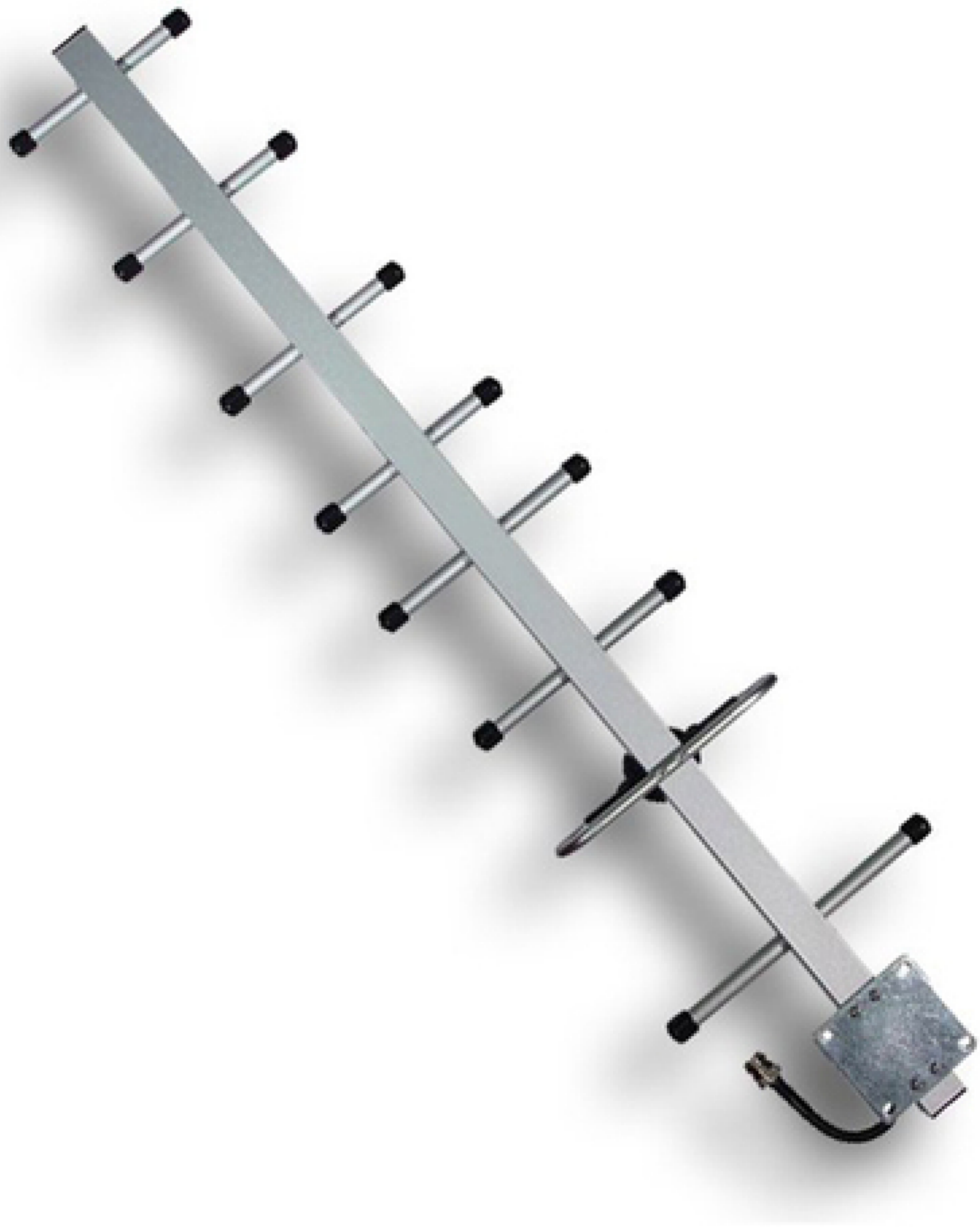
You could also use the other routes (Whonix and Qubes OS) instead of TAILS but in such hostile environments if you want data persistence but this might be riskier. I wouldn't risk it personally unless there was absolutely no other option. If you go for this option, you will only do sensitive activities from a reversible/disposable VM in all cases. Never from the Host OS.

If you resort to this, please keep your online time as short as possible (minutes and not hours).

Be safe and extremely cautious. This is entirely at your own risk.

Appendix Q: Using long range Antenna to connect to Public Wi-Fis from a safe distance:

It's possible to access/connect to remote distant Public Wi-Fis from a distance using a cheap directional Antenna that looks like this:



These antennas are widely available on various online shops for a cheap price (Amazon, AliExpress, Banggood …). The only issue is that they're not really discrete and you might have to find a way to hide it (for instance in a Poster cardboard container in a Backpack). Or in a large enough Bag. Optionally (but riskier) you could even consider using it from your home if you have a nice Window view to various places where some Public Wi-Fi is available.

Such antenna needs to be combined with specific USB adapters that have an external Antenna plug and sufficiently high power to use them.

Personally I would recommend the Alfa brand of adapters in the AWUS036 series (see https://www.alfa.com.tw/). But you could also go with some other brands if you want such as the TP-Link TL-WN722 (see https://www.tp-link.com/us/home-networking/usb-adapter/tl-wn722n/).

See this post for a comparison of various adapters: https://www.wirelesshack.org/best-kali-linux-compatible-usb-adapter-dongles.html (Usually those antennas are used by Penetration Testers to probe Wi-Fis from a distance and are often discussed within the scope of Kali Linux).

The process is really simple:

- Plug in and install your USB adapter on your Host OS.

- **Don't forget to randomize your MAC Address in case you bought this adapter online to prevent traceability (this is enabled by default in TAILS).**
- Connect the Long Range Antenna to the USB adapter (in place of the supplied one).
- Get to a convenient spot where you have a distant view on a place with a Public Wi-Fi available (this can be a rooftop for instance) but you could also imagine hiding the Antenna in some bag and just sit on a Bench somewhere.
- Point the Directional Antenna in the direction of the Public Wi-Fi.
- Connect to the Wi-Fi of your choice.

Do not forget tho that this will only delay a motivated adversary. Your signal can be triangulated easily by a motivated adversary in a matter of minutes once they reach the physical location of the Wi-Fi you're connecting to (for instance using a device such as AirCheck https://www.youtube.com/watch?v=8FV2QZ1BPnw, also see their other products here https://www.netally.com/products/#wifi_s). These products can easily be deployed on mobile units (in a Car for instance) and pinpoint your location in a matter of minutes.

Ideally this should "not be an issue" since this guide provides multiple ways of hiding your origin IP using VPNs and Tor. But if you are in the situation where VPN and Tor are not an option, then this could be your only security.

Appendix R: Installing a VPN on your VM or Host OS.

Download the VPN client installer of your cash paid VPN service and install it on Host OS (Tor over VPN, VPN over Tor over VPN) or the VM of your choice (VPN over Tor).

- Whonix Tutorial (should work with any VPN provider): https://www.whonix.org/wiki/Tunnels/Connecting_to_a_VPN_before_Tor (use the Linux configurations below to get the necessary configuration files)
- Windows Tutorials:
    - Mullvad: https://mullvad.net/en/help/install-mullvad-app-windows/
    - IVPN: https://www.ivpn.net/apps-windows
    - ProtonVPN : https://protonvpn.com/support/protonvpn-windows-vpn-application/
- MacOS :
    - Mullvad : https://mullvad.net/en/help/install-and-use-mullvad-app-macos/
    - IVPN: https://www.ivpn.net/apps-macos/
    - ProtonVPN : https://protonvpn.com/support/protonvpn-mac-vpn-application/
- Linux
    - Mullvad : https://mullvad.net/en/help/install-mullvad-app-linux/
    - IVPN: https://www.ivpn.net/apps-linux/
    - ProtonVPN : https://protonvpn.com/support/linux-vpn-setup/

Important note: Tor does not support UDP and you should use TCP instead with the VPN client in the Tor over VPN cases (on the VMs).

In all cases you should set the VPN to start from boot and enable the "kill switch" if you can. This is an extra-step since this guide proposes solutions that all fall back on Tor network in case of VPN failure. Still recommended IMHO.

Here are some guides provided by the recommended VPN providers in this guide:

- Windows:
    - iVPN https://www.ivpn.net/knowledgebase/general/do-you-offer-a-kill-switch-or-vpn-firewall/
    - ProtonVPN https://protonvpn.com/support/what-is-kill-switch/
    - Mullvad https://mullvad.net/en/help/using-mullvad-vpn-app/#killswitch
- Whonix Workstation: Coming Soon, It's certainly possible but I didn't find a suitable and easy tutorial yet. It's also worth remembering that if your VPN stops on Whonix, you'll still be behind the Tor Network.
- MacOS:
    - Mullvad same as Windows, the option should be in the provided VPN client
    - iVPN same as Windows, the option should be in the provided VPN client

    - ProtonVPN same as Windows with the client, the option should be in the provided VPN client https://protonvpn.com/blog/macos-vpn-kill-switch/
- Linux:
    - Mullvad
        - https://mullvad.net/en/help/wireguard-and-mullvad-vpn/
        - https://mullvad.net/en/help/linux-openvpn-installation/
    - ProtonVPN https://github.com/ProtonVPN/linux-cli/blob/master/USAGE.md#kill-switch
    - iVPN
        - https://www.ivpn.net/knowledgebase/linux/linux-wireguard-kill-switch/
        - https://www.ivpn.net/knowledgebase/linux/linux-kill-switch-using-the-uncomplicated-firewall-ufw/

Appendix S: Check your network for surveillance/censorship using OONI

So what is OONI? OONI stands for Open Observatory of Network Interference and is a sub-project of the Tor Project[198].

First OONI will allow you to check online for surveillance/censorship in your country just by looking at their Explorer that features test results from other people. This can be done here: https://explorer.ooni.org/

But these tests are limited and could not apply to your personal situation. If that is the case, you could consider running the OONI Probe yourself and running the tests yourself.

The problem obviously is that your network providers will be able to see those tests and your attempts at connecting to various services if the network is monitored. The other issue is that there are solutions to prevent OONI from working properly[341].

While this might not be a big deal in a normal environment, this could put you at risk in a hostile environment. **So running these tests can be risky.**

If you are in such a hostile environment where you suspect network activity is actively monitored and the simple fact of trying to access some resources can put you at risk, you should take some precautions before even attempting this:

- **Do not run the tests from your home/work network obviously.**
- **Do not run these tests from a known device or a smartphone but only for a secured OS on an ideally dedicated laptop.**
    - **You will not be able to do this from TAILS as TAILS will try to connect to Tor by default**
    - **You should only do this with the Qubes OS route or the Whonix Route of this guide after completing one of the routes.**
- **Only consider running these tests quickly from a Public Wi-Fi from a safe distance (see Appendix P: Accessing the internet as safely as possible when Tor and VPNs are not an option).**

The probe can be found here: https://ooni.org/install/ for various platforms (iOS, Android, Windows, MacOS, and Linux).

Appendix T: Checking files for malware

Fortunately, most recent anti-virus programs should be able to protect you from most of those threats. Yet if you are in doubt. I encourage you to check any document/image/video/archive/program you intend to run at https://www.virustotal.com/gui/ (just upload the file and wait for the results).

HOWEVER, please be aware that while VirusTotal is very practical for scanning various files, their privacy policy is problematic (see https://support.virustotal.com/hc/en-us/articles/115002168385-Privacy-Policy) and states:

"When you submit Samples to the Services, if you submit Samples to the Services, we will collect all of the information in the Sample itself and information about the act of submitting it."

Be aware that any document you submit to them will be kept, shared and used commercially including the content. So you should not do that with sensitive information and rely on various local AV scanners.

This guide's PDF files were submitted to them because it's meant to be public knowledge.

Concerning PDFs, you could also double-check using PDFID which you can download at https://blog.didierstevens.com/programs/pdf-tools/

- Install Python,
- Download PDFID and
- Extract the files
- run "python pdfid.py file-to-check.pdf" and you should see these at 0 in the case of the PDF files in this repository:

```
/JS              0
/JavaScript      0
/AA              0
/OpenAction      0
/AcroForm        0
/JBIG2Decode     0
/RichMedia       0
/Launch          0
/EmbeddedFile    0
/XFA             0
```

Everything sensitive and such checks should happen in sandboxes and hardened Virtual Machines. This is to prevent any malware for actually accessing your computer.

For other formats, please refer to:

- This practical cheat sheet from SANS: https://digital-forensics.sans.org/media/analyzing-malicious-document-files.pdf (warning, many of those tools will be harder to use to Windows and you might consider using them from a Linux OS such as Tails, Whonix Workstation or a Linux distribution of your choice as explained later in this guide. There are also other guides out there[342] that might be of use.
- This interesting PDF with various links https://www.winitor.com/pdf/Malware-Analysis-Fundamentals-Files-Tools.pdf

Acknowledgements:

- Thanks to GitHub for hosting this project and the many people who starred it
- Thanks to all the people who contributed and shared this guide to others
- Thanks to the people at the Monero project
- Thanks to the people at the Wikipedia project
- Thanks to the people at the TAILS project
- Thanks to the people at the Whonix project
- Thanks to the people at the Qubes OS project
- Thanks to the people at the Veracrypt project
- Thanks to the people at the Tor Project
- Thanks to the people at the EFF
- Thanks to the people at the Privacytools.io project
- Thanks to the people at BlackHat, DEF CON and CCC
- Thanks to the people at Bellingcat and other OSINT/Forensics researchers
- Thanks to the makers of the Social Dilemma documentary **(go watch it if you didn't yet)**
- Thanks to Randall Munroe at XKCD for his great webcomics.
- Thanks to the people at the various few commercial entities who do take privacy seriously

- Thanks to the whole open-source community and especially the Linux community
- Thanks to the many researchers, journalists, lawyers and individuals referenced in this guide for their various research and projects
- **Special Thanks to Edward Snowden and who inspired me to write this guide (buy and read his book please https://en.wikipedia.org/wiki/Permanent_Record_(autobiography))**

[1] YouTube, https://www.youtube.com/watch?v=6DGNZnfKYnU
[2] Wikipedia, OSINT https://en.wikipedia.org/wiki/Open-source_intelligence
[3] YouTube Internet Historian Playlist, HWNDU
https://www.youtube.com/playlist?list=PLna1KTNJu3y09Tu70U6yPn28sekaNhOMY
[4] Wikipedia, 4chan https://en.wikipedia.org/wiki/4chan
[5] PIA, See this good article on the matter https://www.privateinternetaccess.com/blog/how-does-privacy-differ-from-anonymity-and-why-are-both-important/ (disclaimer: this is not an endorsement or recommendation for this commercial service).

[6] This World of Ours, James Mickens https://scholar.harvard.edu/files/mickens/files/thisworldofours.pdf
[7] XKCD, Security https://xkcd.com/538/
[8] Wikipedia, Threat Model https://en.wikipedia.org/wiki/Threat_model
[9] Bellingcat https://www.bellingcat.com/
[10] Wikipedia, Doxing https://en.wikipedia.org/wiki/Doxing
[11] YouTube, Internet Historian, The Bikelock Fugitive of Berkeley https://www.youtube.com/watch?v=muoR8Td44UE
[12] BBC News, Tor Mirror https://www.bbc.com/news/technology-50150981
[13] GitHub, Real World Onion websites https://github.com/alecmuffett/real-world-onion-sites
[14] Tor Project, Who Uses Tor https://2019.www.torproject.org/about/torusers.html.en
[15] Whonix Documentation, The importance of Anonymity https://www.whonix.org/wiki/Anonymity
[16] Wikipedia, IANAL https://en.wikipedia.org/wiki/IANAL
[17] Wikipedia, Trust but verify https://en.wikipedia.org/wiki/Trust,_but_verify
[18] English translation of German Telemedia Act https://www.huntonprivacyblog.com/wp-content/uploads/sites/28/2016/02/Telemedia_Act__TMA_.pdf. Section 13, Article 6, "The service provider must enable the use of Telemedia and payment for them to occur anonymously or via a pseudonym where this is technically possible and reasonable. The recipient of the service is to be informed about this possibility. ".
[19] Wikipedia, Don't be evil https://en.wikipedia.org/wiki/Don%27t_be_evil
[20] Wikipedia, IP Address, https://en.wikipedia.org/wiki/IP_address
[21] Wikipedia; Data Retention https://en.wikipedia.org/wiki/Data_retention
[22] Wikipedia, Tor Anonymity Network https://en.wikipedia.org/wiki/Tor_(anonymity_network)
[23] Wikipedia, VPN https://en.wikipedia.org/wiki/Virtual_private_network
[24] Wikipedia, DNS https://en.wikipedia.org/wiki/Domain_Name_System
[25] Wikipedia, DNS Blocking https://en.wikipedia.org/wiki/DNS_blocking
[26] CensoredPlanet https://censoredplanet.org/
[27] ArXiv, Characterizing Smart Home IoT Traffic in the Wild https://arxiv.org/pdf/2001.08288.pdf
[28] Labzilla.io, Your Smart TV is probably ignoring your Pi-Hole https://labzilla.io/blog/force-dns-pihole
[29] Wikipedia, DNS over HTTPS: https://en.wikipedia.org/wiki/DNS_over_HTTPS
[30] Wikipedia, DNS over TLS, https://en.wikipedia.org/wiki/DNS_over_TLS
[31] Wikipedia, Pi-Hole https://en.wikipedia.org/wiki/Pi-hole
[32] Wikipedia, SNI https://en.wikipedia.org/wiki/Server_Name_Indication
[33] Wikipedia, eSNI https://en.wikipedia.org/wiki/Server_Name_Indication#Encrypted_Client_Hello
[34] Wikipedia, ECH, https://en.wikipedia.org/wiki/Server_Name_Indication#Encrypted_Client_Hello
[35] Mozilla Blog, Encrypted Client Hello: the future of ESNI in Firefox https://blog.mozilla.org/security/2021/01/07/encrypted-client-hello-the-future-of-esni-in-firefox/
[36] Usenix.org, On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention
https://www.usenix.org/system/files/foci19-paper_chai_0.pdf
[37] Wikipedia, CDN https://en.wikipedia.org/wiki/Content_delivery_network
[38] Cloudflare, Good-bye ESNI, hello ECH! https://blog.cloudflare.com/encrypted-client-hello/
[39] ZDNET, Russia wants to ban the use of secure protocols such as TLS 1.3, DoH, DoT, ESNI
https://www.zdnet.com/article/russia-wants-to-ban-the-use-of-secure-protocols-such-as-tls-1-3-doh-dot-esni/
[40] ZDNET, China is now blocking all encrypted HTTPS traffic that uses TLS 1.3 and ESNI https://www.zdnet.com/article/china-is-now-blocking-all-encrypted-https-traffic-using-tls-1-3-and-esni/
[41] KUL, Encrypted DNS==⇒Privacy? A Traffic Analysis Perspective https://www.esat.kuleuven.be/cosic/publications/article-3153.pdf
[42] ResearhGate, Oblivious DNS: Practical Privacy for DNS Queries
https://www.researchgate.net/publication/332893422_Oblivious_DNS_Practical_Privacy_for_DNS_Queries
[43] Nymity.ch, The Effect of DNS on Tor's Anonymity https://nymity.ch/tor-dns/
[44] Wikipedia, IMEI https://en.wikipedia.org/wiki/International_Mobile_Equipment_Identity

[45] Wikipedia, IMSI https://en.wikipedia.org/wiki/International_mobile_subscriber_identity
[46] Android Documentation, Device Identifiers https://source.android.com/devices/tech/config/device-identifiers
[47] Google Privacy Policy, Look for IMEI https://policies.google.com/privacy/embedded?hl=en-US
[48] Wikipedia, IMEI and the Law https://en.wikipedia.org/wiki/International_Mobile_Equipment_Identity#IMEI_and_the_law
[49] Bellingcat, The GRU Globetrotters: Mission London https://www.bellingcat.com/news/uk-and-europe/2019/06/28/the-gru-globetrotters-mission-london/
[50] Bellingcat,"V" For "Vympel": FSB's Secretive Department "V" Behind Assassination Of Georgian Asylum Seeker In Germany https://www.bellingcat.com/news/uk-and-europe/2020/02/17/v-like-vympel-fsbs-secretive-department-v-behind-assassination-of-zelimkhan-khangoshvili/
[51] Wikipedia, CCTV https://en.wikipedia.org/wiki/Closed-circuit_television
[52] Apple, Transparency Report, Device Requests https://www.apple.com/legal/transparency/device-requests.html
[53] The Intercept, How Cops Can Secretly Track Your Phone https://theintercept.com/2020/07/31/protests-surveillance-stingrays-dirtboxes-phone-tracking/
[54] Wikipedia, IMSI Catcher https://en.wikipedia.org/wiki/IMSI-catcher
[55] Wikipedia, Stingray https://en.wikipedia.org/wiki/Stingray_phone_tracker
[56] Gizmodo, Cops Turn to Canadian Phone-Tracking Firm After Infamous 'Stingrays' Become 'Obsolete' https://gizmodo.com/american-cops-turns-to-canadian-phone-tracking-firm-aft-1845442778
[57] Wikipedia, MITM https://en.wikipedia.org/wiki/Man-in-the-middle_attack
[58] Purism, Librem 5 https://shop.puri.sm/shop/librem-5/
[59] Wikipedia, MAC Address https://en.wikipedia.org/wiki/MAC_address
[60] ResearchGate, Tracking Anonymized Bluetooth Devices https://www.researchgate.net/publication/334590931_Tracking_Anonymized_Bluetooth_Devices/fulltext/5d3308db92851cd04675a469/Tracking-Anonymized-Bluetooth-Devices.pdf
[61] Apple, Differential Privacy White Paper https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf
[62] Wikipedia, Differential Privacy https://en.wikipedia.org/wiki/Differential_privacy
[63] Google Android Help, Android Location Services https://support.google.com/accounts/answer/3467281?hl=en
[64] Apple Support, Location Services and Privacy https://support.apple.com/en-us/HT207056
[65] State University of New York, Towards 3D Human Pose Construction Using Wi-Fi https://cse.buffalo.edu/~lusu/papers/MobiCom2020.pdf
[66] Using Metadata to find Paul Revere (https://kieranhealy.org/blog/archives/2013/06/09/using-metadata-to-find-paul-revere/)
[67] Wikipedia, Google SensorVault, https://en.wikipedia.org/wiki/Sensorvault
[68] NRKBeta, My Phone Was Spying on Me, so I Tracked Down the Surveillants https://nrkbeta.no/2020/12/03/my-phone-was-spying-on-me-so-i-tracked-down-the-surveillants/
[69] New York Times https://www.nytimes.com/interactive/2019/12/19/opinion/location-tracking-cell-phone.html
[70] Sophos, Google data puts innocent man at the scene of a crime https://nakedsecurity.sophos.com/2020/03/10/google-data-puts-innocent-man-at-the-scene-of-a-crime/
[71] Wikipedia, Geofence Warrant https://en.wikipedia.org/wiki/Geo-fence_warrant
[72] Wikipedia, Room 641A https://en.wikipedia.org/wiki/Room_641A
[73] Wikipedia, Edward Snowden https://en.wikipedia.org/wiki/Edward_Snowden
[74] Wikipedia, Permanent Record https://en.wikipedia.org/wiki/Permanent_Record_(autobiography)
[75] Wikipedia, XKEYSCORE https://en.wikipedia.org/wiki/XKeyscore
[76] ElectroSpaces, Danish military intelligence uses XKEYSCORE to tap cables in cooperation with the NSA https://www.electrospaces.net/2020/10/danish-military-intelligence-uses.html
[77] Wikipedia, MUSCULAR https://en.m.wikipedia.org/wiki/MUSCULAR_(surveillance_program)
[78] Wikipedia, PRISM https://en.wikipedia.org/wiki/PRISM_(surveillance_program)
[79] Justsecurity, General Hayden https://www.justsecurity.org/10318/video-clip-director-nsa-cia-we-kill-people-based-metadata/
[80] Reuters, Exclusive: Apple dropped plan for encrypting backups after FBI complained – sources https://www.reuters.com/article/us-apple-fbi-icloud-exclusive-idUSKBN1ZK1CT
[81] ZDnet, I asked Apple for all my data. Here's what was sent back https://www.zdnet.com/article/apple-data-collection-stored-request/
[82] Wired, The Strava Heat Map and the End of Secrets https://www.wired.com/story/strava-heat-map-military-bases-fitness-trackers-privacy/
[83] Bellingcat, How to Use and Interpret Data from Strava's Activity Map https://www.bellingcat.com/resources/how-tos/2018/01/29/strava-interpretation-guide/
[84] The Guardian, Fitness tracking app Strava gives away location of secret US army bases https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases
[85] Telegraph, Running app reveals locations of secret service agents in MI6 and GCHQ https://www.telegraph.co.uk/technology/2018/07/08/running-app-exposes-mi6-gchq-workers-whereabouts/

[86] De Correspondent, Here's how we found the names and addresses of soldiers and secret agents using a simple fitness app https://decorrespondent.nl/8481/heres-how-we-found-the-names-and-addresses-of-soldiers-and-secret-agents-using-a-simple-fitness-app/412999257-6756ba27
[87] Washington Post, Alexa has been eavesdropping on you this whole time https://www.washingtonpost.com/technology/2019/05/06/alexa-has-been-eavesdropping-you-this-whole-time/?utm_term=.8514f3a17b1c&itid=lk_interstitial_manual_59
[88] CryptoEngineering, How does Apple (privately) find your offline devices? https://blog.cryptographyengineering.com/2019/06/05/how-does-apple-privately-find-your-offline-devices/
[89] Apple Support https://support.apple.com/en-us/HT210515
[90] XDA, Samsung's Find My Mobile app can locate Galaxy devices even when they're offline https://www.xda-developers.com/samsung-find-my-mobile-app-locate-galaxy-devices-offline/
[91] Apple Support, If your Mac is lost or stolen https://support.apple.com/en-us/HT204756
[92] Wikipedia, BLE https://en.wikipedia.org/wiki/Bluetooth_Low_Energy
[93] Cryptography Engineering Blog, How does Apple (privately) find your offline devices? https://blog.cryptographyengineering.com/2019/06/05/how-does-apple-privately-find-your-offline-devices/
[94] Wikipedia, RFID https://en.wikipedia.org/wiki/Radio-frequency_identification
[95] Wikipedia, NFC https://en.wikipedia.org/wiki/Near-field_communication
[96] Samsonite Online Shop, RFID accessories, https://shop.samsonite.com/accessories/rfid-accessories/
[97] Bellingcat, Joseph Mifsud: Rush for the EXIF https://www.bellingcat.com/news/americas/2018/10/26/joseph-mifsud-rush-exif/
[98] Zoom Support, Adding a watermark https://support.zoom.us/hc/en-us/articles/209605273-Adding-a-Watermark
[99] Zoom Support, Audio Watermark https://support.zoom.us/hc/en-us/articles/360021839031-Audio-Watermark
[100] CreativeCloud Extension, IMATAG https://exchange.adobe.com/creativecloud.details.101789.imatag-invisible-watermark-and-image-monitoring.html
[101] NexGuard, https://dtv.nagra.com/nexguard-forensic-watermarking
[102] Vobile Solutions, https://www.vobilegroup.com/solutions
[103] Cinavia, https://www.cinavia.com/languages/english/pages/technology.html
[104] Imatag, https://www.imatag.com/
[105] Wikipedia, Steganography https://en.wikipedia.org/wiki/Steganography
[106] IEEExplore, A JPEG compression resistant steganography scheme for raster graphics images https://ieeexplore.ieee.org/document/4428921
[107] ScienceDirect, Robust audio watermarking using perceptual masking https://www.sciencedirect.com/science/article/abs/pii/S0165168498000140
[108] IEEExplore, Spread-spectrum watermarking of audio signals https://ieeexplore.ieee.org/abstract/document/1188746
[109] Google Scholar, source camera identification https://scholar.google.com/scholar?q=source+camera+identification
[110] HackerFactor Blog, Deanonymizing Tor Circuits https://www.hackerfactor.com/blog/index.php?/archives/868-Deanonymizing-Tor-Circuits.html
[111] KU Leuven, Website Fingerprinting through Deep Learning https://distrinet.cs.kuleuven.be/software/tor-wf-dl/
[112] DailyDot, How Tor helped catch the Harvard bomb threat suspect https://www.dailydot.com/unclick/tor-harvard-bomb-suspect/
[113] ArsTechnica, How the NSA can break trillions of encrypted Web and VPN connections https://arstechnica.com/information-technology/2015/10/how-the-nsa-can-break-trillions-of-encrypted-web-and-vpn-connections/
[114] ArsTechnica, Does Tor provide more benefit or harm? New paper says it depends https://arstechnica.com/gadgets/2020/11/does-tor-provide-more-benefit-or-harm-new-paper-says-it-depends/
[115] ResearchGate, The potential harms of the Tor anonymity network cluster disproportionately in free countries https://www.pnas.org/content/early/2020/11/24/2011893117
[116] arXiv, An Analysis of Anonymity in the Bitcoin System https://arxiv.org/abs/1107.4524
[117] Bellingcat, How To Track Illegal Funding Campaigns Via Cryptocurrency, https://www.bellingcat.com/resources/how-tos/2019/03/26/how-to-track-illegal-funding-campaigns-via-cryptocurrency/
[118] Wikipedia, KYC https://en.wikipedia.org/wiki/Know_your_customer
[119] YouTube, Breaking Monero https://www.youtube.com/watch?v=WOyC6OB6ezA&list=PLsSYUeVwrHBnAUre2G_LYDsdo-tD0ov-y
[120] Monero, Monero vs Princeton Researchers, https://monero.org/monero-vs-princeton-researchers/
[121] ArXiv, Tracking Mixed Bitcoins, https://arxiv.org/abs/2009.14007
[122] Wikipedia, Exploit https://en.wikipedia.org/wiki/Exploit_(computer_security)
[123] Wikipedia, Freedom Hosting https://en.wikipedia.org/wiki/Freedom_Hosting
[124] Wired, 2013 FBI Admits It Controlled Tor Servers Behind Mass Malware Attack https://www.wired.com/2013/09/freedom-hosting-fbi/

[125] Wikipedia, 2020 United States federal government data breach https://en.wikipedia.org/wiki/2020_United_States_federal_government_data_breach
[126] Wikipedia, Sandbox https://en.wikipedia.org/wiki/Sandbox_(computer_security)
[127] Wikipedia, CPU https://en.wikipedia.org/wiki/Central_processing_unit
[128] Wikipedia, Intel Management Engine https://en.wikipedia.org/wiki/Intel_Management_Engine
[129] Wikipedia, AMD Platform Security Processor https://en.wikipedia.org/wiki/AMD_Platform_Security_Processor
[130] Wikipedia, IME, Security Vulnerabilities https://en.wikipedia.org/wiki/Intel_Management_Engine#Security_vulnerabilities
[131] Wikipedia, IME, Assertions that ME is a backdoor https://en.wikipedia.org/wiki/Intel_Management_Engine#Assertions_that_ME_is_a_backdoor
[132] Wikipedia, IME, Disabling the ME https://en.wikipedia.org/wiki/Intel_Management_Engine#Disabling_the_ME
[133] Libreboot, https://libreboot.org/
[134] Digi.Ninja, Jasager https://digi.ninja/jasager/
[135] Hak5 Shop, Wi-Fi Pineapple https://shop.hak5.org/products/wifi-pineapple
[136] Wikipedia, Deautentication Attack https://en.wikipedia.org/wiki/Wi-Fi_deauthentication_attack
[137] Wikipedia, Capture Portal https://en.wikipedia.org/wiki/Captive_portal
[138] Wired, Why the Security of USB Is Fundamentally Broken https://www.wired.com/2014/07/usb-security/
[139] Wikipedia, Stuxnet https://en.wikipedia.org/wiki/Stuxnet
[140] Superuser.com, How do I safely investigate a USB stick found in the parking lot at work? https://superuser.com/questions/1206321/how-do-i-safely-investigate-a-usb-stick-found-in-the-parking-lot-at-work
[141] Magnet Forensics, Magnet AXIOM https://www.magnetforensics.com/products/magnet-axiom/cloud/
[142] Cellebrite, Unlock cloud-based evidence to solve the case sooner https://www.cellebrite.com/en/ufed-cloud/
[143] ArsTechnica, How the way you type can shatter anonymity—even on Tor https://arstechnica.com/information-technology/2015/07/how-the-way-you-type-can-shatter-anonymity-even-on-tor/
[144] Paul Moore Blog, Behavioral Profiling: The password you can't change. https://paul.reviews/behavioral-profiling-the-password-you-cant-change/
[145] EFF CoverMyAss, https://coveryourtracks.eff.org/
[146] ArsTechnica, Stakeout: how the FBI tracked and busted a Chicago Anon https://arstechnica.com/tech-policy/2012/03/stakeout-how-the-fbi-tracked-and-busted-a-chicago-anon/
[147] Bellingcat MH17 - Russian GRU Commander 'Orion' Identified as Oleg Ivannikov https://www.bellingcat.com/news/uk-and-europe/2018/05/25/mh17-russian-gru-commander-orion-identified-oleg-ivannikov/
[148] Chromium Documentation, Technical analysis of client identification mechanisms https://sites.google.com/a/chromium.org/dev/Home/chromium-security/client-identification-mechanisms#TOC-Machine-specific-characteristics
[149] Mozilla Wiki, Fingerprinting https://wiki.mozilla.org/Fingerprinting
[150] Facebook Research, Deepface https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/
[151] Privacy News Online, Putting the "face" in Facebook: how Mark Zuckerberg is building a world without public anonymity https://www.privateinternetaccess.com/blog/putting-face-facebook-mark-zuckerberg-building-world-without-public-anonymity/
[152] CNBC, "Facebook has mapped populations in 23 countries as it explores satellites to expand internet" https://www.cnbc.com/2017/09/01/facebook-has-mapped-human-population-building-internet-in-space.html
[153] MIT Technology Review, This is how we lost control of our faces, https://www.technologyreview.com/2021/02/05/1017388/ai-deep-learning-facial-recognition-data-history/
[154] Bellingcat, Shadow of a Doubt: Crowdsourcing Time Verification of the MH17 Missile Launch Photo https://www.bellingcat.com/resources/case-studies/2015/08/07/shadow-of-a-doubt/
[155] Brown Institute, Open Source Investigation, https://brown.columbia.edu/open-source-investigation/
[156] NewScientist, Facebook can recognize you in photos even if you're not looking https://www.newscientist.com/article/dn27761-facebook-can-recognise-you-in-photos-even-if-youre-not-looking/
[157] Google Patent, Techniques for emotion detection and content delivery https://patents.google.com/patent/US20150242679
[158] APNews, Chinese 'gait recognition' tech IDs people by how they walk https://apnews.com/article/bf75dd1c26c947b7826d270a16e2658a
[159] TechCrunch, Facial recognition reveals political party in troubling new research https://techcrunch.com/2021/01/13/facial-recognition-reveals-political-party-in-troubling-new-research/
[160] Slate https://slate.com/technology/2018/04/facebook-collects-data-on-non-facebook-users-if-they-want-to-delete-it-they-have-to-sign-up.html
[161] The Conversation https://theconversation.com/shadow-profiles-facebook-knows-about-you-even-if-youre-not-on-facebook-94804
[162] The Verge https://www.theverge.com/2018/4/11/17225482/facebook-shadow-profiles-zuckerberg-congress-data-privacy

[163] ZDNET https://www.zdnet.com/article/anger-mounts-after-facebooks-shadow-profiles-leak-in-bug/
[164] CNET https://www.cnet.com/news/shadow-profiles-facebook-has-information-you-didnt-hand-over/
[165] Anyvision https://www.anyvision.co/
[166] BBC, Met police deploy live facial recognition technology https://www.theguardian.com/uk-news/2020/feb/11/met-police-deploy-live-facial-recognition-technology
[167] YouTube, The Economist, China: facial recognition and state control | The Economist https://www.youtube.com/watch?v=lH2gMNrUuEY
[168] Washington Post, Huawei tested AI software that could recognize Uighur minorities and alert police, report says https://www.washingtonpost.com/technology/2020/12/08/huawei-tested-ai-software-that-could-recognize-uighur-minorities-alert-police-report-says/
[169] The Intercept, How a Facial Recognition Mismatch Can Ruin Your Life https://theintercept.com/2016/10/13/how-a-facial-recognition-mismatch-can-ruin-your-life/
[170] BBC, WhatsApp photo drug dealer caught by 'groundbreaking' work https://www.bbc.com/news/uk-wales-43711477
[171] IMDB, Gattaca 1997, https://www.imdb.com/title/tt0119177/
[172] IMDB, Person of Interest 2011 https://www.imdb.com/title/tt1839578
[173] IMDB, Minority Report 2002, https://www.imdb.com/title/tt0181689
[174] Wikipedia, Deepfake https://en.wikipedia.org/wiki/Deepfake
[175] Econotimes, Deepfake Voice Technology: The Good. The Bad. The Future https://www.econotimes.com/Deepfake-Voice-Technology-The-Good-The-Bad-The-Future-1601278
[176] Wikipedia, Deepfake Events https://en.wikipedia.org/wiki/Deepfake#Example_events
[177] Forbes, A Voice Deepfake Was Used To Scam A CEO Out Of $243,000 https://www.forbes.com/sites/jessedamiani/2019/09/03/a-voice-deepfake-was-used-to-scam-a-ceo-out-of-243000/
[178] Joseph Steinberg, How To Prevent Facial Recognition Technology From Identifying You https://josephsteinberg.com/how-to-prevent-facial-recognition-technology-from-identifying-you/
[179] NIST, Face recognition accuracy with masks using pre-COVID-19 algorithms https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8311.pdf
[180] Wikipedia, Phishing https://en.wikipedia.org/wiki/Phishing
[181] Wikipedia, Social Engineering https://en.wikipedia.org/wiki/Social_engineering_(security)
[182] Grayshirt, https://www.grayshift.com/
[183] Securephones.io, Data Security on Mobile Devices: Current State of the Art, Open Problems, and Proposed Solutions https://securephones.io/main.pdf
[184] Wikipedia, Gag Order, https://en.wikipedia.org/wiki/Gag_order
[185] Wikipedia, National Security Letter https://en.wikipedia.org/wiki/National_security_letter
[186] Heise Online (German), https://www.heise.de/news/Gericht-zwingt-Mailprovider-Tutanota-zu-Ueberwachungsfunktion-4972460.html
[187] PCMag, Did PureVPN Cross a Line When It Disclosed User Information? https://www.pcmag.com/opinions/did-purevpn-cross-a-line-when-it-disclosed-user-information
[188] Internet Archive, Wipeyourdata, "No logs" EarthVPN user arrested after police finds logs https://archive.is/XNuVw#selection-230.0-230.1
[189] Internet Archive, Invisibler, What Everybody Ought to Know About HideMyAss https://archive.is/ag9w4#selection-136.0-136.1
[190] Wikipedia, Lavabit Suspension and Gag order, https://en.wikipedia.org/wiki/Lavabit#Suspension_and_gag_order
[191] Wikipedia, Warrant Canary https://en.wikipedia.org/wiki/Warrant_canary
[192] Washington Post, The intelligence coup of the century https://www.washingtonpost.com/graphics/2020/world/national-security/cia-crypto-encryption-machines-espionage/
[193] Swissinfo.ch, Second Swiss firm allegedly sold encrypted spying devices https://www.swissinfo.ch/eng/second-swiss-firm-allegedly-sold-encrypted-spying-devices/46186432
[194] Wired, Mind the Gap: This Researcher Steals Data With Noise, Light, and Magnets https://www.wired.com/story/air-gap-researcher-mordechai-guri/
[195] Ben Nassi, Lamphone, https://www.nassiben.com/lamphone
[196] Wikipedia, Rubber-hose Cryptanalysis https://en.wikipedia.org/wiki/Rubber-hose_cryptanalysis
[197] Defuse.ca, TrueCrypt's Plausible Deniability is Theoretically Useless https://defuse.ca/truecrypt-plausible-deniability-useless-by-game-theory.htm
[198] Wikipedia, OONI, https://en.wikipedia.org/wiki/OONI
[199] Privacy International, Timeline of SIM Card Registration Laws https://privacyinternational.org/long-read/3018/timeline-sim-card-registration-laws
[200] NYTimes, Lost Passwords Lock Millionaires Out of Their Bitcoin Fortunes https://www.nytimes.com/2021/01/12/technology/bitcoin-passwords-wallets-fortunes.html

[201] Usenix.org, Shedding too much Light on a Microcontroller's Firmware Protection https://www.usenix.org/system/files/conference/woot17/woot17-paper-obermaier.pdf
[202] Wikipedia, TAILS, https://en.wikipedia.org/wiki/Tails_(operating_system)
[203] Veracrypt Documentation, Trim Operations https://www.veracrypt.fr/en/Trim%20Operation.html
[204] Coreboot, https://www.coreboot.org/
[205] Qubes OS, Anti-Evil Maid, https://github.com/QubesOS/qubes-antievilmaid
[206] XKCD, Password Strength https://xkcd.com/936/
[207] QubesOS FAQ, https://www.qubes-os.org/faq/#is-secure-boot-supported
[208] Wikipedia, Secure Boot, https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface#Secure_boot
[209] Wikipedia, Booting https://en.wikipedia.org/wiki/Booting
[210] Wired https://www.wired.com/2013/12/better-data-security-nail-polish/
[211] Wikipedia, Virtual Machine https://en.wikipedia.org/wiki/Virtual_machine
[212] Sneak.Berlin , Your computer is not yours https://sneak.berlin/20201112/your-computer-isnt-yours/
[213] Thenextweb, Apple apps on Big Sur bypass firewalls and VPNs — this is terrible https://thenextweb.com/plugged/2020/11/16/apple-apps-on-big-sur-bypass-firewalls-vpns-analysis-macos/
[214] Wikipedia, APNS https://en.wikipedia.org/wiki/Apple_Push_Notification_service
[215] Wikipedia, Plausible Deniability https://en.wikipedia.org/wiki/Plausible_deniability
[216] Privacytools.io, Don't use Windows 10 - It's a privacy nightmare https://privacytools.io/operating-systems/#win10
[217] Wikipedia, Deniable Encryption https://en.wikipedia.org/wiki/Deniable_encryption
[218] Wikipedia, Key Disclosure Laws https://en.wikipedia.org/wiki/Key_disclosure_law
[219] GP Digital, World map of encryption laws and policies https://www.gp-digital.org/world-map-of-encryption/
[220] Wikipedia, Bitlocker https://en.wikipedia.org/wiki/BitLocker
[221] Wikipedia, Evil Maid Attack https://en.wikipedia.org/wiki/Evil_maid_attack
[222] Wikipedia, Cold Boot Attack https://en.wikipedia.org/wiki/Cold_boot_attack
[223] CITP 2008 (https://www.youtube.com/watch?v=JDaicPIgn9U)
[224] ResearchGate, Defeating Plausible Deniability of VeraCrypt Hidden Operating Systems https://www.researchgate.net/publication/318155607_Defeating_Plausible_Deniability_of_VeraCrypt_Hidden_Operating_Systems
[225] SANS.org, Mission Implausible: Defeating Plausible Deniability with Digital Forensics https://www.sans.org/reading-room/whitepapers/forensics/mission-implausible-defeating-plausible-deniability-digital-forensics-39500
[226] SourceForge, Veracrypt Forum https://sourceforge.net/p/veracrypt/discussion/technical/thread/53f33faf/
[227] Microsoft, BitLocker Countermeasures https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-countermeasures
[228] SANS, Windows ShellBag Forensics in-depth https://www.sans.org/reading-room/whitepapers/forensics/windows-shellbag-forensics-in-depth-34545
[229] University of York, Forensic data recovery from the Windows Search Database https://eprints.whiterose.ac.uk/75046/1/Forensic_Data_Recovery_From_The_Windows_Search_Database_preprint_DIIN328.pdf
[230] A forensic insight into Windows 10 Jump Lists https://cyberforensicator.com/wp-content/uploads/2017/01/1-s2.0-S1742287616300202-main.2-14.pdf
[231] Wikipedia, Gatekeeper https://en.wikipedia.org/wiki/Gatekeeper_(macOS)
[232] Wikipedia Veracrypt https://en.wikipedia.org/wiki/VeraCrypt
[233] OSTIF Veracrypt Audit, 2016, https://ostif.org/the-veracrypt-audit-results/
[234] Veracrypt Documentation, Unencrypted Data in RAM https://www.veracrypt.fr/en/Unencrypted%20Data%20in%20RAM.html
[235] Veracrypt Documentation, Data Leaks https://www.veracrypt.fr/code/VeraCrypt/plain/doc/html/Data%20Leaks.html
[236] Wikipedia, Trim https://en.wikipedia.org/wiki/Trim_(computing)
[237] Veracrypt Documentation, Trim Operations https://www.veracrypt.fr/en/Trim%20Operation.html
[238] Veracrypt Documentation, Rescue Disk https://www.veracrypt.fr/en/VeraCrypt%20Rescue%20Disk.html
[239] St Cloud State University, Forensic Research on Solid State Drives using Trim Analysis https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1141&context=msia_etds
[240] WindowsCentral, Trim Tutorial https://www.windowscentral.com/how-ensure-trim-enabled-windows-10-speed-ssd-performance
[241] Veracrypt Documentation, Trim Operation https://veracrypt.eu/en/docs/trim-operation/
[242] Black Hat 2018, Perfectly Deniable Steganographic Disk Encryption https://i.blackhat.com/eu-18/Thu-Dec-6/eu-18-Schaub-Perfectly-Deniable-Steganographic-Disk-Encryption.pdf
[243] Milan Broz's Blog, TRIM & dm-crypt ... problems? http://asalor.blogspot.com/2011/08/trim-dm-crypt-problems.html
[244] Veracrypt Documentation, Rescue Disk https://www.veracrypt.fr/en/VeraCrypt%20Rescue%20Disk.html

[245] Wikipedia, Virtualbox https://en.wikipedia.org/wiki/VirtualBox
[246] VirtualBox Ticket 17987 https://www.virtualbox.org/ticket/17987
[247] Whonix Documentation, Spectre Meltdown, https://www.whonix.org/wiki/Spectre_Meltdown#VirtualBox
[248] Wikipedia, Whonix https://en.wikipedia.org/wiki/Whonix
[249] Oracle Virtualbox Manual, Snapshots https://docs.oracle.com/en/virtualization/virtualbox/6.0/user/snapshots.html
[250] Utica College, FORENSIC RECOVERY OF EVIDENCE FROM DELETED ORACLE VIRTUALBOX VIRTUAL MACHINES https://programs.online.utica.edu/sites/default/files/Neal_6_Gonnella_Forensic_Recovery_of_Evidence_from_Deleted_Oracle_VirtualBox_Virtual_Machine.pdf
[251] Wikipedia, Spectre https://en.wikipedia.org/wiki/Spectre_(security_vulnerability)
[252] Wikipedia, Meltdown https://en.wikipedia.org/wiki/Meltdown_(security_vulnerability)
[253] Wikipedia, TOTP https://en.wikipedia.org/wiki/Time-based_One-time_Password_algorithm
[254] Wikipedia, Multi-Factor Authentication https://en.wikipedia.org/wiki/Multi-factor_authentication
[255] Qubes OS, System Requirements https://www.qubes-os.org/doc/system-requirements/
[256] Qubes OS Issues, Simulate Hibernation / Suspend-To-Disk #2414 https://github.com/QubesOS/qubes-issues/issues/2414
[257] Wikipedia, TOTP https://en.wikipedia.org/wiki/Time-based_One-time_Password_algorithm
[258] Wikipedia, Multi-Factor Authentication https://en.wikipedia.org/wiki/Multi-factor_authentication
[259] Wikipedia, Captcha https://en.wikipedia.org/wiki/CAPTCHA
[260] Wikipedia, Turing Test https://en.wikipedia.org/wiki/Turing_test
[261] Google reCaptcha https://www.google.com/recaptcha/about/
[262] hCaptcha https://www.hcaptcha.com/
[263] hCaptcha hCaptcha Is Now the Largest Independent CAPTCHA Service, Runs on 15% Of The Internet https://www.hcaptcha.com/post/hcaptcha-now-the-largest-independent-captcha-service
[264] Nearcyan.com, You (probably) don't need ReCAPTCHA https://nearcyan.com/you-probably-dont-need-recaptcha/
[265] ArsTechnica, "Google's reCAPTCHA turns "invisible," will separate bots from people without challenges" https://arstechnica.com/gadgets/2017/03/googles-recaptcha-announces-invisible-background-captchas/
[266] BlackHat Asia 2016, "I'm not a human: Breaking the Google reCAPTCHA", https://www.blackhat.com/docs/asia-16/materials/asia-16-Sivakorn-Im-Not-a-Human-Breaking-the-Google-reCAPTCHA-wp.pdf
[267] Google Blog https://security.googleblog.com/2014/12/are-you-robot-introducing-no-captcha.html
[268] Tor Project Community, Cloudflare Captcha Monitoring https://community.torproject.org/gsoc/cloudflare-captcha-monitoring/
[269] Cloudflare Blog, Cloudflare supports Privacy Pass https://blog.cloudflare.com/cloudflare-supports-privacy-pass/
[270] Privacy International, Timeline of SIM Card Registration Laws https://privacyinternational.org/long-read/3018/timeline-sim-card-registration-laws
[271] Wikipedia, Device Fingerprinting https://en.wikipedia.org/wiki/Device_fingerprint
[272] Developers Google Blog,
Guidance to developers affected by our effort to block less secure browsers and applications
https://developers.googleblog.com/2020/08/guidance-for-our-effort-to-block-less-secure-browser-and-apps.html
[273] IDMB, The Social Dilemma https://www.imdb.com/title/tt11464826/
[274] Google Help, Access age-restricted content & features https://support.google.com/accounts/answer/10071085
[275] Wikipedia, Dark Pattern https://en.wikipedia.org/wiki/Dark_pattern
[276] The Verge, Tinder will give you a verified blue check mark if you pass its catfishing test https://www.theverge.com/2020/1/23/21077423/tinder-photo-verification-blue-checkmark-safety-center-launch-noonlight
[277] DigitalInformationWorld, Facebook will now require you to Create a Video Selfie for Identity Verification https://www.digitalinformationworld.com/2020/03/facebook-is-now-demanding-some-users-to-create-a-video-selfie-for-identity-verification.html
[278] Vice.com, PornHub Announces 'Biometric Technology' to Verify Users https://www.vice.com/en/article/m7a4eq/pornhub-new-verification-policy-biometric-id
[279] Wikipedia, Trust but verify https://en.wikipedia.org/wiki/Trust,_but_verify
[280] Wikipedia, Espionage, Organization https://en.wikipedia.org/wiki/Espionage#Organization
[281] Wikipedia, Sim Swapping https://en.wikipedia.org/wiki/SIM_swap_scam
[282] Whonix Documentation, https://www.whonix.org/wiki/Tor#Edit_Tor_Configuration
[283] Tor Browser Documentation, https://support.torproject.org/tbb/tbb-editing-torrc/
[284] Facebook Onion Website http://facebookcorewwwi.onion
[285] Google Help https://support.google.com/accounts/answer/114129?hl=en
[286] Google Help https://support.google.com/google-ads/answer/7474263?hl=en
[287] Google, Your account is disabled https://support.google.com/accounts/answer/40695
[288] Google, Request to restore the account https://support.google.com/accounts/contact/disabled2
[289] Jumio, ID verification features https://www.jumio.com/features/

[290] Privacytools.io Recommended E-mail Providers https://privacytools.io/providers/email/
[291] ProtonMail Human Verification System https://ProtonMail.com/support/knowledge-base/human-verification/
[292] Twitter Appeal Form https://help.twitter.com/forms/general
[293] KnowYourMeme, Good Luck, I'm Behind 7 Proxies https://knowyourmeme.com/memes/good-luck-im-behind-7-proxies
[294] Youwave, https://youwave.com/
[295] Bluestacks, https://www.bluestacks.com/
[296] Wikipedia, TPM https://en.wikipedia.org/wiki/Trusted_Platform_Module
[297] Wikipedia, Wear Leveling https://en.wikipedia.org/wiki/Wear_leveling
[298] Wikipedia, Trim https://en.wikipedia.org/wiki/Write_amplification#TRIM
[299] Wikipedia, Write Amplification https://en.wikipedia.org/wiki/Write_amplification
[300] Wikipedia, Trim Disadvantages https://en.wikipedia.org/wiki/Trim_(computing)#Disadvantages
[301] Wikipedia, Garbage Collection https://en.wikipedia.org/wiki/Write_amplification#Garbage_collection
[302] Techgage, Too TRIM? When SSD Data Recovery is Impossible https://techgage.com/article/too_trim_when_ssd_data_recovery_is_impossible/
[303] ResearchGate, Live forensics method for acquisition on the Solid State Drive (SSD) NVMe TRIM function https://www.researchgate.net/publication/341761017_Live_forensics_method_for_acquisition_on_the_Solid_State_Drive_SSD_NVMe_TRIM_function
[304] ElcomSoft, Life after Trim: Using Factory Access Mode for Imaging SSD Drives https://blog.elcomsoft.com/2019/01/life-after-trim-using-factory-access-mode-for-imaging-ssd-drives/
[305] Forensic Focus, Forensic Acquisition Of Solid State Drives With Open Source Tools https://www.forensicfocus.com/articles/forensic-acquisition-of-solid-state-drives-with-open-source-tools/
[306] ResearchGate, Solid State Drive Forensics: Where Do We Stand? https://www.researchgate.net/publication/325976653_Solid_State_Drive_Forensics_Where_Do_We_Stand
[307] Wikipedia, Parted Magic https://en.wikipedia.org/wiki/Parted_Magic
[308] Wikipedia, hdparm https://en.wikipedia.org/wiki/Hdparm
[309] GitHub, nvme-cli https://github.com/linux-nvme/nvme-cli
[310] PartedMagic Secure Erase, https://partedmagic.com/secure-erase/
[311] Partedmagic NVMe Secure Erase, https://partedmagic.com/nvme-secure-erase/
[312] UFSExplorer, Can I recover data from an encrypted storage? https://www.ufsexplorer.com/solutions/data-recovery-on-encrypted-storage.php
[313] Apple Developer Documentation, https://developer.apple.com/library/archive/documentation/FileManagement/Conceptual/APFS_Guide/FAQ/FAQ.html
[314] EFF, How to: Delete Your Data Securely on MacOS https://ssd.eff.org/en/module/how-delete-your-data-securely-macos
[315] Privacytools.io, Productivity tools https://www.privacytools.io/software/productivity/
[316] Whonix Documentation, Scrubbing Metadata https://www.whonix.org/wiki/Metadata
[317] TAILS documentation, MAT https://gitlab.tails.boum.org/tails/blueprints/-/wikis/doc/mat/
[318] Praxis Films, Open Letter from Laura Poitras https://www.praxisfilms.org/open-letter-from-laura-poitras/
[319] Wikipedia, SecureDrop https://en.wikipedia.org/wiki/SecureDrop
[320] DuckDuckGo help, Cache https://help.duckduckgo.com/duckduckgo-help-pages/features/cache/
[321] DuckDuckGo help, Sources https://help.duckduckgo.com/duckduckgo-help-pages/results/sources/
[322] Wikipedia, Dead Drop https://en.wikipedia.org/wiki/Dead_drop
[323] Wikipedia, Secure Communication Obfuscation https://en.wikipedia.org/wiki/Obfuscation#Secure_communication
[324] Wikipedia, Steganography https://en.wikipedia.org/wiki/Steganography
[325] Wikipedia, Kleptography https://en.wikipedia.org/wiki/Kleptography
[326] Wikipedia, Koalang https://en.wikipedia.org/wiki/Koalang
[327] Tor Documentation, Snowflake, https://gitlab.torproject.org/legacy/trac/-/wikis/doc/Snowflake/
[328] Privacytools.io, Operating Systems https://privacytools.io/operating-systems/
[329] Brave Support, What is a Private Window with Tor? https://support.brave.com/hc/en-us/articles/360018121491-What-is-a-Private-Window-with-Tor-
[330] Medium.com, The Windows USN Journal https://medium.com/velociraptor-ir/the-windows-usn-journal-f0c55c9010e
[331] Medium.com, Digging into the System Resource Usage Monitor (SRUM) https://medium.com/velociraptor-ir/digging-into-the-system-resource-usage-monitor-srum-afbadb1a375
[332] SANS, Timestamped Registry & NTFS Artifacts from Unallocated Space https://www.sans.org/blog/timestamped-registry-ntfs-artifacts-from-unallocated-space/
[333] DBAN, https://dban.org/
[334] NYTimes, Lost Passwords Lock Millionaires Out of Their Bitcoin Fortunes https://www.nytimes.com/2021/01/12/technology/bitcoin-passwords-wallets-fortunes.html
[335] CrystalDiskInfo https://crystalmark.info/en/software/crystaldiskinfo/

[336] Wikipedia, Faraday Cage, https://en.wikipedia.org/wiki/Faraday_cage
[337] Edith Cowan University, A forensic examination of several mobile device Faraday bags & materials to test their effectiveness materials to test their effectiveness https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1165&context=adf
[338] YouTube, Fingerprinting Paper - Forensic Education https://www.youtube.com/watch?v=sO98kDLkh-M
[339] Wikipedia, Touch DNA, https://en.wikipedia.org/wiki/Touch_DNA
[340] TheDNAGuide, DNA from Postage Stamps or Hair Samples? Yeeesssss..... https://www.yourdnaguide.com/ydgblog/dna-hair-samples-postage-stamps
[341] GitHub, Mhinkie, OONI-Detection https://github.com/mhinkie/ooni-detection
[342] Lenny Zeltser, Analyzing Malicious Documents Cheat Sheet https://zeltser.com/analyzing-malicious-documents/