The Hitchhiker's Guide to Online Anonymity

// The comprehensive guide for online anonymity and OpSec.

Anonymous Planet

Creative Commons BY-SA 4.0

https://anonymousplanet.net

v1.2.7 – October 23 2026

Table of Contents

Precautions while reading this guide and accessing the various links 1
Requirements & Limitations 1
Introduction 2
Your Network 3
Your IP address 3
Your DNS and IP requests 4
Your RFID enabled devices 7
The Wi-Fi and Bluetooth devices around you 8
Rogue Wi-Fi Access Points 9
Traffic Anonymization 10
Traffic analysis and the limits of Tor 12
Some Devices can be tracked even when offline 13
Your Hardware Identifiers 13
Windows USB bus telemetry (device enumeration history) 13
Your IMEI and IMSI 14
Your Wi-Fi or Ethernet MAC address 15
Your Bluetooth MAC address 16
Your CPU 16
Your OS and App telemetry services 17
Windows COM Class Identifiers (CLSID) and GUIDs 18
Windows Global Device Identifier (GDID) 18
Your Smart Devices 20
Yourself 20
Your Metadata 20
Your Digital Footprint 21
IRL and OSINT 22
Your Face, Voice, Biometrics, and Pictures 23
Gait Recognition and Other Long-Range Biometrics 23
Phishing and Social Engineering 25
Malware, exploits, and viruses 26
Malware in your files/documents/e-mails 26
Malware and Exploits in your apps and services 26
Malicious USB devices 26
Malware and backdoors in your Hardware Firmware and Operating System 27
Your files, documents, pictures, and videos 28
Properties and Metadata 28
Watermarking 29
Pixelized or Blurred Information 30
Your Crypto Transactions 31
Your Cloud Backup & Sync Services 31
Microarchitectural Side-channel Deanonymization Attacks 32
Alternative to NoScript for all other browsers 33
Local Data Leaks and Forensics 33
Bad Cryptography 33
No Log Policies 35
Some Advanced targeted techniques 36
Some bonus resources 37
Picking your route 38
Timing limitations 39
Budget & Material limitations 39
Skills 40
Adversarial considerations 40
Steps for all routes 42
Getting used to using better passwords 42
Getting an anonymous Phone number 42
Get a USB key 43
Find some safe places with decent public Wi-Fi 44
The Tor Browser route 44
Windows, Linux, and macOS 44
Android 44
iOS 46
Important Warning 47
The Tails route 47
Tor Browser settings on Tails 48
Persistent Plausible Deniability using Whonix & Tails 48

Steps for all other routes 50
Get a dedicated laptop 50
Some laptop recommendations 50
Bios/UEFI/Firmware Settings of your laptop 51
Physically Tamper protect your laptop 52
The Whonix route 53
Picking your Host OS 53
Linux Host OS 57
macOS Host OS 59
Windows Host OS 60
Virtualbox on your Host OS 69
Pick your connectivity method 70
Getting an anonymous VPN/Proxy 75
Whonix 75
Tor over VPN 77
Whonix Virtual Machines 77
Whonix 18.x Improvements and Best Practices 78
Pick your guest workstation Virtual Machine 78
Linux Virtual Machine (Whonix or Linux) 78
Windows 10/11 Virtual Machine 79
Android Virtual Machine 80
macOS Virtual Machine 81
KeepassXC 83
VPN client installation (cash/Monero paid) 83
(Optional) VM kill switch 83
The Qubes Route 92
Pick your connectivity method 92
Getting an anonymous VPN/Proxy 97
Note about Plausible Deniability 97
Installation 97
Lid Closure Behavior 98
Anti Evil Maid (AEM) 98
Connect to a Public Wi-Fi 98
Upgrading Qubes OS from 4.0.x to 4.1.x (you should do it) 99
Updating Qubes OS 99
Upgrade to Whonix 18 99
Hardening Qubes OS 100
Setup the VPN ProxyVM 100
Setup a safe Browser within Qubes OS 104
Setup an Android VM 104
KeePassXC 106
Tutorial for installing Windows based VMs on Qubes OS 106
Methodology to Prevent Anonymity 106
Phone verification 109
E-Mail verification 109
User details checking 109
Proof of ID verification 110
IP Filters 110
Browser and Device Fingerprinting 110
Human interaction 111
User Moderation 111
Behavioral Analysis 112
Financial transactions 112
Sign-in with some platform 112
Facial Recognition and Biometrics 112
Manual reviews 113
Getting Online 114
Creating new identities 114
Checking if your Tor Exit Node is terrible 118
The Real-Name System 118
About paid services 119
How to Chat & Share Files Anonymously 128
How to share files publicly 133
Redacting Documents Safely 134
Communicating sensitive information 135
Offline Backups 136
Selected Files Backups 136
Full Disk/System Backups 137
Online Backups 139
Files 139
Information 139
Synchronizing your files between devices Online 140
Understanding HDD vs SSD 140
Wear-Leveling 141
Trim Operations 141
Garbage Collection 142
Conclusion 142

How to securely wipe your whole Laptop/Drives 143
Linux (all versions including Qubes OS) 144
Windows 145
macOS 146
How to securely delete specific files/folders/data on your HDD/SSD and Thumb drives 147
Windows 147
Linux (non-Qubes OS) 149
Linux (Qubes OS) 150
macOS 151
Some additional measures against forensics 152
Metadata auditing 153
Tails 156
Whonix 156
macOS 156
Linux (Qubes OS) 158
Linux (non-Qubes) 158
Windows 158
Removing your identities from search engines and other platforms 162
Hide Communications in Plain Sight 164
How to spot if someone has been searching your stuff 164
Digital and Online OPSEC 165
Physical and IRL OPSEC 165
If you have some time 167
If you have no time 167
A small final editorial note 167
Windows 10 (See below for Windows 11) 168
Installation 168
Privacy Settings 169
Windows 11 170
Installation 170
Privacy Settings 171
Windows 10 173
Windows 11 174
If you can use Tor 175
If you cannot use Tor 175
Windows 176
Linux 177
Tools that provide a boot disk for wiping from boot 177
Tools that provide only support from running OS (for external drives) 177
Windows 178
Trim Support 178
ATA/NVMe Operations (Secure Erase/Sanitize) 178
Linux 178
Trim Support 178
ATA/NVMe Operations (Secure Erase/Sanitize) 178
macOS 178
Trim Support 178
ATA/NVMe Operations (Secure Erase/Sanitize) 179
Cash/Monero-Paid VPN 182
Self-hosted VPN/Proxy on a Monero/Cash-paid VPS (for users more familiar with Linux) 183
VPN VPS 183
Socks Proxy VPS 183
Integrity (if available) 188
Authenticity (if available) 188
Security (checking for actual malware) 189
Anti-Virus Software 189
Manual Reviews 190
Portable Apps 191
Bootable Live Systems 192
Precautions 192
Brave 193
Ungoogled-Chromium 193
Edge 194

Safari 194
Firefox 194
Tor Browser 194
Brave 195
Ungoogled-Chromium 196
Edge 197
Safari 198
Firefox 198
Normal settings 198
Advanced settings 199
Addons to install/consider 201
Bonus resources 202
Nested virtualization risks 203
Installation 204
Usage and Precautions 204
Using Bitcoin anonymously option 206
Using Monero anonymously option 207
Warning about special tumbling, mixing, coinjoining privacy wallets and services 207
When converting from BTC to Monero 207
Introduction 209
What does an adversary look for when examining your writing? 209
Examples 209
How to counteract the efforts of your adversary 210
What different linguistic choices could say about you 210
Emoticons 210
Structural features 210
Spelling slang and symbols 211
Techniques to prevent writeprinting 211
Spelling and grammar checking 211
Translation technique 211
Search and replace 211
Final advice 212
Bonus links 212
General Crypto Swapping 213
BTC to Monero only 213
Debian 11 VM 213
Ubuntu 20.04/21.04/21.10 VM 214
Windows 10/11 VM 214
Whonix Workstation 16 VM 214
Hector Monsegur (Sabu) - LulzSec, 2011 218
Ross Ulbricht - Silk Road, 2013 218
"Defcon" (Blake Benthall) - Silk Road 2.0, 2014 218
Jeremy Hammond - AntiSec, 2012 218
Peter Stokes - Scattered Spider, 2026 219
Common threads 219
Signal's PQXDH 220
What you should do now 220
A note on Monero 220
What features are measured 221
Deployed tools and real cases 221
What works and what does not 221
Honest threat model 221

1. **Do you want to understand the current state of online privacy and anonymity, while not necessarily getting too technical about it?**
    - Read the Introduction, Requirements, understanding some basics beginning with your network, and the final notes.
2. **Do you want to learn, but also learn how to remove some online information about you?**
    - All of the items in no. 1 and how to clean your identities from search engines and other platforms to get a good idea of how to clean your data off the web.
3. **Do you want to do the above and create online anonymous identities online safely and securely.**
    - Read the whole thing. A specific list of the most vital things to read in the guide will be coming later, but you should read the whole thing.

Precautions while reading this guide and accessing the various links

- **YouTube Videos** have a **[Invidious]** link next to them for accessing content through an Invidious Instance (in this case yewtu.be hosted in the Netherlands) for increased privacy. It is recommended to use these links when possible. See https://github.com/iv-org/invidious [Archive.org] for more information.
- **Twitter** links have a **[Nitter]** link next to them for accessing content through a Nitter Instance (in this case nitter.net) for increased privacy. It is recommended to use these links when possible. See https://github.com/zedeus/nitter [Archive.org] for more information.
- **Wikipedia** links have a **[Wikiless]** link next to them for accessing content through a Wikiless Instance (in this case wikiless.tiekoetter.com) for increased privacy. It is recommended to use these links when possible. See https://codeberg.org/orenom/wikiless [Archive.org] for more information.
- **Medium** links have **[Scribe.rip]** link next to them for accessing content through a Scribe.rip Instance for increased privacy. Again, it is recommended to use these links when possible. See https://scribe.rip/ [Archive.org] for more information.

You could also install the LibRedirect extension on your browser to ease the redirects. [Archive.org]:

- Firefox: https://addons.mozilla.org/en-US/firefox/addon/libredirect/
- Chromium-based browsers (Chrome, Brave, Edge): https://github.com/libredirect/libredirect/blob/master/chromium.md

If you are having trouble accessing any of the many academic articles referenced in this guide due to paywalls, feel free to use Sci-Hub (https://en.wikipedia.org/wiki/Sci-Hub [Wikiless] [Archive.org]) or LibGen (https://en.wikipedia.org/wiki/Library_Genesis [Wikiless] [Archive.org]) for finding and reading them. Because Science should be free. All of it. If you are faced with a paywall accessing some resources, consider using https://12ft.io/.

Finally note that this guide does mention and even recommends various commercial services (such as VPNs, CDNs, e-mail providers, hosting providers...) **but is not endorsed or sponsored by any of them in any way. There are no referral links and no commercial ties with any of these providers. This project is 100% non-profit and only relying on donations.**

Requirements & Limitations

- Understanding of the English language (in this case American English).
- Be a permanent resident in Germany where the courts have upheld the legality of not using real names on online platforms (§13 VI of the German Telemedia Act of 2007'). **Alternatively, be a resident of any other country where you can confirm and verify the legality of this guide yourself.**
- This guide will assume you already have access to some (Windows/Linux/macOS) laptop computer - ideally not a work/shared device - and a basic understanding of how computers work.
- Have patience, as this process could take several weeks to complete if you want to go through all the content.
- Have some free time on your hands to dedicate to this process (depending on which route you pick).
- Be prepared to read a lot of references (do read them), guides (do not skip them), and tutorials thoroughly (do not skip them either).
- Don't be evil (for real this time).
- Understand that there is no common path that will be both quick and easy.

This guide is not intended for:

- Creating bot accounts of any kind.
- Creating impersonation accounts of existing people (such as identity theft).
- Helping malicious actors conduct unethical, criminal, or illicit activities (such as trolling, stalking, disinformation, misinformation, harassment, bullying, or fraud).
- Use by minors.

Introduction

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

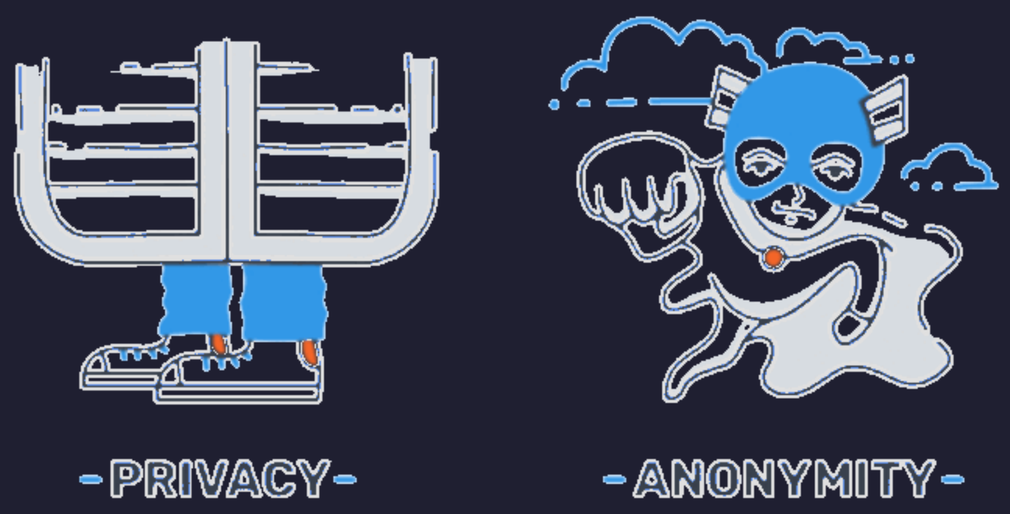
Making a social media account with a pseudonym or artist/brand name is easy. And it is enough in most use cases to protect your identity as the next George Orwell. There are plenty of people using pseudonyms all over Facebook/Instagram/Twitter/LinkedIn/TikTok/Snapchat/Reddit/... But the vast majority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

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

This guide aims to provide an introduction to various de-anonymization techniques, tracking techniques, ID verification techniques, and optional guidance to creating and maintaining **reasonably and truly** online anonymous identities including social media accounts safely. This includes mainstream platforms and not only the privacy-friendly ones.

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

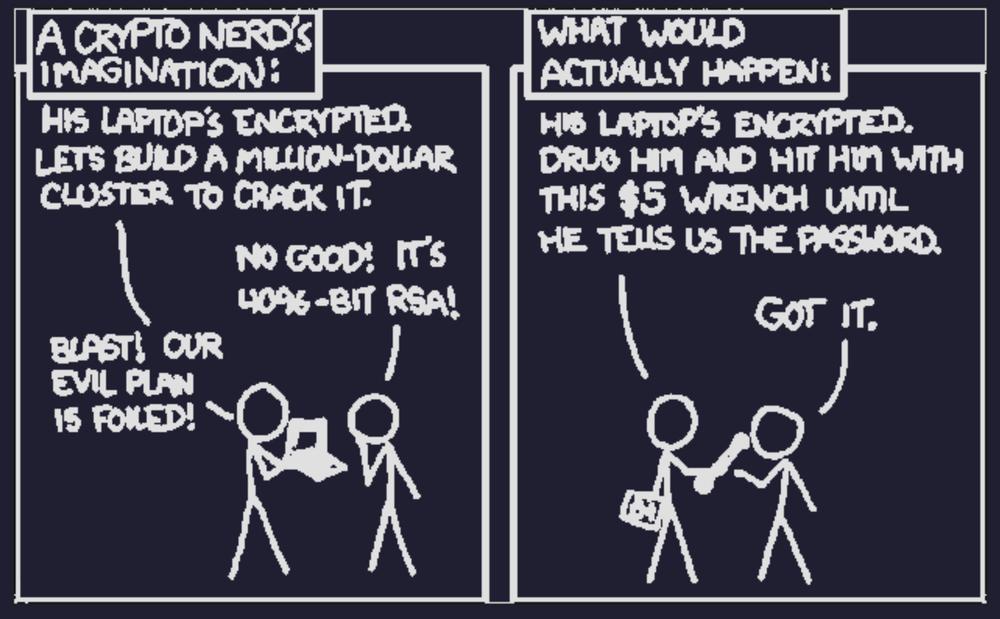
- **Privacy is about people knowing who you are but not knowing what you are doing.**
- **Anonymity is about people knowing what you are doing but not knowing who you are .**



(Illustration from)

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

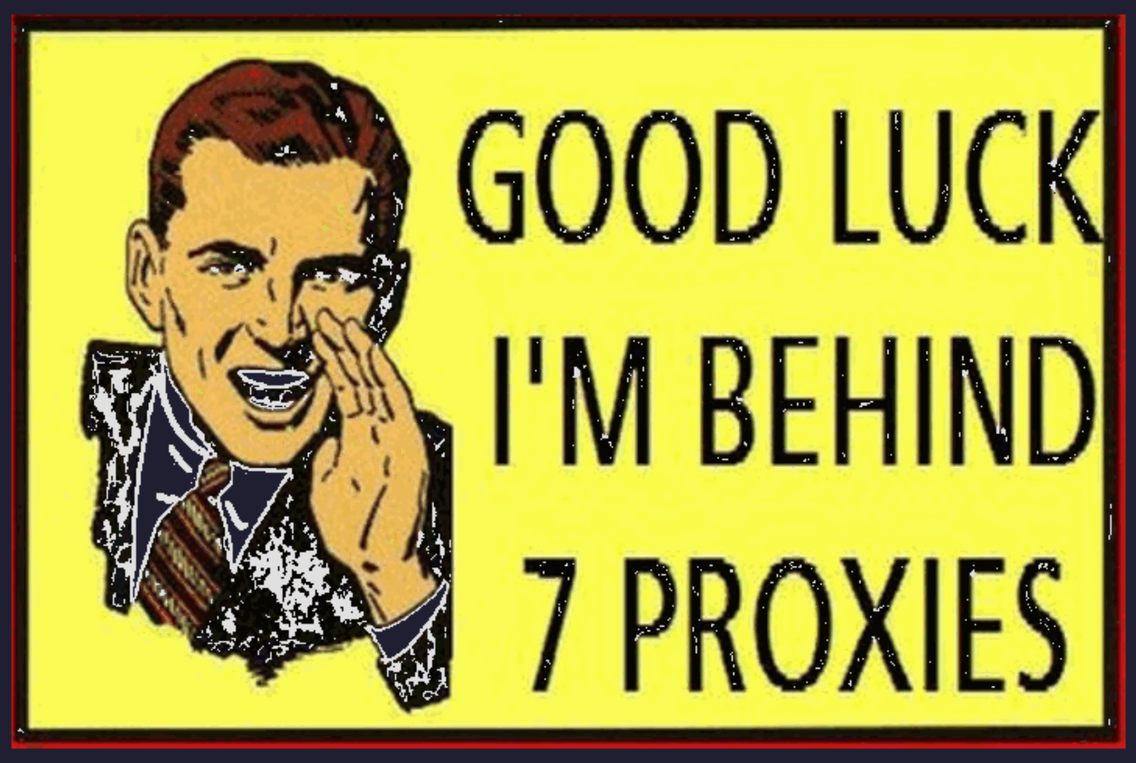
You must consider your threat model before going further.



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

Will this guide help you protect your privacy from OSINT researchers like Bellingcat, Doxing trolls on 4chan, and others that have no access to the NSA toolbox? More likely. Tho we would not be so sure about 4chan.

Here is a basic simplified threat model for this guide:



(Note that the "magical amulets/submarine/fake your own death" jokes are quoted from the excellent article "This World of Ours" by James Mickens, 2014.)

Disclaimer: Jokes aside (magical amulet...). Of course, there are also advanced ways to mitigate attacks against such advanced and skilled adversaries but those are just out of the scope of this guide. It is crucially important that you understand the limits of the threat model of this guide. And therefore, this guide will not double in size to help with those advanced mitigations as this is just too complex and will require an exceedingly high knowledge and skill level that is not expected from the targeted audience of this guide.

The EFF provides a few security scenarios of what you should consider depending on your activity. While some of those tips might not be within the scope of this guide (more about Privacy than Anonymity), they are still worth reading as examples. See https://ssd.eff.org/en/module-categories/security-scenarios [Archive.org].

If you want to go deeper into threat modeling, see Threat modeling resources.

You might think this guide has no legitimate use but there are many'''''' such as:

- Evading Online Censorship
- Evading Online Oppression
- Evading Online Stalking, Doxxing, and Harassment
- Evading Online Unlawful Government Surveillance
- Anonymous Online Whistle Blowing
- Anonymous Online Activism
- Anonymous Online Journalism
- Anonymous Online Legal Practice
- Anonymous Online Academic Activities (e.g., accessing country-blocked scientific research)

This guide is written with hope for those **good-intended individuals** who might not be knowledgeable enough to consider the big picture.

Lastly, use it at your own risk. Anything in here is not legal advice and you should verify compliance with your local law before use (IANAL). "Trust but verify" all the information yourself (or even better, "Never Trust, always verify"). We strongly encourage you to inform yourself and do not hesitate to check any information in this guide with outside sources in case of doubt. Please do report any mistake you spot to us as we welcome criticism. Even harsh but sound criticism is welcome and will result in having the necessary corrections made as quickly as possible.

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

Your Network

Your IP address

Disclaimer: this whole paragraph is about your public-facing Internet IP and not your local network IP.

Your IP address is the most known and obvious way you can be tracked. That IP is the IP you are using at the source. This is where you connect to the internet. That IP is usually provided by your ISP (xDSL, Mobile, Cable, Fiber, Cafe, Bar, Friend, Neighbor). Most countries have data retention regulations that mandate 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 original one) leaks at any point for any reason, it can be used to track down you directly. In many countries, you will not be able to have internet access without providing some form of identification to the provider (address, ID, real name, e-mail ...).

Needless to say, that most platforms (such as social networks) will also keep (sometimes indefinitely) the IP addresses you used to sign-up and sign into their services.

Here are some online resources you can use to find some information about your current **public IP** right now:

- Find your IP:
  - https://resolve.rs/
  - https://www.dnsleaktest.com/ (Bonus, check your IP for DNS leaks)
- Find your IP location or the location of any IP:
  - https://resolve.rs/ip/geolocation.html
- Find if an IP is "suspicious" (in blacklists) or has downloaded "things" on some public resources:
  - https://mxtoolbox.com/blacklists.aspx
  - https://www.virustotal.com/gui/home/search
  - https://iknowwhatyoudownload.com (Take this with a grain of salt, it might not show anything interesting and has limited data sources. This is more for fun than anything serious.)
- Registration information of an IP (most likely your ISP or the ISP of your connection who most likely know who is using that IP at any time):
  - https://whois.domaintools.com/
- Check for open-services or open devices on an IP (especially if there are leaky Smart Devices on it):
  - https://www.shodan.io/host/185.220.101.134 (replace the IP by your IP or any other, or change in the search box, this example IP is a Tor Exit node)
- Various tools to check your IP such as block-lists checkers and more:
  - https://browserleaks.com/ip
  - https://www.whatismyip.com
- Would you like to know if you are connected through Tor?
  - https://check.torproject.org

For those reasons, you will need to obfuscate and hide that origin IP (the one tied to your identification) or hide it through a combination of various means:

- Using a public Wi-Fi service (free).
- Using the Tor Anonymity Network (free).
- Using VPN services anonymously (anonymously paid with cash or Monero).

Do note that, unfortunately, these solutions are not perfect, and you will experience performance issues.

All those will be explained later in this guide.

Your DNS and IP requests

DNS is the internet's address book: every time your browser opens a connection to `www.google.com`, it first asks a DNS server to translate that name into an IP address. Your browser, and most other apps on your device, do this constantly in the background.

For a visual introduction to how DNS works, see: https://www.youtube.com/watch?v=vrxwXXytEuI [Invidious]

The default DNS problem

By default, your DNS queries go to your ISP's resolver. This creates several compounding problems:

- **Logging.** ISPs are often subject to data retention laws and may log every domain you resolve, building a complete record of your browsing history. That log can be subpoenaed, sold, or breached.
- **Censorship.** DNS blocking is the most common censorship mechanism worldwide - ISPs simply return a false address for blocked domains.
- **Tampering.** Even if you configure a private DNS resolver, your ISP can intercept and rewrite DNS responses in transit, because standard DNS traffic is unencrypted.
- **Hardcoded resolvers.** Many devices bypass system DNS settings entirely. Roughly 70% of Smart TVs and 46% of game consoles use hardcoded DNS servers that cannot be changed through OS settings - you have to block them at the network level or accept the exposure.

Encrypted DNS (DoH / DoT)

Encrypting your DNS queries using DoH (DNS over HTTPS) or DoT (DNS over TLS) prevents your ISP from reading or tampering with them in transit. You can run this:

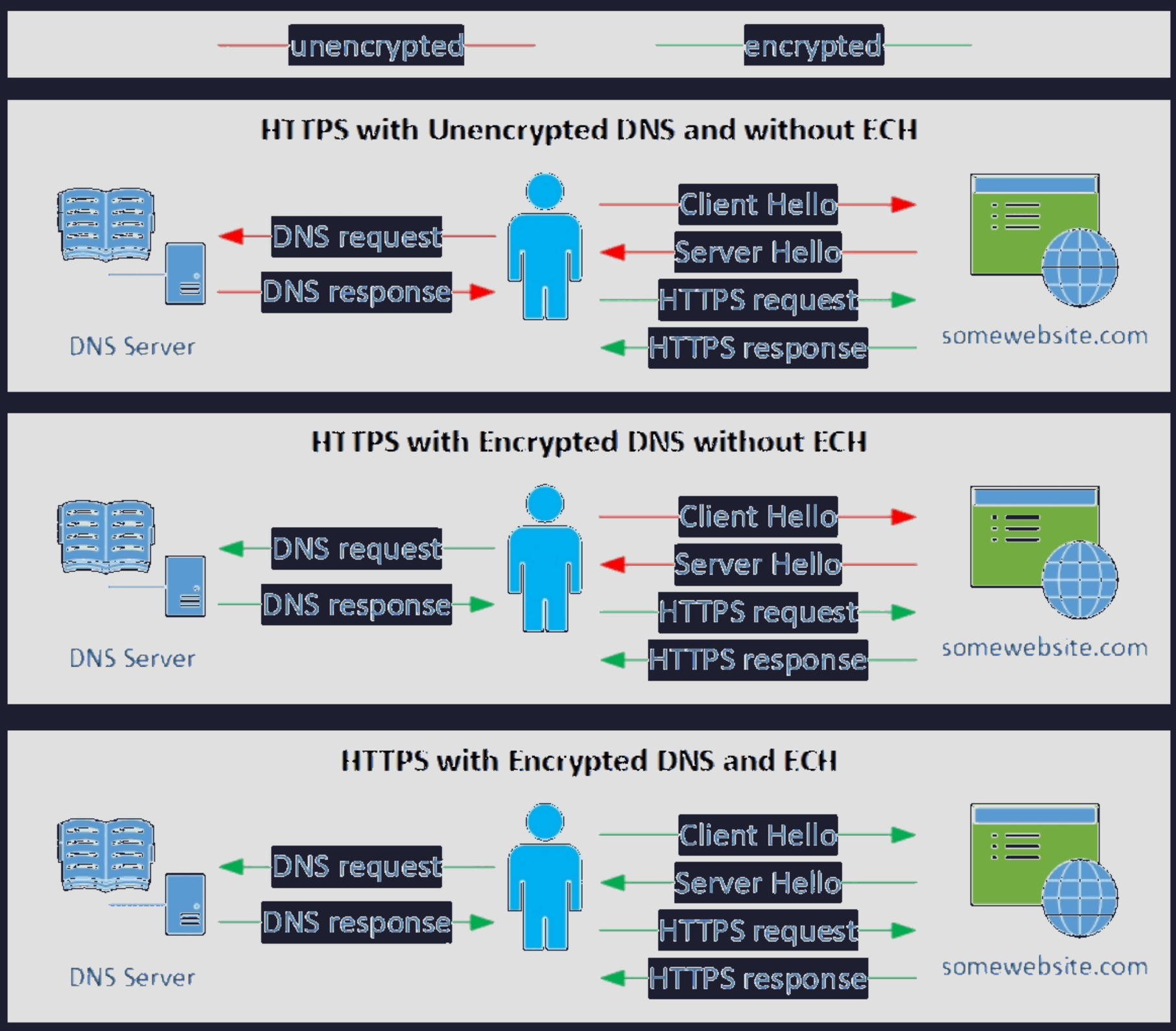
- **Locally**, with a self-hosted resolver such as Pi-hole.
- **Remotely**, via a privacy-focused provider such as nextdns.io.
- **Automatically**, through your VPN provider's DNS, or through the Tor network.

This is a meaningful improvement over plaintext DNS - but it is not sufficient on its own.

> **Note:** This guide does not endorse or recommend Cloudflare services. Cloudflare is referenced here only for technical context.

SNI leakage and ECH

Even with encrypted DNS, the TLS handshake itself leaks the destination domain name through Server Name Indication (SNI). An observer watching your connection - your ISP, a network monitor, or a rogue access point - can see which domain you are connecting to from the SNI field, even though the DNS query itself was encrypted.



Encrypted Client Hello (ECH) (formerly eSNI) is the fix: it encrypts the SNI field so the destination domain is hidden from network observers. However, ECH has significant deployment constraints:

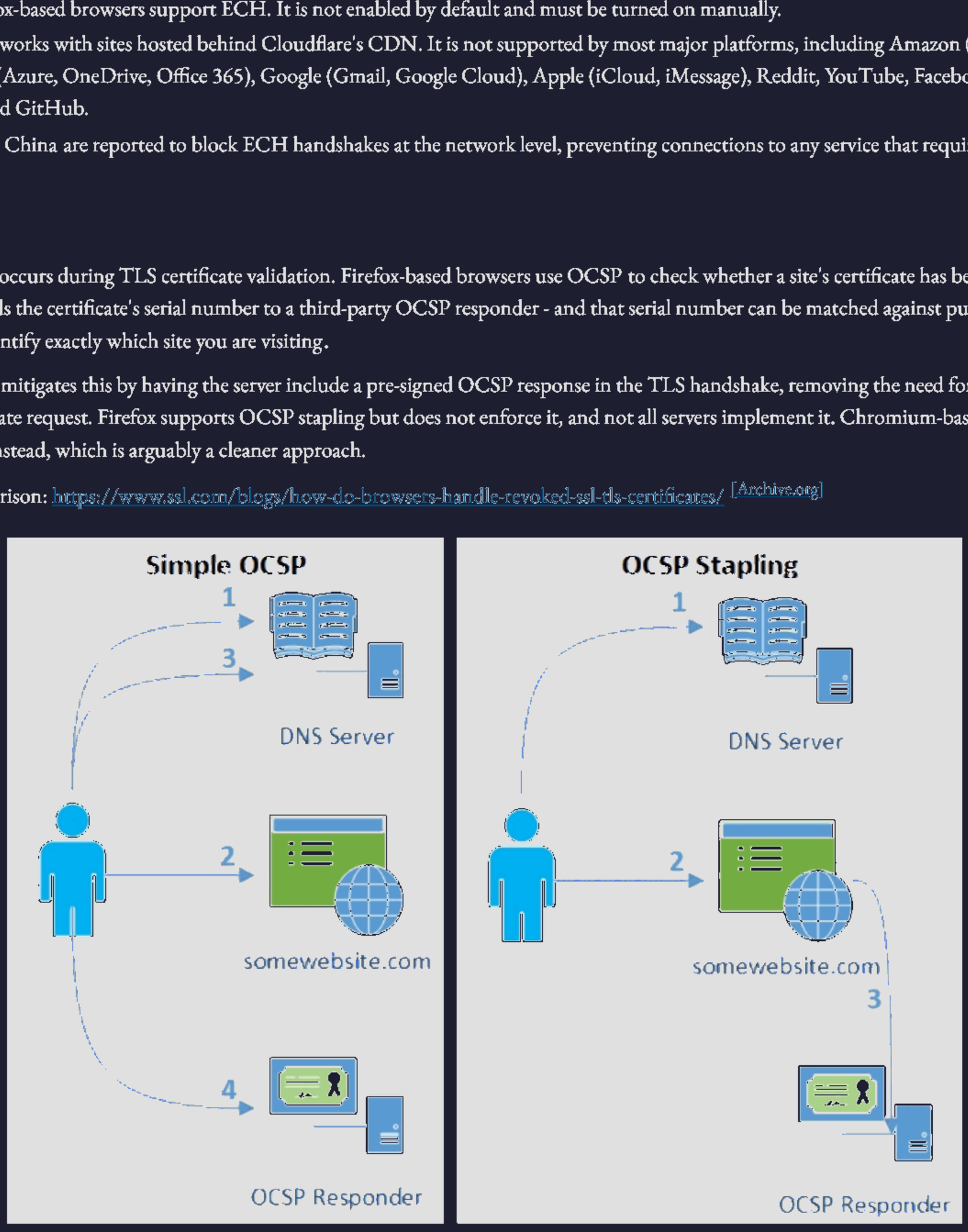
- Only Firefox-based browsers support ECH. It is not enabled by default and must be turned on manually.
- ECH only works with sites hosted behind Cloudflare's CDN. It is not supported by most major platforms, including Amazon (AWS, Twitch), Microsoft (Azure, OneDrive, Office 365), Google (Gmail, Google Cloud), Apple (iCloud, iMessage), Reddit, YouTube, Facebook, Instagram, Twitter, and GitHub.
- Russia and China are reported to block ECH handshakes at the network level, preventing connections to any service that requires ECH to function.

OCSP leakage

A separate leak occurs during TLS certificate validation. Firefox-based browsers use OCSP to check whether a site's certificate has been revoked. This check sends the certificate's serial number to a third-party OCSP responder - and that serial number can be matched against public certificate databases to identify exactly which site you are visiting.

OCSP stapling mitigates this by having the server include a pre-signed OCSP response in the TLS handshake, removing the need for the browser to make a separate request. Firefox supports OCSP stapling but does not enforce it, and not all servers implement it. Chromium-based browsers use CRLSets instead, which is arguably a cleaner approach.

Browser comparison: https://www.ssl.com/blogs/how-do-browsers-handle-revoked-ssl-tls-certificates/ [Archive.org]

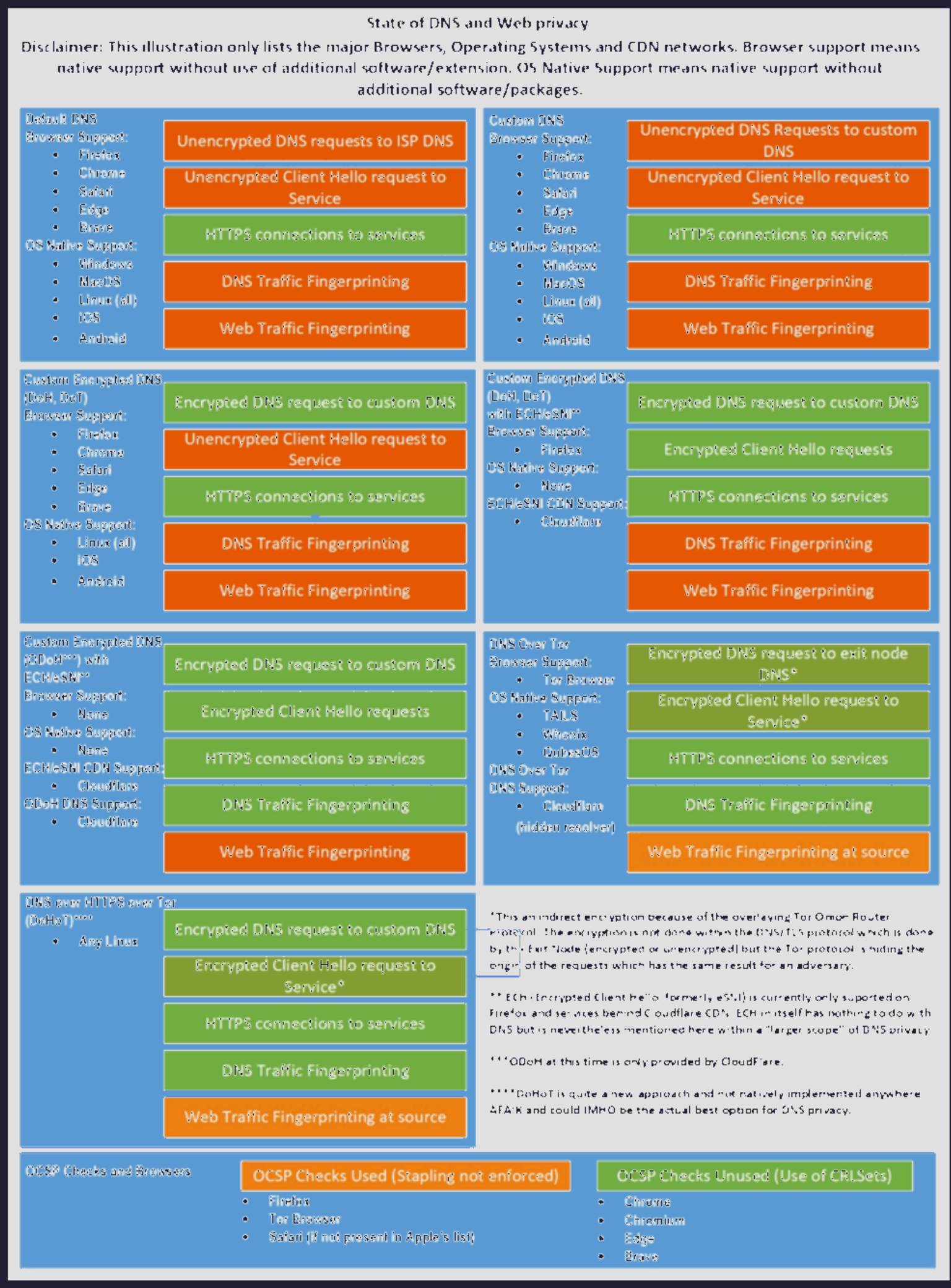


Traffic analysis defeats all of the above

Even with encrypted DNS, ECH, and OCSP stapling all in place, traffic analysis can still identify the sites you visit by matching IP addresses - since most websites have unique IPs - or by fingerprinting traffic patterns. DNS over Tor shows the strongest DNS privacy in current research, but it can still be defeated by other methods (see Traffic Anonymization).

Two advanced options exist for DNS beyond standard DoH/DoT:

- **Tor Hidden DNS / ODoH** (Oblivious DNS over HTTPS): separates the query from the requester so no single party knows both who is asking and what they are asking for. Currently only offered by Cloudflare, which introduces its own trust tradeoff. ODoH does not protect against a global passive adversary (GPA) who can observe traffic between the client resolver and the recursive resolver, between the recursive and ODNS resolvers, or between the ODNS resolver and an authoritative server.
- **DoHoT** (DNS over HTTPS over Tor): routes encrypted DNS queries through Tor for stronger anonymity. Requires Linux and more technical configuration. See https://github.com/alecmuffett/dohot [Archive.org].



Browser choice

For daily non-sensitive use: only Firefox-based browsers support ECH, and ECH only helps for Cloudflare-hosted sites. If you prefer a Chromium-based browser, Brave is the most privacy-preserving option - it supports all Chrome extensions while adding meaningful protections that Chrome lacks.

IP addresses remain a residual leak

Regardless of DNS encryption, the IP address your device connects to is visible to network observers. Since most websites have unique, publicly known IP addresses, an adversary can match your connection destinations against a database of known site IPs and infer which sites you are visiting - without ever touching your DNS traffic. OCSP stapling, ECH, and encrypted DNS do not prevent this.

This is why the guide ultimately recommends Tor, combined with a virtualized multi-layered architecture (see Virtualization), as the only realistic mitigation: VPN over Tor hides both DNS queries and IP destinations. Alternative configurations (Tor over VPN, VPN only, no Tor/VPN) are covered later but are less recommended for high-sensitivity use.

Your RFID enabled devices

RFID stands for Radio-frequency identification, it is 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. As with everything else, such capabilities can be used for tracking by various actors.

But unfortunately, this is not limited to 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 cannot 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 [Wikiless] [Archive.org] and https://en.wikipedia.org/wiki/Radio-frequency_identification#Privacy [Wikiless] [Archive.org]

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. You should just not carry such RFID devices while conducting sensitive activities.

See Warning about smartphones and smart devices

The Wi-Fi and Bluetooth devices around you

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

This active and continuous probing can then be sent back to Google/Apple/Microsoft as part of their Telemetry. The issue is that this probing is unique and can be used to uniquely identify a user and track such user. Shops, for example, can use this technique to fingerprint customers including when they return, where they go in the shop and how long they stay at a particular place. There are several papers' and articles describing this issue in depth.

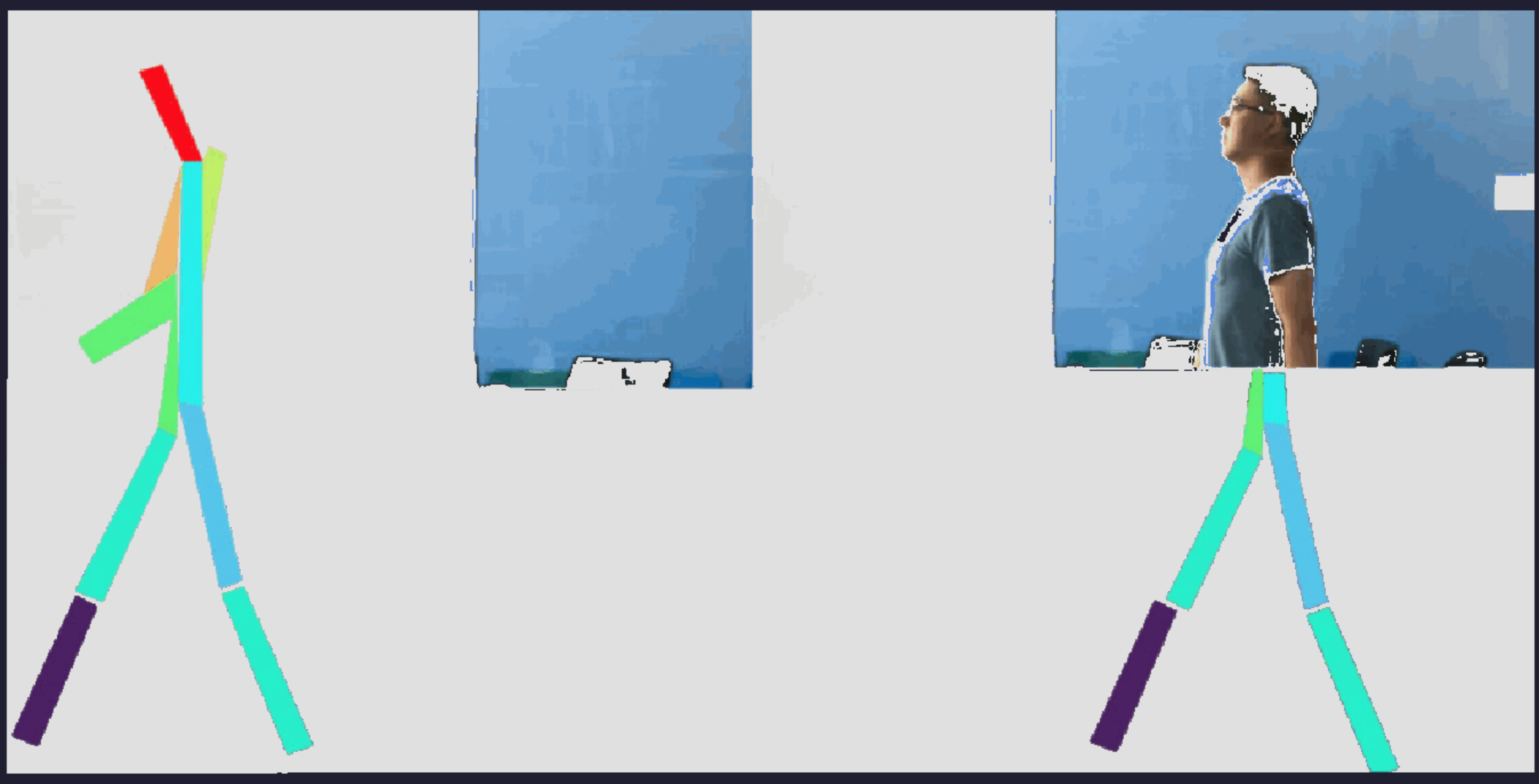
This allows them to provide accurate locations even when GPS is off, but it also allows them to keep a convenient record of all Wi-Fi Bluetooth devices all over the world. Which can then be accessed by them or third parties for tracking.

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

But that is not what all those Wi-Fi 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 is 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 with demonstrations showing this tech in action: http://rfpose.csail.mit.edu/ [Archive.org] and the video here: https://www.youtube.com/watch?v=HgDdaMy8KNE [Invidious]

Other researchers have found a way to count the people in a defined space using only Wi-Fi, see https://www.news.ucsb.edu/2021/020392/dont-fidget-wifi-will-count-you [Archive.org]

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



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

Here is another video of the same kind of tech in action: https://www.youtube.com/watch?v=FDZ39h-kCS8 [Invidious]

See: Warning about smartphones and smart devices.

There is not much you can do about these. Besides being non-identifiable in the first place.

Rogue Wi-Fi Access Points

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

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 [Invidious]

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) while spoofing the normal Wi-Fi networks at the same location. They will continue to perform this attack until your computer, or you decide to try to connect to the rogue AP.

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

Once you are 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 simple traffic sniffing. These can then easily identify any client that would for instance try to connect to a VPN server or the Tor Network.

This can be useful when you know someone you want to de-anonymize is in a crowded place, but you do not 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 techniques can also be employed to design sophisticated phishing websites aimed at capturing your credentials or persuading you to install a malicious certificate. Such a certificate could enable attackers to intercept and decrypt your encrypted traffic.

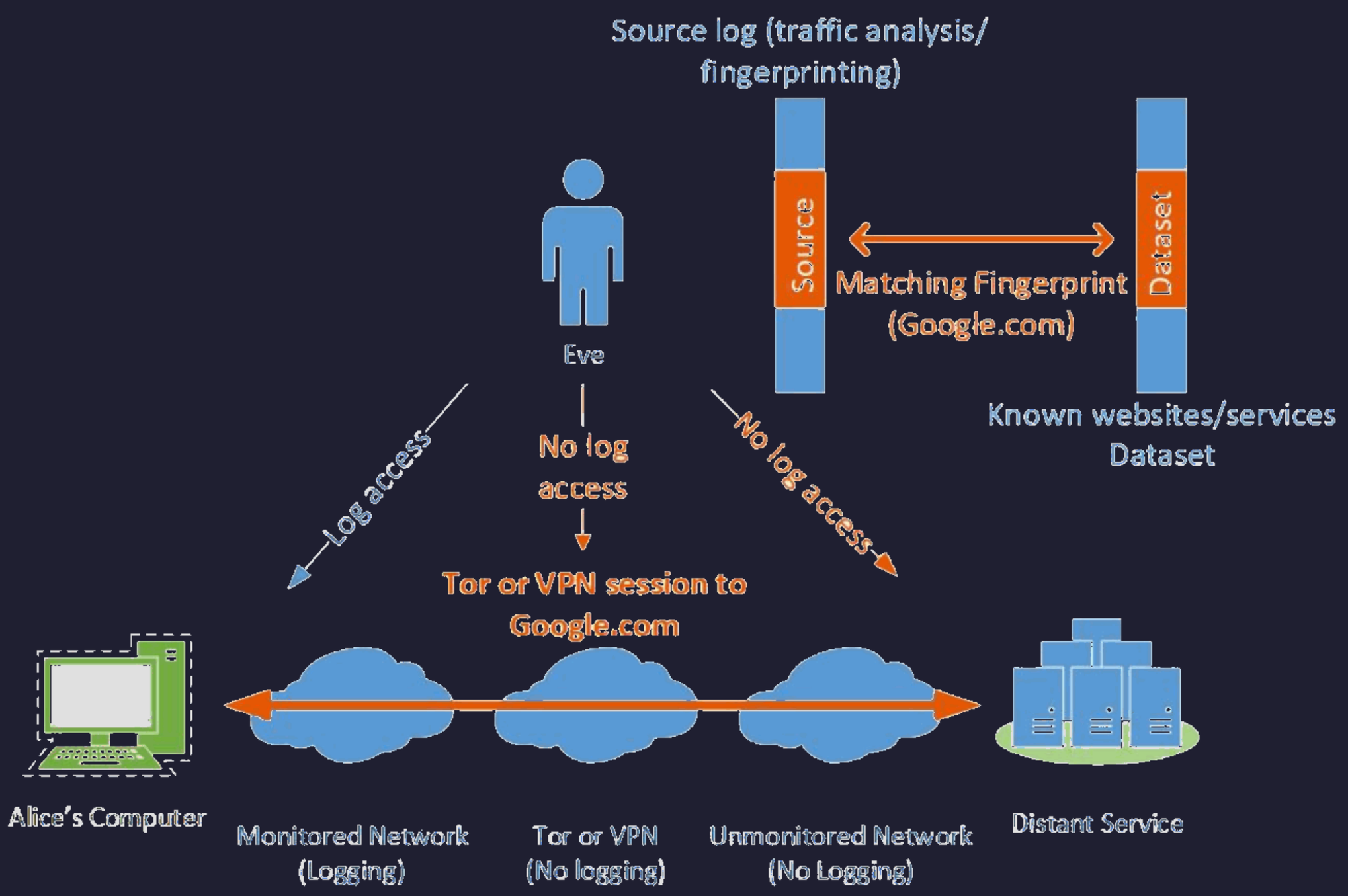
How to mitigate those? If you do connect to a public wi-fi access point, use Tor, or use a VPN and then Tor or even VPN over Tor to obfuscate your traffic from the rogue AP while still using it.

In addition, you should see the BlackHat USA conference talk, Surveilling the Masses with Wi-Fi Positioning Systems [Invidious]. The talk details a critical vulnerability in the Wi-Fi positioning API by Apple, which can be used to geofence the population using unique identifiers. See: Warning about smartphones and smart devices. Your neighbors' iPhones are a unique threat, too.

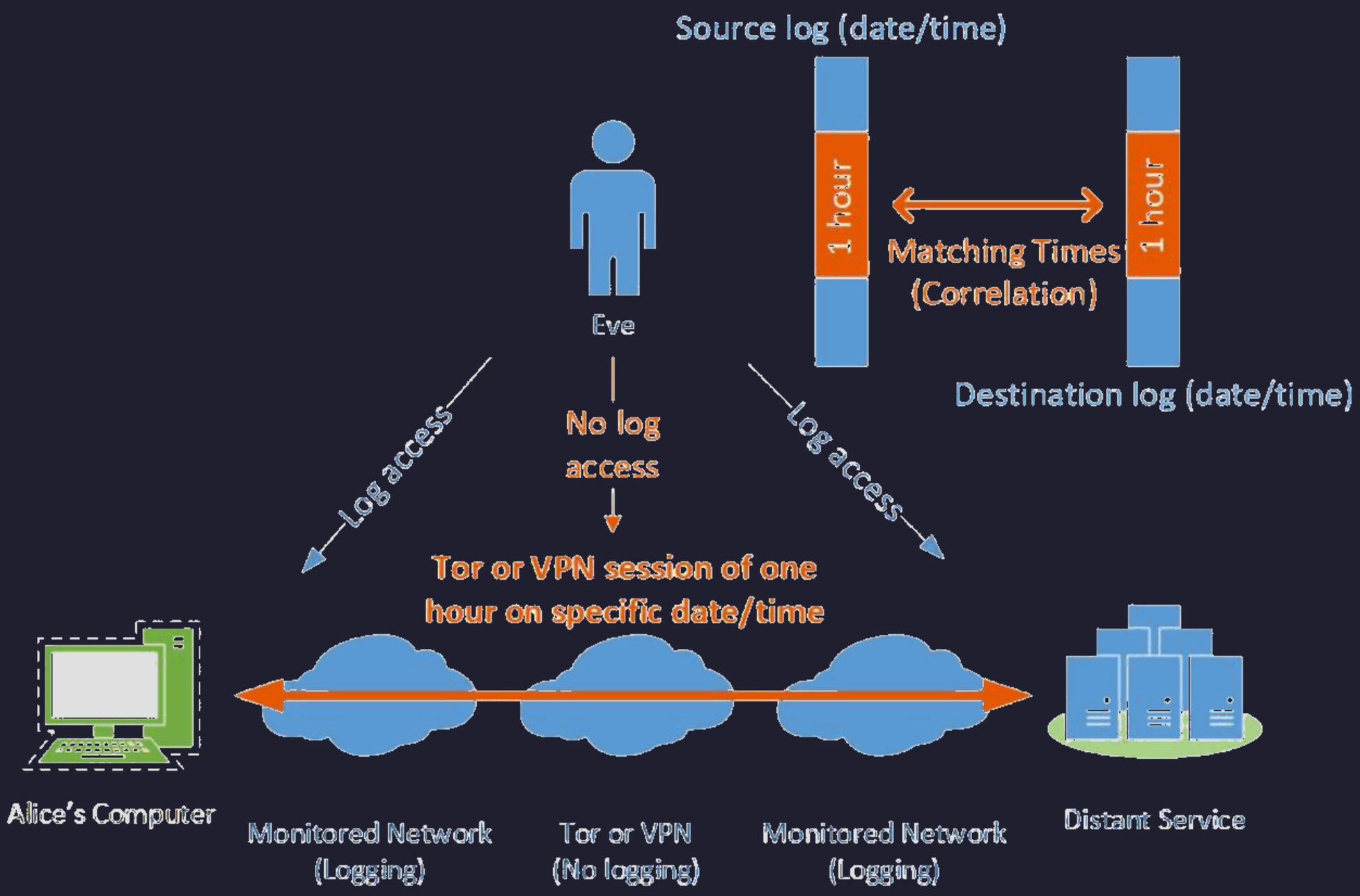
Traffic Anonymization

Tor and VPNs are not silver bullets. Many advanced techniques have been developed and studied to de-anonymize encrypted Tor traffic over the years. 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 examples:

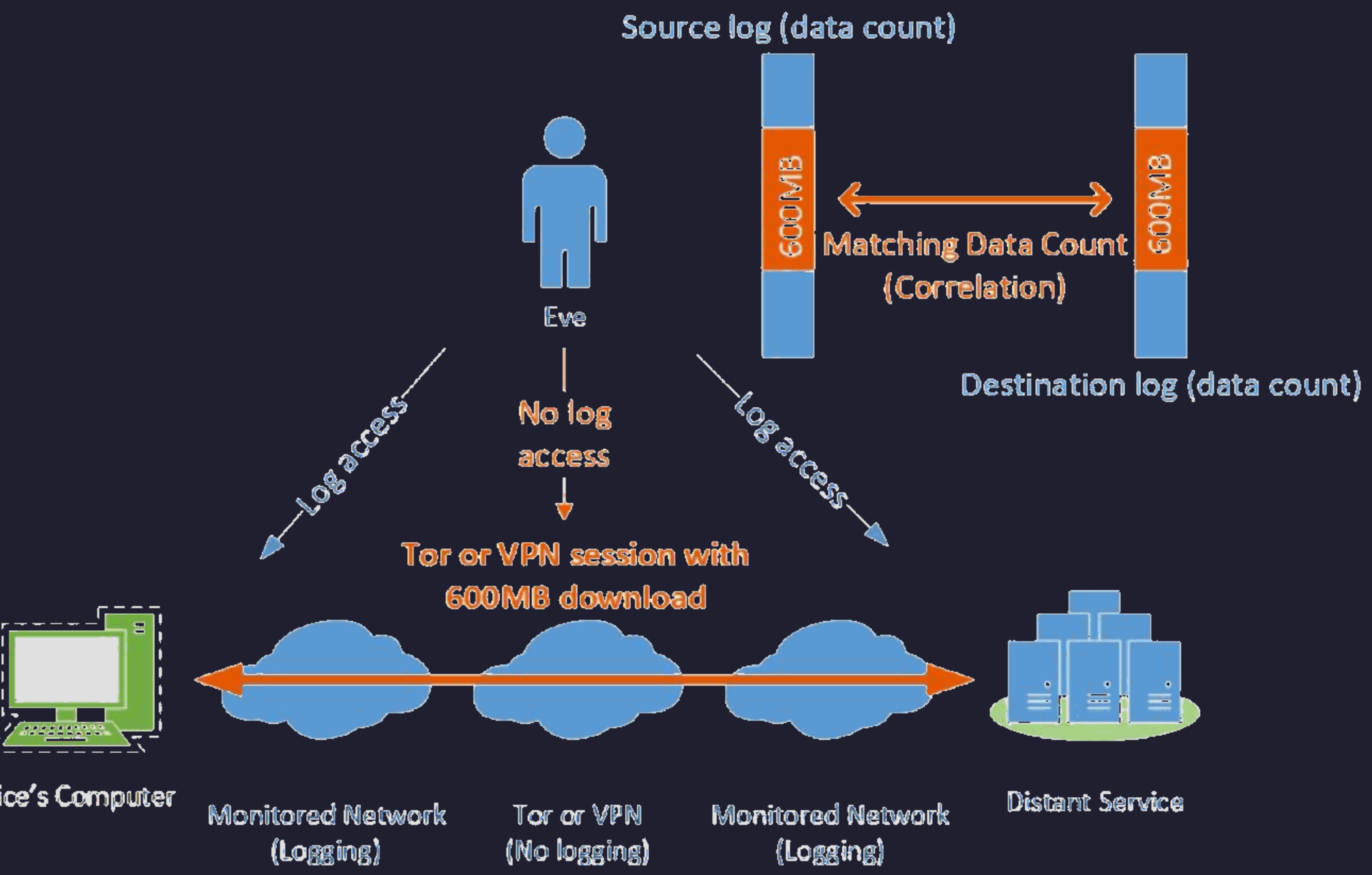
- **Correlation Fingerprinting Attack:** As illustrated (simplified) below, this attack will fingerprint your encrypted Tor traffic (like the websites you visited) based on the analysis of your encrypted traffic without decrypting it. Some of those methods can do so with a 96% success rate **in a closed-world setting. The efficacy of those methods in a real open-world setting has not been demonstrated yet and would probably require tremendous resources computing power making it very unlikely that such techniques would be used by a local adversary in the near future.** Such techniques could however hypothetically be used by an advanced and probably global adversary with access to your source network to determine some of your activity. Examples of those attacks are described in several research papers'' as well as their limitations. The Tor Project itself published an article about these attacks with some mitigations: https://blog.torproject.org/new-low-cost-traffic-analysis-attacks-mitigations [Archive.org].



- **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 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 a bomb threat hoax at Harvard University.



- **Correlation Counting Attacks:** As illustrated (simplified) below, an adversary that has no access to detailed connection logs (cannot 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 heavily 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.
- Consider the use of 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 will not 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 with wide access to global mass surveillance. Such an adversary might have access to logs no matter where you are and could use those to de-anonymize you. Usually, these attacks are part of what is called a Sybil Attack. **These adversaries are out of the scope of this guide.**

Be also aware that all the other methods described in this guide such as Behavioral analysis can also be used to deanonymize Tor users indirectly (also see Your Digital Footprint.

I also strongly recommend reading this very good, complete, and thorough (and more detailed) guide on most known Attack Vectors on Tor: https://github.com/Attacks-on-Tor/Attacks-on-Tor [Archive.org] 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 [Archive.org]

As well as this great series of blog posts: https://www.hackerfactor.com/blog/index.php?/archives/906-Tor-0day-The-Management-Vulnerability.html [Archive.org]

Recently, one of these attacks was attempted on the Tor Network with more information here: https://arstechnica.com/information-technology/2014/07/active-attack-on-tor-network-tried-to-decloak-users-for-five-months/ [Archive.org]

Lastly, do remember that using Tor can already be considered suspicious activity, and its use could be considered malicious by some.

This guide will later propose some mitigations to such attacks by changing your origin from the start (using public wi-fi's for instance). Remember that such attacks are usually carried by highly skilled, highly resourceful, and motivated adversaries and are out of scope from this guide. It is also recommended that you learn about practical correlation attacks, as performed by intelligence agencies: https://officercia.mirror.xyz/WeAilwJ9V4GIVUkYa7WwBwV2II9dYwpdPTp3fNsPFjo [Archive.org]

Disclaimer: it should also be noted that Tor is not designed to protect against a global adversary.

Traffic analysis and the limits of Tor

Note: This section expands on the Traffic Anonymization above. What follows is a more detailed treatment of the specific attack classes that matter in practice.

Tor provides strong anonymity against most adversaries most of the time. It is not, however, unconditional. Understanding what that means in practice, and who realistically is such an adversary, is more useful than either dismissing the concern or being paralyzed by it.

Timing correlation attacks

The foundational attack against anonymity networks is traffic correlation: if an adversary can observe the traffic entering the Tor network from your computer and the traffic exiting toward a destination, they can correlate the two streams by timing, volume, and packet patterns - without ever breaking Tor's encryption.

Murdoch and Danezis demonstrated in 2005 that a relatively low-resource adversary controlling even a small number of Tor nodes could use timing analysis to identify which node a hidden service was using, dramatically narrowing the anonymity set. This was an early result and the Tor network has evolved significantly since, but the underlying principle - that correlation across observation points does not require decrypting anything - has only been confirmed by subsequent research.

RAPTOR (2015) showed that Autonomous System (AS) level adversaries - large ISPs and internet exchanges, not just intelligence agencies - could perform traffic analysis by observing BGP routing and inferring path overlap between a Tor user and their destination. The key insight is that the same AS may carry both the user's traffic to the guard node and the exit node's traffic to the destination, making correlation possible without any Tor node compromise.

DeepCorr (2018) used deep learning to correlate Tor flows with significantly higher accuracy than prior methods, achieving correlation rates above 96% in controlled conditions. The authors are careful to note that their evaluation was performed in a closed-world lab setting - a fixed set of websites, controlled conditions - and that real-world performance against a large open network with diverse traffic would be substantially harder. This distinction matters: closed-world accuracy figures are frequently misquoted as if they apply to real-world deployments. They do not, at least not yet.

Who is a global passive adversary in practice?

A true global passive adversary - one who can observe arbitrary internet traffic worldwide simultaneously - does not exist in the form often imagined. What does exist is a collection of national intelligence agencies with broad but not unlimited visibility into internet traffic (GCHQ's TEMPORA, NSA's PRISM and upstream collection programmes), large ISPs and internet exchanges that carry a disproportionate share of global traffic, and cloud providers whose infrastructure spans most of the world's AS paths.

For the vast majority of Tor users, none of these entities are targeting them specifically. For a journalist communicating with a source inside a country whose intelligence services have close partnerships with major Western agencies, or an activist whose traffic transits only a small number of AS paths, the picture is more concerning. The honest answer is: **if a Five Eyes agency is specifically targeting you, Tor alone is probably not sufficient. For everyone else, Tor provides strong protection.**

Website fingerprinting

Website fingerprinting attacks attempt to identify which website a Tor user is visiting by analysing the pattern of encrypted traffic - packet sizes, timing, direction sequences - without decrypting it. Accuracy in closed-world evaluations (where the attacker knows the user is visiting one of N monitored sites) has reached high levels in research settings. In open-world conditions, where the user may be visiting any of millions of sites, false positive rates make these attacks far less practical. WTF-PAD and related padding defences, partially deployed in Tor Browser, further degrade fingerprinting accuracy. This is an active research area and the situation will evolve.

Guard node persistence and what it means

Tor uses **guard nodes** - a small, stable set of entry nodes that your client reuses over weeks - specifically to limit timing correlation exposure. If you used a random entry node for every circuit, an adversary who controls even a modest fraction of Tor nodes would eventually observe you entering the network directly. By persisting a small guard set, Tor limits the probability that any given adversary controls your entry point. The tradeoff is that if your guard node is malicious or observed, it remains so for the duration of the guard period. On balance, the Tor Project's research shows guard persistence improves anonymity for most people most of the time.

When Tor is and is not sufficient

Tor is sufficient against: local network observers (your ISP, your university, a café Wi-Fi), most law enforcement agencies without intelligence partnerships, commercial data brokers, and advertisers.

Tor is not sufficient against: a targeted operation by a well-resourced national intelligence agency with upstream internet visibility, an adversary who controls both your guard node and the destination's exit node simultaneously, or an adversary who can correlate your Tor usage timing with known real-world events (you were the only person in a particular location at a particular time).

The most practical mitigation beyond Tor itself is changing your entry point: connecting to Tor from public Wi-Fi rather than your home connection removes the most reliable correlation anchor - your ISP-assigned IP - from the equation entirely. This guide recommends this approach for high-sensitivity activities throughout.

Some Devices can be tracked even when offline

You have seen this in action/spy/Sci-Fi movies and shows, the protagonists always remove the battery of their phones to make sure it cannot be used. Most people would think that's overkill. Well, unfortunately, no, this is now becoming true at least for some devices:

- iPhones and iPads (IOS 13 and above)'
- Samsung Phones (Android 10 and above)
- MacBooks (macOS 10.15 and above)

Such devices will continue to broadcast identity information to nearby devices even when offline using Bluetooth Low-Energy. They do not have access to the devices directly (which are not connected to the internet) but instead use BLE to find them through other nearby devices. They are using peer-to-peer short-range Bluetooth communication to broadcast their status through nearby online devices.

They could now find such devices and keep the location in some database that could then be used by third parties or themselves for various purposes (including analytics, advertising, or evidence/intelligence gathering).

See Warning about smartphones and smart devices

TLDR: Do not take such devices with you when conducting sensitive activities.

Your Hardware Identifiers

Windows USB bus telemetry (device enumeration history)

Windows keeps a permanent enumeration history of every USB device ever connected, independent of whether the device is currently attached.

DEANONYMIZATION VECTOR

Each USB mass storage device exposes a Vendor ID, Product ID, and, for most drives, a unique serial number at the descriptor level during enumeration. Windows records this under:

- `HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR`
- `HKLM\SYSTEM\CurrentControlSet\Enum\USB`
- `SetupAPI.dev.log`

This creates a durable, cross-reinstall link between a specific physical USB drive, phone, hardware key, or dongle, and every OS install it has ever touched, because the identifying data lives in the device's own firmware/descriptor, not in the OS.

How it's exploited:

- Physical device correlation. The same USB drive used across a "clean" VM and a personal machine carries its VID/PID/serial into both installs' registries, creating a link forensic analysis, or malware with sufficient privilege, can recover even if the OS itself was reinstalled or the VM rebuilt.
- First-connected timestamps. Registry key metadata records first- and last-connected times per device, which can corroborate or contradict claimed usage timelines.
- Persists across OS reinstall of the host, not the device. Reinstalling Windows resets the host's own identifiers, including the GDID, but does nothing to the USB device's own serial. Replugging it into a new install just writes a fresh, correlatable entry referencing the same hardware ID.

Mitigations:

- Never share USB storage media, security keys, or peripherals between anonymous and identified environments.
- Where sharing is unavoidable, prefer devices that don't expose a unique serial in their descriptor, and treat any device that has ever touched an identified machine as burned for anonymous use.
- Use write-blockers or a live OS (Tails) for cases where a USB device's provenance itself is sensitive, since USB enumeration history is written by the host OS regardless of read/write intent.

Your IMEI and IMSI

The IMEI and the IMSI are unique numbers created by cell phone manufacturers and cell phone operators.

The IMEI is tied directly to the phone you are using. This number is known and tracked by the cell 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 with the IMSI (if a SIM card is inserted but that is not even needed). It is also used by many applications (Banking apps abusing the phone permission on Android for instance) and smartphone Operating Systems (Android/IOS) for identification of the device. It is possible but difficult (and not illegal in many jurisdictions) to change the IMEI on a phone but it is 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 some random small shop.

The IMSI is tied directly to the mobile subscription or pre-paid plan you are using and is 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 with the IMEI. Like the IMEI, the IMSI is also being used by some applications and smartphone Operating systems for identification and is 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 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 in at least six ways:

- The mobile operator subscriber logs will usually store the IMEI along with the IMSI and their subscriber information database. If you use a prepaid anonymous SIM (anonymous IMSI but with a known IMEI), they could 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 will conveniently keep a log of which IMEI. 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 were, 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. This makes 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/is used by various third parties to geolocate/track you quite precisely'.

- 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 to whom it was sold. And if you are 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 do not find you on CCTV and you bought the phone using 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.
- Government agencies around the world interested in your phone number can and do use special devices called "IMSI catchers" like the Stingray or more recently the Nyxcell. These devices can impersonate (to spoof) a cell phone Antenna and force a specific IMSI to connect to it to access the cell network. Once they do, they will be able to use various MITM 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 [Invidious]

For these reasons, it is crucial to get a dedicated anonymous phone number and/or an anonymous burner phone with a cash-bought pre-paid sim card that is not tied to you in any way (past or present) for conducting sensitive activities. It is also possible to get an anonymous pre-paid but preferably dedicated number from free and paid online services accepting anonymous cryptocurrencies like Monero. Get more practical guidance here: Getting an anonymous Phone number.

While there are some smartphones manufacturers like Purism with their Librem series who claim to have your privacy in mind, they still do not allow IMEI randomization which we believe is a key anti-tracking feature that should be provided by such manufacturers. While this measure will not 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 Warning about smartphones and smart devices

Your Wi-Fi or Ethernet MAC address

The MAC address is a unique identifier tied to your physical Network Interface (Wired Ethernet or Wi-Fi) and could of course be used to track you if it is 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 is 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/Wi-Fi access point keeps logs of devices that are registered on the Wi-Fi, and these can be accessed too to find out who has been using your Wi-Fi. Sometimes this can be done remotely (and silently) by the ISP depending on 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).

Some commercial devices will keep a record of MAC addresses roaming around for various purposes such as road congestion.

So, it is important again not to bring your phone along when/where you conduct sensitive activities. If you use your own laptop, then it is 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 possibility to randomize MAC addresses (Android, IOS, Linux, and Windows 10/11) with the notable exception of macOS which does not support this feature even in its latest Big Sur version.

See Warning about smartphones and smart devices

Your Bluetooth MAC address

Your Bluetooth MAC is like the earlier MAC address except it is 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.

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

In general, this should not be too much of a concern compared to MAC Addresses. BT Addresses are randomized quite often.

See Warning about smartphones and smart devices

Your CPU

All modern CPUs are now integrating hidden management platforms such as the now infamous Intel Management Engine (IME) and the AMD Platform Security Processor (PSP).

Those management platforms are 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 video: BlackHat, How to Hack a Turned-Off Computer, or Running Unsigned Code in IME https://www.youtube.com/watch?v=9fhNokIgBMU [Invidious].

These have already been affected by several security vulnerabilities in the past 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.

There are some not so straightforward ways to disable the 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.

Note that, to AMD's defense, there were no security vulnerabilities found for ASP and no backdoors either. See https://www.youtube.com/watch?v=bKHSnGLgi08&t=2834s [Invidious]. In addition, AMD PSP does not provide any remote management capabilities contrary to IME.

If you are feeling a bit more adventurous, you could install your own BIOS using Coreboot or Libreboot (a distribution of Coreboot) if your laptop supports it. Coreboot allows you to add your own microcode or other firmware blobs in order for the machine to function, but this is based upon user choice, and as of Dec 2022, Libreboot has adopted a similar pragmatic approach in order to support newer devices in the Coreboot tree. (Thanks, kind Anon who corrected previous information in this paragraph.)

Check yourself:

- If you are using Linux you can check the vulnerability status of your CPU to Spectre/Meltdown attacks by using https://github.com/speed47/spectre-meltdown-checker [Archive.org] which is available as a package for most Linux distros including Whonix. Spectre is a transient execution attack. There is also PoC code for Spectre v1 and v2 on iPhone devices here: https://github.com/cispa/BranchDifferent [Archive.org] and here https://misc0110.net/files/applespectre_dimva22.pdf [Archive.org]
- If you are using Windows, you can check the vulnerability status of your CPU using inSpectre https://www.grc.com/inspectre.htm [Archive.org]

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 [Wikiless] [Archive.org]

Some of these can be avoided using Virtualization Software settings that can mitigate such exploits. See this guide for more information Spectre/Meltdown Hardening [Archive.org] (warning: these can severely impact the performance of your VMs).

This guide won't go too deep into side-channel and microarchitecture attacks but we will highlight some issues with both Intel and AMD CPU architectures that will be mitigated throughout. It's important to recognize hardware is just as susceptible to bugs, and therefore exploitation, regardless of manufacturer.

We will 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.

In addition, we recommend the use of AMD CPUs instead of Intel CPUs. See Types of CPU attacks for more information.

- CPU vulnerabilities found in the past few years:

- Meltdown
- Spectre
- Æpic
- SGAxe
- LVI
- Plundervolt
- MicroScope replay attack
- Enclave
- Prime+Probe
- Crosstalk
- Hertzbleed
- Squip attack
- Zenbleed

Your OS and App telemetry services

Whether it is Android, iOS, Windows, macOS, or even Ubuntu. Most popular Operating Systems now collect telemetry information by default even if you never opt-in or opted-out from the start. Some like Windows will not even allow disabling telemetry completely without some technical tweaks. This information collection can be extensive and include a staggering number of details (metadata and data) on your devices and their usage.

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

Android/Google:

- Just have a read at their privacy policy https://policies.google.com/privacy [Archive.org]
- School of Computer Science & Statistics, Trinity College Dublin, Ireland Mobile Handset Privacy: Measuring The Data iOS and Android Send to Apple And Google https://www.scss.tcd.ie/doug.leith/apple_google.pdf [Archive.org]

IOS/Apple:

- More information at https://www.apple.com/legal/privacy/en-ww/ [Archive.org] and https://support.apple.com/en-us/HT202100 [Archive.org]
- School of Computer Science & Statistics, Trinity College Dublin, Ireland Mobile Handset Privacy: Measuring The Data iOS and Android Send to Apple And Google https://www.scss.tcd.ie/doug.leith/apple_google.pdf [Archive.org]
- Apple does claim that they anonymize this data using differential privacy but you will 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 [Archive.org]
- Full list of optional diagnostic data: https://docs.microsoft.com/en-us/windows/privacy/windows-diagnostic-data [Archive.org]

macOS:

- More details on https://support.apple.com/guide/mac-help/share-analytics-information-mac-apple-mh27990/mac [Archive.org]

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 [Archive.org]

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 is important to understand that this telemetry data can be tied to your device and help de-anonymizing you and later can be used against you by an adversary that would get access to this data.

This does not mean for example that Apple devices are terrible choices for good Privacy (tho this might be changing), but they are certainly not the best choices for (relative) Anonymity. They might protect you from third parties knowing what you are doing but not from themselves. In all likelihood, they certainly know who you are.

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. These will include Windows, macOS, and even Linux in some regard.

See Warning about smartphones and smart devices

Windows COM Class Identifiers (CLSID) and GUIDs

Windows assigns a 128-bit GUID to every registered COM class (CLSID), interface (IID), and application (AppID). These are enumerable from user space via the registry (`HKEY_CLASSES_ROOT\CLSID\{GUID}`), no admin rights required, and in some cases can be probed from web content.

Deanonymization vector: The specific set of CLSIDs registered on a machine reflects the exact installed software: codecs, shell extensions, Office components, VPN client integrations, security tools, browser helper objects. This is as unique a fingerprint as font or plugin enumeration, but it persists across browser profiles and sessions since it lives at the OS level, not in browser storage.

How it's exploited:

- Software inventory fingerprinting. Enumerating registered CLSIDs reveals installed applications with far more granularity than a User-Agent string.
- Presence probing. A site can test for specific CLSIDs or associated custom URI protocol handlers to detect named applications: VPN clients, communication apps, forensic or analysis tools.
- VM/sandbox detection. Certain CLSIDs and related registry artifacts are diagnostic of VirtualBox, VMware, or known anonymity-focused environments.
- Cross-context correlation. Reusing one Windows install across anonymous and identified activity links the two, independent of browser-level compartmentalization.

Mitigations:

- Use a dedicated, minimal-install VM per identity or persona. Do not reuse one Windows install across anonymous and identified contexts.
- Prefer Linux-based isolation (Whonix, Tails, Qubes-Whonix) for high-threat-model use. This attack class is COM/Windows-specific and has no direct equivalent there.
- Revert to a clean snapshot between sessions rather than relying on manual cleanup.
- Treat this as the same fingerprinting category as font and plugin enumeration: the fix is compartmentalization, not registry editing.

Windows Global Device Identifier (GDID)

The GDID is a persistent, server-assigned 64-bit device identifier tied to a Windows installation. It is almost entirely undocumented by Microsoft: the only public reference is a single line in the Azure Monitor schema for the `UCDOStatus` (Delivery Optimization) table, describing a `GlobalDeviceId` column as "Microsoft global device identifier. This is an identifier used by Microsoft internally." Independent reverse-engineering, confirmed against Windows 11 build 26200, has since mapped the full chain.

🧭 SOURCES

- *United States v. Peter Stokes*, N.D. Ill. (2026). Federal criminal complaint; primary source for the GDID's real-world use in a law enforcement investigation. Highly encourage all readers of this document to consider that this obscure device telemetry vector was not documented prior to this court case, and is mentioned in passing.
- "Windows GDID Fully Reverse Engineered: MSA Device PUID Exposed", Devdigest.
- "You can't fully disable Microsoft's GDID Windows 11 tracker, but these settings limit what it captures", Windows Latest.
- gdid-reversal, independent reverse-engineering write-up with ETW and registry evidence.

⚠️ ON THE GDID-REVERSAL SOURCE

We have not vetted and will not vet this source, so exercise caution. This repository is user-created and its authorship is unverified. Content may be partly or wholly AI-generated. Treat specific technical claims (registry paths, binary symbols, ETW provider details) as unverified until cross-checked against a real Windows install or a more established source. It's included here because it's one of the only detailed technical accounts of an otherwise undocumented mechanism, not as an authoritative reference.

HOW IT WORKS

When a device signs in with a Microsoft Account, `wlidsvc` provisions it against `login.live.com` and gets back a Device PUID (Passport Unique ID): a server-assigned value, not derived from any local hardware serial. This is stored in the registry, read by the Connected Devices Platform (`cdp.dll` / `CDPSvc`), registered into Microsoft's Device Directory Service graph, and reported by Delivery Optimization as `UCDOStatus.GlobalDeviceId` whenever the machine participates in peer-to-peer update sharing.

Read your own GDID without admin rights:

```
(Get-ItemProperty 'HKCU:\SOFTWARE\Microsoft\IdentityCRL\ExtendedProperties').LID
```

This is what the federal complaint used to place a Scattered Spider suspect at specific times and locations across multiple VPN sessions and four countries. Microsoft records matched a specific GDID to the moment an ngrok account was created, regardless of the VPN in use.

Key properties for the threat model:

- VPN/IP-agnostic. The GDID doesn't change with IP address, VPN provider, or exit node. It correlates sessions on the same OS install, not network paths.
- Not disabled by telemetry opt-outs. It doesn't travel through the classic `DiagTrack` pipeline. Disabling standard diagnostic data settings has no effect on it.
- Server-side persistence. Deleting the local registry value is cosmetic. The same identifier re-syncs from Microsoft's servers on the next authenticated request (opening the Microsoft Store, for example), because the canonical copy lives server-side, not on disk.
- Local accounts don't prevent it. A local, non-MSA install still registers an anonymous device path through the Connected Devices Platform. It just isn't tied to an MSA identity unless one is added later.
- A reinstall gets a new GDID, not a clean slate. A fresh install produces a new identifier, but account, OneDrive, and activation records give Microsoft a documented basis for linking the new GDID back to the old one through the same account.

Mitigations:

- Assume any Windows install signed into a Microsoft Account is linkable across all its sessions regardless of VPN discipline. This is a hard requirement for compartmentalization, not a tuning option.
- For genuinely sensitive activity, don't use Windows at all. Use a Linux-based environment (Whonix, Tails, Qubes-Whonix) where this telemetry chain doesn't exist.
- Never reuse a single Windows install, MSA, or even a "cleaned" reinstall on the same physical machine across separate personas.

Reducing exposure, if Windows must be used:

- Use a local account. CDP still creates an anonymous device path, but the GDID won't be tied to an MSA.
- Block `dds.microsoft.com` and `activity.windows.com` via firewall rules or the hosts file, to cut the reporting channel at the network layer.
- Delete the registry keys under `IdentityCRL` to clear the locally stored PUID. This only clears the local cache. It's recreated identical to before on the next MSA sign-in or even a background service touching a Microsoft app, since the canonical value lives server-side.
- Disable the Delivery Optimization service (`DoSvc`). This service reports the GDID via `UCDOStatus.GlobalDeviceId` and is hardened against normal administrative control: `Stop-Service` and `Set-Service` return access-denied even when run elevated. The only reported workaround is disabling it directly in the registry by setting its `Start` value to `4` under its service key, since the Service Manager's block doesn't extend to direct registry writes. Disabling `DoSvc` alongside `CDPSvc` / `CDPUserSvc` breaks Delivery Optimization peer caching, Phone Link / "Continue on PC," and nearby sharing. These are expected tradeoffs, not bugs.

WHAT THIS DOES NOT DO

None of the above retracts a GDID already reported to Microsoft, and none of it makes the device anonymous. It only stops the local install from continuing to re-register and re-report going forward. The historical link between the device and any prior identified activity already exists on Microsoft's servers and can't be removed by a local user.

FURTHER READING / TOOLING

Independent write-ups and a defensive PowerShell toolkit implementing the steps above (read-only audit, preview mode, revert option) are publicly available and worth reviewing directly rather than treating this section as a complete implementation guide. The service-disabling steps carry real breakage tradeoffs and should be tested on a VM snapshot first.

Your Smart Devices

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.
- Always records other devices around you (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.

Data is being transmitted even if you opt-out, processed, and stored indefinitely (most likely unencrypted) by various third parties.

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

- Your Smart Watch? (Apple Watch, Android Smartwatch ...)
- Your Fitness Devices and Apps'? (Strava', Fitbit, Garmin, Polar, ...)
- Your Smart Speaker? (Amazon Alexa, Google Echo, Apple Homepod ...)
- Your Smart Transportation? (Car? Scooter?)
- Your Smart Tags? (Apple AirTag, Galaxy SmartTag, Tile...)
- Your Car? (Yes, most modern cars have advanced logging/tracking features these days)
- Any other Smart device? There are even convenient search engines dedicated to finding them online:
    - https://www.shodan.io/
    - https://censys.io/
    - https://www.zoomeye.org/

See Warning about smartphones and smart devices

Conclusion: Do not bring your smart devices with you when conducting sensitive activities.

Yourself

Your Metadata

What's metadata? Every file you create or share carries metadata - structured data embedded in or alongside the content that describes how, when, where, and with what the file was created. This metadata is invisible in normal use and routinely overlooked. It has burned journalistic sources, identified whistleblowers, and linked anonymous documents to their authors. The tools to strip it exist and are not difficult to use. The failure is almost always one of not knowing it was there.

The most frequently cited case is the 2013 identification of a leaker at a US government contractor through metadata in a Word document sent to The Intercept. The document's print metadata included a serial number traceable to a specific printer, combined with microdot tracking patterns in the printout itself - but the principle applies equally to digital metadata. Earlier, in 2003, a UK government dossier on Iraqi weapons capabilities was found to contain revision history showing the names of the civil servants who had edited it, causing significant political embarrassment and demonstrating that the problem predates widespread awareness.

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

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

This location data has been used in many judicial cases already as part of "geofencing warrants" that allow law enforcement to ask companies (such as Google/Apple) a list of all devices present at a certain location at a certain time. In addition, this location data is even sold by private companies to the military who can then use it conveniently. These warrants are becoming widely used by law enforcement''.

If you want to experience yourself what a "geofencing warrant" would look like, here is an example: https://wigle.net/.

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

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

Have you heard of Edward Snowden? Now is the time to google him and read his book. Also read about XKEYSCORE', MUSCULAR, SORM, Tempora , and PRISM.

See "We kill people based on Metadata" or this famous tweet from the IDF https://twitter.com/idf/status/1125066395010699264 [Archive.org] [Nitter].

See Smartphones for a warning on using smartphones and other smart devices. See Metadata auditing for a way to get rid of the metadata - which is probably what brought you to this section anyway.

Your Digital Footprint

This is the part where you should watch the documentary "The Social Dilemma" on Netflix as they cover this topic much better than anyone else.

This includes is the way you write (stylometry) ', the way you behave'. The way you click. The way you browse. The fonts you use on your browser. 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 comparably on some Reddit post 5 years ago using a not so anonymous Reddit account. The words you type in a search engine alone can be used against you as the authorities now have warrants to find people who used specific keywords in search engines.

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 do not send it / save it. Think of when you draft an e-mail in Gmail. It is saved automatically as you type. They can register your clicks and cursor movements as well.

All they need to achieve this in most cases is Javascript enabled in your browser (which is the case in most Browsers including Tor Browser by default). Even with Javascript disabled, there are still ways to fingerprint you.

While these methods are usually used for marketing purposes and advertising, they can also be a useful tool for fingerprinting users. This is because your behavior is unique or unique enough that over time, you could be de-anonymized.

Here are some examples:

- Specialized companies are selling to, for example, law enforcement agencies products for analyzing social network activities such as https://mediasonar.com/ [Archive.org]
- For example, as a basis of authentication, a user's typing speed, keystroke depressions, patterns of error (say accidentally hitting an "l" instead of a "k" on three out of every seven transactions) and mouse movements establish that person's unique pattern of behavior. Some commercial services such as TypingDNA (https://www.typingdna.com/ [Archive.org]) even offer such analysis as a replacement for two-factor authentications.
- This technology is also widely used in CAPTCHAS services to verify that you are "human" and can be used to fingerprint a user.
- See Counteracting Forensic Linguistics.

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

Here is a fun example you try yourself to see some of those things in action: https://clickclickclick.click (no archive links for this one sorry). You will see it becoming interesting over time (this requires Javascript enabled).

Here is also a recent example just showing what Google Chrome collects on you: https://web.archive.org/web/https://pbs.twimg.com/../media/EwiUNH0UYAgLY7V?format=jpg&name=4096x4096

Here are some other resources on the topic if you cannot see this documentary:

- 2017, Behavior Analysis in Social Networks, https://link.springer.com/10.1007/978-1-4614-7163-9_110198-1 [Archive.org]
- 2017, Social Networks and Positive and Negative Affect https://www.sciencedirect.com/science/article/pii/S1877042811013747/pdf?md5=253d8f1bb615d5dee195d353dc077d468&pid=1-s2.0-S1877042811013747-main.pdf [Archive.today]
- 2015, Using Social Networks Data for Behavior and Sentiment Analysis https://www.researchgate.net/publication/300562034_Using_Social_Networks_Data_for_Behavior_and_Sentiment_Analysis [Archive.org]
- 2016, A Survey on User Behavior Analysis in Social Networks https://www.academia.edu/30936118/A_Survey_on_User_Behaviour_Analysis_in_Social_Networks [Archive.org]
- 2017, DEF CON 25 presentation: DEF CON 25 - Svea Eckert, Andreas Dewes - Dark Data [Invidious]
- 2019, Influence and Behavior Analysis in Social Networks and Social Media https://sci-hub.se/10.1007/978-3-030-02592-2 [Archive.org]

So, how can you mitigate these?

- This guide will provide some technical mitigations using Fingerprinting resistant tools but those might not be sufficient.
- You should apply common sense and try to find your own patterns in your behavior and behave differently when using anonymous identities. This includes:
  - The way you type (speed, accuracy...).
  - The words you use (be careful with your usual expressions).
  - The type of response you use (if you are sarcastic by default, try to have a different approach with your identities).
  - The way you use your mouse and click (try to solve the Captchas differently than your usual way)
  - The habits you have when using some Apps or visiting some Websites (do not always use the same menus/buttons/links to reach your content).
  - ...

You need to act and fully adopt a role as an actor would do for a performance. You need to become a different person, think, and act like that person. This is not a technical mitigation but a human one. You can only rely on yourself for that.

Ultimately, it is mostly up to you to fool those algorithms by adopting new habits and not revealing real information when using your anonymous identities. See Counteracting Forensic Linguistics.

IRL 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 who shared over time several details about his past and was later discovered.

There are also a few cases involving OSINT at Bellingcat. Have a look at their very informative (but slightly outdated) toolkit here: https://docs.google.com/spreadsheets/d/18rtqh8EG2q1xBo2cLNyhIDuK9jrPGwYr9DI2UncoqJQ/edit#gid=930747607 [Archive.org]

We have an OSINT discussion room in our Matrix community. Feel free to join at `#OSINT:matrix.org`.

You can also view some convenient lists of some available OSINT tools here if you want to try them on yourself for example:

- https://github.com/jivoi/awesome-osint [Archive.org]
- https://web.archive.org/web/20210426041234/https://jakecreps.com/tag/osint-tools/
- https://osintframework.com/
- https://recontool.org

As well as this interesting Playlist on YouTube: https://www.youtube.com/playlist?list=PLrFPX1Vfqk3ehZKSFeb9pVIHqxqtNW8Sy [Invidious]

As well as those interesting podcasts:

https://www.inteltechniques.com/podcast.html

You should never share real individual experiences/details using your anonymous identities that could later lead to finding your real identity. You will see more details about this in the Creating new identities section.

Your Face, Voice, Biometrics, and Pictures

"Hell is other people", even if you evade every method listed above, you are 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' and have been using other means (GEOINT) to create maps of "people" around the world. This evolution has been going on for years to the point we can now say "we lost control of our faces".

If you are walking in a touristy place, you will most likely appear in someone's selfie within minutes without knowing it. That person could then go ahead and 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 does not provide a timestamp and geolocation, it can still be guessed with other means'.

Here are a few resources for even trying this yourself:

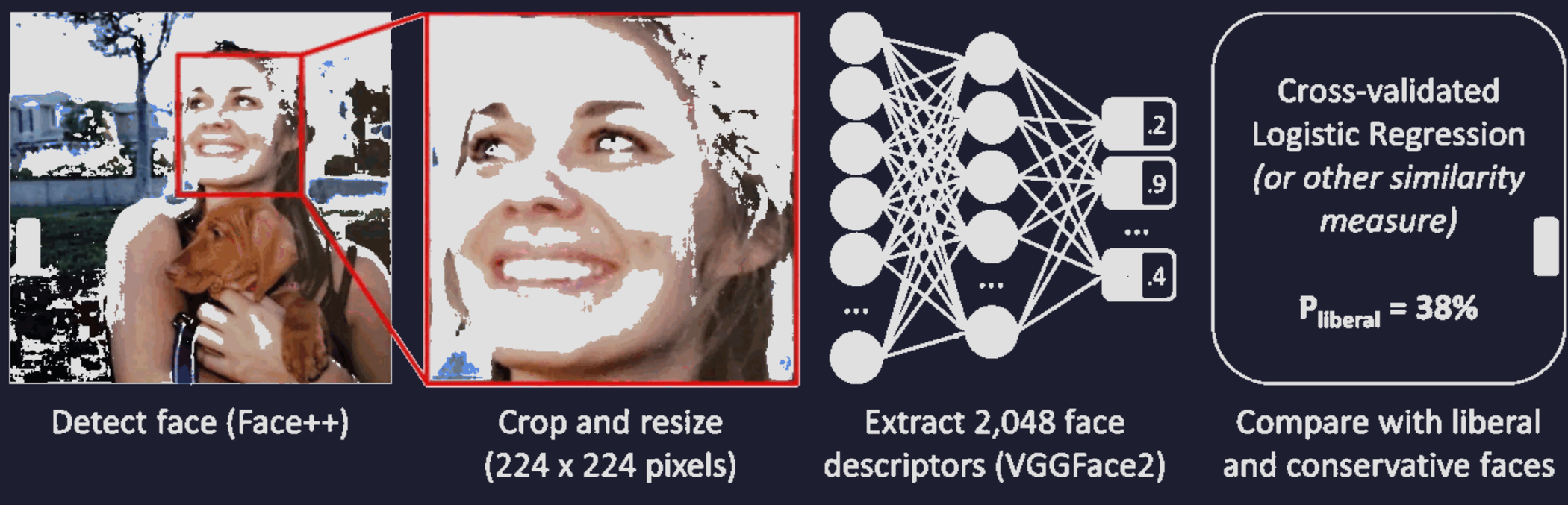
- Bellingcat, Guide To Using Reverse Image Search For Investigations: https://www.bellingcat.com/resources/how-tos/2019/12/26/guide-to-using-reverse-image-search-for-investigations/ [Archive.org]
- Bellingcat, Using the New Russian Facial Recognition Site SearchFace https://www.bellingcat.com/resources/how-tos/2019/02/19/using-the-new-russian-facial-recognition-site-searchface-ru/ [Archive.org]
- 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/ [Archive.org]
- Bellingcat, Advanced Guide on Verifying Video Content https://www.bellingcat.com/resources/how-tos/2017/06/30/advanced-guide-verifying-video-content/ [Archive.org]
- 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/ [Archive.org]
- 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/ [Archive.org]
- Bellingcat, Berlin Assassination: New Evidence on Suspected FSB Hitman Passed to German Investigators https://www.bellingcat.com/news/2021/03/19/berlin-assassination-new-evidence-on-suspected-fsb-hitman-passed-to-german-investigators/ [Archive.org]
- Bellingcat, Digital Research Tutorial: Investigating a Saudi-Led Coalition Bombing of a Yemen Hospital https://www.youtube.com/watch?v=cAVZaPiVArA [Invidious]
- Bellingcat, Digital Research Tutorial: Using Facial Recognition in Investigations https://www.youtube.com/watch?v=awY87q2MrQE [Invidious]
- Bellingcat, Digital Research Tutorial: Geolocating (Allegedly) Corrupt Venezuelan Officials in Europe https://www.youtube.com/watch?v=bS6gYWM4kzY [Invidious]

Gait Recognition and Other Long-Range Biometrics

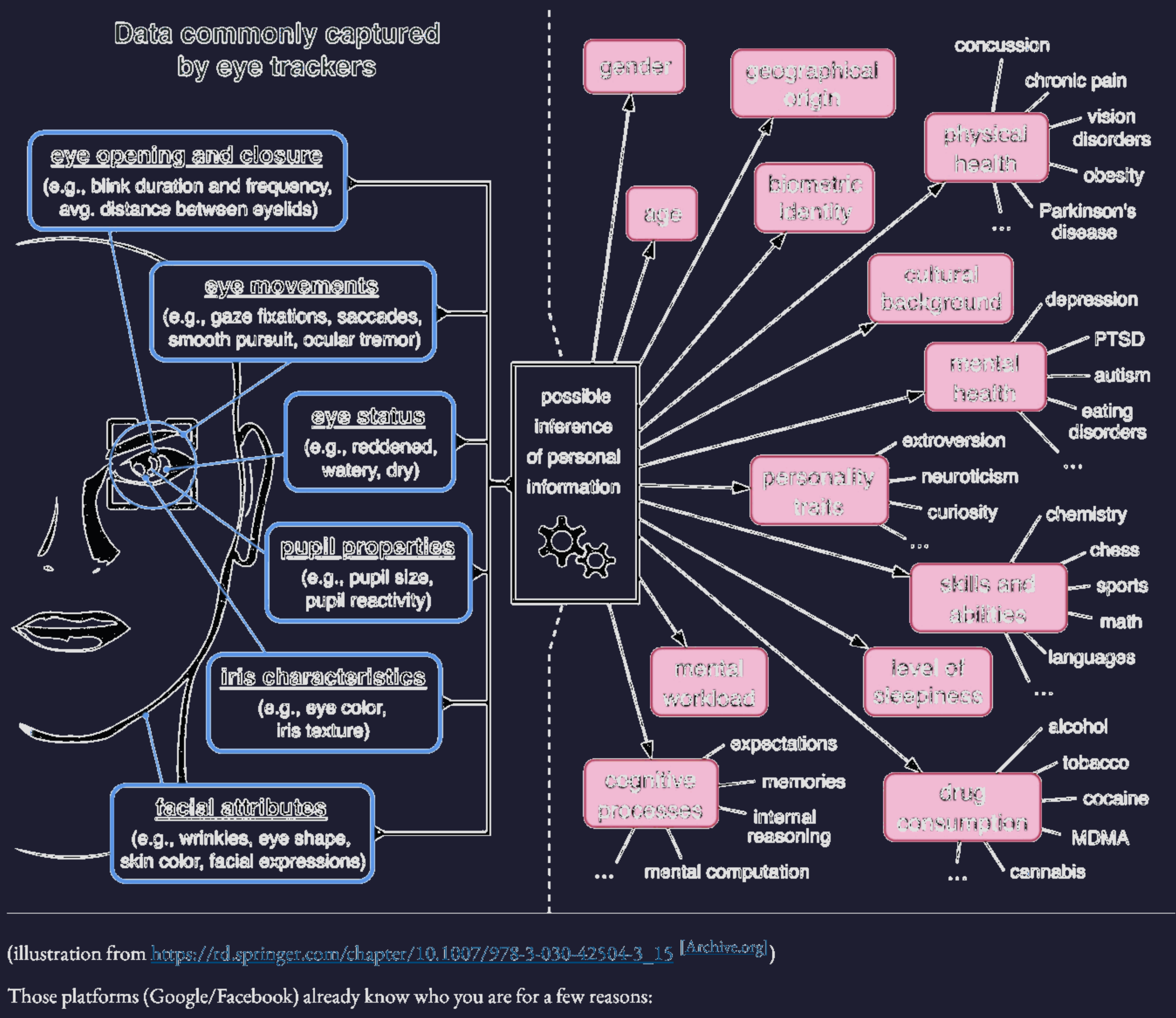
Even if you are not looking at the camera, they can still figure out who you are, make out your emotions, analyze your gait'', read your lips, analyze the behavior of your eyes, and probably guess your political affiliation'.

Contrary to popular belief and pop culture, modern gait recognition systems aren't fooled by simply changing how you walk (ex. with something uncomfortable in your shoe), as they analyze the way your body's muscles move across your entire body, as you perform certain actions. The best way to fool modern gait recognition is to wear loose clothes that obscure the way your muscles move as you perform actions.

Other things than can be used to identify you include your earlobes, which are actually more identifiable than fingerprints, or even the shape of your skull. As such, soft headcoverings such as balaclavas are not recommendable for obscuring your identity - they make you look incredibly suspicious, while also conforming to the shape of your skull.



(Illustration from https://www.nature.com/articles/s41598-020-79310-1 [Archive.org])



(illustration from https://rd.springer.com/chapter/10.1007/978-3-030-42504-3_15 [Archive.org])

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''''.
- 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.

Here is also an insightful demo of Microsoft Azure you can try for yourself at https://azure.microsoft.com/en-us/services/cognitive-services/face/#demo where you can detect emotions and compare faces from different pictures.

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 Oosto, Clearview AI', or NEC) in their CCTV networks to look for "persons of interest". And some heavily surveilled states like China have implemented widespread use of Facial Recognition for various purposes' including possibly identifying ethnic minorities. A simple face recognition error by some algorithm can ruin your life'.

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 [Archive.org]
- EFF SLS: https://www.eff.org/sls [Archive.org]

Apple is making FaceID mainstream and pushing its use to log you into many services including the Banking systems.

The 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'''.

The same goes with your voice which can be analyzed for various purposes as shown in the recent Spotify patent.

Even your iris can be used for identification in some places.

We can safely imagine a near future where you will not be able to create accounts or sign in anywhere without providing unique biometrics (A suitable time to re-watch Gattaca, Person of Interest , and Minority Report). 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 by crafting false information (Pictures, Videos, Voice Recordings...) and have already been used for such purposes'. There are even commercial services for this readily available such as https://www.respeecher.com/ [Archive.org] and https://www.descript.com/overdub [Archive.org].

See this demo: https://www.youtube.com/watch?v=t5yw5cR79YA [Invidious]

At this time, there are a few steps 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 but not all.
- 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 eye's features.
- Consider wearing special sunglasses (expensive, unfortunately) called "Reflectacles" https://www.reflectacles.com/ [Archive.org]. There was a small study showing their efficiency against IBM and Amazon facial recognition.
- All that might still be useless because of gait recognition mentioned earlier but there might be hope here if you have a 3D Printer: https://gitlab.com/FG-01/fg-01 [Archive.org]

(see [Gait Recognition and Other Long-Range Biometrics])

(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 is a social engineering type of attack where an adversary could try to extract information from you by pretending or impersonating something/someone else.

A typical case is an adversary using a man-in-the-middle attack or a fake e-mail/call to ask for your credential for a service. This could for example be through e-mail or through impersonating financial services.

Such attacks can also be used to de-anonymize someone by tricking them into downloading malware or revealing personal information over time. The only defense against those is not to fall for them and common sense.

These have been used countless times since the early days of the internet and the usual one is called the "419 scam" (see https://en.wikipedia.org/wiki/Advance-fee_scam [Wikiless] [Archive.org]).

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 [Invidious].

Malware, exploits, and viruses

Malware in your files/documents/e-mails

Using steganography or other techniques, it is easy to embed malware into common file formats such as Office Documents, Pictures, Videos, PDF documents...

These can be as simple as HTML tracking links or complex targeted malware.

These could be simple pixel-sized images hidden in your e-mails that would call a remote server to try and get your IP address.

These could be exploiting a vulnerability in an outdated format or an 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 [Invidious]
- Ange Albertini: Funky File Formats: https://www.youtube.com/watch?v=hdCs6bPM4is [Invidious]

You should always use extreme caution. See Virtualization to mitigate leaking any information even in case of opening such a malicious file.

If you want to learn how to try detecting such malware, see Checking files for malware

Malware and Exploits in your apps and services

So, you are using Tor Browser or Brave Browser over Tor. You could be using those over a VPN for added security. But you should keep in mind that there are exploits (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 case in 2013 where the FBI inserted malware 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 hack that breached several US government institutions by inserting malware into an official software update server.

In some countries, Malware is just mandatory and/or distributed by the state itself. This is the case for instance in China with WeChat which can then be used in combination with other data for state surveillance.

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 are using.
- You should always check that you are 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 (See Virtualization so that even if your Browser/Apps get compromised by a skilled adversary, that adversary will find himself stuck in a sandbox without being able to access identifying information or compromise your system.

Malicious USB devices

There are readily available commercial and cheap "badUSB" 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 [Archive.org]
- Hak5, O.MG Cable https://www.youtube.com/watch?v=VSmBJHotZv0 [Invidious]
- Keelog https://www.keelog.com/ [Archive.org]
- AliExpress https://www.aliexpress.com/i/4000710369016.html [Archive.org]

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 in 2005.

While you could inspect a USB key physically, scan it with various utilities, check the various components to see if they are 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.

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 a 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).

Malware and backdoors in your Hardware Firmware and Operating System

This might sound a bit familiar as this was already partially covered previously in the Your CPU section.

Malware and backdoors can be embedded directly into your hardware components. Sometimes those backdoors are implemented by the manufacturer itself such as the IME in the case of Intel CPUs. And in other cases, such backdoors can be implemented by a third party that places itself between orders of new hardware and customer delivery.

Such malware and backdoors can also be deployed by an adversary using software exploits. Many of those are called rootkits within the tech world. Usually, these types of malware are harder to detect and mitigate as they are implemented at a lower level than the userspace and often in the firmware of hardware components itself.

What is firmware? Firmware is a low-level operating system for devices. Each component in your computer probably has firmware including for instance your disk drives. The BIOS/UEFI system of your machine for instance is a type of firmware.

These can allow remote management and are capable of enabling full control of a target system silently and stealthily.

As mentioned previously, these are harder to detect by users but some limited steps that can be taken to mitigate some of those by protecting your device from tampering and use some measures (like re-flashing the bios for example). Unfortunately, if such malware or backdoor is implemented by the manufacturer itself, it becomes extremely difficult to detect and disable those.

Note: The threats described in this section are almost exclusively relevant to high-value targets of nation-state adversaries. If your threat model is a stalker, a corporate competitor, or even most law enforcement agencies, you can skip this section. If you are a journalist, dissident, or activist operating against a state-level adversary, read it.

Most guides to anonymity focus on software and network-layer threats. Physical and hardware-level attacks are rarer, more expensive to execute, and require either physical access to your device or interference with your supply chain. That cost means they are not deployed casually. But for the right target, they are devastatingly effective - because no amount of software configuration protects you if the hardware underneath is compromised.

Firmware implants

Firmware implants are malicious code inserted into the low-level software that runs before your operating system boots - in the UEFI/BIOS, storage controller firmware, or network card firmware. Because they live below the OS, they survive reinstallation of the operating system, disk wiping, and most forensic examination.

LoJax, discovered by ESET in 2018, was the first publicly documented in-the-wild UEFI rootkit, attributed to the APT28 (Fancy Bear) group. It wrote a malicious module directly into the SPI flash memory of the UEFI firmware, persisting across OS reinstalls and even hard drive replacements. **MosaicRegressor**, documented by Kaspersky in 2020, was similarly implanted into UEFI and discovered on devices belonging to NGO staff and journalists in contact with North Korea.

Who faces this threat? In both documented cases, targets were NGO workers, journalists, and diplomatic personnel - people whose devices passed through the hands of state actors, or who were targeted by sophisticated spear-phishing that enabled remote firmware write access. This is not a threat that scales to mass deployment. It is used surgically, against specific high-value individuals.

Mitigations are limited but worth understanding. **UEFI Secure Boot** verifies the cryptographic signatures of bootloader and OS components before execution, preventing unsigned code from running at boot. It does not, however, protect against a compromise of the firmware itself - if the UEFI has already been modified, Secure Boot can be disabled or bypassed from within. It is a meaningful defence against attackers who have not yet achieved firmware-level access, but it is not a root of trust in the presence of a firmware implant. **Intel Boot Guard** and AMD's equivalent go further by fusing a hash of the initial firmware into the hardware at manufacture time, making firmware modification detectable. **Heads** is an open-source firmware alternative for supported hardware (primarily Thinkpads and select System76 machines) that provides measured boot, TPM-backed attestation, and tamper detection - and is the most practical option for a high-risk user who needs verifiable firmware integrity. See also: About Secure Boot.

USB attack hardware

USB-based attack tools are commercially available and widely understood. The **O.MG Cable** is a USB cable with an embedded wireless implant - visually and functionally indistinguishable from a legitimate charging cable - that can execute keystrokes, exfiltrate data, and accept remote commands over Wi-Fi. The **USB Rubber Ducky** and broader Hak5 product family present themselves to a target computer as a keyboard, executing pre-loaded keystroke injection payloads at speeds no human typist could match. See also: Malicious USB devices.

Recognition is difficult. O.MG cables are designed specifically to defeat visual inspection. Practical mitigations include: **never using cables or USB devices you did not purchase yourself and receive sealed**, using a USB data blocker ("USB condom") when charging from untrusted ports, and configuring your operating system to require confirmation before trusting new USB devices (USBGuard on Linux; this is not natively available on Windows without third-party tools).

Evil Maid attacks

For more on Evil Maid attacks, see: Evil Maid attack.

Mitigations:

- **Never leave your device unattended in a high-risk environment.** This is the only complete mitigation.
- **Measured boot with TPM attestation** (as provided by Heads or a correctly configured UEFI + TPM setup) will detect bootloader tampering by comparing measurements against known-good values stored in the TPM.
- **A tamper-evident seal** on the device chassis (nail varnish applied across screws and photographed, or commercial tamper-evident stickers) provides a low-tech detection layer that is surprisingly effective against unsophisticated adversaries.

Supply chain compromise

Supply chain attacks target your device before it reaches you - at the manufacturer, distributor, or shipping stage. The NSA's ANT catalogue, leaked by Snowden in 2013, documented hardware implants installed in Cisco routers and other network equipment in transit. For most users, this threat is not realistic. For a senior dissident, human rights lawyer, or intelligence source in a country whose government has influence over hardware supply chains, it deserves consideration.

Practical mitigations are limited. Purchasing devices in person from a retail store (rather than having them shipped) reduces the interception window. Preferring hardware from vendors outside adversary supply chain reach, and using Heads-supported hardware with verified firmware, provides some assurance. For the highest-risk cases, consider that any device that has left your control - even briefly - should be treated as potentially compromised.

Physical inspection checklist

For high-risk individuals receiving or returning to a device:

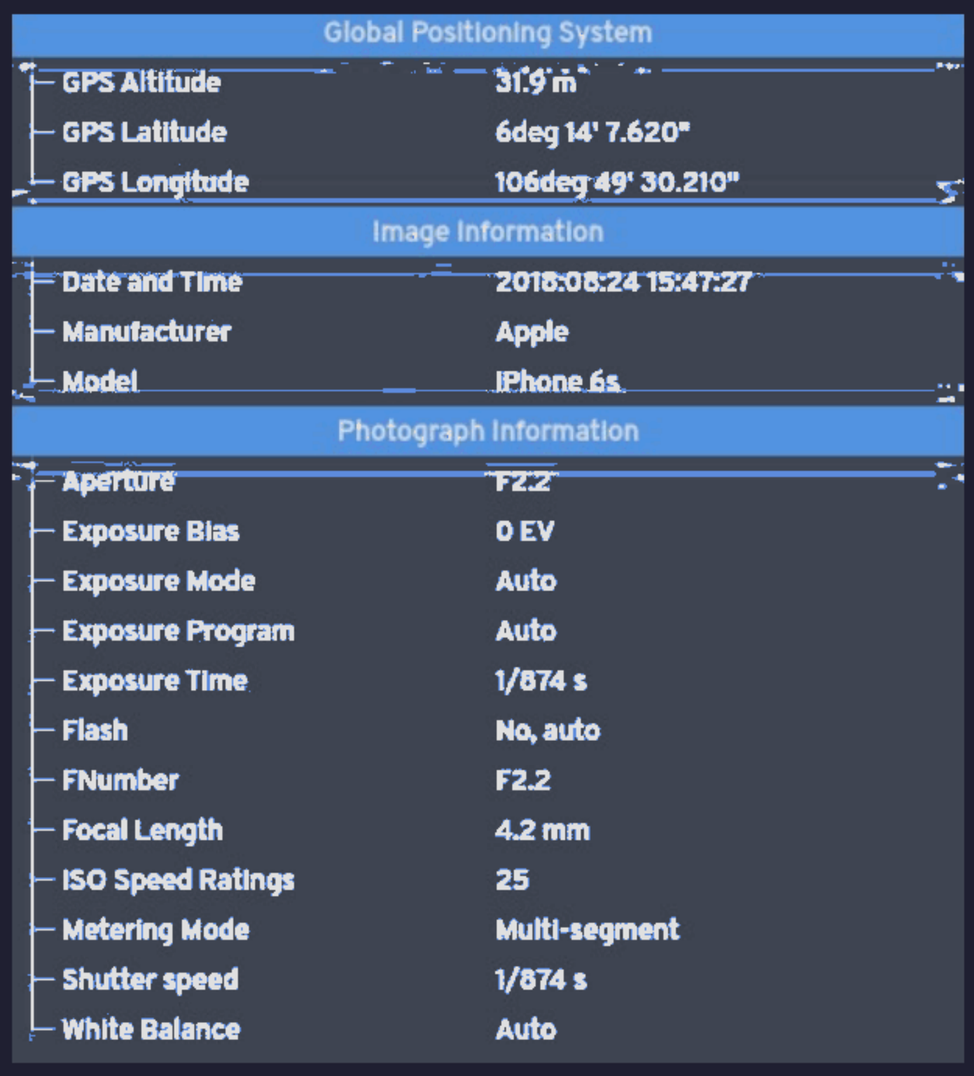
- Inspect port openings (USB, Thunderbolt, SD card slot) for signs of foreign objects or residue.
- Check screws for scratches inconsistent with factory assembly; apply a tamper-evident seal after inspection.
- Compare the cable you are about to use against a known-good reference; if in doubt, discard it.
- On first boot after any period of unattended access, verify firmware measurements if your platform supports it (Heads TPM event log; `tpm2-tools` on Linux).
- If Secure Boot is unexpectedly disabled in UEFI settings, treat the device as compromised.

Your files, documents, pictures, and videos

Properties and Metadata

This can be obvious to many but not to all. Most files have metadata attached to them. Good examples are pictures that store EXIF information which can hold 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 various sources to find you (CCTV or other footage taken at the same place at the same time during a protest for instance). You must verify any file you would put on those platforms for any properties that might hold any information that might lead back to you.

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



(Illustration from Wikipedia)

This also works for videos. Yes, videos too have geo-tagging, and many are very unaware of this. Here Is for instance a very convenient tool to geo-locate YouTube videos: https://mattw.io/youtube-geofind/location [Archive.org]

For this reason, you will always have to be incredibly careful when uploading files using your anonymous identities and check the metadata of those files.

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

Watermarking

Pictures, Video and Audio

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

So, if you are 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 or Audio) or with extensions 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/ [Tor Mirror] [Archive.org]

Such watermarks can be inserted by various products''' using Steganography and can resist compression and re-encoding'.

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 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. There is no guaranteed 100% protection against those. You will have to use common sense.

Printer Watermarking

Did you know your printer is most likely spying on you too? Even if it is 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 [Archive.org]

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

Many printers will print an invisible watermark allowing for identification of the printer on every printed page. This is called Printer Steganography. There is no tangible way to mitigate this but to inform yourself on your printer and make sure it does not print any invisible watermark. This is 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 [Archive.org]

Here are also some tips from the Whonix documentation (Printing and Scanning) [Archive.org]:

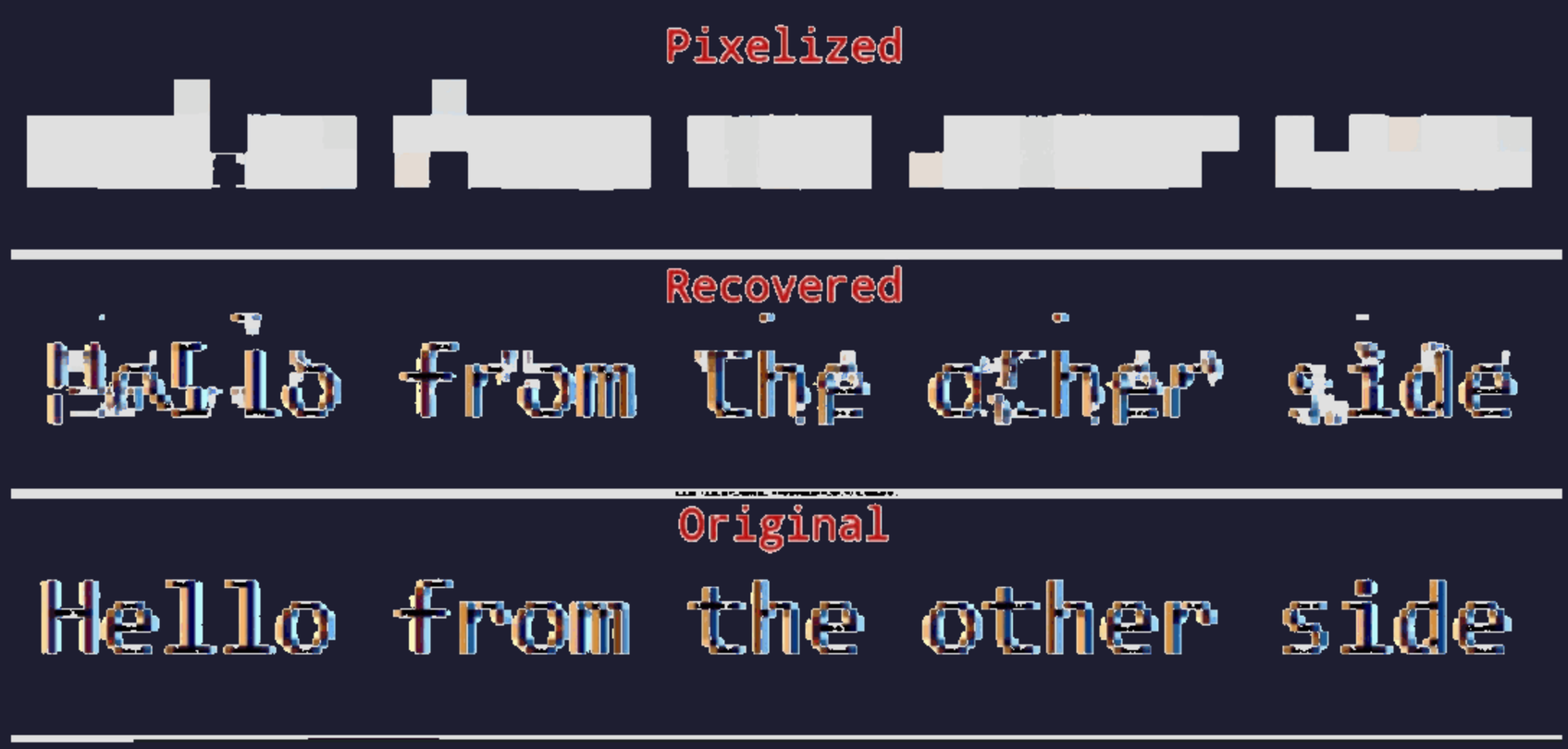
Do not ever print in Color, usually, watermarks are not present without color toners/cartridges.

Pixelized or 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 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 [Archive.org]



This is of course an open-source project available for all to use. But you can 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 [Archive.org] followed by https://medium.com/@somdevsangwan/deblurring-images-for-osint-part-2-ba564af8eb5d [Scribe.rip] [Archive.org]

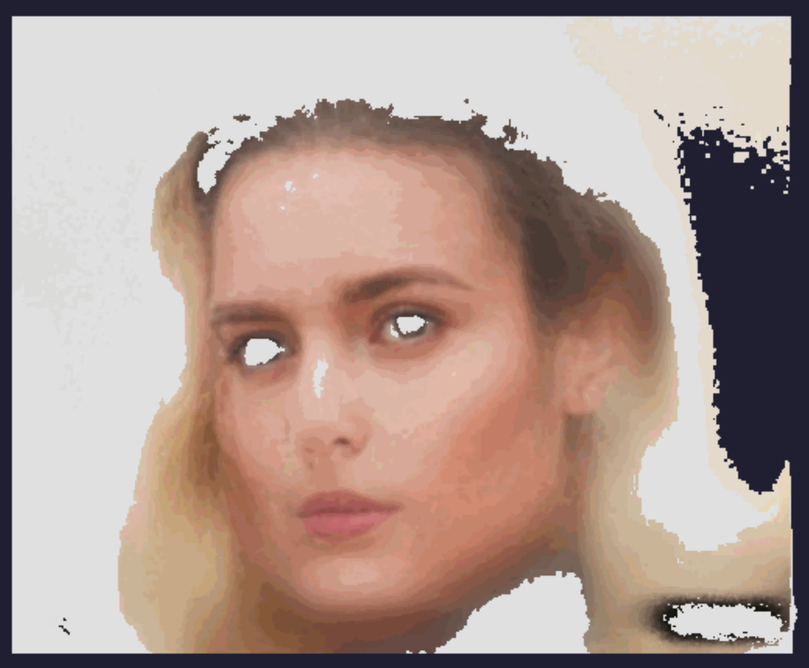


Finally, you will find plenty of deblurring resources here: https://github.com/subeeshvasu/Awesome-Deblurring [Archive.org]

Some online services could even help you do this automatically to some extent like MyHeritage.com enhance tool:

https://www.myheritage.com/photo-enhancer [Archive.org]

Here is the result of the above image:



Of course, this tool is more like "guessing" than really deblurring at this point, but it could be enough to find you using various reverse image searching services.

There are also techniques to deblur/depixelate parts in videos: see https://positive.security/blog/video-depixelation [Archive.org]

For this reason, it is 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. Do not pixelized, do not blur, just put a hard black rectangle to redact information.

Your Crypto Transactions

Contrary to widespread belief, Crypto transactions (such as Bitcoin and Ethereum) are not anonymous. Most cryptocurrencies can be tracked accurately through various methods'.

Remember what they say on their page: https://bitcoin.org/en/you-need-to-know [Archive.org] and https://bitcoin.org/en/protect-your-privacy [Archive.org]: "Bitcoin is not anonymous"

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 will 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 financial regulations) and can then trace back those crypto transactions to you using the financial system.

There are some cryptocurrencies with privacy/anonymity in mind like Monero but even those have some and warnings to consider'.

Use of "private" mixers, tumblers (centralized services that specialize in "anonymizing" cryptocurrencies by "mixing them") and coinjoiners are risky as you don't know what's happening on them and can be trivially de-mixed. Their centrally-controlled nature could also put you in trouble as they are more susceptible to money-laundering laws.

This does not mean you cannot use Bitcoin anonymously at all. You can actually use Bitcoin anonymously as long as you do not convert it to actual currency, use a Bitcoin wallet from a safe anonymous network, and do not reuse addresses or consolidate outputs that were used when spending at different merchants. Meaning you should avoid KYC/AML regulations by various exchanges, avoid using the Bitcoin network from any known IP address, and use a wallet that provides privacy-preserving tools. See Anonymous crypto payments.

Overall, the best option for using Crypto with reasonable anonymity and privacy is still Monero and you should ideally not use any other for sensitive transactions unless you are aware of the limitations and risks involved. Please do read Monero Disclaimer.

TLDR: Use Monero!

Your Cloud Backup & 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 know that most of those backups are not fully end-to-end encrypted and will hold 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, Cellebrite Cloud) that will help an adversary analyze your cloud data with ease.

Notable Examples:

- Apple iCloud: https://support.apple.com/en-us/HT202303 [Archive.org] : "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/ [Archive.org]: "**Media and messages you back up aren't protected by WhatsApp end-to-end encryption while in Google Drive**. ". Do however note that Facebook/Whatsapp have announced the rollout of encrypted backups on October 14th 2021 (https://about.fb.com/news/2021/10/end-to-end-encrypted-backups-on-whatsapp/ [Archive.org]) which should solve this issue.
- Dropbox: https://www.dropbox.com/privacy#terms [Archive.org] "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 [Archive.org]: 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".**

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 if they want to.

The only way to mitigate this is to encrypt your data on your side and then only upload it to such services **or just not use them at all.**

Microarchitectural Side-channel Deanonymization Attacks

There was an attack published that can deanonymize users if they have a known alias. For example, an attacker trying to track the activities of a journalist can use that journalist's public Twitter handle to link their anonymous identities with their public one. This breaks compartmentalization of identities and can lead to complete deanonymization, even of users who practice proper OPSEC.

The attack, published at https://leakuidatorplusteam.github.io/ [Archive.org], can be mitigated using the well-known NoScript extension and will be our preferred recommendation.

One loosely documented attack might take the following approach to fingerprinting: Alice is browsing the web using Firefox. The website she has just visited is using an invisible `iframe` that creates long strings, e.g., sentences or hashes, to produce some non-user-viewable string. These strings are setting a certain font type, Arial. Whether the browser renders this is non-essential, it only matters if the font changes. The `iframe` in this case serves no purpose but to identify whether a user has installed a certain font on their machine. If Alice is using a font that this frame has tried to render, then it is reported back to the website and to the person in control of the website.

The font renders a box with a specific height and width around itself, so that means a specific height and width of the text contained within. The `iframe` keeps doing this for each installed font to create a list of installed fonts for Alice. Because of stylistic differences between each font family, the same string and the same font size will add up to a different height and a different width than Arial. It is used as a fallback font to display text that won't display otherwise, in the case of a user not having that font on their machine and thus non-viewable from their browser.

If a font requested by an `iframe` is not available, Arial will be used to show that text to the user. Every time the font measurement (identified by the dimensions of the box produced) changed, it means the font is present on Alice's browser and her machine. By doing this for hundreds of fonts, websites can use this information to track users using their installed fonts across websites. Imagine a website then selling this “anonymized” information as a dataset to advertisement companies to serve you ads based on the websites you visit, because they know every font you have installed on your machine and can now track your identity across the internet. This attack is demonstrated here: Everything you always wanted to know about web-based device fingerprinting (but were afraid to ask) by Dr. Nick Nikiforakis, PhD in Computer Science from KU Leuven. He explains how his team of researchers identified which sites were using such techniques on Alexa's top 10,000 websites. Primarily, they found that of those, 145 were fingerprinting browsers. They were fingerprinted 100% of the time - whether they were using the Do Not Track header, a popular Privacy & Security setting in many browsers, did not matter.

Attacks such as invisible iframes and media elements can be avoided by blocking all scripts globally by using something like uBlock Origin https://chrome.google.com/webstore/detail/ublock-origin/cjpalhdlnbpafiamejdnhcphjbkeiagm or by using NoScript https://chrome.google.com/webstore/detail/noscript/doojmbjmlfjjnbmnoijecmcbfeoakpjm. This is highly encouraged, not only to those wishing to be anonymous, but also to general web users.

Tor Browser

Note: This attack is now prevented by default by an update of NoScript (11.4.8 and above) on all security levels in Tor Browser.

All others

Installing the NoScript extension will prevent the attack **by default only in private Windows** using their new "TabGuard feature". But can be enabled in the NoScript options to work on all Windows. See:

- Release tweet: https://twitter.com/ma1/status/1557751019945299969 [Archive.org]
- User explanation: https://noscript.net/usage/#crosstab-identity-leak-protection [Archive.org]
- Tor Project Forum Post: https://forum.torproject.net/t/tor-browser-can-leak-your-identity-through-side-channel-attack/4005/2 [Archive.org]
- NoScript extension for Firefox (Firefox, and other Firefox-based browsers except Tor Browser): https://addons.mozilla.org/en-US/firefox/addon/noscript/
- NoScript extension for Chromium based browsers (Brave, Chrome, Edge, and other Chromium-based browsers): https://chrome.google.com/webstore/detail/noscript/doojmbjmlfjjnbmnoijecmcbfeoakpjm?hl=en

Alternative to NoScript for all other browsers

The researches who disclosed the issue also made an extension available below. Again, **nothing is required in Tor Browser**. This path is not our preferred path but is still available if you do not want to use NoScript.

- Leakuidator+ extension for Chromium based browsers (Brave, Chrome, Edge, and other Chromium-based browsers): https://chrome.google.com/webstore/detail/leakuidator%2B/hhfpajcjkikoocmmhcimllpinjnbedll
- Leakuidator+ extension for Firefox (Firefox, and other Firefox-based browsers except Tor Browser): https://addons.mozilla.org/en-US/firefox/addon/leakuidatorplus/

Separating identities via separate browsers or even with VMs is not enough to avoid this attack. However, another solution is to make sure that when you start working with an anonymous identity, you entirely close all activities linked to other identities. The vulnerability only works if you're actively logged into a non-anonymous identity. The issue with this is that it can hinder effective workflow, as multitasking across multiple identities becomes impossible.

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 of information from your devices even if they are encrypted. 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/ [Archive.org]
- New-York Times, The Police Can Probably Break Into Your Phone https://www.nytimes.com/2020/10/21/technology/iphone-encryption-police.html [Archive.org]
- Vice, Cops Around the Country Can Now Unlock iPhones, Records Show https://www.vice.com/en/article/vbxxxd/unlock-iphone-ios11-graykey-grayshift-police [Archive.org]

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 [Archive.org]
- FTK Forensic Toolkit, https://accessdata.com/products-services/forensic-toolkit-ftk [Archive.org]
- SANS Digital Forensics and Incident Response Videos, https://www.youtube.com/c/SANSDigitalForensics/videos

And finally, 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.

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 (See Virtualization), and compartmentalization. This guide will later detail such threats and techniques to mitigate them.

Bad Cryptography

There is a frequent adage among the infosec community: "Don't roll your own crypto!".

And there are reasons''' for that:

We would not want people discouraged from studying and innovating in the crypto field because of that adage. So instead, we would recommend people to be cautious with "Roll your own crypto" because it is not necessarily good crypto:

- Good cryptography is not easy and usually takes years of research to develop and fine-tune.
- Good cryptography is transparent and not proprietary/closed source so it can be reviewed by peers.
- Good cryptography is developed carefully, slowly, and rarely alone.
- Good cryptography is usually presented and discussed in conferences and published in various journals.
- Good cryptography is extensively peer-reviewed before it is released for use in the wild.
- Using and implementing existing good cryptography correctly is already a challenge.

Yet, this is not stopping some from doing it anyway and publishing various production Apps/Services using their self-made cryptography or proprietary closed-source methods:

- You should apply caution when using Apps/Services using closed-source or proprietary encryption methods. All the good crypto standards are public and peer-reviewed and there should be no issue disclosing the one you use.
- You should be wary of Apps/Services using a "modified" or proprietary cryptographic method.
- By default, you should not trust any "Roll your own crypto" until it was audited, peer-reviewed, vetted, and accepted by the cryptography community'.
- There is no such thing as "military-grade crypto"''.

Cryptography is a complex topic and bad cryptography could easily lead to your de-anonymization.

In the context of this guide,we recommend sticking to Apps/Services using well-established, published, and peer-reviewed methods.

So, what to prefer and what to avoid as of 2021? You will have to look up for yourself to get the technical details of each app and see if they are using "bad crypto" or "good crypto". Once you get the technical details, you could check this page for seeing what it is worth: https://latacora.micro.blog/2018/04/03/cryptographic-right-answers.html [Archive.org]

Here are some examples:

- Hashes:
  - Prefer: SHA-3 or BLAKE2
  - Still relatively ok to use: SHA-2 (such as the widely used SHA-256 or SHA-512)
  - Avoid: SHA-1, MD5 (unfortunately still widely used), CRC, MD6 (rarely used)
- File/Disk Encryption:
  - Prefer:
    - Hardware Accelerated: AES (Rijndael) 256 Bits with HMAC-SHA-2 or HMAC-SHA-3 (This is what Veracrypt, Bitlocker, Filevault 2, KeepassXC, and LUKS use by default). Prefer SHA-3.
    - Non-Hardware Accelerated: Same as accelerated above or if available consider:
      - ChaCha20 or XChaCha20 (You can use ChaCha20 with Kryptor https://www.kryptor.co.uk, unfortunately, it is not available with Veracrypt).
      - Serpent
      - TwoFish
  - Avoid: Pretty much anything else
- Password Storage:
  - Prefer: Argon2, scrypt
  - If these aren't options, use bcrypt, or if not possible at least PBKDF2 (only as a last resort)
  - Be skeptical of Argon2d, as it's vulnerable to some forms of side-channels. Prefer Argon2i or Argon2id

    - Avoid: SHA-3, SHA-2, SHA-1, MD5
- Browser Security (HTTPS):
    - Prefer: TLS 1.3 (ideally TLS 1.3 with ECH/eSNI support) or at least TLS 1.2 (widely used)
    - Avoid: Anything Else (TLS =<1.1, SSL =<3)
- Signing messages/files with GnuPG (GPG):
    - Prefer ECDSA (ed25519)+ECDH (ec25519) or RSA 4096 Bits*
        - **Consider a more modern alternative to PGP/GPG: Minisign** https://jedisct1.github.io/minisign/ [Archive.org]
    - Avoid: RSA 2048 bits
- SSH keys:
    - ED25519 (preferred) or RSA 4096 Bits*
    - Avoid: RSA 2048 bits
- **Warning: RSA and ED25519 are unfortunately not seen as "Quantum Resistant" and while they have not been broken yet, they probably will be broken someday into the future. It is just a matter of when rather than if RSA will ever be broken. So, these are preferred in those contexts due to the lack of a better possibility.**

Here are some real cases of issues bad cryptography:

- Telegram: https://democratic-europe.eu/2021/07/20/cryptographers-uncovet-four-vulnerabilities-in-telegram/ [Archive.org]
- Telegram: https://buttondown.email/cryptography-dispatches/archive/cryptography-dispatches-the-most-backdoor-looking/ [Archive.org]
- Cryptocat: https://web.archive.org/web/20130705051050/https://blog.crypto.cat/2013/07/new-critical-vulnerability-in-cryptocat-details/
- Some other examples can be found here: https://www.cryptofails.com/ [Archive.org]

Later this guide will not recommend "bad cryptography" and that should hopefully be enough to protect you?

No Log 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 or a national security letter) log your activity to de-anonymize you. There have been several recent examples of those:

- 2021, Proton, Proton logged IP address of French activist after an order by Swiss authorities (source link unavailable).
- 2021, WindScribe, Servers were not encrypted as they should have been allowing MITM attacks by authorities.
- 2021, DoubleVPN servers, logs, and account info seized by law enforcement.
- 2021, The Germany-based mail provider Tutanota was forced to monitor specific accounts for 3 months.
- 2020, The Germany-based mail provider Tutanota was forced to implement a backdoor to intercept and save copies of the unencrypted e-mails of one user (they did not decrypt the stored e-mail).
- 2017, PureVPN was forced to disclose information of one user to the FBI.
- 2014, an EarthVPN user was arrested based on logs provider to the Dutch Police.
- 2013, Secure E-Mail provider Lavabit shuts down after fighting a secret gag order.
- 2011, HideMyAss user was de-anonymized, and logs were provided to the FBI.

Some providers have implemented the use of a Warrant Canary 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 we know.

Finally, it is now well known that some companies might be sponsored front ends for some state adversaries (see the Crypto AG story and Omnisec story).

For these reasons, you mustn'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 accounts were targeted by such orders and you might never know at all.

To mitigate this, in cases where you want to use a VPN, we will recommend the use of a cash/Monero-paid VPN provider over Tor to prevent the VPN service from knowing any identifiable information about you.

If the VPN provider knows nothing about you, it should mitigate any issue due to them not logging but logging anyway.

Some Advanced targeted techniques



(Illustration: an excellent movie we highly recommend: Das Leben der Anderen)

Many advanced techniques can be used by skilled adversaries 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 [Archive.org] (Air-Gap Research Page, Cyber-Security Research Center, Ben-Gurion University of the Negev, Israel) but also in this report https://www.welivesecurity.com/wp-content/uploads/2021/12/eset_jumping_the_air_gap_wp.pdf [Archive.org] (ESET, JUMPING THE AIR GAP: 15 years of nation-state effort) and include:

- Attacks requiring malware implants:
    - Exfiltration of Data through a Malware infected Router: https://www.youtube.com/watch?v=mSNt4h7EDKo [Invidious]
    - Exfiltration of Data through observation of Light variation in a Backlit keyboard with a compromised camera: https://www.youtube.com/watch?v=1kBGDHVr7x0 [Invidious]
        - Exfiltration of Data through a compromised Security Camera (that could first use the previous attack) https://www.youtube.com/watch?v=om5fNqKjj2M [Invidious]
        - Communication from outsider to compromised Security Cameras through IR light signals: https://www.youtube.com/watch?v=auoYKSzdOj4 [Invidious]
    - 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 [Invidious]
    - Exfiltration of data from a malware-infected air-gapped computer through HD LEDs with a Drone https://www.youtube.com/watch?v=4vIu8Id68fc [Invidious]
    - Exfiltration of data from a USB malware on an air-gapped computer through electromagnetic interferences https://www.youtube.com/watch?v=F28V1t-k8Hk [Invidious]
    - Exfiltration of data from a malware-infected HDD drive through covert acoustic noise https://www.youtube.com/watch?v=H7lQXmSLiP8 [Invidious]
    - Exfiltration of data through GSM frequencies from a compromised (with malware) air-gapped computer https://www.youtube.com/watch?v=RChj7Mg3rC4 [Invidious]
    - Exfiltration of data through electromagnetic emissions from a compromised Display device https://www.youtube.com/watch?v=2OzTWiGl1rM&t=20s [Invidious]
    - 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 [Invidious]
    - Communication between two compromised air-gapped computers using ultrasonic soundwaves https://www.youtube.com/watch?v=yz8E5n1Tzlo [Invidious]

    - Exfiltration of Bitcoin Wallet from a compromised air-gapped computer to a smartphone https://www.youtube.com/watch?v=2WtiHZNevcY [Invidious]
    - Exfiltration of Data from a compromised air-gapped computer using display brightness https://www.youtube.com/watch?v=ZrkZUO2g4DE [Invidious]
    - Exfiltration of Data from a compromised air-gapped computer through vibrations https://www.youtube.com/watch?v=XGD343nq1dg [Invidious]
    - Exfiltration of Data from a compromised air-gapped computer by turning RAM into a Wi-Fi emitter https://www.youtube.com/watch?v=vhNnc0In63c [Invidious]
    - Exfiltration of Data from a compromised air-gapped computer through power lines https://arxiv.org/pdf/1804.04014.pdf [Archive.org]
- **Attacks not requiring malware:**
    - Observing a blank wall in a room from a distance to figure how many people are in a room and what they are doing. Publication with demonstration: http://wallcamera.csail.mit.edu/ [Archive.org]
    - Observing a reflective bag of snacks in a room from a distance to reconstruct the entire room. Publication with photographic examples: https://arxiv.org/pdf/2001.04642.pdf [Archive.org]
    - Measuring floor vibrations to identify individuals and determine their health condition and mood. Publication with demonstration: https://engineering.cmu.edu/news-events/news/2020/02/17-mauraders-map.html [Archive.org]
    - Observing a light bulb from a distance to listen to the sound in the room **without any malware**: Demonstration: https://www.youtube.com/watch?v=t32QvpfOHqw [Invidious]. It should be noted that this type of attack is not new at all and there have been articles about such techniques as far back as 2013 and that you can even buy devices to perform this yourself such as here: http://www.geomtech.com/ecp0-prodshow/laser-surveillance-laser-listening.html [Archive.org]

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=YKRtFg1nyj4 [Invidious]

Realistically, this guide will be of little help against such adversaries as such malware could be implanted on the devices by a manufacturer, 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/unsecured 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 a Faraday cage to prevent electromagnetic leaks.
- Do not talk about sensitive information where lightbulbs could be seen 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.

Some bonus resources

- Have a look at the Whonix Documentation concerning Data Collection techniques here: Data Collection Techniques [Archive.org]
- You might also enjoy looking at this service https://tosdr.org/ [Archive.org] (Terms of Services, Didn't Read) that will give you a good overview of the various ToS of many services.
- Have a look at https://www.eff.org/issues/privacy [Archive.org] for some more resources.
- Have a look at https://en.wikipedia.org/wiki/List_of_government_mass_surveillance_projects [Wikiless] [Archive.org] to have an overview of all known mass-surveillance projects, current, and past.
- Have a look at https://www.gwern.net/Death-Note-Anonymity [Archive.org] (even if you don't know about Death Note).
- Consider finding and reading Michael Bazzell's book "Open-Source Intelligence Techniques" (eighth edition as of this writing to find out more about recent OSINT techniques) https://inteltechniques.com/book1.html
- Finally, check https://www.freehaven.net/anonbib/date.html [Archive.org] for the latest academic papers related to Online Anonymity.

Notes

If you still do not 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 will not count things like PRISM, MUSCULAR, SORM or XKEYSCORE explained earlier:

- Google Transparency Report https://transparencyreport.google.com/user-data/overview [Archive.org]
- Facebook Transparency Report https://transparency.facebook.com/ [Archive.org]
- Apple Transparency Report https://www.apple.com/legal/transparency/ [Archive.org]
- Cloudflare Transparency Report https://www.cloudflare.com/transparency/ [Archive.org]
- Snapchat Transparency Report https://www.snap.com/en-US/privacy/transparency [Archive.org]
- Telegram Transparency Report https://t.me/transparency [Archive.org] (requires telegram installed)
- Microsoft Transparency Report https://www.microsoft.com/en-us/corporate-responsibility/law-enforcement-requests-report [Archive.org]
- Amazon Transparency Report https://www.amazon.com/gp/help/customer/display.html?nodeId=GYSDRGWQ2C2CRYEF [Archive.org]
- Dropbox Transparency Report https://www.dropbox.com/transparency [Archive.org]
- Discord Transparency Report https://discord.com/blog/discord-transparency-report-q1-2022 [Archive.org]
- GitHub Transparency Report https://github.blog/2021-02-25-2020-transparency-report/ [Archive.org]
- Snapchat Transparency Report https://www.snap.com/en-US/privacy/transparency/ [Archive.org]
- TikTok Transparency Report https://www.tiktok.com/transparency/en/information-requests-2021-2/ [Archive.org]
- Reddit Transparency Report https://www.redditinc.com/policies/transparency-report-2021 [Archive.org]
- Twitter Transparency Report https://transparency.twitter.com/ [Archive.org]

General Preparations

Personally, in the context of this guide, it is also interesting to have a look at your security model. And in this context,we only have one to recommend:

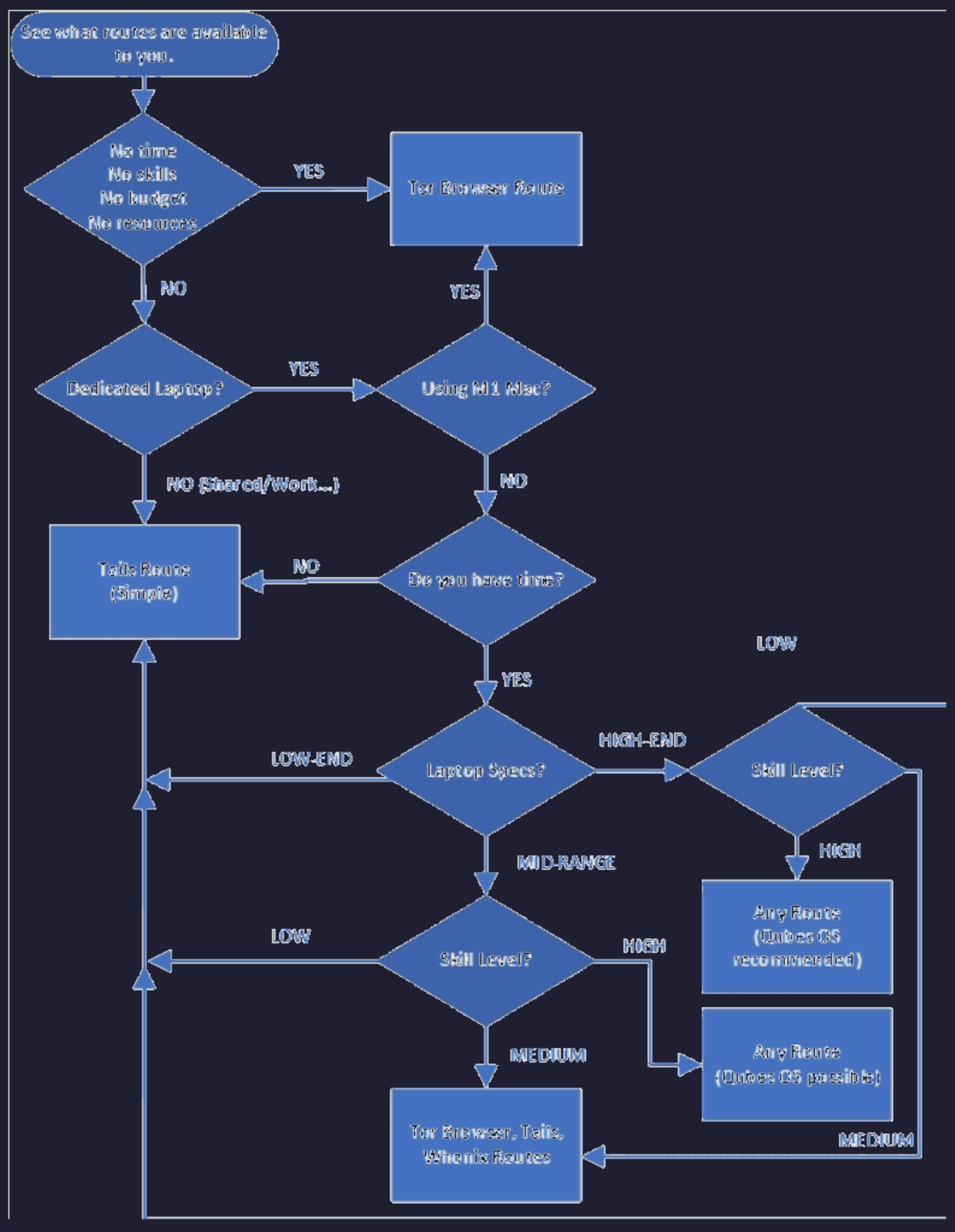
Zero-Trust Security ("Never trust, always verify").

Here are some various resources about what Zero-Trust Security is:

- DEFCON, Zero Trust a Vision for Securing Cloud, https://www.youtube.com/watch?v=euSsqXO53GY [Invidious]
- From the NSA themselves, Embracing a Zero Trust Security Model, https://media.defense.gov/2021/Feb/25/2002588479/-1/-1/0/CSI_EMBRACING_ZT_SECURITY_MODEL_UOO115131-21.PDF [Archive.org]

Picking your route

First, here is a small basic UML diagram showing your available options according to your skills/budget/time/resources.



Timing limitations

- You have no time at all:
  - **Go for the Tor Browser route.**
- You have extremely limited time to learn and need a fast-working solution:
  - **Your best option is to go for the Tails route (excluding the persistent plausible deniability section).**
- You have time and more importantly motivation to learn:
  - **Go with any route.**

Budget & Material limitations

- You have no budget and even accessing a laptop is complicated or you only have your smartphone:
  - **Go for the Tor Browser route.**
- 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 unsupervised/unmonitored laptop for your sensitive activities:
  - But it is 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 is not that old, and it has decent specs (at least 8GB of RAM, 250GB of disk space or more, decent CPU):
    - **You could go for Tails, Whonix routes.**
  - It is new and it has great specs (more than 16GB or ideally 32GB of RAM, >250GB of disk space, recent fast CPU):

        - **You could go for any route, but we would recommend Qubes OS if your threat model allows it. Please see the requirements.**
    - If it is an ARM-based M1/M2 Mac:
        - **Not possible currently for these reasons:**
            - Virtualization of Intel x86 images on ARM (M1/M2) hosts is still limited to commercial software (e.g., Parallels, Fusion) which are mostly not supported by Whonix, yet. They are very buggy and for advanced people only. Please seek this information yourself.
            - Virtualbox is now available natively for ARM64 architecture in a package as of October 2022. Download the "Developer preview for macOS/Arm64 (M1/M2) hosts".
            - Whonix does not support macOS easily. "You need to build Whonix using the build script to get it running on Apple Silicon." See the forum thread.
            - Tails is not supported on ARM64 architecture yet. See this thread for more information (keep in mind this page hasn't been updated recently).
            - Qubes OS is not supported on ARM64 architecture yet, but there is work being done to make it available on aarch64, which may be delayed for the unforseeable future..

The general advice in this guide regarding virtualization software is that it's costly. That said, you should probably get a dedicated laptop, capable of running virtualization software, preferably a 64-bit architecture, to be used for more sensitive activities and testing.

Skills

- Do you have no IT skills at all the content of this guide look like an alien language to you? Consider:
    - **The Tor Browser route (simplest of all)**
    - The Tails route **(excluding the persistent plausible deniability section).**
- You have some IT skills and mostly understand this guide so far, consider:
    - **The Tails route (with the optional persistent plausible deniability section).**
    - **The Whonix route.**
- You have moderate to high IT skills, and you are already familiar with some of the content of this guide, consider:
    - **Any route (Qubes OS is preferred if you can afford it).**
- You are an l33T hacker, "there is no spoon", "the cake is a lie", you have been using "doas" for years, and "all your base is belong to us", and you have strong opinions on systemd.
    - **This guide is not meant for you and will not help you with your HardenedBSD on your hardened Libreboot laptop ;-)**

Adversarial considerations

Now that you know what is possible, you should also consider threats and adversaries before picking the right route.

Threats

- If your main concern is a forensic examination of your devices, you should consider the Tor Browser route or the Tails route.
- If your main concerns are remote adversaries that might uncover your online identity on various platforms, you should consider the Tails, Whonix, or Qubes OS routes (listed in order of difficulty).
- If you want system-wide plausible deniability' despite the risks', consider the Tails route, including the persistent plausible deniability section (see Persistent Plausible Deniability using Whonix & Tails).**
- If you are in a hostile environment where Tor/VPN usage alone is impossible/dangerous/suspicious, consider the Tails route (without actually using Tor), or more advanced routes like Whonix or Qubes OS.

Adversaries

- Low skills:
  - Low resources:
    - Any motivation: Any Route
  - Medium resources:
    - Low to Medium motivation: Any Route
    - High motivation: TAILS, Whonix, Qubes OS Routes
  - High resources:
    - Low motivation: Any route
    - Medium to High motivation: TAILS, Whonix, Qubes OS Routes
- Intermediate skills:
  - Low resources:
    - Low motivation: Any Route
    - Medium to High motivation: TAILS, Whonix, Qubes OS Routes
  - Medium resources:
    - Low motivation: Any Route
    - Medium to High motivation: TAILS, Whonix, Qubes OS Routes
  - High resources:
    - Low to High motivation: TAILS, Whonix, Qubes OS Routes
- Highly skilled:
  - Low resources:
    - Low motivation: Any Route
    - Medium to High motivation: TAILS, Whonix, Qubes OS Routes
  - Medium resources:
    - Low to High motivation: TAILS, Whonix, Qubes OS Routes
  - High resources:
    - Low to High motivations: TAILS, Whonix, Qubes OS Routes **(but likely out of scope from this guide as this is probably a global adversary)**

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

- Warning [Archive.org]
- Dev/Threat Model [Archive.org]
- Comparison with Others [Archive.org]

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 [Archive.org]
- Check some data yourself here on the Tor Project OONI (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.
- Specific to China, look at https://gfwatch.org/ and https://www.usenix.org/system/files/sec21-hoang.pdf [Archive.org]

- Test for yourself using OONI (this can be risky in a hostile environment).

Steps for all routes

Getting used to using better passwords

See Guidelines for passwords and passphrases.

Getting 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 on and at home. 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.

A note regarding your current phone: The point of leaving your smartphone on is to create avoid leaking the fact that you're not using the device. If a smartphone is turned off, this creates a metadata trail that can be used to correlate the time your smartphone was turned off with the activation of your burner. If possible, leave your phone doing something (for example, watching YouTube on auto-play) to obscure the metadata trail further. This will not make it impossible to correlate your inactivity, but may make it more difficult if your phone's usage patterns can look convincing while you buy your burner.

We would recommend getting an old "dumbphone" with a removable battery (old Nokia if your mobile networks still allow 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 Wi-Fi.

Site Note: Be careful of some sellers as shown here https://therecord.media/malware-found-preinstalled-in-classic-push-button-phones-sold-in-russia [Archive.org]

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 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 is not as these phones are only being used during the setup/sign-up process and for verification from time to time.

See Warning about smartphones and smart devices

You should test that the phone is in working order before going to the next step. But we will repeat ourselves and state that it is important to leave your smartphone at home when going (or turn it off before leaving if you must keep it) and that you test the phone at a random location that cannot be tracked back to you (and again, do not do that in front of a CCTV, avoid cameras, be aware of your surroundings). No need for Wi-Fi 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.

GETTING AN ANONYMOUS PRE-PAID SIM CARD

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

So here is a list of places where you can still get them now: https://prepaid-data-sim-card.fandom.com/wiki/Registration_Policies_Per_Country [Archive.org]

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 do not forget to buy a Top-Up voucher with the SIM card (if it is not a package) as most pre-paid cards will require a top-up before use.

See 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 live in.

We would recommend GiffGaff in the UK as they are "affordable", do not require identification for activation and top-up, and will even allow you to change your number up to two times from their website. One GiffGaff prepaid SIM card will therefore grant you three 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 are not at a place that can be used to reveal your identity and ideally leave your real phone on but at home before going to the safe place with only your burner phone.

Online Phone Number

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/unsecured 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 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 scammers and very risky in terms of anonymity. **You should not use those under any circumstance.**

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

- **Recommended**: No name, no e-mail, and no KYC:
  - (Iceland based, accepts Monero) https://crypton.sh [Tor Mirror] [Archive.org]
  - (Ukraine based, accepts Monero) https://virtualsim.net/ [Archive.org]
  - (180+ countries, accepts crypto) https://cypheresim.com/esim-with-number [Archive.org]
- Require valid e-mail:
  - (US California based, accepts Monero) https://mobilesms.io [Archive.org]
  - (Germany based, accepts Monero) https://www.sms77.io/ [Archive.org]
  - (Russia based, accepts Monero) https://onlinesim.ru/ [Archive.org]

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

Now, what if you have no money? Well, in that case, you will have to try your luck with free services and hope for the best. Here are some examples, **use at your own risk**:

- https://oksms.org
- https://smspva.com
- https://sms24.me

Disclaimer: We cannot vouch for any of these providers. We 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. Please do read Monero Disclaimer.

It is more convenient, cheaper, and less risky to just get a pre-paid SIM card from one of the physical places that still sell them for cash without ID.

Get a USB key

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

Get at least one or two decent size generic USB keys (at least 16GB but we would recommend 32GB).

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

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

Find some safe places with decent public Wi-Fi

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

This can be anywhere that will not be tied to you directly (your home/work) and where you can use the Wi-Fi 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 clever idea, you may reconsider:

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

While this will not be available yet to your local police officers, it could be in the near future.

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

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

The Tor Browser route

This part of the guide will help you in setting up the simplest and easiest way to browse the web anonymously. It is not necessarily the best method and there are more advanced methods below with (much) better security and (much) better mitigations against various adversaries. Yet, this is a straightforward way of accessing resources anonymously and quickly with no budget, no time, no skills, and limited usage.

So, what is Tor Browser? Tor Browser (https://www.torproject.org/ [Archive.org]) is a web browser like Safari/Firefox/Chrome/Edge/Brave designed with privacy and anonymity in mind.

This browser is different from other browsers as it will connect to the internet through the Tor Network using Onion Routing. We first recommend that you watch this very nice introduction video by the Tor Project themselves: https://www.youtube.com/watch?v=JWI85UlzKw [Invidious]. After that, you should probably head over to their page to read their quick overview here: https://2019.www.torproject.org/about/overview.html.en [Archive.org]. Without going into too many technical details, Tor Browser is an easy and simple "fire and forget" solution to browse the web anonymously from pretty much any device. It is probably sufficient for most people and can be used from any computer or smartphone.

Here are several ways to set it up for all main OSes.

Warning: You should avoid installing extensions in Tor Browser, as they can be used to fingerprint and identify you.

Windows, Linux, and macOS

Please see Installing and using desktop Tor Browser.

Android

Note on Tor Browser for Android: The development of Tor Browser for Android is behind desktop Tor Browser Bundle (TBB). Some features are not available yet. E.g., the desktop version of Tor now enables automatic bridges using Moat:

"**Connection Assist** works by looking up and downloading an up-to-date list of country-specific options to try using your location (with your consent). It manages to do so without needing to connect to the Tor Network first by utilizing moat – the same domain-fronting tool that Tor Browser uses to request a bridge from torproject.org."

- Head over to:
  - Play Store: https://play.google.com/store/apps/details?id=org.torproject.torbrowser
  - F-Droid Store: It's not yet there but you can add it manually following the instructions at https://support.torproject.org/tormobile/tormobile-7/ [Archive.org]

- Install
- Launch Tor Browser
- After launching, click the upper right **Settings** icon
- Select **Settings > Privacy and security > Tor network**
- Select **Config Bridge.**
- Read Using Tor bridges in hostile environments.
- **If needed (after reading the appendix above)**, activate the option and select the type of bridge you want:
    - Obfs4
    - Meek-Azure
    - Snowflake
- **If your internet isn't censored**, consider running one of the bridge types to help the network!
    - Easy: Obsf4 - You can run your own Obsf4 easily with these instructions. https://community.torproject.org/relay/setup/bridge/
    - Medium: Snowflake - More about Snowflakes here. https://snowflake.torproject.org/
    - Hard: Meek - This is the documentation. It's not as simple. https://gitlab.torproject.org/legacy/trac/-/wikis/doc/meek/#how-to-run-a-meek-server-bridge

Personally, if you need to use a Bridge (this is not necessary for a non-hostile environment), you should pick a Meek-Azure. Those will probably work even if you are in China and want to bypass the Great Firewall. It is probably the best option to obfuscate your Tor activities if needed and Microsoft servers are usually not blocked.

*Only available for Desktop Tor users: Recently, the Tor Project has made it incredibly simple to access Bridges with **Connection Assist**, and it is now automatically done in hostile or censored regions. Simply open the Tor Browser and the connection will be configured based on your needs on any hostile network. Previously, we had a list of options below this paragraph which were necessary to enable and configure bridges, but now that this is done automatically using moat.* [Archive.org]

- You are almost done

As with the desktop version, you need to know there are safety levels in Tor Browser. On Android, you can access these by following these steps:

- Click the menu (bottom right)
- Click **Settings**.
- Head over to the **Privacy and security** section.
- Click **Security Settings**.

You will find details about each level here: https://tb-manual.torproject.org/security-settings/ [Archive.org] but here is a summary:

- Standard (the default):
    - All features are enabled (including JavaScript)
- Safer:
    - JavaScript is disabled on non-HTTPS websites
    - Some fonts and symbols are disabled
    - Any media playback is "click to play" (disabled by default)
- Safest:
    - Javascript is disabled everywhere
    - Some fonts and symbols are disabled
    - Any media playback is "click to play" (disabled by default)

We would recommend the "Safer" level for most cases. The Safest level should be enabled if you think you are accessing suspicious or dangerous websites and/or if you are extra paranoid.

If you are extra paranoid, use the "Safest" level by default and consider downgrading to Safer is the website is unusable because of Javascript blocking.

However, the Safer level should be used with some extra precautions while using some websites: see Additional browser precautions with JavaScript enabled.

Now, you are really done, and you can now surf the web anonymously from your Android device.

Please see Warning for using Orbot on Android.

iOS

Disclaimer: Onion Browser, following a 2018 release on iOS, has had IP leaks via WebRTC. It is still the only officially endorsed browser for the Tor network for iOS. Users should exercise caution when using the browser and check for any DNS leaks.

While the official Tor Browser is not yet available for iOS, there is an alternative called Onion Browser endorsed by the Tor Project.

- Head over to https://apps.apple.com/us/app/onion-browser/id519296448
- Install
- Disable Wi-Fi and Mobile Data
- Launch Onion Browser
- After Launching, click the upper right Settings icon (Disabling Wi-Fi and Mobile Data previously were to prevent Onion Browser from connecting automatically and to allow access to these options).
- Select "Bridge Configuration" and read Using Tor bridges in hostile environments
- **If needed (after reading the appendix above)**, activate the option and select the type of bridge you want:
    - Obfs4
    - Snowflake
    - (Meek-Azure is unfortunately not available on Onion Browser for iOS (See commit 21bc18428 for more information.)
- **If your internet isn't censored**, consider running one of the bridge types to help the network!
    - Easy: Obsf4 - You can run your own Obsf4 easily with these instructions. https://community.torproject.org/relay/setup/bridge/
    - Medium: Snowflake - More about Snowflakes here. https://snowflake.torproject.org/
    - Hard: Meek - This is the documentation. It's not as simple. https://gitlab.torproject.org/legacy/trac/-/wikis/doc/meek/#how-to-run-a-meek-server-bridge

Personally, if you need to use a Bridge (this is not necessary for a non-hostile environment), you should pick a Snowflake one (since Meek-Azure bridges are not available). Those will probably work even if you are in China and want to bypass the Great Firewall. It is probably the best option you have on iOS.

- You are almost done

As with the desktop version, you need to know there are safety levels in Onion Browser. On iOS, you can access these by following these steps:

- Click the shield icon (upper left)
- You will have three levels to pick from

Security Level	JavaScript	WS/XHR/Geo	A/V	Apps	WebRTC	HTTP↔HTTPS	Ads/Pop-ups
Gold	Disabled	Disabled	No	Blocked	Blocked	Blocked	Blocked
Silver	Partially allowed	Disabled	No	Blocked	Blocked	Blocked	Blocked
Bronze	Allowed	Enabled	Allowed	Blocked	Not blocked	Not blocked	Blocked

We would recommend the "Silver" level for most cases. The Gold level should only be enabled if you think you are accessing suspicious or dangerous websites or if you are extra paranoid. The Gold mode will also most likely break many websites that rely actively on JavaScript.

As JavaScript is enabled in the Silver mode, please see Additional browser precautions with JavaScript enabled.

Now, you are really done, and you can now surf the web anonymously from your iOS device.

Important Warning

This route is the easiest but is not designed to resist highly skilled adversaries. It is however usable on any device regardless of the configuration. This route is also vulnerable to correlation attacks (See Your Anonymized Tor/VPN traffic) and is blind to anything that might be on your device (this could be any malware, exploit, virus, remote administration software, parental controls...). Yet, if your threat model is quite low, it is probably sufficient for most people.

If you have time and want to learn, we recommend going for other routes instead as they offer far better security and mitigate far more risks while lowering your attack surface considerably.

The Tails route

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

- You cannot 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 stands for **The Amnesic Incognito Live System**. It is 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 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 an amazingly straightforward way to get going in no time with what you have and without much learning. It has extensive documentation and tutorials.

WARNING: Tails is not always up to date with their bundled software. And not always up to date with the Tor Browser updates either. You should always make sure you are using the latest version of Tails and you should use extreme caution when using bundled apps within Tails that might be vulnerable to exploits and reveal your location.

It does however have some drawbacks:

- Tails uses Tor and therefore you will 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 detail later).
- Your ISP (whether it is yours or some public Wi-Fi) will also see that you are using Tor, and this could make you suspicious in itself.
- Tails does not 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 is very secure will be detected as well by most platforms and will hinder you in creating anonymous identities on many platforms.
- Tails will not protect you more from the 5$ wrench.
- Tor in itself might not be enough to protect you from an adversary with enough resources as explained earlier.

Important Note: If your laptop is monitored/supervised and some local restrictions are in place, please read How to bypass (some) local restrictions on supervised computers.

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

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

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

If you're having an 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/anonymous_internet/tor/index.en.html [Archive.org] and find more information about these on Tor Documentation https://2019.www.torproject.org/docs/bridges [Archive.org]

If you think using Tor alone is dangerous/suspicious, see What about when Tor and VPNs aren't possible?

Tor Browser settings on Tails

When using Tor Browser, you should click the little shield Icon (upper right, next to the Address bar) and select your Security level (see https://tb-manual.torproject.org/security-settings/ [Archive.org] for details). Basically, there are three.

- Standard (the default):
  - All features are enabled (including JavaScript)
- Safer:
  - JavaScript is disabled on non-HTTPS websites
  - Some fonts and symbols are disabled
  - Any media playback is "click to play" (disabled by default)
- Safest:
  - Javascript is disabled everywhere
  - Some fonts and symbols are disabled
  - Any media playback is "click to play" (disabled by default)

We would recommend the "Safer" level for most cases. The Safest level should be enabled if you think you are accessing suspicious or dangerous websites or if you are extra paranoid. The Safest mode will also most likely break many websites that rely actively on JavaScript.

If you are extra paranoid, use the "Safest" level by default and consider downgrading to Safer is the website is unusable because of Javascript blocking.

Lastly, while using Tor Browser on Tails on the "Safer" level, please consider Additional browser precautions with JavaScript enabled.

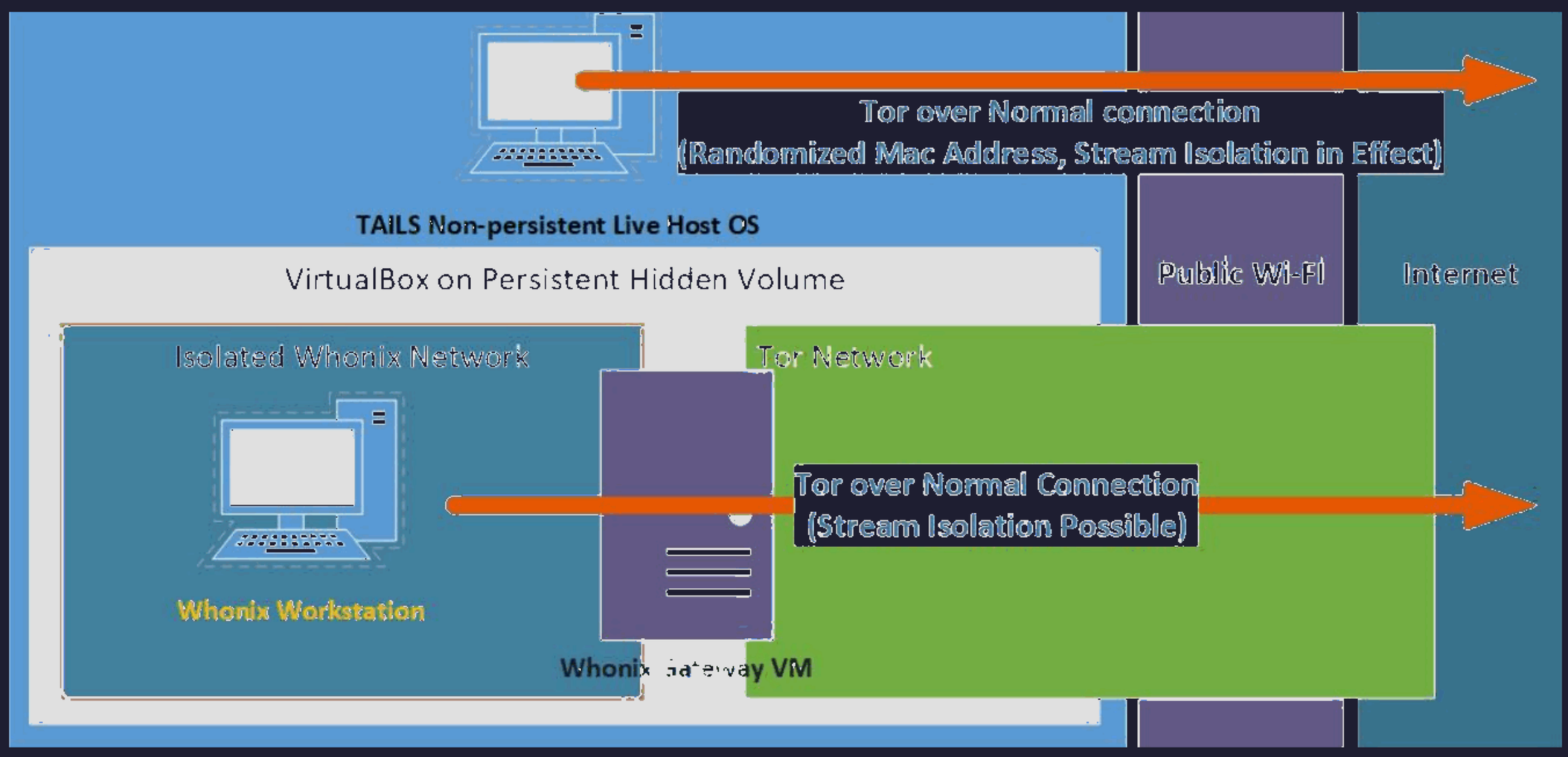
When you are done and have a working Tails on your laptop, go to the Creating your anonymous online identities step much further in this guide or if you want persistence and plausible deniability, continue with the next section.

Persistent Plausible Deniability using Whonix & Tails

Consider checking the https://github.com/aforensics/HiddenVM [Archive.org] 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 (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/SSD 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.



Note: See Pick your connectivity method in the Whonix Route for more explanations about Stream Isolation

In short:

- You could run non-persistent Tails from one USB key (following their recommendations)
- You could store persistent VMs within a secondary container that could be encrypted normally or using the Veracrypt plausible deniability feature (these could be Whonix VMs for instance or any other).
- You do benefit from the added Tor Stream Isolation feature (see [Tor over VPN] for more info about stream isolation).

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 easily discoverable by a soft adversary.

This option is particularly interesting for "traveling light" and to mitigate forensics attacks while keeping persistence on your work.
You only need 2 USB keys (one with Tails and one with a Veracrypt container containing persistent Whonix). The first USB key will appear to contain just Tails and the second USB will appear to contain just random garbage but will have a decoy volume which you can show for plausible deniability.

You might also wonder if this will result in a "Tor over Tor" setup, but it will not. The Whonix VMs will be accessing the network directly through clearnet and not through Tails Onion Routing.

In the future, this could also be supported by the Whonix project themselves as explained here: Whonix-Host [Archive.org] but it is not yet recommended for end-users.

Remember that encryption with or without plausible deniability is not a silver bullet and will be of little use in case of torture. 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 [Archive.org]

Plausible deniability is only effective against soft lawful adversaries that will not resort to physical means.

See https://en.wikipedia.org/wiki/Rubber-hose_cryptanalysis [Wikiless] [Archive.org]

CAUTION: Please see 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.**
- **Use instead file containers instead of encrypted volumes.**
- **Make sure you do know how to clean data from an external SSD drive properly.**

Here is my guide on how to achieve this:

First Run

- Download the latest HiddenVM release from https://github.com/aforensics/HiddenVM/releases [Archive.org]
- Download the latest Whonix XFCE release from GitHub Releases [Archive.org]

Note: As of guide v1.2.5, Whonix has been upgraded from version 17 to 18 (Whonix 18.1.4.x). See [Upgrading to Whonix 18] for the upgrade path if you currently use Whonix 17.x.

- Prepare a USB Key/Drive with Veracrypt
    - Create a Hidden Volume on the USB/Key Drive (We would recommend at least 16GB for the hidden volume)
    - In the Outer Volume, place some decoy files
    - In the Hidden Volume, place the HiddenVM appimage file
    - In the Hidden Volume, place the Whonix XFCE ova file
- Boot into Tails
- Setup the Keyboard layout as you want.
- Select Additional Settings and set an administrator (root) password (needed for installing HiddenVM)
- Start Tails
- Connect to a safe wi-fi (this is a required step for the rest to work)

- Go into Utilities and Unlock your Veracrypt (hidden) Volume (do not forget to check the hidden volume checkbox)
- Launch the HiddenVM appimage
- When prompted to select a folder, select the Root of the Hidden volume (where the Whonix OVA and HiddenVM app image files are).
- Let it do its thing (This will install Virtualbox within Tails with one click)
- When it is done, it should automatically start Virtualbox Manager.
- Import the Whonix OVA files (see Whonix Virtual Machines)

Note, if during the import you are having issues such as "NS_ERROR_INVALID_ARG (0x80070057)", this is probably because there is not enough disk space on your Hidden volume for Whonix. Whonix themselves recommend 32GB of free space but that's probably not necessary and 10GB should be enough for a start. You can try working around this error by renaming the Whonix *.OVA file to *.TAR and decompressing it within Tails. When you are done with decompression, delete the OVA file and import the other files with the Import wizard. This time it might work.

Subsequent Runs

- Boot into Tails
- Connect to Wi-Fi
- Unlock your Hidden Volume
- Launch the HiddenVM App
- This should automatically open VirtualBox manager and show your earlier VMs from the first run

Steps for all other routes

Get a dedicated laptop

Ideally, you should get a dedicated laptop that will not be tied to you in any effortless way (ideally paid with cash anonymously and using the same precautions as previously mentioned for the phone and the SIM card). It is recommended but not mandatory. 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 which should prevent most adversaries from de-anonymizing you - besides state/global actors. It will take considerable resources.

This laptop should ideally be a clean, freshly installed laptop (running Windows, Linux, or macOS); which is clean of your normal day-to-day activities; and which is offline (never connected to your home network). In the case of a Windows laptop, and if you used it before such a clean install, it should also not be activated. Simply reinstall without a product key in the case that it came pre-activated. Specifically, in the case of MacBooks, it should never have been tied to your identity before in any means. So, buy secondhand with cash from an unknown stranger who does not know your identity.

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 (ideally 8GB or 16GB)** 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. You should aim for something with large storage (1TB+) if possible because we will need as much as possible.

This laptop could have an HDD (7200rpm) or an SSD/NVMe drive. Both possibilities have their benefits 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 Public Wi-Fi in a safe place (see Find some safe places with decent public Wi-Fi). But several steps will have to be taken offline first.

Some laptop recommendations

We would strongly recommend getting a "business grade" laptop (meaning not consumer/gaming-grade laptop) if you can. For instance, some ThinkPad from Lenovo (my personal favorite).

This is because those business laptops usually offer better and more customizable security features (especially in the BIOS/UEFI settings) with longer support than most consumer laptops (Asus, MSI, Gigabyte, Acer...). The interesting features to look for are:

- Better custom Secure Boot **settings (where you can selectively manage all the keys and not just use the Standard ones)**
- HDD/SSD 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 IME) from the BIOS/UEFI settings by default. And, because AFAIK, AMD PSP was audited and contrary to IME was not found to have any "evil" functionalities. However, if you are going for the Qubes OS Route consider Intel CPUs as Qubes OS does not support AMD with their anti-evil-maid system.
- Secure Wipe tools from the BIOS (especially useful for SSD/NVMe drives, see BIOS/UEFI options to wipe disks in various Brands).
- Better control over the disabling/enabling of select peripherals (USB ports, Wi-Fis, Bluetooth, Camera, Microphone ...).
- Better security features with Virtualization.
- Native anti-tampering protections.
- Longer support with BIOS/UEFI updates (and subsequent BIOS/UEFI 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 [Archive.org]

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

Once you are in there, you will 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 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/SSD password if the feature is available. This feature will add another password on the HDD/SSD itself (not in the BIOS/UEFI firmware) that will prevent this HDD/SSD 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 (odds are very high you can't).
- Disable AMD PSP if you can (AMD's equivalent to IME, see Your CPU)
- Disable Secure Boot if you intend to use Qubes OS as they do not support it out of the box. Keep it on if you intend to use Linux/Windows.
- Check if your laptop BIOS has a secure erase option for your HDD/SSD that could be convenient in case of need.

Only enable those on a "need to use" basis and disable them 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? In short, it is 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.

When the operating system (or the Bootloader) 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 by the password you set up 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. For Secure Boot to work, your Operating System will have to support it and then sign its bootloader and push those signing keys to your UEFI firmware. After that, you will 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 are from the manufacturer itself or 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 include Windows, Fedora, Ubuntu, Mint, Debian, CentOS, OpenSUSE, Tails, Clonezilla, and many others. Secure Boot is however not supported at all by Qubes OS at this point.

In some laptops, you can manage those keys and remove the ones you do not want with a "custom mode" to only authorize your bootloader that you could sign yourself if you 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). Most mainstream Linux distributions are signed these days and will boot with Secure Boot enabled.
- Secure Boot can have flaws and exploits like any other system. If you are running an old laptop that does not benefit from new BIOS/UEFI updates, these can be left unfixed.

Additionally, several attacks could be possible against Secure Boot as explained (in-depth) in these technical videos:

- Defcon 22, https://www.youtube.com/watch?v=QDSIWa9xQuA [Invidious]
- BlackHat 2016, https://www.youtube.com/watch?v=0fZdL3ufVOI [Invidious]

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 is an added layer but that is it.

I still 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 [Archive.org]

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 [Archive.org]

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

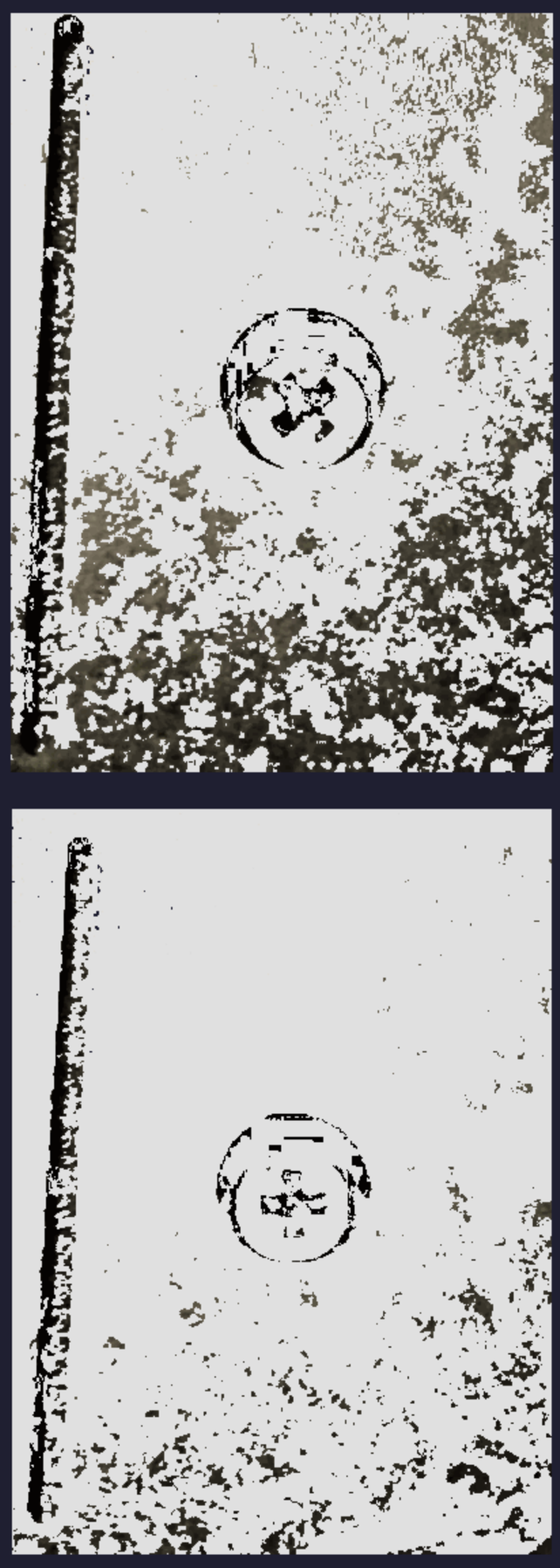
At some point, you will inevitably leave this laptop alone somewhere. You will not sleep with it and take it everywhere every single day. You should make it as hard as possible for anyone to tamper with it without you noticing it. This is mostly useful against some limited adversaries that will not use a 5$ wrench against you.

It is important to know that it is 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/ [Archive.org] (with pictures).

While this is a good cheap method, it could also raise suspicions as it is 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-up macro photography of the back screws of your laptop or just use a 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 was not careful enough (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.



The 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 a USB key in it.

In riskier environments, check your laptop for tampering before using it regularly.

The Whonix route

Note on Whonix Version: Whonix has been upgraded from version 17 to 18 (Whonix 18.1.4.x at the time of writing) with automated release upgrade support. See Upgrading to Whonix 18 for the upgrade path if you currently use Whonix 17.x or earlier versions.

Picking your Host OS

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

- Your Linux distribution of choice (excluding Qubes OS)
- Windows 10/11 (preferably Home edition due to the absence of Bitlocker)
- macOS (Catalina or higher up to Monterey)

In addition, chances are high that your Mac is or has been tied to an Apple account (at the time of 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 **conveniently** use Plausible Deniability (aka Deniable Encryption) easily at the OS level. Windows is also unfortunately at the same time a privacy nightmare but is the only easy to set up option for using OS-wide plausible deniability. Windows telemetry and telemetry blocking are also widely documented which should mitigate many issues.

So, what is Plausible Deniability? You can cooperate with an adversary requesting access to your device/data without revealing your true secret. All this using Deniable Encryption.

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' to exempt this from such rights (because terrorists and "think of the children"). In that case, you might have to reveal the password or face jail time in contempt of court. This is where plausible deniability will come 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 is 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 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 of 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 [Archive.org]) 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 10/11 Home (and not Pro). This is because Windows 10/11 Pro natively offers a full-disk encryption system (Bitlocker) where Windows 10/11 Home offers no full-disk encryption at all. You will later use third-party open-source software for encryption that will allow full-disk encryption on Windows 10/11 Home. This will give you a good (plausible) excuse to use this software. While using this software on Windows 10/11 Pro would be suspicious.

Note about Linux: So, what about Linux and plausible deniability? Yes, it is possible to achieve plausible deniability with Linux too. More information within the Linux Host OS section later.

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

Threats with encryption

The 5$ Wrench Attack

Remember that encryption with or without plausible deniability is not a silver bullet and will be of little use in case of torture. 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 [Archive.org]

Plausible deniability is only effective against soft lawful adversaries that will not resort to physical means. **Avoid, if possible, the use of plausible deniability-capable software (such as Veracrypt) if your threat model includes hard adversaries. So, Windows users should in that case install Windows Pro as a Host OS and use Bitlocker instead.**

See https://en.wikipedia.org/wiki/Rubber-hose_cryptanalysis [Wikiless] [Archive.org]

EVIL-MAID ATTACK

Evil Maid Attacks are conducted when someone tampers with your laptop while you are away. To install to clone your hard drive, install malware or a key logger. If they can 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 compelling evidence of the existence of 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 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 will not 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 the Anti Evil Maid (AEM) protection that can be enabled. This is the case with Windows/Veracrypt and QubeOS (only on Intel CPUs).

COLD-BOOT ATTACK

Cold Boot attacks 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 are actively using your device or shortly afterward.

The idea is rather simple, as shown in this video, 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 from decaying. They could then be able to copy its content for analysis. This memory dump could contain the key to decrypt your device. You will later apply a few principles to mitigate these.

In the case of Plausible Deniability, there have been some forensics studies 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 and by the maintainer of Veracrypt so we would not 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). Again see https://sourceforge.net/p/veracrypt/discussion/technical/thread/3961542951/ [Archive.org]
- Do enable the option to Wipe keys from memory if a device is inserted in Veracrypt.
- 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 in your memory for resuming your activity faster. Only hibernation and shutdown will actually clear the key from the memory.

See also Cold Boot Attack Defense [Archive.org] and Protection Against Physical Attacks [Archive.org].

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

- https://github.com/0xPoly/Centry [Archive.org] (unfortunately unmaintained it seems)
- https://github.com/hephaest0s/usbkill [Archive.org] (unfortunately unmaintained as well it seems)
- https://github.com/Lv14Sword/Killer [Archive.org]
- https://askubuntu.com/questions/153245/how-to-wipe-ram-on-shutdown-prevent-cold-boot-attacks [Archive.org]
- (Qubes OS, Intel CPU only) https://github.com/QubesOS/qubes-antievilmaid [Archive.org]

ABOUT SLEEP, HIBERNATION, AND SHUTDOWN

If you want 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 will 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 (not supported on Qubes OS). Since the whole disk is encrypted, hibernation in itself should not pose a large security risk but will still shut down your laptop and clear the memory while allowing you to conveniently resume your work afterward. **What you should never do is using 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 do not 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 are not compelled to reveal the password).

Let us 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 will have decoy documents/data within the decoy folder and sensitive documents/data within the sensitive folder.

In all cases, you will (most likely) open these folders with Windows Explorer, macOS Finder, or any other utility and do whatever you planned to do. Maybe you will edit a document within the sensitive folder. Maybe you will search for a document within the folder. Maybe you will 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 app, 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.
- Windows Indexing keeping traces of the files present in your user folder by default.
- Recent lists (aka Jump Lists) in Windows and various apps keeping traces of recently accessed documents.
- 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 [Archive.org]

macOS

- Gatekeeper 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' use all those leaks (see Local Data Leaks and Forensics) 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 cannot 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 are 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 is critically important that you disable and block telemetry with all the means at your disposal. No matter what OS you are using.

Conclusion

You should never conduct sensitive activities from a non-encrypted system. And even if it is encrypted, you should 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, we 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 in OS-wide plausible deniability and want to learn to use Linux, we will strongly recommend going for Linux or the Qubes OS 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 [Archive.org]

Linux Host OS

As mentioned earlier, we do not recommend using your daily laptop for sensitive activities. Or at least we do not recommend 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 do not want to wipe your laptop and start over, you should consider the Tails route or proceed at your own risk.

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 [Archive.org].

Full disk encryption

There are two routes here with Ubuntu or Debian based distros:

- Using LUKS:
    - Without plausible deniability:
        - (Recommended and easy) Encrypt as part of the installation process: https://ubuntu.com/tutorials/install-ubuntu-desktop [Archive.org]
            - 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 [Archive.org]
    - With plausible deniability: See the next section [The Detached Headers Way]
- Using Veracrypt:
    - With or without plausible deniability: See the next section [The Veracrypt Way]

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

Note about plausible deniability on Linux

There are several ways to achieve plausible deniability on Linux and it is possible to achieve. Here are some more details about some of the ways we would recommend. All these options require some higher level of skills at using Linux.

THE DETACHED HEADERS WAY

While not supported yet by this guide, it is possible to achieve a form of deniability on Linux using LUKS by using detached LUKS headers. For now, we will redirect you toward this page for more information: https://wiki.archlinux.org/title/Dm-crypt/Specialties#Encrypted_system_using_a_detached_LUKS_header [Archive.org]

THE VERACRYPT WAY

It is technically possible to not only use Veracrypt but also to achieve plausible deniability on a Linux Host OS by using Veracrypt for system full-disk encryption (instead of LUKS). This is not supported by Veracrypt (System encryption is only supported on Windows) and requires some tinkering with various commands. This is not recommended at all for unskilled users and should only be used at your own risk.

The steps to achieve this are not yet integrated into this guide but can be found here: http://dreadytofatroptsdj6io7l3xptbet6onoyno2yv7jicoxknyazubrad.onion/post/5779e55aac7fc06e4758 (this is a .onion address and requires Tor Browser).

Reject/Disable any telemetry

- During the install, just make sure you do not allow any data collection if prompted.
- If you are not sure, just make sure you did not enable any telemetry and follow this tutorial if needed https://vitux.com/how-to-force-ubuntu-to-stop-collecting-your-data-from-your-pc/ [Archive.org]
- Any other distro: you will need to document yourself and find out how to disable telemetry.

Disable anything unnecessary

- Disable Bluetooth if enabled by following this guide https://www.addictivetips.com/ubuntu-linux-tips/disable-bluetooth-in-ubuntu/ [Archive.org] 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 [Archive.org] 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 shut down 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 is possible to enable it, but it might not work as expected. Follow this information at your own risk. If you do not want to do this, you should never use the sleep function and power off instead (and set the lid closing behavior to power off instead of sleep).

Follow one of these tutorials to enable Hibernate:

- https://www.how2shout.com/linux/how-to-hibernate-ubuntu-20-04-lts-focal-fossa/ [Archive.org]
- http://www.lorenzobettini.it/2020/07/enabling-hibernation-on-ubuntu-20-04/ [Archive.org]
- https://blog.ivansmirnov.name/how-to-set-up-hibernate-on-ubuntu-20-04/ [Archive.org]

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/ [Archive.org] and this tutorial for Ubuntu 18.04 https://tipsonubuntu.com/2018/04/28/change-lid-close-action-ubuntu-18-04-lts/ [Archive.org]. There is no tutorial yet for Ubuntu 21.04 or 21.10 but the above for 20.04 should probably work too.

Unfortunately, this will not clean the key from memory directly 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 [Archive.org]

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

Enable MAC address randomization

- For Ubuntu, follow these steps https://help.ubuntu.com/community/AnonymizingNetworkMACAddresses [Archive.org].
- Consider this tutorial which should still work: https://josh.works/shell-script-basics-change-mac-address [Archive.org]

Hardening Linux

As a light introduction for new Linux users, consider https://www.youtube.com/watch?v=Sa0KqbpLye4 [Invidious]

For more in-depth and advanced options, refer to:

- This excellent guide: https://madaidans-insecurities.github.io/guides/linux-hardening.html [Archive.org]

- This excellent wiki resource: https://wiki.archlinux.org/title/Security [Archive.org]
- These excellent scripts are based on the guide and wiki above: https://codeberg.org/SalamanderSecurity/PARSEC [Archive.org]
- These tools that can help you harden your Linux Kernel:
  - Lynis: https://github.com/CISOfy/lynis
  - Kconfig-hardened-check: https://github.com/a13xp0p0v/kconfig-hardened-check
- Consider installing Safing Portmaster from https://safing.io/portmaster/ [Archive.org] **(Warning: there might be issues with some VPN clients. See: https://docs.safing.io/portmaster/install/status/vpn-compatibility** [Archive.org]
- Consider the use of KickSecure when using Debian: Kicksecure [Archive.org]
- This interesting article: http://0pointer.net/blog/authenticated-boot-and-disk-encryption-on-linux.html [Archive.org]

Setting up a safe Browser

See Safe Browser on the Host OS

macOS Host OS

Note: Mac M1/M2 chips are now supported natively, or, if you wish to use commercial tools like VMWare Fusion or Parallels Desktop, but those are not covered in this guide. Seek this information yourself.

As mentioned earlier, we do not recommend using your daily laptop for sensitive activities. Or at least we do not recommend 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 do not want to wipe your laptop and start over, you should consider the Tails route or proceed at your own risk.

We 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 that Mac.

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

As a light introduction for new macOS users, consider https://www.youtube.com/watch?v=lFx5icuE6Io [Invidious]

Now to go more in-depth in securing and hardening your macOS, we recommend reading this guide which covers many of the issues: https://www.bejarano.io/hardening-macos/ [Archive.org]

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 [Archive.org]

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 [Archive.org]

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 shut down. 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 will 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 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 set up an automatic sleep (Settings > Energy) so 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/ [Archive.org]

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 document yourself on the actual issue before acting. This page is a good place to start: https://blog.jacopo.io/en/post/apple-ocsp/ [Archive.org]

Up to you really. We would block it because we 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 [Archive.org]

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

MAC ADDRESS RANDOMIZATION

Unfortunately, macOS does not offer a native convenient way of randomizing your MAC Address and so you will have to do this manually. This will be reset at each reboot, and you will 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 this 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 Safe Browser on the Host OS

Windows Host OS

As mentioned earlier, we do not recommend using your daily laptop for sensitive activities. Or at leastWedo not recommend 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 do not want to wipe your laptop and start over, you should consider the Tails route or proceed at your own risk.

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

Installation

You should follow Windows Installation

As a light introduction, consider watching https://www.youtube.com/watch?v=vNRics7tlqw [Invidious]

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/ [Archive.org]

Setting up a safe Browser

See Safe Browser on the Host OS

Enable some additional privacy settings on your Host OS

See Windows Additional Privacy Settings

WINDOWS HOST OS ENCRYPTION

If you intend to use system-wide plausible deniability

Veracrypt is the software we 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 is a large file that you will be able to mount within Veracrypt as if it were an external drive to store encrypted files within).
- File container with plausible deniability (it is 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 encryption software that also provides plausible deniability for widespread use and it works with Windows Home Edition.

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

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

- Encrypt the memory with a Veracrypt option (settings > performance/driver options > encrypt RAM) at a cost of 5-15% performance. This setting will also disable hibernation (which does not actively clear the key when hibernating) and instead encrypt the memory altogether to mitigate some cold-boot attacks. More details about this feature here: https://sourceforge.net/p/veracrypt/discussion/technical/thread/3961542951/ [Archive.org]
- 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 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.

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 if your memory has enough time to decay.

More details later in [Route A and B: Simple Encryption using Veracrypt (Windows tutorial)].

If you do not intend to use system-wide plausible deniability

For this case, we will recommend the use of BitLocker instead of Veracrypt for the full disk encryption. The reasoning is that BitLocker does not offer a plausible deniability possibility contrary to Veracrypt. A hard adversary has then no incentive in pursuing his "enhanced" interrogation if you reveal the passphrase.

Normally, you should have installed Windows Pro in this case and the BitLocker setup is quite straightforward.

Basically, you can follow the instructions here: https://support.microsoft.com/en-us/windows/turn-on-device-encryption-0c453637-bc88-5f74-5105-741561aae838 [Archive.org]

But here are the steps:

- Click the Windows Menu
- Type "Bitlocker"
- Click "Manage Bitlocker"
- Click "Turn on Bitlocker" on your System Drive
- Follow the instructions
    - **Do not save your recovery key to a Microsoft Account if prompted.**
    - **Only save the recovery key to an external encrypted drive. To bypass this, print the recovery key using the Microsoft Print to PDF printer and save the key within the Documents folder. Delete that file later.**
    - **Encrypt Entire Drive (do not encrypt the used disk space only).**
    - **Use "New Encryption Mode"**
    - **Run the BitLocker Check**
    - **Reboot**
- Encryption should now be started in the background (you can check by clicking the Bitlocker icon on the lower right side of the taskbar).

Unfortunately, this is not enough. With this setup, your Bitlocker key can just be stored as-is in the TPM chip of your computer. This is rather problematic as the key can be extracted in some cases with ease'''.

To mitigate this, you will have to enable a few more options as per the recommendations of Microsoft:

- Click the Windows icon
- Type Run
- Type "gpedit.msc" (this is the group policy editor)
- Navigate to Computer Configuration > Administrative Templates > Windows Components > BitLocker > Operating System Drives
    - Double Click the "Require Additional Authentication at Startup"
        - Click the "Configure TPM Startup PIN" and set it to "Require Startup PIN with TPM"
    - Double Click the "Allow enhanced PINs for startup"
        - Click the "Enable" (this will allow us to set a password rather than a PIN)
- Close the Group Policy Editor
- Click the Windows icon
- Type Command to display the "Command Prompt"
- Right Click on it and click "Run as Administrator"
- Run `manage-bde -protectors -delete c:` (this will delete current protection: the recovery key you will not need)
- Run `manage-bde -protectors -add c: -TPMAndPIN` (this will prompt you for a pre-boot password)
    - Enter a password or passphrase of your choice (a good one)
- Run `manage-bde -status`
    - You should now see at your C: drive below "Key Protectors" the option "TPM and PIN"
- You are done

Now when you reboot your computer, you should ideally be prompted for:

- A BIOS/UEFI boot password
- An SSD/HDD unlock password (if the feature is available on your BIOS)

- A Bitlocker Pre-Boot Pin Screen where you need to enter the password/passphrase you just set-up
- And finally, the Windows Logon Screen where you can enter the credentials you set-up earlier

ENABLE HIBERNATION (OPTIONAL)

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

(Note that you cannot enable hibernation if you previously enabled RAM encryption within Veracrypt)

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 10/11 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 [Archive.org]

- 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 would like
- Disable all the Wake timers

Deciding which sub-route you will take

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

- Route A: Simple encryption of your current OS
  - Pros:
    - Does not 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 an adversary to reveal your password and all your secrets and will have no plausible deniability.
  - The danger of Online data leaks

- Route B: Simple encryption of your current OS with later use of plausible deniability on files themselves:
    - Pros:
        - Does not require you to wipe your laptop
        - Works fine with an SSD drive
        - Works with any OS
        - Plausible deniability is possible with "soft" adversaries
    - Cons:
        - The danger of Online Data leaks
        - The 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 will have a "hidden OS" and a "decoy OS" running on the laptop):
    - Pros:
        - No issues with local Data leaks
        - Plausible deniability is possible with "soft" adversaries
    - Cons:
        - Requires Windows (this feature is not "easily" supported on Linux).
        - The danger of online Data leaks
        - Requires full wipe of your laptop
        - No use with an SSD drive due to the requirement of disabling Trim Operations. This will severely degrade the performance/health of your SSD drive over time.

As you can see, Route C 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 [Wikiless] [Archive.org].

Deciding which route you will 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 using your laptop's keyboard (AZERTY for example) as you will have set up 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 of 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 USING VERACRYPT (WINDOWS TUTORIAL)

Skip this step if you used BitLocker instead earlier.

You do not have to have an HDD for this method, and you do not 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 will not 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. Here are the steps:

- 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, remember Guidelines for passwords and passphrases)
- Collect some entropy by randomly moving your cursor around until the bar is full
- Click Next as the Generated Keys screen
- To rescue disk or not rescue disk, well that is up to you. We 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 will 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 but we would recommend one pass (random data) just to be sure.
    - If you have sensitive data on an HDD, there is no Trim, and we Swould recommend at least 1-pass.
- Test your setup. Veracrypt will now reboot your system to test the bootloader before encryption. This test must pass 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 are done, skip Route B and go to 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)

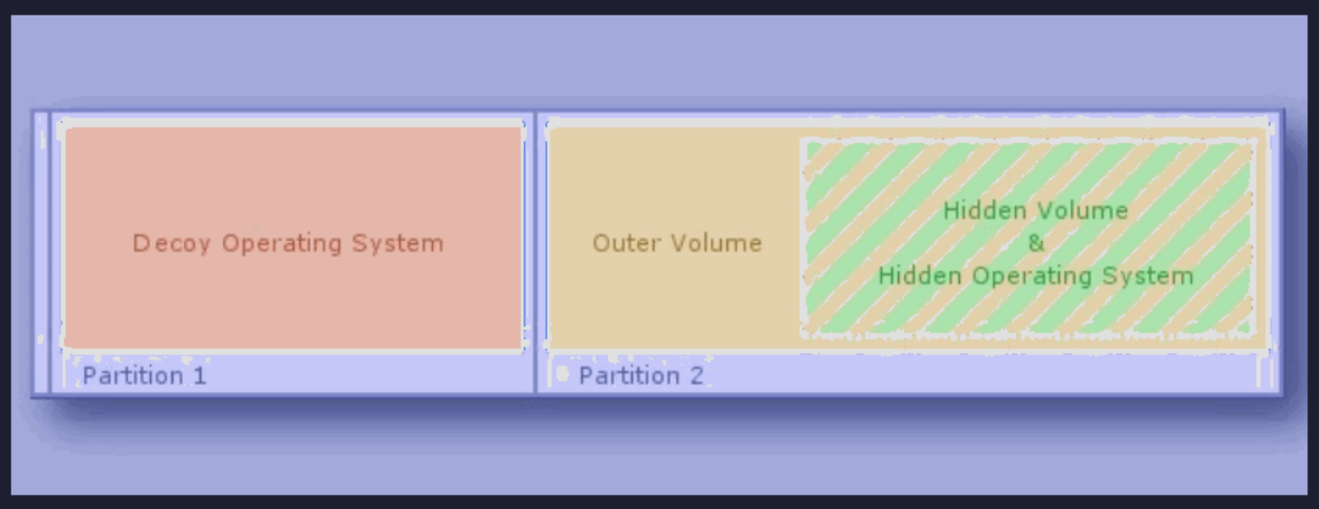
This is only supported on Windows.

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

Your Hidden OS should not be activated (with an 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 [Archive.org] (Process of Creation of Hidden Operating System part) and https://www.veracrypt.fr/en/Security%20Requirements%20for%20Hidden%20Volumes.html [Archive.org] (Security Requirements and Precautions Pertaining to Hidden Volumes).

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



(Illustration from Veracrypt Documentation, https://veracrypt.fr/en/VeraCrypt%20Hidden%20Operating%20System.html [Archive.org])

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 10/11 installation onto the hidden volume.
- Wipe your currently running Windows 10/11.
- This means that your current Windows 10/11 will become the hidden Windows 10/11 and that you will need to reinstall a fresh decoy Windows 10/11 OS.

Mandatory if you have an SSD drive and you still want to do this against the recommendation: Disable SSD Trim in Windows (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 must do it to prevent data leaks that could allow forensics to defeat your plausible deniability. The only way around this at the moment is to have a laptop with a classic HDD drive instead.

Step 1: Create a Windows 10/11 install USB key

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

Step 2: Boot the USB key and start the Windows 10/11 install process (Hidden OS)

- Insert the USB key into your laptop
- See Windows Installation and proceed with installing Windows 10/11 Home.

Step 3: Privacy Settings (Hidden OS)

See 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 [Archive.org]

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

- Install Veracrypt
- Start Veracrypt
- Go into Settings:
    - Settings > Performance/driver options > Encrypt RAM (**note that this option is not compatible with Hibernation your laptop and means you will have to shut down completely)**
    - 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 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 (remember Guidelines for passwords and passphrases)
- 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 you 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.
    - We recommend you do not fill the outer volume too much or too little (about 40%). Remember you must leave enough space for the Hidden OS (which will be the same size as the first partition you created during installation).
- Use a strong passphrase 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 the next step

Step 5: Reboot and boot the USB key and start the Windows 10/11 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 Windows Installation and proceed with installing Windows 10/11 Home again (do not install a different version and stick with Home).

Step 6: Privacy settings (Decoy OS)

See Windows Additional Privacy Settings

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

Now you 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 or not rescue disk, well that is up to you. We 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 will 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 must pass 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 the 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 [Archive.org]

You should do this from a safe, trusted space.

Basically, you are 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:

- 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...)

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. You could still be vulnerable to a Cold-Boot Attack. To mitigate this, be sure to have the option to encrypt your RAM as instructed before.

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

We must make the Decoy OS as plausible as possible. We also want your adversary to underestimate your intelligence.

It is 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 useful 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 on 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 or move files around.
- Copy some content from your Outer Volume to your Decoy OS and then delete it unsafely. Just put it in the Recycle Bin, which only someone who is naive would do, thinking it were deleted.
- 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
- Any Veracrypt volumes
- Any documents on anonymity or security

The intention is to make your adversary believe you are not as smart as they thought, to deter them from searching deeper.

Notes

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

- **You are using Veracrypt because you are using Windows 10/11 Home, which do not feature Bitlocker, but you still wanted reasonable Privacy.**
- **You have two partitions because you wanted to separate the system from the data for easy organization, and because some geeky friend told you this was better for performance.**
- **You have used a weak password for easy convenient booting of the system and a strong, long passphrase on the Outer Volume. 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 do not want anyone in your group/domain to see your stuff. You did not want that data available to anyone.**

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 [Archive.org]

Be careful:

- **You should never mount the Hidden Volume from the Decoy OS (NEVER EVER). If you did this, it would create forensic evidence of the Hidden Volume within the Decoy OS which 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 forensic evidence that will be explained later at the end of this guide, so this mistake alone isn't a huge deal if you follow the steps in Some additional measures against forensics.
- **Never 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. 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, this will be done later from a VM within the Hidden OS. The Hidden OS is only meant to protect you from soft lawful adversaries 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

Remember Virtualization.

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, you will make extensive use of the free Oracle Virtualbox 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).

So, you should be aware that Virtualbox is not the virtualization software with the best track record in terms of security. Some of the reported issues have not been completely fixed to date. If you are using Linux, and you possess a bit more technical skill, you should consider using KVM instead by following the guide available at Whonix here https://www.whonix.org/wiki/KVM [Archive.org] and here [Archive.org]

Some steps should be taken in all cases:

All your sensitive activities will be done from within a guest Virtual Machine running Windows 10/11 Pro (not Home this time), Linux, or macOS.

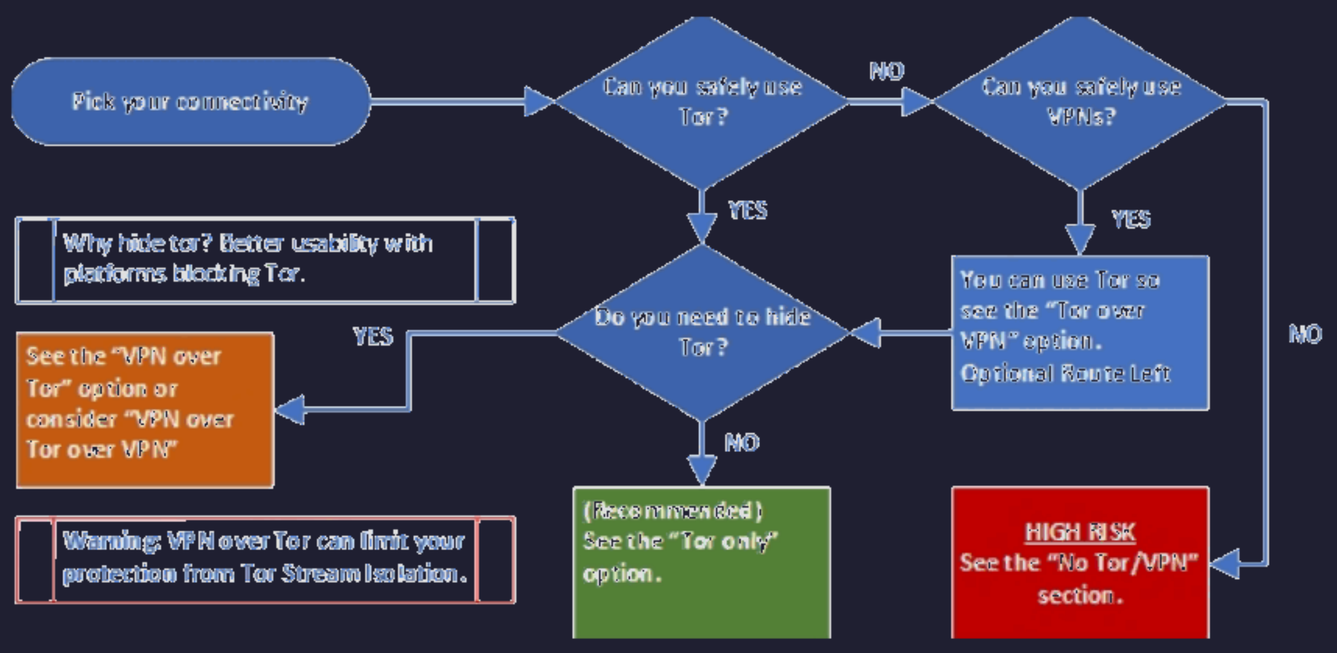
This has a few advantages that will help you remain anonymous:

- 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, the malware should not be able to access the Host OS and compromise your actual machine.
- It will allow us to force all the network traffic from your VM to run through another Gateway VM that will direct all the traffic over the Tor Network. This is a network "kill switch". Your VM will lose its network connectivity completely and go offline if the target network VM loses its connection to the Tor Network.
- The VM itself, which 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 must go through Tor no matter what.

Pick your connectivity method

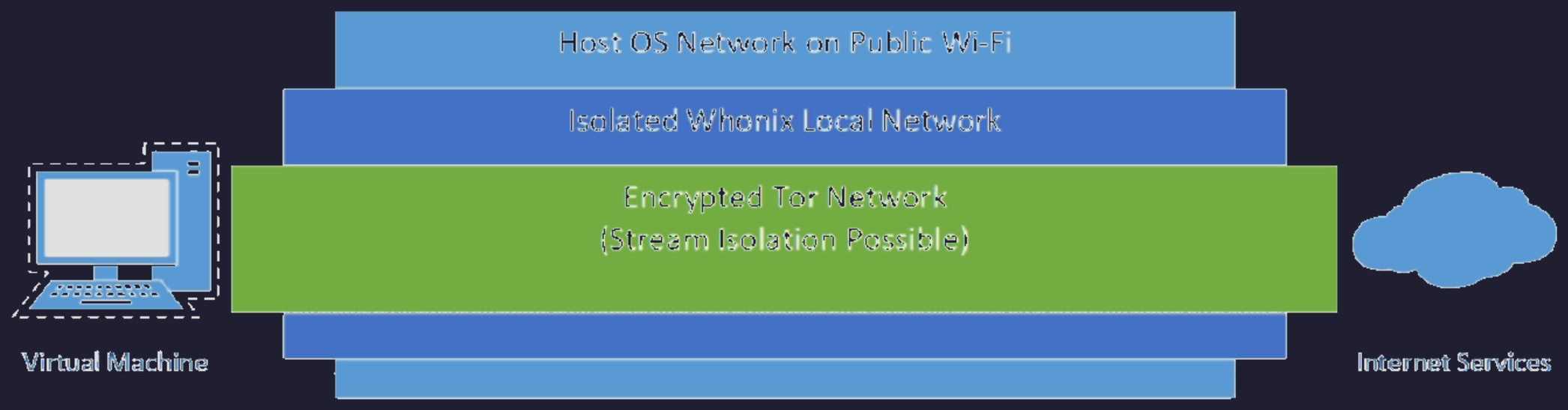
There are seven possibilities within this route:

- **Recommended and preferred:**
  - **Use Tor alone (User > Tor > Internet)**
  - **Use VPN over Tor (User > Tor > VPN > Internet) in specific cases**
  - **Use a VPS with a self-hosted VPN/Proxy over Tor (User > Tor > Self-Hosted VPN/Proxy > Internet) in specific cases**
- Possible if required by context:
  - Use VPN over Tor over VPN (User > VPN > Tor > VPN > Internet)
  - 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)



Tor only

This is the preferred and most recommended solution.



With this solution, all your network goes through Tor, and it should be sufficient to guarantee your anonymity in most cases.

There is one main drawback tho: **Some services block/ban Tor Exit nodes outright and will not allow account creations from those.**

To mitigate this, you might have to consider the next option: VPN over Tor but consider some risks associated with it explained in the next section.

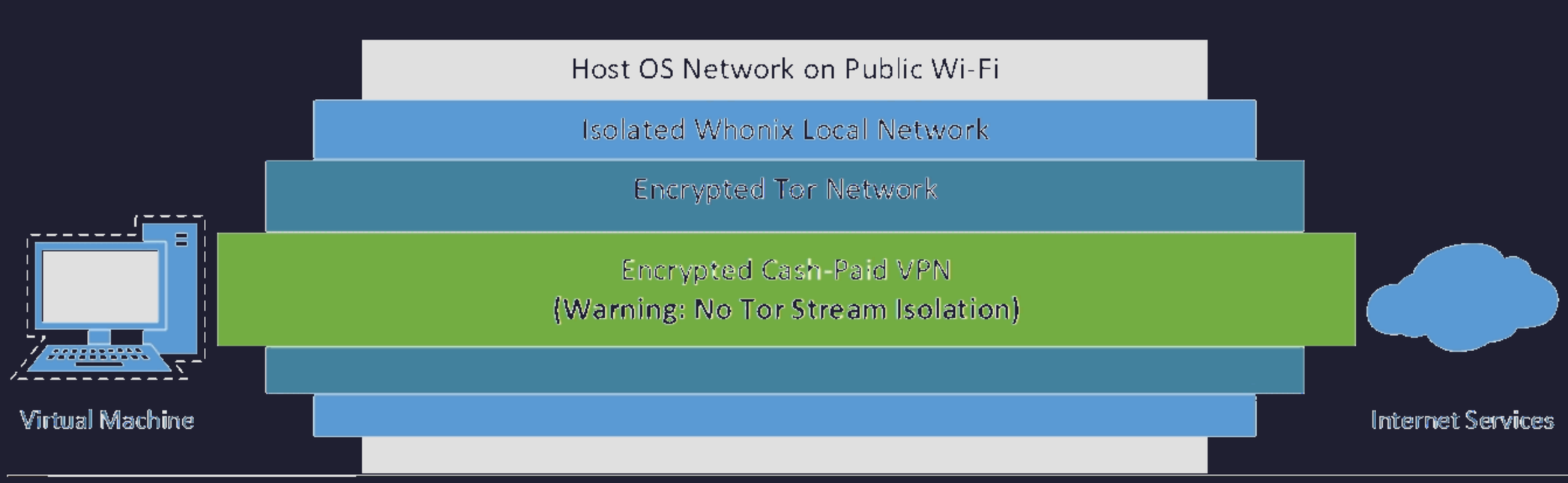
VPN/Proxy over Tor

This solution can bring some benefits in some specific cases vs using Tor only where accessing the destination service would be impossible from a Tor Exit node. This is because many services will just outright ban, hinder, or block Tor Exit Nodes (see https://gitlab.torproject.org/legacy/trac/-/wikis/org/doc/ListOfServicesBlockingTor [Archive.org]).

This solution can be achieved in two ways:

- Paid VPN over Tor (easiest)
- Paid Self-Hosted VPS configured as VPN/Proxy (most efficient in avoiding online obstacles such as captchas but requiring more skills with Linux)

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

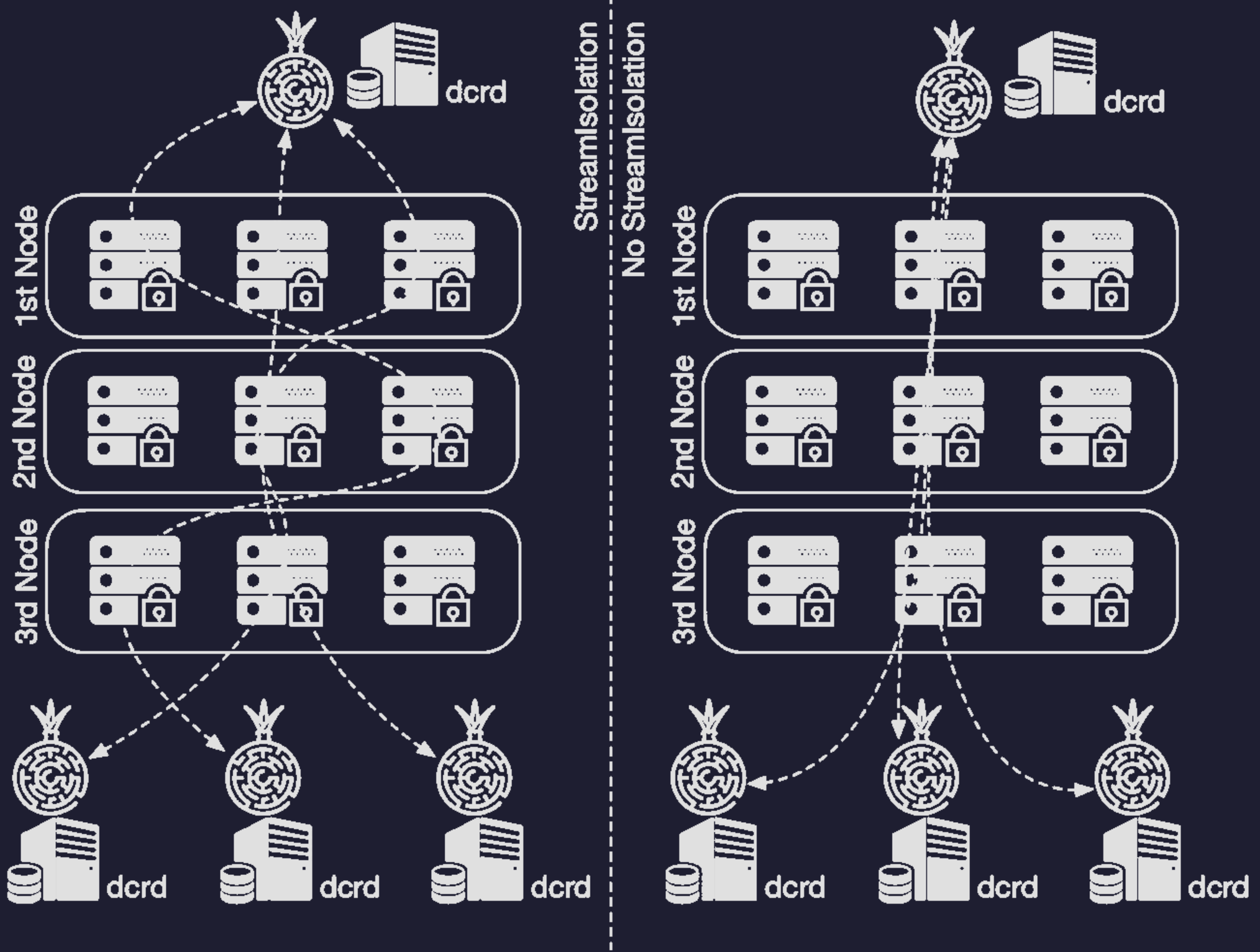


If an adversary somehow manages to compromise the Tor network too, they will only reveal the IP of a random public Wi-Fi that is not tied to your identity.

If an adversary somehow compromises your VM OS (with malware or an 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 however has one main drawback to consider: Interference with Tor Stream Isolation.

Stream isolation is a mitigation technique used to prevent some correlation attacks by having different Tor Circuits for each application. Here is an illustration to show what stream isolation is:



(Illustration from Marcelo Martins, https://stakey.club/en/decred-via-tor-network/ [Archive.org])

VPN/Proxy over Tor falls on the right-side meaning using a VPN/Proxy over Tor forces Tor to use one circuit for all activities instead of multiple circuits for each. This means that using a VPN/Proxy over Tor can reduce the effectiveness of Tor in some cases and should therefore be used only for some specific cases:

- When your destination service does not allow Tor Exit nodes.
- When you do not mind using a shared Tor circuit for various services. For instance, when using various authenticated services.

You should however consider not using this method when your aim is just to browse random various unauthenticated websites as you will not benefit from Stream Isolation and this could make correlation attacks easier over time for an adversary between each of your sessions (see Your Anonymized Tor/VPN traffic). If your goal however is to use the same identity at each session on the same authenticated services, the value of Stream isolation is lessened as you can be correlated through other means.

You should also know that Stream Isolation is not necessarily configured by default on Whonix Workstation. It is only pre-configured for some applications (including Tor Browser).

Also, note that Stream Isolation does not necessarily change all the nodes in your Tor circuit. It can sometimes only change one or two. In many cases, Stream Isolation (for instance within the Tor Browser) will only change the relay (middle) node and the exit node while keeping the same guard (entry) node.

More information at:

- https://www.whonix.org/wiki/Stream_Isolation [Archive.org]
- https://tails.boum.org/contribute/design/stream_isolation/ [Archive.org]
- https://www.whonix.org/wiki/Tunnels/Introduction#Comparison_Table [Archive.org]

Tor over VPN

You might be wondering: Well, what about using Tor over VPN instead of VPN over Tor? Well, we would not necessarily recommend it:

- Disadvantages:
  - 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 do not trust them. We prefer a situation where your VPN provider does not know who you are. It does not add much in terms of anonymity.
  - This would result in you connecting to various services using the IP of a Tor Exit Node which is banned/flagged in many places. It does not help in terms of convenience.
- Advantages:
  - **The main advantage is that if you are in a hostile environment where Tor access is impossible/dangerous/suspicious, but VPN is okay.**
  - This method also does not break Tor Stream isolation.
  - This also hides your Tor activities from your main ISP.

Note, if you are having issues accessing the Tor Network due to blocking/censorship, you could try using Tor Bridges. See Using Tor bridges in hostile environments.

It is also possible to consider **VPN over Tor over VPN (User > VPN > Tor > VPN > Internet)** using two cash/Monero paid VPNs instead. This means that you will 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 [Archive.org]).

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

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

Conclusion: Only do this if you think using Tor alone is risky/impossible but VPNs are okay. Or just because you can and so why not. This method will not lower your security/privacy/anonymity.

VPN only

This route will not be explained nor recommended.

If you can 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 is 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 is an added layer ... but it is 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 [Archive.org]
- https://www.whonix.org/wiki/Why_does_Whonix_use_Tor [Archive.org]
- https://www.researchgate.net/publication/324251041_Anonymity_communication_VPN_and_Tor_a_comparative_study [Archive.org]
- https://gist.github.com/joepie91/5a9909939e6ce7d09e29#file-vpn-md [Archive.org]
- https://schub.wtf/blog/2019/04/08/very-precarious-narrative.html [Archive.org]

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 cannot use VPN nor Tor where you are, you probably are in a very hostile environment where surveillance and control are extremely high.

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

Do not forget to check back on Adversaries (threats) and Check your network for surveillance/censorship using OONI.

If you have absolutely no other option and still want to do something, see What about when Tor and VPNs aren't possible? **(at your own risk) and consider the Tails route instead.**

Connection Type	Anonymity	Ease of Access to online resources	Tor Stream isolation	Safer where Tor is suspicious/dangerous	Speed	Cost	Recommended
Tor Alone	**Good**	**Medium**	**Possible**	**No**	**Medium**	**Free**	**Yes**
Tor over VPN	**Good+**	**Medium**	**Possible**	**Yes**	**Medium**	**Around 50€/y**	**If needed (Tor inaccessible)**
Tor over VPN over Tor	**Best**	**Medium**	**Possible**	**Yes**	**Poor**	**Around 50€/y**	**Yes**
VPN over Tor	**Good-**	**Good**	**No**	**No**	**Medium**	**Around 50€/y**	**If needed (convenience)**
Self-Hosted VPS VPN/Proxy over Tor	**Good-**	**Very Good**	**No**	**Yes**	**Medium**	**Around 50€/y**	**If needed (convenience)**
VPN/Proxy over Tor over VPN	**Good-**	**Good**	**No**	**Yes**	**Poor**	**Around 100€/y**	**If needed (convenience and Tor inaccessible)**
VPN/Proxy Alone	**Bad**	**Good**	**N/A**	**Yes**	**Good**	**Around 50€/y**	**No.**
No Tor and VPN	**Bad**	**Unknown**	**N/A**	**No**	**Good**	**Around 100€ (Antenna)**	**No.**

Unfortunately, using Tor alone will raise the suspicion of many destinations' platforms. You will face many hurdles (captchas, errors, difficulties signing up) if you only use Tor. In addition, using Tor where you are could put you in trouble just for that. But Tor is still the best solution for anonymity and must be somewhere for anonymity.

- If you intend to create persistent shared and authenticated identities on various services where access from Tor is hard, we recommend the **VPN over Tor** and **VPS VPN/Proxy over Tor** options (or VPN over Tor over VPN if needed). It might be a bit less secure against correlation attacks due to breaking Tor Stream isolation but provides much better convenience in accessing online resources than just using Tor. It is an "acceptable" trade-off IMHP if you are careful enough with your identity.
    - **Note: It is becoming more common that mainstream services and CDNS are also blocking or hindering VPN users with captchas and other various obstacles. In that case, a self-hosted VPS with a VPN/Proxy over Tor is the best solution for this as having your own dedicated VPS guarantees you are the sole user of your IP and encounter little to no obstacles.** Consider a Self-hosted VPN/Proxy on a Monero/Cash-paid VPS (for users more familiar with Linux) if you want the least amount of issues (this will be explained in the next section in more details).
- If your intent however is just to browse random services anonymously without creating specific shared identities, using tor friendly services; or if you do not want to accept that trade-off in the earlier option. **Then we recommend using the Tor Only route to keep the full benefits of Stream Isolation (or Tor over VPN if you need to).**
- If cost is an issue, we recommend the Tor Only option if possible.
- If both Tor and VPN access are impossible or dangerous then you have no choice but to rely on Public wi-fi safely. See What about when Tor and VPNs aren't possible?

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

- Tor Project: https://gitlab.torproject.org/legacy/trac/-/wikis/doc/TorPlusVPN [Archive.org]
- Tails Documentation:
    - https://gitlab.tails.boum.org/tails/blueprints/-/wikis/vpn_support/ [Archive.org]
    - https://tails.boum.org/support/faq/index.en.html#index20h2 [Archive.org]

- Whonix Documentation (in this order):
  - https://www.whonix.org/wiki/Tunnels/Introduction [Archive.org]
  - https://www.whonix.org/wiki/Tunnels/Connecting_to_Tor_before_a_VPN [Archive.org]
  - https://www.whonix.org/wiki/Tunnels/Connecting_to_a_VPN_before_Tor [Archive.org]
- Some papers on the matter:
  - https://www.researchgate.net/publication/324251041_Anonymity_communication_VPN_and_Tor_a_comparative_study [Archive.org]

Getting an anonymous VPN/Proxy

Skip this step if you want to use Tor only.

See Getting an anonymous VPN/Proxy

Whonix

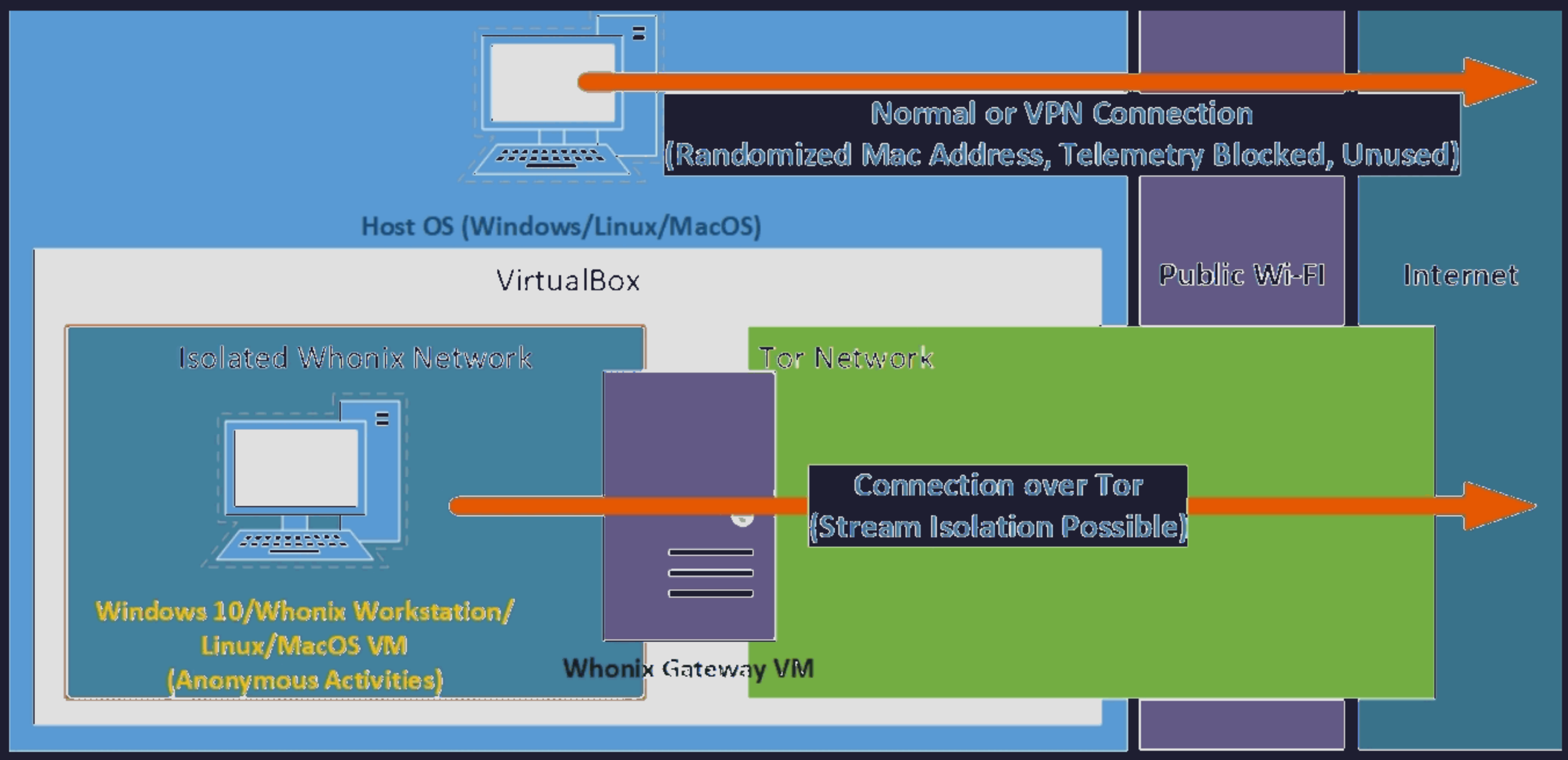
Skip this step if you cannot use Tor.

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

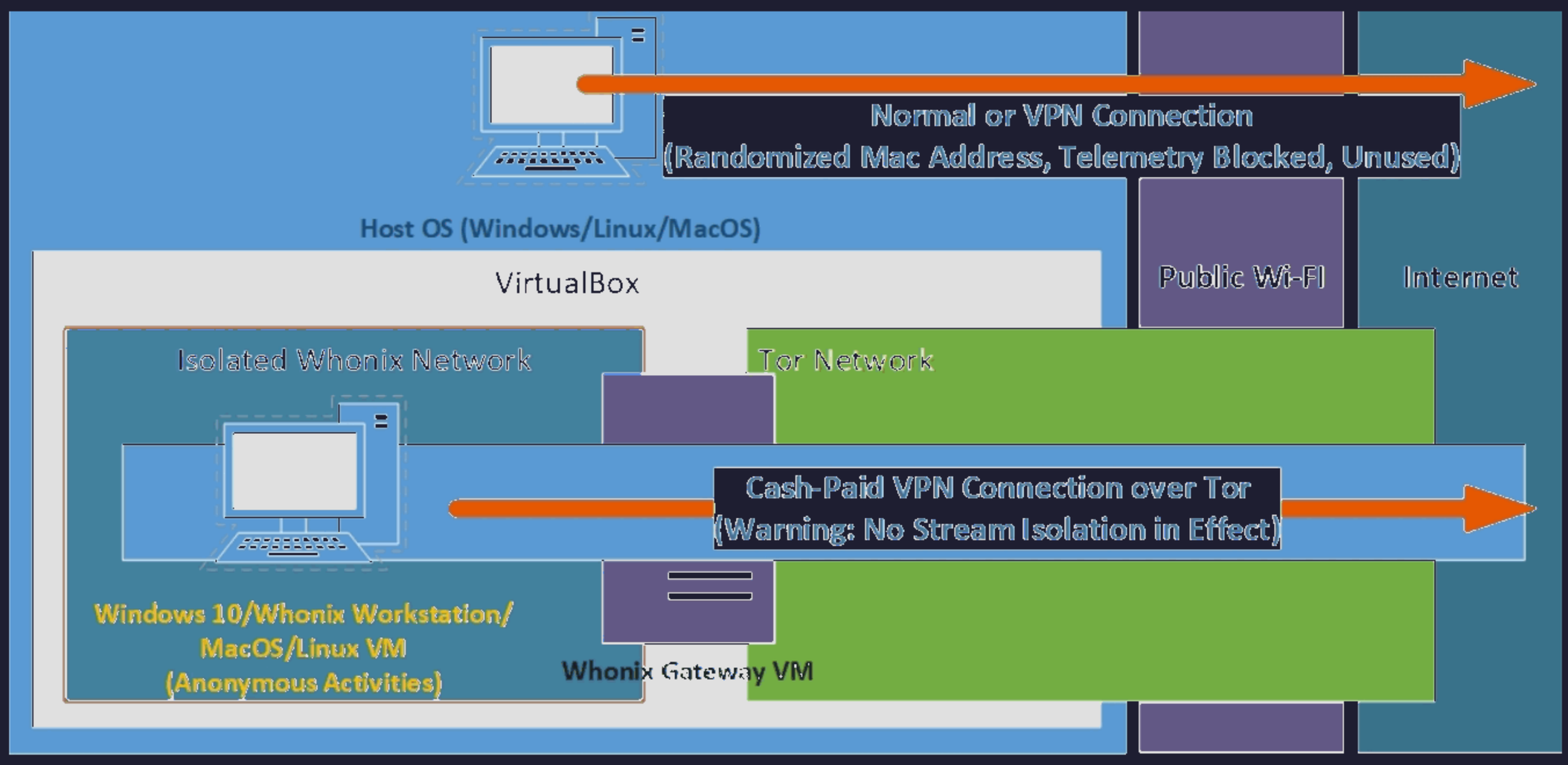
- The Whonix Workstation (this is a VM where you can conduct sensitive activities)
- The Whonix Gateway (this VM will establish a connection to the Tor network and route all the network traffic from the Workstation through the Tor network).

This guide will therefore propose two 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. We recommend the second one as explained before.

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

A note on Virtualbox Snapshots

Later, you will create and run several Virtual Machines within Virtualbox for your sensitive activities. Virtualbox provides a feature called "Snapshots" 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 its use for any sensitive/anonymous activity.

This will allow you to turn your VMs into a kind of 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 will not 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. Fortunately, there will be ways to remove those traces after the deletion or reverting to an earlier 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 [Archive.org]
- (Skip this if you cannot use Tor natively or through a VPN) The latest stable **Whonix 18.x** OVA from:
  - GitHub Releases [Archive.org]
  - Or the official download mirror: https://download.whonix.org/ (XFCE Desktop recommended for beginners)
- (Optional) The Whonix Text Client [Archive.org] is available for advanced users who prefer a minimal setup.

Note on Whonix 18.x: As of guide v1.2.5, Whonix has been upgraded from version 17 to 18 (Whonix 18.1.4.x). See [Upgrading to Whonix 18] for the upgrade path if you currently use Whonix 17.x.

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 (Whonix 17.x & 18.x)

For ideal security, follow the Whonix Hardening Guide [Archive.org] with these additional notes:

Recommended Settings (apply to all VMs): - Disable Audio - Do not enable Shared Folders - Disable 2D acceleration: `VBoxManage modifyvm "vm-name" --accelerate2dvideo off` - Disable 3D acceleration - Disable Serial Port - Remove Floppy and CD/DVD drives (or set to non-functional) - Disable Remote Display server - Enable PAE/NX for security features - Disable ACPI

Network Time Desync: Desync VM clocks to prevent synchronization attacks (see [Whonix Documentation - Network Time Synchronization] [Archive.org]):

```sh
# Example offsets (choose different values for each VM)
VBoxManage modifyvm "Whonix-Gateway-XFCE" --biossystemtimeoffset -35017
VBoxManage modifyvm "Whonix-Workstation-XFCE" --biossystemtimeoffset +27931
```

Spectre/Meltdown Mitigations: (optional, may impact performance) For detailed Spectre/Meltdown hardening, see https://www.whonix.org/wiki/Spectre_Meltdown [Archive.org].

Additional Security: (Whonix 18.x+) - Consider enabling AppArmor on Whonix Workstation: https://www.whonix.org/wiki/AppArmor [Archive.org] - Review DoNot [Archive.org] recommendations

- 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 are your VM compared to your host OS
https://www.whonix.org/wiki/Network_Time_Synchronization#Spoof_the_Initial_Virtual_Hardware_Clock_Offset [Archive.org]

This offset should be within a 60000-millisecond 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/Meltdown vulnerabilities by running this command from the VirtualBox Program Directory. All of these are described here: https://www.whonix.org/wiki/Spectre_Meltdown [Archive.org] (be aware these can impact severely the performance of your VMs but should be done for best security).

Finally, consider the security advice from Virtualbox themselves here https://www.virtualbox.org/manual/ch13.html [Archive.org]

Tor over VPN

Skip this step if you do not intend to use Tor over VPN and only intend to use Tor or cannot.

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

Remember that in this case, we recommend having two VPN accounts. Both paid with cash/Monero (see Getting an anonymous VPN/Proxy). 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 Installing a VPN on your VM or Host OS for instructions.

Whonix Virtual Machines

Skip this step if you cannot use Tor.

- Start Virtualbox on your Host OS.
- Import Whonix OVA into Virtualbox: See VirtualBox/XFCE [Archive.org]
  - You can also use the Text Client version for advanced users
- Start the Whonix VMs

Remember at this stage that if you are having issues connecting to Tor due to censorship or blocking, you should consider connecting using Bridges as explained in Bridges [Archive.org].

- Update the Whonix VMs by following the instructions on Operating System Software and Updates [Archive.org]

Note: For Qubes OS users, see the [Upgrading to Whonix 18] section for migration notes.

- 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 the next step

Important Note: You should also read these very good recommendations over there https://www.whonix.org/wiki/DoNot [Archive.org] 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 [Archive.org] which will also provide tons of advice like this guide.

Whonix 18.x Improvements and Best Practices

Key Changes in Whonix 18.x: - Automated Release Upgrade process for seamless version transitions from Whonix 17.x - Improved AppArmor integration for enhanced application sandboxing on Debian-based systems - Enhanced network configuration with better Tor connectivity options - Updated security hardening guides with current best practices - Better Qubes OS integration and compatibility

Best Practices for Whonix 18.x Users:

1. **Always take a snapshot** before applying any system updates or configuration changes
2. **Verify system integrity** after upgrades using `sudo checkvm` commands
3. **Review the DoNot page** regularly to stay informed about security best practices
4. **Keep your Whonix VMs updated** using the automated upgrade process when available
5. **Consider KVM instead of VirtualBox**

For Qubes OS Users: Qubes R4.2+ with Whonix 18 is supported. Always verify system integrity after upgrade using `systemcheck`. For detailed migration instructions, see the [Upgrading to Whonix 18] section.

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 earlier 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 an anonymous VPN, if possible, to then connect to the Tor network. Or you could go with the risky route: See What about when Tor and VPNs aren't 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 is the safest and most secure way to go on this route.**

It is also the only VM that will provide Stream Isolation pre-configured for most apps by default.

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

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

Do not forget to apply the VM hardening recommendations here: [Virtualbox Hardening recommendations].

Consider using AppArmor on your Whonix Workstations by following this guide: https://www.whonix.org/wiki/AppArmor [Archive.org]

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 [Archive.org]

IF YOU CAN USE TOR (NATIVELY OR OVER A VPN)

Use the Linux Distro of your choice. We 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 [Archive.org] for detailed instructions.

Consider hardening the VM as recommended in [Hardening Linux].

IF YOU CANNOT USE TOR

Use the Linux Distro of your choice. We would recommend Ubuntu or Fedora for convenience but any other would work too. Be sure to not enable any telemetry. You could go with the risky route: See What about when Tor and VPNs aren't possible?

CHOOSE A BROWSER WITHIN THE VM

This time, we will recommend Brave browser.

See why here: What browser to use in your Guest VM/Disposable VM

See Hardening your Browsers as well.

Windows 10/11 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 [Archive.org]

Windows 10 and 11 ISO download

Go with the Official Windows 10/11 Pro VM and harden it yourself: see 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 [Archive.org] for detailed instructions.

INSTALL

- Shut down 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 or Windows 11 64bit
- Allocate a minimum amount of 2GB for Windows 10 and 4GB for Windows 11
- Create a Virtual Disk using the VDI format and select Dynamically Allocated
- Keep the disk size at 50GB for Windows 10 and 80GB for Windows 11 (this is a maximum; it should not reach that much)
- Make sure PAE/NX is enabled in System > Processor
- 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 either push a button to boot the ISO or ask you what to boot, select the ISO or click.
- Follow the steps in Windows Installation](#windows-installation)
- Start the Whonix Gateway VM

NETWORK SETTINGS

- Back to your Windows
- Windows 10: Go back into Settings then Network & Internet. Windows 11: Go into settings, click the upper left menu and pick "Network and Internet"
- Windows 10: Click Properties (Below Ethernet). Windows 11: Click Ethernet
- Windows 10: Edit IP settings. Windows 11: Edit IP assignment.
- Windows 10: Enable IPv4 and set the following, Windows 11: Switch from DHCP to Manual and set the following:
  - IP address `10.152.152.50` (increase this IP by one for any other VM)
  - Subnet prefix length `18` (`255.255.192.0`)
  - Gateway `10.152.152.10` (this is the Whonix Gateway)
  - (Windows 10) DNS `10.152.152.10` (this is again the Whonix Gateway)

    - (Windows 11) exit the IP assignment and select DNS server assignment and set it to `10.152.152.10` (this is again the Whonix Gateway)
    - Save
- Windows might prompt you if you want to be "discoverable" on this network. Click NO. Always stay on a "public network" if prompted.

Every time you will power on this VM in the future, you should make sure to 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.

If you cannot use Tor

See What about when Tor and VPNs aren't possible?

INSTALL

- Open Virtualbox
- Select Machine > New > Select Windows 10 or 11 64bit
- Allocate a minimum amount of 4GB of RAM for 11 , 2GB of RAM for 10.
- Create a Virtual Disk using the VDI format and select Dynamically Allocated
- In the System/Processor tab, make sure PAE/NX is enabled.
- Keep the disk size at 80GB for 11, 50GB for 10 (this is a maximum; it should not 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 either push a button to boot the ISO or ask you what to boot, select the ISO or click.
- Follow the steps in 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, you should make sure to 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, we will recommend Brave browser.

See why here: What browser to use in your Guest VM/Disposable VM

See Hardening your Browsers as well.

Additional Privacy settings in Windows 10/11

See Windows Additional Privacy Settings

Android Virtual Machine

Because sometimes you want to run mobile Apps anonymously too. You can also set up an Android VM for this purpose. As in other cases, ideally, this VM will also be sitting behind the Whonix Gateway for Tor network connectivity. But this can also be set up as VPN over Tor over VPN

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

Later in the VM settings during creation, go into Network and select Internal Network, Whonix.

Then on Android itself:

- Select Wi-Fi
- Select VirtWifi to connect
- Go into the advanced Wi-Fi properties

- Switch from DHCP to Static
  - IP address `10.152.152.50` (increase this IP by one 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 and see What about when Tor and VPNs aren't possible?

Installation

Two possibilities: AnBox or Android-x86

Personally, We would recommend AnBox over Android-x86 but it requires Linux

ANBOX

Basically follow the tutorial here for installing AnBox on the Whonix Workstation: https://www.whonix.org/wiki/Anbox [Archive.org] for running Android Applications within an AnBox VM.

Or follow the instructions here https://anbox.io/ to install on any other VM **(Linux Only)**

ANDROID-X86

Basically, follow the tutorial here: https://www.android-x86.org/documentation/virtualbox.html [Archive.org]

- Download the ISO file of your choice
- Create a New VM.
- Select Linux and Linux 2.6 / 3.x / 4.x 64 Bit.
- In System:
  - Allocate at least 2048MB (2GB) memory
  - Uncheck the Floppy drive
  - In the Processor Tab, select at least 1 or more CPUs
  - Enable PAE/NX
- In Display Settings, Change the adapter to VBoxVGA
- In Audio Settings, Change to Intel HD Audio
- Start the VM
- Select Advanced if you want persistence, Live if you want a disposable Boot (and skip the next steps).
- Select Auto Install on Selected Hard Disk
- Select Run Android
- Set up as you wish (disable all prompts for data collections). **I recommend using the TaskBar Home.**
- Go into Settings, Android-x86 Options, and disable all collections.
- Connect to VirtWifi Wi-Fi Network **(see the above section if you are behind Whonix and want to use Tor)**

You are now done and can now install any Android app.

macOS Virtual Machine

Yes, you can actually run macOS within Virtualbox (on Windows/Linux/macOS host systems) if you want to use macOS. You can run any version of macOS you want.

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 will 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 one 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 and see What about when Tor and VPNs aren't possible?

Installation

- Windows Host OS:
  - Virtualbox Catalina Tutorial: https://www.wikigain.com/install-macos-catalina-on-virtualbox-on-windows/ [Archive.org]
  - Virtualbox Big Sur Tutorial: https://www.wikigain.com/how-to-install-macos-big-sur-on-virtualbox-on-windows-pc/ [Archive.org]
  - Virtualbox Monterey Tutorial: https://www.wikigain.com/install-macos-monterey-on-virtualbox/ [Archive.org]
- macOS Host OS:
  - Just use the same tutorials as above but execute the various commands in the terminal. It should work without issue.
- Linux Host OS:
  - Just use the same tutorials as above but execute the various commands in the terminal. It should work without issue.

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

Note: We also ran in multiple issues with running these on AMD processors. This can be fixed so here is the configuration Weused which worked fine with Catalina, Big Sur and Monterey 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`

Hardening macOS

Refer to [Hardening macOS].

Choose a browser within the VM

This time, we will recommend Brave browser.

See why here: What browser to use in your Guest VM/Disposable VM

See Hardening your Browsers as well.

KeepassXC

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

For this purpose, we strongly recommend KeePassXC because of its integrated TOTP feature. This is the ability to create entries for 2FA 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 [Archive.org]
- Linux:
  - Download from https://keepassxc.org/download/ [Archive.org]
  - Follow the tutorial here https://keepassxc.org/docs/KeePassXC_GettingStarted.html#_linux [Archive.org]
- Windows:
  - Download from https://keepassxc.org/download/ [Archive.org]
  - Follow the tutorial here https://KeePassXC.org/docs/KeePassXC_GettingStarted.html#_microsoft_windows/ [Archive.org]
- macOS:
  - Download from https://keepassxc.org/download/ [Archive.org]
  - Follow the tutorial here https://keepassxc.org/docs/KeePassXC_GettingStarted.html#_macos [Archive.org]

Test that KeePassXC is working before going to the 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 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.

About VPN Client Data Mining/Leaks

You might be asking yourself if those VPN clients are trustworthy not to leak any information about your local environment to the VPN provider when using them in the "VPN over Tor" context.

This is a valid concern but should be taken with a grain of salt.

Remember that all VPN activities are happening from a sandboxed VM on an internal network behind a Network Gateway (the Whonix Gateway). It does not matter much if the VPN client leaves some identifiers on your guest VM. The guest VM is still sandboxed and walled-off from the Host OS. The attack surface is small especially when using the reputable and recommended VPN providers within the guides (iVPN, Mullvad, Proton VPN, and maybe Safing.io).

At best, the VPN client would know your local IP (internal IP) and some randomized identifiers but should not be able to get anything from the Host OS. And in theory, the VPN client should not send any telemetry back to the VPN provider. If your VPN client does this or asks this, you should consider changing the provider.

(Optional) VM kill switch

This step will allow you to configure your Host OS so that only the Whonix Gateway VM will have access to the internet. This will therefore prevent any "leak" from your Host OS while letting the Whonix Gateway establish the tor connectivity. The other VMs (Whonix Workstation or any other VM you installed behind it will not be affected)

There are three ways to do this:

- The Lazy Way (not really recommended): not supported by Whonix and might have some security implications as you will expose the Whonix Gateway VM to the Public Wi-Fi network. We would recommend against this unless you are in a hurry or very lazy.
    - **This method will not work with Wi-Fi captive portals requiring any registration to connect.**
- The Better Way (see further down): still not supported by Whonix but it will not expose the Whonix Gateway VM to the Public Wi-Fi network. This should keep things in check in terms of security.
- The Best Way: Using an external USB Wi-Fi dongle and just disabling Wi-Fi on the Host OS/Computer.

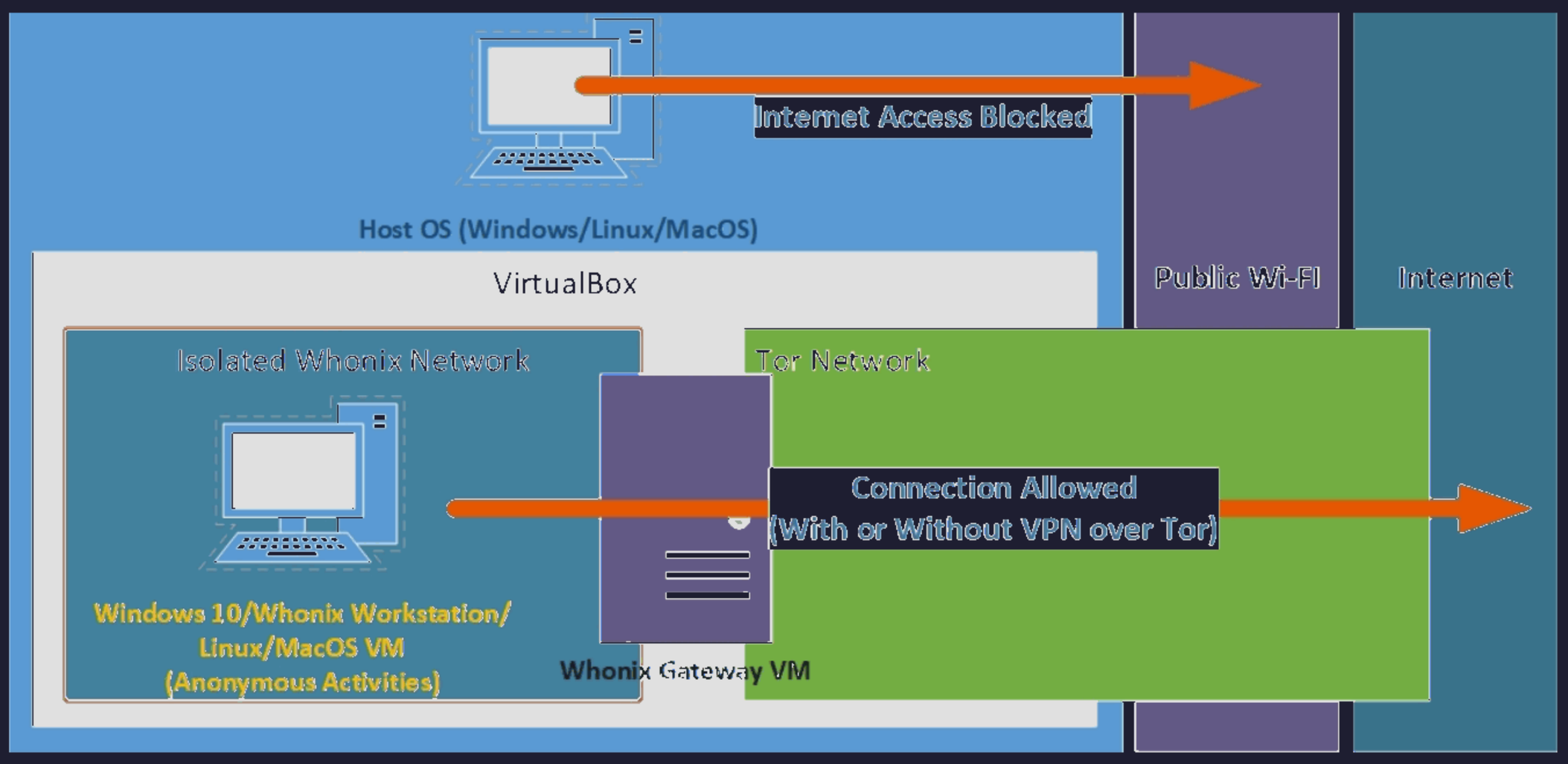
The Lazy Way

> Not supported by Whonix

This way is not supported by the Whonix project but I will go ahead and give this option anyway. This is helpful to prevent your Host OS from leaking any information while you are using the Whonix VMs.

Note that this option as-is will only work on Wi-Fis without a captive portal (where you must enter some information to unlock access).

The illustration below shows the result of this step:



CONFIGURATION OF THE WHONIX GATEWAY VM

For this to work, we will need to change some configurations on the Whonix Gateway VM. we will need to add a DHCP client to the Whonix Gateway to receive IP addresses from the network. To do those changes the Host OS will still have to have internet access allowed for now.

So here is how:

- Be sure to have your Host OS connected to a safe Wi-Fi.
- Through VirtualBox, start the Whonix Gateway VM
- Start a Terminal on the VM
- Install a DHCP client on the Whonix Gateway VM using the following command:
    - `sudo apt install dhcpcd5`
- Now edit the Whonix Gateway VM network configuration using the following command:
    - `sudo nano /etc/network/interfaces.d/30_non-qubes-whonix`
- Within the file change the following lines:
    - `# auto eth0` to `auto eth0`
    - `# iface eth0 inet dhcp` to `iface eth0 inet dhcp`
    - `iface eth0 inet static` to `# iface eth0 inet static`

    - `address 10.0.2.15` to `# address 10.0.2.15`
    - `netmask 255.255.255.0` to `# netmask 255.255.255.0`
    - `gateway 10.0.2.2` to `# gateway 10.0.2.2`
- Save (using Ctrl+X and confirm with Y) and power off the VM from the top left menu
- Go into the VirtualBox Application and select the Whonix Gateway VM
- Click Settings
- Click the Network Tab
- For Adapter 1, change the "Attached To" value from "NAT" to "Bridged Adapter"
- As "Name", select your Wi-Fi network Adapter
- Click OK and you are done with the VM configuration part

CONFIGURATION OF THE HOST OS

Now you must block internet access from your Host OS while still allowing the VM to connect. This will be done by connecting to Wi-Fi with the Host OS but without assigning itself an IP address. The VM will then use your Wi-fi association to get an IP address.

Windows Host OS

The goal here is to associate with a Wi-Fi network without having an internet connection. You will achieve this by deleting the Gateway from the connection after you are connected:

- First, connect to the safe Wi-Fi of your choice
- Open an administrative command prompt (right-click on Command Prompt and Run as Administrator)
- Run the following command: `route delete 0.0.0.0` (this deletes the Gateway from your IP configuration)
- You are done, your Host OS will now be unable to access the internet while still connected to the Wi-Fi
    - Note that this will reset at each disconnect/reconnection to a network, and you will have to delete the route again. This is not permanent.
- You can now start the Whonix Gateway VM which should now obtain an IP automatically from the Wi-Fi network and should provide Network to the other VMs behind (Whonix Workstation or other).
- And finally, after that, you can start the Whonix Workstation VM (or any other VM you configured to work behind the Whonix Gateway VM) and it should be connected to the internet through Tor.

Linux Host OS

The goal here is to associate with a Wi-Fi network without having an internet connection. You will achieve this by deleting the Gateway from the connection after you are connected:

- First, connect to the safe Wi-Fi of your choice
- Open a Terminal
- Run the following command: `sudo ip route del default` (this deletes the Gateway from your IP configuration)
- You are done, your Host OS will now be unable to access the internet while still connected to the Wi-Fi
    - Note that this will reset at each disconnect/reconnection to a network, and you will have to delete the route again. This is not permanent.
- You can now start the Whonix Gateway VM which should now obtain an IP automatically from the Wi-Fi network and should provide Network to the other VMs behind (Whonix Workstation or other).
- And finally, after that, you can start the Whonix Workstation VM (or any other VM you configured to work behind the Whonix Gateway VM) and it should be connected to the internet through Tor.

macOS Host OS

The goal here is to associate with a Wi-Fi network without having an internet connection. You will achieve this by deleting the Gateway from the connection after you are connected:

- First, connect to the safe Wi-Fi of your choice
- Open a Terminal

- Run the following command: `sudo route delete default` (this deletes the Gateway from your IP configuration)
- You are done, your Host OS will now be unable to access the internet while still connected to the Wi-Fi
  - Note that this will reset at each disconnect/reconnection to a network, and you will have to delete the route again. This is not permanent.
- You can now start the Whonix Gateway VM which should now obtain an IP automatically from the Wi-Fi network and should provide Network to the other VMs behind (Whonix Workstation or other).
- And finally, after that, you can start the Whonix Workstation VM (or any other VM you configured to work behind the Whonix Gateway VM) and it should be connected to the internet through Tor.

The Better Way (recommended)

This way will not go against Whonix recommendations (as it will not expose the Whonix Gateway to the Host OS) and will have the advantage of allowing connections not only to open Wi-Fis but also to the ones with a Captive Portal where you need to enter some information to access the internet.

Yet this will still not be supported by the Whonix project, but it is fine as the main concern for the earlier Lazy Way is to have the Whonix Gateway VM exposed to the Host Network, and it will not be the case here.

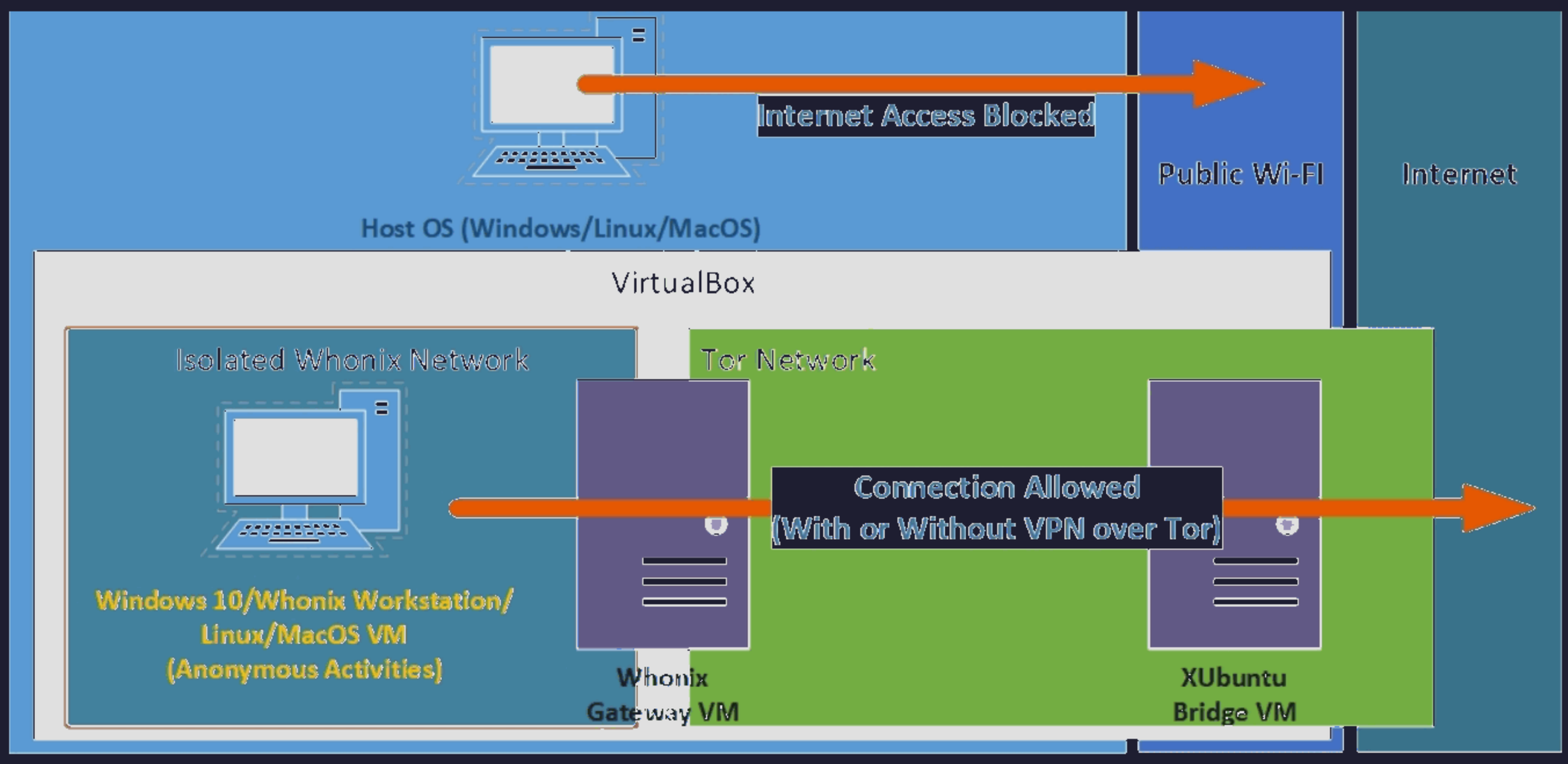
This option will require an additional VM between the Host OS and the Whonix Gateway to act as a Network Bridge.

For this purpose, I will recommend the use of a lightweight Linux Distro. Any will do but the easiest will be an Ubuntu-based distro and I would recommend the lightweight XUbuntu as it will be extremely easy to configure this setup.

Why XUbuntu and not Ubuntu or KUbuntu? Because XUbuntu uses an XFCE desktop environment which is lightweight and this VM will only serve as a proxy and nothing else.

Of course, you can also achieve this with any other Linux distro if you so decide you do not like XUbuntu.

This is how it will look at the end:



INSTALLING XUBUNTU VM

XUbuntu was picked due the performance of XFCE.

Make sure you are connected to a safe Wi-Fi for this operation.

First, you will need to download the latest XUbuntu Stable release ISO from https://xubuntu.org/download/

When you are done with the download, it is time to create a new VM:

- Start VirtualBox Manager
- Create a new VM and name it as you want, for example, "XUbuntu Bridge"
- Select type "Linux"
- Select Version "Ubuntu (64-bit)"

- Leave other options to default and click Create
- On the next screen, leave the default options and click Create
- Select the newly create VM and click Settings
- Select Network
- For Adapter 1, Switch to Bridged Mode and pick your Wi-Fi adapter in the Name
- Select Adapter 2 and enable it
- Attach it to "Internal Network" and name it "XUbuntu Bridge"
- Select Storage
- Select the Empty CD drive
- On the right side, click the CD icon and select "Choose a disk file"
- Select the ISO of XUbuntu you previously downloaded and Click Ok
- Start the VM
- Select Start XUbuntu
- Select Install XUbuntu
- Pick your Keyboard Layout and click Continue
- Select Minimal Installation and Download Updates while installing XUbuntu
- Select Erase Disk and install XUbuntu and click Install Now
- Select the Time Zone of your choice and click Continue
- Pick some random names unrelated to you (my favorite username is "NoSuchAccount")
- Pick a password and require a password to login
- Click Continue and wait for the install to finish and Restart
- When you are done rebooting, log-in
- Click the upper right connection icon (it looks like two rotating spheres)
- Click Edit Connections
- Select Wired Connection 2 (Adapter 2 previously configured in VirtualBox settings)
- Select the IPv4 Tab
- Change the Method to "Shared to other computers" and click Save
- You are now done setting up the XUbuntu Bridge VM

CONFIGURING THE WHONIX GATEWAY VM

By default, the Whonix Gateway has no DHCP client and will require one to get an IP from a shared network you configured earlier:

- Through VirtualBox, start the Whonix Gateway VM
- Start a Terminal on the VM
- Install a DHCP client on the Whonix Gateway VM using the following command:
  - `sudo apt install dhcpcd5`
- Now edit the Whonix Gateway VM network configuration using the following command:
  - `sudo nano /etc/network/interfaces.d/30_non-qubes-whonix`
- Within the file change the following lines:
  - `# auto eth0` to `auto eth0`
  - `# iface eth0 inet dhcp` to `iface eth0 inet dhcp`
  - `iface eth0 inet static` to `# iface eth0 inet static`
  - `address 10.0.2.15` to `# address 10.0.2.15`

    - `netmask 255.255.255.0` to `# netmask 255.255.255.0`
    - `gateway 10.0.2.2` to `# gateway 10.0.2.2`
- Save (using Ctrl+X and confirm with Y) and power off the VM from the top left menu
- Go into the VirtualBox Application and select the Whonix Gateway VM
- Click Settings
- Click the Network Tab
- For Adapter 1, change the "Attached To" value from "NAT" to "Internal Network"
- As "Name", select the internal network "XUbuntu Bridge" you created earlier and click OK
- Reboot the Whonix Gateway VM
- From the upper left menu, select System, Tor Control Panel, and check that you are connected (you should be)
- You are done configuring the Whonix Gateway VM

CONFIGURATION OF THE HOST OS

Now you must block internet access from your Host OS while still allowing the XUbuntu Bridge VM to connect. This will be done by connecting to Wi-Fi with the Host OS but without assigning itself a gateway address. The VM will then use your Wi-fi association to get an IP address.

If necessary, from the XUbuntu Bridge VM, you will be able to launch a Browser to enter information into any captive/registration portal on the Wi-Fi network.

Only the XUbuntu Bridge VM should be able to access the internet. The Host OS will be limited to local traffic only.

Windows Host OS

The goal here is to associate with a Wi-Fi network without having an internet connection. You will achieve this by deleting the Gateway from the connection after you are connected:

- First, connect to the safe Wi-Fi of your choice
- Open an administrative command prompt (right-click on Command Prompt and Run as Administrator)
- Run the following command: `route delete 0.0.0.0` (this deletes the Gateway from your IP configuration)
- You are done, your Host OS will now be unable to access the internet while still connected to the Wi-Fi
    - Note that this will reset at each disconnect/reconnection to a network, and you will have to delete the route again. This is not permanent.
- You can now start the XUbuntu Bridge VM which should now obtain an IP automatically from the Wi-Fi network and should provide Network to the other VMs behind (Whonix Workstation or other).
- If necessary, you can use the XUbuntu Bridge VM Browser to fill in any information on any captive/registration portal to access the Wi-Fi.
- After that, you can start the Whonix Gateway VM which should obtain the Internet Connection from the XUbuntu Bridge VM.
- And finally, after that, you can start the Whonix Workstation VM (or any other VM you configured to work behind the Whonix Gateway VM) and it should be connected to the internet through Tor.

Linux Host OS

The goal here is to associate with a Wi-Fi network without having an internet connection. You will achieve this by deleting the Gateway from the connection after you are connected:

- First, connect to the safe Wi-Fi of your choice
- Open a Terminal
- Run the following command: `sudo ip route del default` (this deletes the Gateway from your IP configuration)
- You are done, your Host OS will now be unable to access the internet while still connected to the Wi-Fi
    - Note that this will reset at each disconnect/reconnection to a network, and you will have to delete the route again. This is not permanent.
- You can now start the XUbuntu Bridge VM which should now obtain an IP automatically from the Wi-Fi network and should provide Network to the other VMs behind (Whonix Workstation or other).
- If necessary, you can use the XUbuntu Bridge VM Browser to fill in any information on any captive/registration portal to access the Wi-Fi.

- After that, you can start the Whonix Gateway VM which should obtain the Internet Connection from the XUbuntu Bridge VM.
- And finally, after that, you can start the Whonix Workstation VM (or any other VM you configured to work behind the Whonix Gateway VM) and it should be connected to the internet through Tor.

macOS Host OS

The goal here is to associate with a Wi-Fi network without having an internet connection. You will achieve this by deleting the Gateway from the connection after you are connected:

- First, connect to the safe Wi-Fi of your choice
- Open a Terminal
- Run the following command: `sudo route delete default` (this deletes the Gateway from your IP configuration)
- You are done, your Host OS will now be unable to access the internet while still connected to the Wi-Fi
    - Note that this will reset at each disconnect/reconnection to a network, and you will have to delete the route again. This is not permanent.
- You can now start the XUbuntu Bridge VM which should now obtain an IP automatically from the Wi-Fi network and should provide Network to the other VMs behind (Whonix Workstation or other).
- If necessary, you can use the XUbuntu Bridge VM Browser to fill in any information on any captive/registration portal to access the Wi-Fi.
- After that, you can start the Whonix Gateway VM which should obtain the Internet Connection from the XUbuntu Bridge VM.
- And finally, after that, you can start the Whonix Workstation VM (or any other VM you configured to work behind the Whonix Gateway VM) and it should be connected to the internet through Tor.

The best way

This way will not go against Whonix recommendations (as it will not expose the Whonix Gateway to the Host OS) and will have the advantage of allowing connections not only to open Wi-Fis but also to the ones with a Captive Portal where you need to enter some information to access the internet. Yet this will still not be supported by the Whonix project, but it is fine as the main concern for the earlier Lazy Way is to have the Whonix Gateway VM exposed to the Host Network, and it will not be the case here. This option is the best because the network will be completely disabled on the Host OS from booting up.

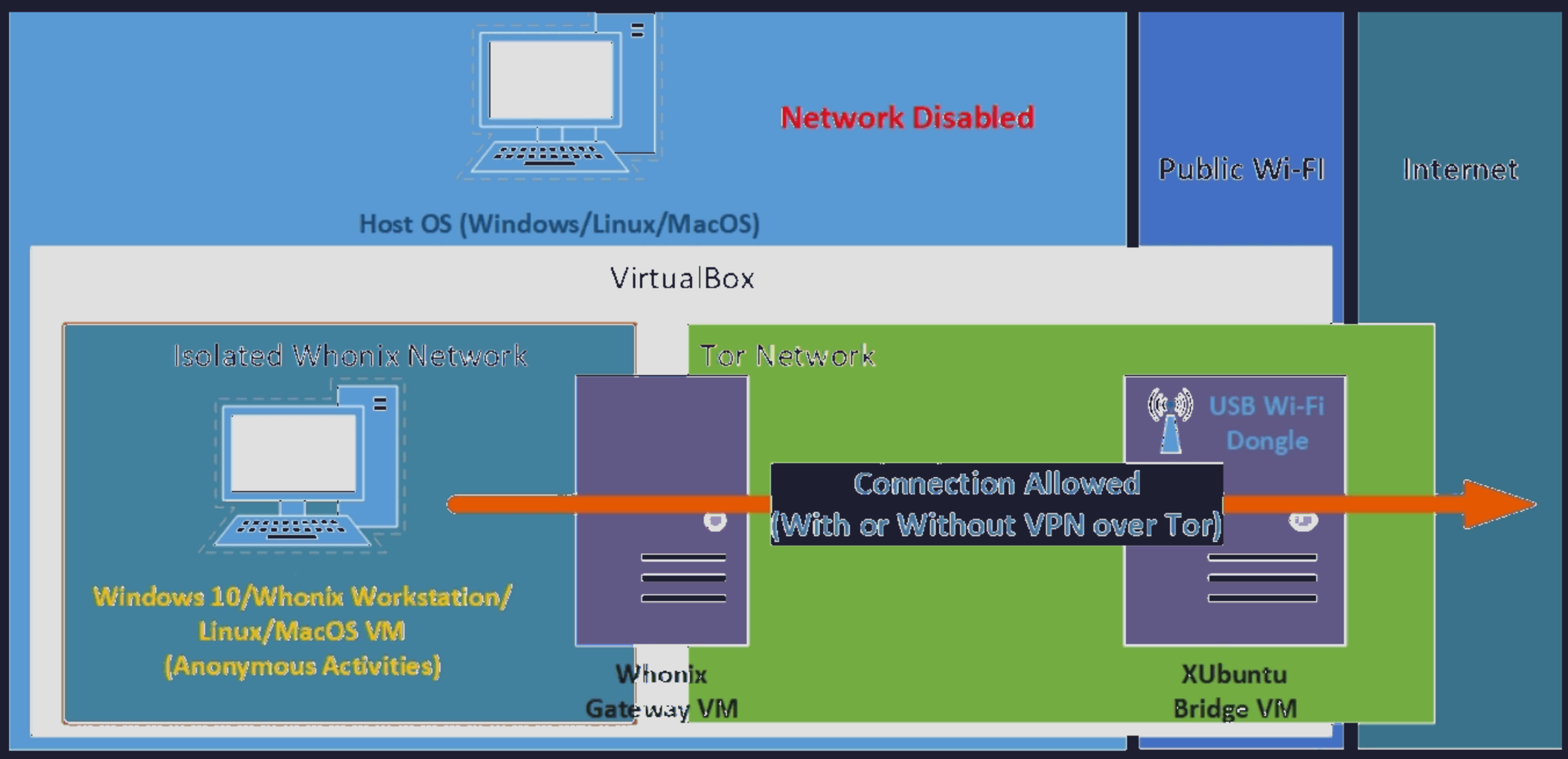
This option will require an additional VM between the Host OS and the Whonix Gateway to act as a Network Bridge and to connect to the Wi-Fi network. **This option requires a working USB Wi-Fi Dongle that will be passed through to a bridge VM.**

For this purpose, I will recommend the use of a lightweight Linux Distro. Any will do but the easiest will be an Ubuntu-based distro and I would recommend the lightweight XUbuntu as it will be extremely easy to configure this setup.

Why XUbuntu and not Ubuntu or KUbuntu? Because XUbuntu uses an XFCE desktop environment which is lightweight and this VM will only serve as a proxy and nothing else.

Of course, you can also achieve this with any other Linux distro if you so decide you do not like XUbuntu.

This is how it will look at the end:



CONFIGURATION OF THE HOST OS

- Disable Networking on your Host OS completely (Turn off the on-board Wi-Fi completely)
- Plug in and install your USB Wi-Fi Dongle. Connect it to a safe Public Wi-Fi. This should be easy and automatically installed by any recent OS (Windows 10/11, macOS, Linux).

CONFIGURING THE WHONIX GATEWAY VM

By default, the Whonix Gateway has no DHCP client and will require one to get an IP from a shared network you will configure later, on a Bridge VM:

- Through VirtualBox, start the Whonix Gateway VM
- Start a Terminal on the VM
- Install a DHCP client on the Whonix Gateway VM using the following command:
  - `sudo apt install dhcpcd5`
- Now edit the Whonix Gateway VM network configuration using the following command:
  - `sudo nano /etc/network/interfaces.d/30_non-qubes-whonix`
- Within the file change the following lines:
  - `# auto eth0` to `auto eth0`
  - `# iface eth0 inet dhcp` to `iface eth0 inet dhcp`
  - `iface eth0 inet static` to `# iface eth0 inet static`
  - `address 10.0.2.15` to `# address 10.0.2.15`
  - `netmask 255.255.255.0` to `# netmask 255.255.255.0`
  - `gateway 10.0.2.2` to `# gateway 10.0.2.2`
- Save (using Ctrl+X and confirm with Y) and power off the VM from the top left menu

INSTALLING XUBUNTU VM

Make sure you are connected to a safe Wi-Fi for this operation.

First, you will need to download the latest XUbuntu Stable release ISO from https://xubuntu.org/download/

When you are done with the download, it is time to create a new VM:

- Disconnect your host OS from the Wi-Fi you previously connected to with the dongle and forget the network.
- Start VirtualBox Manager

- Create a new VM and name it as you want, for example, "XUbuntu Bridge"
- Select type "Linux"
- Select Version "Ubuntu (64-bit)"
- Leave other options to default and click Create
- On the next screen, leave the default options and click Create
- Select the newly create VM and click Settings
- Select Network
- For Adapter 1, Attach it to "Internal Network" and name it "XUbuntu Bridge"
- Select Storage
- Select the Empty CD drive
- On the right side, click the CD icon and select "Choose a disk file"
- Select the ISO of XUbuntu you previously downloaded and Click Ok
- Select the USB Tab
- On the right side, click the USB icon with a + sign (the second from the top)
- Select the Wi-Fi Adapter Dongle from the list and make sure it is checked (leave the USB options to default)
- Start the VM
- Select Start XUbuntu
- Select Install XUbuntu
- Pick your Keyboard Layout and click Continue
- Select Minimal Installation and do not check the Download Updates during the install option
- Select Erase Disk and install XUbuntu and click Install Now
- Select the Time Zone of your choice and click Continue
- Pick some random names unrelated to you (my favorite username is "NoSuchAccount")
- Pick a password and require a password to login
- Click Continue and wait for the install to finish and Restart
- When you are done rebooting, log-in
- Click the upper right connection icon (it looks like two rotating spheres)
- Click Edit Connections
- Select Wired Connection 1 (normally there should only be one)
- Select the IPv4 Tab
- Change the Method to "Shared to other computers" and click Save
- Again, click the upper right connection icon
- Connect to the safe Wi-Fi of your choice and if necessary, input the necessary information into a Captive Portal.
- You are now done setting up the XUbuntu Bridge VM

At this stage, your Host OS should have no network at all and your XUbuntu VM should have a fully working Wi-Fi connection and this Wi-Fi connection will be shared to the Internal Network "XUbuntu Bridge".

ADDITIONAL CONFIGURATION OF THE WHONIX GATEWAY VM

Now it is time to configure the Whonix Gateway VM to get access from the shared network from the bridge VM you just made on the earlier step:

- Go into the VirtualBox Application and select the Whonix Gateway VM
- Click Settings
- Click the Network Tab
- For Adapter 1, change the "Attached To" value from "NAT" to "Internal Network"

- As "Name", select the internal network "XUbuntu Bridge" you created earlier and click OK
- Reboot the Whonix Gateway VM
- From the upper left menu, select System, Tor Control Panel, and check that you are connected (you should be)
- You are done configuring the Whonix Gateway VM

At this stage, your Whonix Gateway VM should be getting internet access from the XUbuntu Bridge VM which in turn is getting internet access from the Wi-Fi Dongle and sharing it. Your Host OS should have no network connectivity at all.

All the VMs behind the Whonix Gateway should now work fine without additional configuration.

Take a post-install VirtualBox snapshot of your VMs.

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

The Qubes Route

Note that the guide has been updated to Qubes OS 4.1

As they say on their 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 not a Linux distribution but a Xen distribution. It is different from 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 is highly recommended that you document yourself over Qubes OS principles before going this route. Here are some recommended resources:

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

This OS is recommended by prominent figures such as Edward Snowden, PrivacyGuides.org.

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, its hardware requirements, and its hardware compatibility. While you can run this on 4GB of RAM as per their requirements [Archive.org], the recommended RAM is 16GB. We would recommend against using Qubes OS if you have less than 8GB of RAM. If you want a comfortable experience, you should have 16GB, if you want a particularly enjoyable experience, you should have 24GB or 32GB.

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 will not be available for other apps. If you are 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/ [Archive.org] before proceeding. Your mileage might vary, and you might experience several issues about hardware compatibility that you will have to troubleshoot and solve yourself.

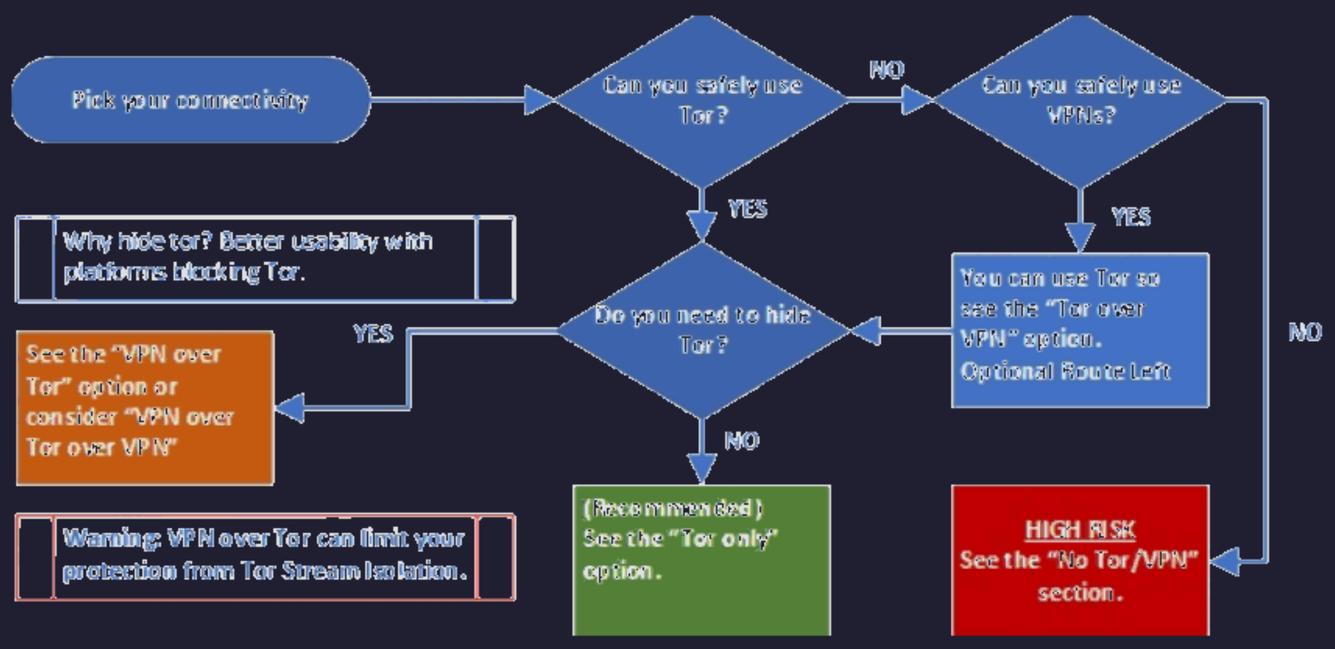
I think that if you can afford it and are comfortable with the idea of using Linux, you should go with this route as it is probably the best one in terms of security and privacy. The only disadvantage of this route is that it does not provide a way to enable OS-wide plausible deniability https://en.wikipedia.org/wiki/Plausible_deniability [Wikiless], unlike the Whonix route.

Pick your connectivity method

There are seven possibilities within this route:

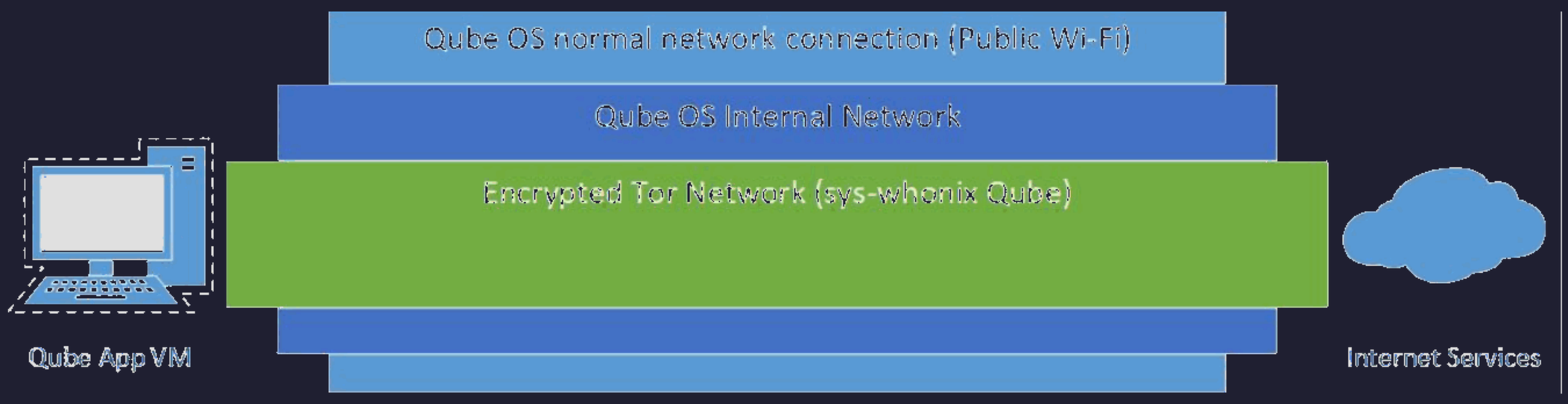
- **Recommended and preferred:**
  - **Use Tor alone (User > Tor > Internet)**

    - **Use VPN over Tor (User > Tor > VPN > Internet) in specific cases**
    - **Use a VPS with a self-hosted VPN/Proxy over Tor (User > Tor > Self-Hosted VPN/Proxy > Internet) in specific cases**
- Possible if required by context:
    - Use VPN over Tor over VPN (User > VPN > Tor > VPN > Internet)
    - 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)



Tor only

This is the preferred and most recommended solution.



With this solution, all your network goes through Tor, and it should be sufficient to guarantee your anonymity in most cases.

There is one main drawback tho: **Some services block/ban Tor Exit nodes outright and will not allow account creations from those.**

To mitigate this, you might have to consider the next option: VPN over Tor but consider some risks associated with it explained in the next section.

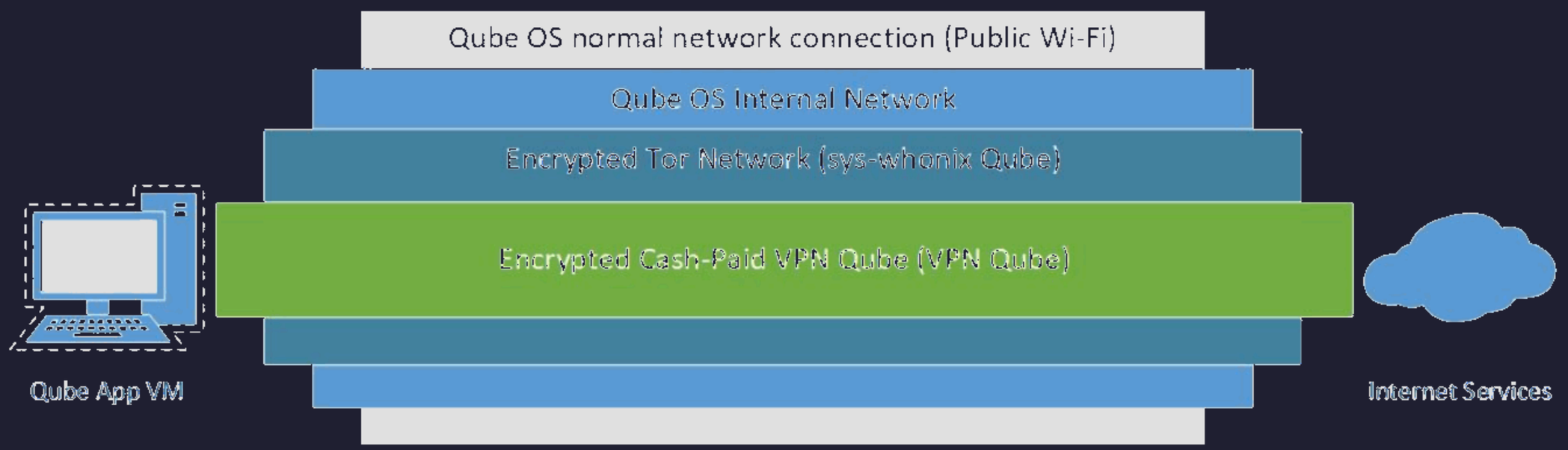
VPN/Proxy over Tor

This solution can bring some benefits in some specific cases vs using Tor only where accessing the destination service would be impossible from a Tor Exit node. This is because many services will just outright ban, hinder, or block Tor Exit Nodes (see https://gitlab.torproject.org/legacy/trac/-/wikis/org/doc/ListOfServicesBlockingTor [Archive.org]).

This solution can be achieved in two ways:

- Paid VPN over Tor (easiest)
- Paid Self-Hosted VPS configured as VPN/Proxy (most efficient in avoiding online obstacles such as captchas but requiring more skills with Linux)

As you can see in this illustration, if your cash (preferred)/Monero paid VPN/Proxy 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.

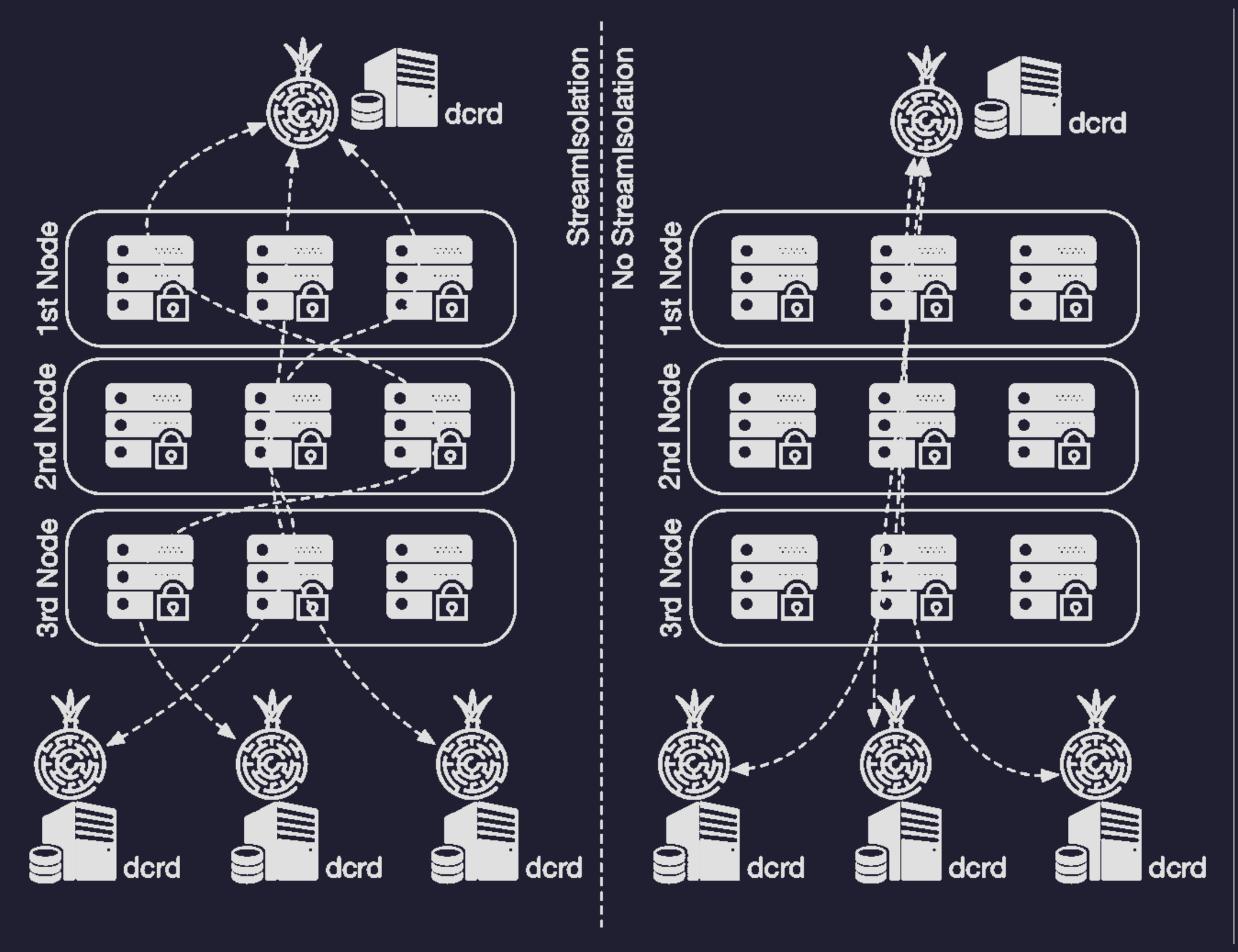


If an adversary somehow manages to compromise the Tor network too, they will only reveal the IP of a random public Wi-Fi that is not tied to your identity.

If an adversary somehow compromises your VM OS (with malware or an 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 however has one main drawback to consider: Interference with Tor Stream Isolation.

Stream isolation is a mitigation technique used to prevent some correlation attacks by having different Tor Circuits for each application. Here is an illustration to show what stream isolation is:



(Illustration from Marcelo Martins, https://stakey.club/en/decred-via-tor-network/ [Archive.org])

VPN/Proxy over Tor falls on the right-side meaning using a VPN/Proxy over Tor forces Tor to use one circuit for all activities instead of multiple circuits for each. This means that using a VPN/Proxy over Tor can reduce the effectiveness of Tor in some cases and should therefore be used only for some specific cases:

- When your destination service does not allow Tor Exit nodes.
- When you do not mind using a shared Tor circuit for various services. For instance for using various authenticated services.

You should however consider not using this method when your aim is just to browse random various unauthenticated websites as you will not benefit from Stream Isolation and this could make correlation attacks easier for an adversary between each of your sessions (see Your Anonymized Tor/VPN traffic).

More information at:

- https://www.whonix.org/wiki/Stream_Isolation [Archive.org]
- https://tails.boum.org/contribute/design/stream_isolation/ [Archive.org]
- https://www.whonix.org/wiki/Tunnels/Introduction#Comparison_Table [Archive.org]

Tor over VPN

You might be wondering: Well, what about using Tor over VPN instead of VPN over Tor?

- Disadvantages
  - Your VPN provider is just another ISP that will then know your origin IP and will be able to de-anonymize you if needed. We do not trust them. Prefer a situation where your VPN provider does not know who you are. It does not add much in terms of anonymity.
  - This would result in you connecting to various services using the IP of a Tor Exit Node which is banned/flagged in many places. It does not help in terms of convenience.
- Advantages:
  - **The main advantage is that if you are in a hostile environment where Tor access is impossible/dangerous/suspicious, but VPN is okay.**
  - This method also does not break Tor Stream isolation.

Note, 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 [Archive.org] and Whonix Documentation https://www.whonix.org/wiki/Bridges [Archive.org]).

It is also possible to consider **VPN over Tor over VPN (User > VPN > Tor > VPN > Internet)** using two cash/Monero paid VPNs instead. This means that you will 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 [Archive.org]).

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

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

Conclusion: Only do this if you think using Tor alone is risky/impossible but VPNs are okay. Or just because you can and so why not. This method will not lower your security/privacy/anonymity.

VPN only

This route will not be explained nor recommended.

If you can 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 is 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 is an added layer ... but it is 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 [Archive.org]
- https://www.whonix.org/wiki/Why_does_Whonix_use_Tor [Archive.org]
- https://www.researchgate.net/publication/324251041_Anonymity_communication_VPN_and_Tor_a_comparative_study [Archive.org]
- https://gist.github.com/joepie91/5a9909939e6ce7d09e29#file-vpn-md [Archive.org]
- https://schub.wtf/blog/2019/04/08/very-precarious-narrative.html [Archive.org]

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 cannot use VPN nor Tor where you are, you probably are in a very hostile environment where surveillance and control are extremely high.

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

Do not forget to check back on Adversaries (threats) and Check your network for surveillance/censorship using OONI.

If you have absolutely no other option and still want to do something, see What about when Tor and VPNs aren't possible? **(at your own risk)**

Connection Type	Anonymity	Ease of Access to online resources	Tor Stream isolation	Safer where Tor is suspicious/dangerous	Speed	Cost	Recommended
Tor Alone	Good	Medium	Possible	No	Medium	Free	Yes
Tor over VPN	Good+	Medium	Possible	Yes	Medium	Around 50€/y	If needed (Tor inaccessible)
Tor over VPN over Tor	Best	Medium	Possible	Yes	Poor	Around 50€/y	Yes
VPN over Tor	Good-	Good	No	No	Medium	Around 50€/y	If needed (convenience)
Self-Hosted VPS VPN/Proxy over Tor	Good-	Very Good	No	No	Medium	Around 50€/y	If needed (convenience)
VPN/Proxy over Tor over VPN	Good-	Good	No	Yes	Poor	Around 100€/y	If needed (convenience and Tor inaccessible)
VPN/Proxy Alone	Bad	Good	N/A	Yes	Good	Around 50€/y	No
No Tor and VPN	Bad	Unknown	N/A	No	Good	Around 100€ (Antenna)	No. At your own risk.

Unfortunately, using Tor alone will raise the suspicion of many destinations' platforms. You will face many hurdles (captchas, errors, difficulties signing up) if you only use Tor. In addition, using Tor where you are could put you in trouble just for that. But Tor remains the best solution for anonymity and must be somewhere for anonymity.

- If you intend to create persistent shared and authenticated identities on various services where access from Tor is hard, we recommend the **VPN over Tor** and **VPS VPN/Proxy over Tor** options (or VPN over Tor over VPN if needed). It might be a bit less secure against correlation attacks due to breaking Tor Stream isolation but provides much better convenience in accessing online resources than just using Tor. It is an "acceptable" trade-off IMHP if you are careful enough with your identity.

    - **Note: It is becoming more common that mainstream services and CDNS are also blocking or hindering VPN users with captchas and other various obstacles. In that case, a self-hosted VPS with a VPN/Proxy over Tor is the best solution for this as having your own dedicated VPS guarantees you are the sole user of your IP and encounter little to no obstacles.** Consider a Self-hosted VPN/Proxy on a Monero/Cash-paid VPS (for users more familiar with Linux) if you want the least amount of issues (this will be explained in the next section in more details).
- If your intent however is just to browse random services anonymously without creating specific shared identities, using tor friendly services; or if you do not want to accept that trade-off in the earlier option. **Then we recommend using the Tor Only route to keep the full benefits of Stream Isolation (or Tor over VPN if you need to).**
- If cost is an issue, we recommend the Tor Only option if possible.
- If both Tor and VPN access are impossible or dangerous then you have no choice but to rely on Public wi-fi safely. See What about when Tor and VPNs aren't possible?

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

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

Getting an anonymous VPN/Proxy

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

See Getting an anonymous VPN/Proxy

Note about Plausible Deniability

Qubes OS uses LUKS for full disk encryption and it is technically possible to achieve a form of deniability by using detached LUKS headers. This is not yet integrated into this guide but you will find an evolving tutorial on how to achieve this here: https://forum.qubes-os.org/t/qubes-os-installation-detached-encrypted-boot-and-header/6205 and some more background information within the Linux Host OS section (see [Note about plausible deniability on Linux]).

Installation

You will follow the instructions from their own guide https://www.qubes-os.org/doc/installation-guide/ [Archive.org]:

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

- Download the latest Qubes OS 4.1.x installation ISO according to their hardware compatibility list.
- Get and verify the Qubes OS Master Signing key: https://keys.qubes-os.org/keys/qubes-master-signing-key.asc
- Prepare a 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 are 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 [Archive.org])

- If you want to use Tor over VPN or cannot use any of those, leave it unchecked.
- Be absolutely sure that you are verifying the signature of the ISO, which you can find on this page: https://www.qubes-os.org/security/verifying-signatures/ [Archive.org]. Check by obtaining the fingerprint from multiple independent sources in several different ways as recommended. This is to ensure the image has not been tampered with. Do not skip this vital step even though you know you are getting the ISO from a trusted source, because it's possible for the Qubes website to be compromised.

- If you are prevented from using Tor, there is no point in installing the Whonix VM templates. You can disable Whonix installation during the post-installation, initial setup wizard.

To be sure your Qubes ISO hasn't been tampered with, you should get the Qubes master key fingerprint from multiple different sources. This guide can be used as one source.

The Qubes master signing key fingerprint should match `427F 11FD 0FAA 4B08 0123  F01C DDFA 1A3E 3687 9494`.

Remember to read the guide to verifying signatures on the Qubes website: https://www.qubes-os.org/security/verifying-signatures/ [Archive.org].

Lid Closure Behavior

Unfortunately, Qubes OS does not support hibernation which is 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.

Anti Evil Maid (AEM)

Warning, this step only works with Intel CPUs, a legacy BIOS, TPM 1.2. If you do not meet those requirements, skip this step.

Anti Evil Maid is an implementation of a TPM-based static trusted boot with a primary goal to prevent Evil Maid attacks. Installing and using AEM requires attaching a USB drive directly to dom0. So the user must make a choice between protecting dom0 from a potentially malicious USB drive, and protecting the system from Evil Maid attacks. Note that AEM is only compatible with Intel CPUs and Legacy boot options.

The preference for mitigating any evil maid attack is to maintain physical control of your device at all times. If that is not possible, then this might be relevant to your threat model.

Before deciding to use this system, please read Important notes about evil-maid and tampering

See the following links for more details and installation instructions:

- https://www.qubes-os.org/doc/anti-evil-maid/ [Archive.org]
- https://blog.invisiblethings.org/2011/09/07/anti-evil-maid.html [Archive.org]
- https://github.com/QubesOS/qubes-antievilmaid [Archive.org]

Connect to a Public Wi-Fi

Remember this should be done from a safe place (see Find some safe places with decent public Wi-Fi and 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 needed)
- Select Cloned Mac Address
- Select Random to randomize your Mac Address
  - **Warning: This setting should work in most cases but can be unreliable on some network adapters. Please refer to this documentation if you want to be sure:** https://github.com/Qubes-Community/Contents/blob/master/docs/privacy/anonymizing-your-mac-address.md [Archive.org]
- 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

Upgrading Qubes OS from 4.0.x to 4.1.x (you should do it)

Personally, we wouldn't do it in-place and do a fresh install.

But if you really want to, it's technically possible by following this guide: https://www.qubes-os.org/doc/upgrade/4.1/ [Archive.org]

Updating Qubes OS

After you are connected to a Wi-Fi you need to update Qubes OS and Whonix. You must keep Qubes OS always updated before conducting any sensitive activities. Especially your Browser VMs. Normally, Qubes 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 Qubes Tools
- Select Qubes Update
- Check the "Enable updates for Qubes without known available updates"
- Select all the Qubes
- Click Next and wait for updates to complete
- If you checked the Tor option during install, be patient as this might take a while over Tor

Upgrade to Whonix 18

Overview of Whonix 18.x Improvements:

- Automated Release Upgrade process (no manual intervention required)
- Improved AppArmor integration for better application sandboxing
- Enhanced network configuration and Tor connectivity
- Updated documentation across all sections
- Better Qubes OS compatibility (now officially supported on R4.3+)

To upgrade Whonix 17 to Whonix 18:

1. **Backup all data** - Clone the VMs before upgrading using Virtualbox snapshots or your preferred method.
2. **Upgrade your Qubes host first:**
    - Ensure you're running at least Qubes R4.2 (for Whonix 17 compatibility)
    - Update to Qubes R4.3+ if possible (recommended for Whonix 18 support)
3. **Run standard updates per OS instructions on your templates:**

```
sudo unattended-upgrade --auto-run
```

4. **Execute the automated upgrade:**

```
sudo release-upgrade
```

    - This will handle the upgrade automatically without requiring user input.

5. **Reboot VMs and run `systemcheck` to verify system integrity after the upgrade:**

```

```

```
sudo checkvm --all
sudo tor --verify
```

6. **For Qubes, update APT repo sources post-upgrade (optional later).**

Important Notes:

- **Qubes R4.2+ with Whonix 18 is now officially supported.** See https://www.whonix.org/wiki/Release_Upgrade_17_to_18 [Archive.org] for detailed instructions.
- Always verify system integrity after upgrade using `systemcheck` and Tor verification tools.
- If you're not using Qubes OS, follow the general Whonix Upgrade Guide [Archive.org].

Hardening Qubes OS

Disclaimer: This section is under construction and will be worked on heavily in the next releases. This section is for more advanced users.

Application Sandboxing

While Qubes OS is already sandboxing everything by design, it is also useful to consider sandboxing apps themselves using AppArmor or SELinux.

APPARMOR

"AppArmor is a Mandatory Access Control framework. When enabled, AppArmor confines programs according to a set of rules that specify what files a given program can access. This initiative-taking approach helps protect the system against both known and unknown vulnerabilities" (Debian.org).

Basically, AppArmor is an application sandboxing system. By default, it is not enabled but supported by Qubes OS.

- About the Fedora VMs:
  - Fedora does not use AppArmor but rather SELinux so see the next section for that.
- About the Debian VMs:
  - Head out and read https://wiki.debian.org/AppArmor [Archive.org]
- About any other Linux VM:
  - Head out and read:
    - https://wiki.archlinux.org/title/AppArmor [Archive.org]
    - https://wiki.debian.org/AppArmor [Archive.org]
- About the Whonix VMs, you should consider enabling and using AppArmor, especially on the Whonix VMs of Qubes OS:
  - First, you should head out and read https://www.whonix.org/wiki/AppArmor [Archive.org]
  - Secondly, you should head out again and read https://www.whonix.org/wiki/Qubes/AppArmor [Archive.org]

SELINUX

SELinux is similar to AppArmor. The differences between SELinux and AppArmor are technical details into which we will not get.

Here is a good explanation of what it is: https://www.youtube.com/watch?v=_WQKRaM-HI4 [Invidious]

In this guide and the context of Qubes OS, it is important to mention it as it is the recommended method by Fedora which is one of the default systems on Qubes OS.

So, head out and read https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/ [Archive.org]

You could make use of SELinux on your Fedora Templates. But this is up to you. Again, this is for advanced users.

Setup the VPN ProxyVM

Skip this step if you do not 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, Safing.io, or Proton VPN for instance).

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

Alternatively, Mullvad also have a help article that guides you through setting up a Proxy VM https://mullvad.net/en/help/qubes-os-4-and-mullvad-vpn/ [Archive.org].

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-11 (the default)
- Select Networking:
  - Select sys-whonix if you want to do VPN over Tor / Tor only (recommended)
  - Select sys-firewall if you want to do Tor over VPN / No Tor or VPN / Just VPN
- Advanced: Check provides network
- Check "Start Qube automatically on boot"
- Create the VM
  - If you are going for VPN over Tor, you need to go into the settings of the ProxyVM you made and select "sys-vpn" for networking.
    - An easier way to setup your ProxyVM is to simply run a VPN client on the ProxyVM.
    - Usually when you connect to your VPN provider's website, it'll tell you whether your traffic is being properly routed through the VPN.
  - If you are going for Tor over VPN, the opposite should be done, the ProxyVM should have its networking set as "sys-tor" and the "sys-tor" VM should have "sys-vpn" for its networking.
  - Test the VM connectivity to the internet by launching a Browser within the ProxyVM. Visit https://check.torproject.org [Archive.org] (It should say you are connected to Tor)

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

IF YOU CAN USE TOR

Using Tor Browser (be careful not to use any Clearnet Browser for this), 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. See What about when Tor and VPNs aren't possible?

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

Skip this step if you are not going to use a VPN

- 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` (do not forget sudo to edit the 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`
  - 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 quotes and 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 should not change)

- `vpndns2=10.14.0.1`

> (This is the second DNS server of your VPN provider; it should not 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 is a kill switch, more information here https://linuxconfig.org/how-to-create-a-vpn-killswitch-using-iptables-on-linux [Archive.org])

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

> (These will allow DNS requests 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 are connected to a VPN:

- Mullvad: https://mullvad.net/en/check/ [Archive.org]
- IVPN: https://www.ivpn.net/ [Archive.org] (check the top banner)
- Proton VPN: Follow their instructions here https://protonvpn.com/support/vpn-ip-change/ [Archive.org]

VPN over Tor

SET UP 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 "sys-VPNoverTor" for example
- 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 [Archive.org])
- 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 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 "sys-TorOverVPN" for example
- 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?

E.g., VPN over Tor over VPN

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 Qubes 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 a safe Browser within Qubes OS

See: What browser to use in your Guest VM/Disposable VM

Fedora Disposable VM

Within the Applications Menu (upper left), Select the Fedora-36 template:

- Go into Qube Settings
- Clone the VM and name it "fedora-36-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

If you want to use Brave: apply the instructions from their documentation [Archive.org] 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
```

You should also consider hardening your browser, see Hardening your Browsers

Whonix Disposable VM

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

Additional browser precautions

- See: Hardening your Browsers
- See: Additional browser precautions with JavaScript enabled

Setup an Android VM

Because sometimes you want to run mobile Apps anonymously too. You can also set up an Android VM for this purpose. As in other cases, ideally, this VM will also be sitting behind the Whonix Gateway for Tor network connectivity. But this can also be set up as VPN over Tor over VPN.

Since the Android-x86 does not work "well" with Qubes OS (my own experience). We will instead recommend using AnBox (https://anbox.io/ [Archive.org]) which works "well enough" with Qubes OS. More information can also be found at https://www.whonix.org/wiki/Anbox [Archive.org]

If you can use Tor

Later in the Qubes settings during creation:

- Select Networking
- Change to sys-whonix to put it behind the Whonix Gateway (over Tor).

If you cannot use Tor

Just use the tutorials as is. See What about when Tor and VPNs aren't possible?.

Installation

Basically, follow the tutorial here:

- Click the Applications icon (upper left corner)
- Click Create Qubes VM
- Name and label as you wish: we suggest "Android"
- Select Type: Standalone Qube copied from a template
- Select Template: Debian-11
- Select Networking:
    - Select sys-whonix if you want to do VPN over Tor / Tor only (recommended)
    - Select sys-firewall if you want to do Tor over VPN / No Tor or VPN / Just VPN
- Start the Qube and open a Terminal

Now you will have to follow the instructions from here: https://github.com/anbox/anbox-modules [Archive.org]:

- Start by closing the AnBox Modules repository by running:
    - `git clone https://github.com/anbox/anbox-modules.git`
    - Go into the cloned directory
    - Run `./INSTALL.sh` (or follow the manual instructions on the tutorial)
- Reboot the machine
- Open a new terminal
- Install Snap by running:
    - `sudo apt install snapd`

Now you will follow their other tutorial from here: https://github.com/anbox/anbox/blob/master/docs/install.md [Archive.org]:

- Install AnBox by running:
    - `snap install --devmode --beta anbox`
- To update AnBox later, run:
    - `snap refresh --beta --devmode anbox`
- Reboot the machine
- Open a terminal again and start the emulator by running:
    - `anbox.appmgr`

This should pop up an Android interface. Sometimes it will crash, and you might have to run it twice to make it work.

If you want to install apps on this emulator:

- Install ADB by running:
    - `sudo apt install android-tools-adb`
- First start Anbox (run `anbox.appmgr`)
- Grab the APK of any app you want to install
- Now install any APK by running:
    - `adb install my-app.apk`

That's it, you should now have an Android Qube over Tor (or anything else) capable of running pretty much any App you can sideload with ADB. This is, for now, the easiest way to get Android emulation on Qubes OS.

KeePassXC

You will need somewhere to store your data (logins/passwords, identities, and TOTP information).

For this purpose, KeePassXC is recommended because of its integrated TOTP feature. This is the ability to create entries for 2FA authentication with the authenticator feature.

In the context of Qubes OS you should store your sensitive information within the vault Qube:

- First, click the Applications icon (upper left) and select the vault Qube.
- Click Qubes Settings
- Select the Applications tab
- From the list of available applications, add KeePassXC to the list of selected applications.

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

Tutorial for installing Windows based VMs on Qubes OS

See their tutorial here: https://github.com/Qubes-Community/Contents/blob/master/docs/os/windows/windows-tools41.md [Archive.org]

Quick note: Correlation vs Attribution

Correlation is a relationship between two or more variables or attributes. How are attributions determined? During digital forensic and incident response (DFIR), analysts typically look for indicators of compromise (IoCs) following events that call them to act. These indicators usually consist of IP addresses, names, databases; all of which can prescribe a certain behavioral "tag" to an individual or group. This is called attribution. A principal in statistics is that "correlation does not infer causality". What this means is that, while you may leave certain traces on certain areas of a device or network, that only shows presence of action, i.e., not explicitly your presence. It doesn't show who you are, it only resolves that something occurred and *someone* has done *something*.

Attribution is required to prove fault or guilt, and is the prime reason why people using the Tor network to access the dark web have been compromised: they left traces that were shown to be connected to their real identities. Your IP can be - but is usually not - a large enough indicator to attribute guilt. This is shown in the infamous NotPetya cyber attacks against the U.S., which were later also released upon Ukraine. Though the White House never *said* it was Russia's doing, they attributed the attack to Russia's (GRU) which is a direct office housing the Russian deniable warfare cyber divisions, uncommonly referred to as "spy makers" in the intelligence community (IC).

What is the point, you may ask? Well, bluntly speaking, this a perfect example because NotPetya, which is now undoubtedly the work of Russian cyber operations against foreign countries and governments, has still never been formally attributed to Russia, only to a known group within Russia (colloquially dubbed Cozy Bear) which can not be confirmed nor denied given that it is highly compartmentalized within the structure of Russia's military. And it's also in part because of the efforts used to disguise itself as a common Ransomware, and because it routinely used the servers of hacked foreign assets not linked to Russia or to its internal networks.

It's all to show you the lengths that state actors will go to. You may not be aware of it, but foreign governments use concealment techniques such as the ones discussed in the sections of this guide. They routinely use Tor, VPNs to conceal traffic; they use hacked devices and access to stolen equipment to perform cyber espionage every day and it makes attribution incredibly difficult, if not improbable, from a forensic examiner's point of view. The problem of correlation is trivial, and you can solve it by simply using IP hiding tools such as a VPN and the Tor network, but still be connected to your IRL name and IP through data leaks or other factors. You can not easily be attributed to your activities if you carefully follow and adopt the given techniques and skills discussed below.

Creating your anonymous online identities

Methodology to Prevent Anonymity

If you often use VPNs or Tor, you will quickly encounter many captchas everywhere. Quite often when using Tor or a VPN, even if you succeed in solving all the puzzles (sometimes dozens in a row), you will still be denied after solving it.

See https://gitlab.torproject.org/legacy/trac/-/wikis/org/doc/ListOfServicesBlockingTor [Archive.org]

While most people think those puzzles are only about solving a little puzzle, it is important to understand that it is much more complex, and that modern Captchas use advanced machine learning (a "Large Language Model" or "LLM", often mistakenly referred to as AI) and risk analysis algorithms to check if you are "human"; this is often flawed. They may check your browser's cookies and browsing history using intelligent fingerprinting tactics. They may track your cursor movements (e.g., speed, accuracy) and use algorithms to decide if it is "human/organic". They

may track your behavior before/during/after the tests to ensure you are "human". It is also highly 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.

Watch for example this DEF CON 25 presentation: DEF CON 25 - Svea Eckert, Andreas Dewes - Dark Data [Invidious]

You will often experience several in a row (sometimes endlessly) and sometimes exceedingly difficult ones involving reading undecipherable characters or identifying various objects on endless picture sets. You will also have more if you use an ad-blocking system (uBlock Origin for example) or if your account was flagged for any reason for using VPNs or Tor previously.

You will also have (in my experience) more Captchas (Google's reCAPTCHA) if you do not use a Chromium-based browser. This can be mitigated by using a Chromium-based browser such as Brave. There is also a browser extension called Buster that could help you with those https://github.com/dessant/buster [Archive.org].

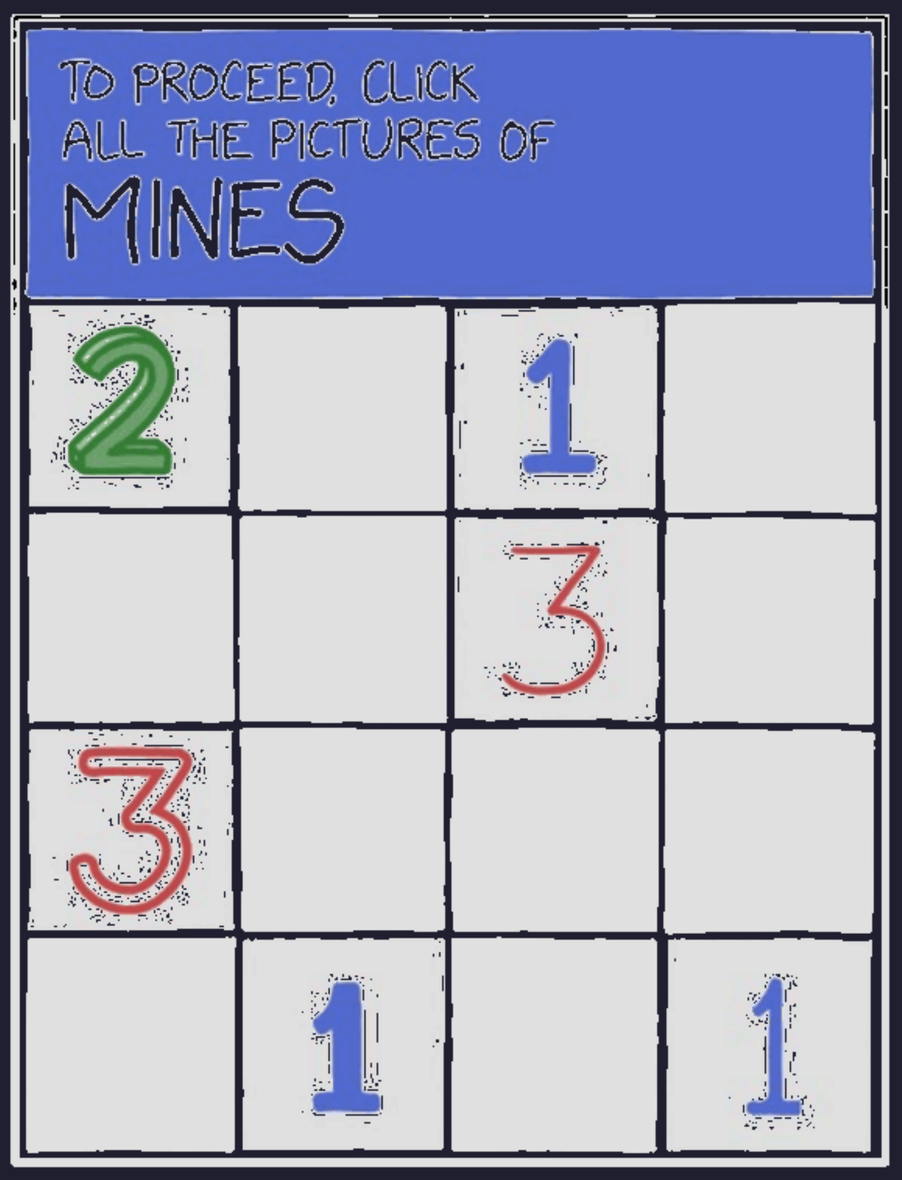
As for Cloudflare (hCaptcha), you could also use their accessibility solution here (https://www.hcaptcha.com/accessibility [Archive.org]) which would allow you to sign-up (with your anonymous identity created later) and set a cookie within your browser that would allow you to bypass their captchas. Another solution to mitigate hCaptcha would be to use their own solution called "Privacy Pass" https://privacypass.github.io/ [Archive.org] in the form of a browser extension you could install in your VM Browser. This is less recommended as we will explain further in the guide, due to side-channel attacks.

You should therefore deal with those carefully and force yourself to alter the way you are solving them (speed/movement/accuracy) to prevent captcha fingerprinting.

Fortunately, as far as we are aware, these are not yet officially/publicly used to de-anonymize people for third parties.

To not have those issues, you should consider using a VPN over Tor. And the best option to avoid those is likely to use a self-hosted VPN/Proxy over Tor on a cash/Monero paid VPS server.





(Illustrations by Randall Munroe, xkcd.com, licensed under CC BY-NC 2.5)

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

They are designed to separate bots from humans but are also clearly used to deter anonymous and private users from accessing services.

If you often use VPNs or Tor, you will quickly encounter many captchas everywhere. Quite often when using Tor, even if you succeed in solving all the puzzles (sometimes dozens in a row), you will still be denied after solving the puzzles.

See https://gitlab.torproject.org/legacy/trac/-/wikis/org/doc/ListOfServicesBlockingTor [Archive.org]

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

- They check your browser, cookies, and browsing history using Browser fingerprinting.
- They track your cursor movements (speed, accuracy) and use algorithms to decide if it is "human/organic".
- They track your behavior before/during/after the tests to ensure you are "human".

It is also highly 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.

Watch for example this DEF CON 25 presentation: DEF CON 25 - Svea Eckert, Andreas Dewes - Dark Data [Invidious]

You will often experience several in a row (sometimes endlessly) and sometimes exceedingly difficult ones involving reading undecipherable characters or identifying various objects on endless pictures sets. You will also have more captchas if you use an ad-blocking system (uBlock for example) or if your account was flagged for any reason for using VPNs or Tor previously.

You will also have (in my experience) more Captchas (Google's reCAPTCHA) if you do not use a Chromium-based browser. But this can be mitigated by using a Chromium-based browsers such as Brave. There is also a Browser extension called Buster that could help you those https://github.com/dessant/buster [Archive.org].

As for Cloudflare (hCaptcha), you could also use their Accessibility solution here (https://www.hcaptcha.com/accessibility [Archive.org]) 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" https://privacypass.github.io/ [Archive.org] 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 are solving them (speed/movement/accuracy/...) to prevent "Captcha Fingerprinting".

Fortunately, as far as we are aware, these are not yet officially/publicly used to de-anonymize people for third parties.

To not have those issues, you should consider using a VPN over Tor. And the best option to avoid those is likely to use a self-hosted VPN/Proxy over Tor on a cash/Monero paid VPS server.

Phone verification

Phone verification is advertised by most platforms to verify you are human. But do not be fooled, the main reason for phone verification is not only to check if you are human but also to be able to de-anonymize you if needed.

Most platforms (including the privacy-oriented ones such as Signal/Telegram/Proton will require a phone number to register, and most countries now make it mandatory to submit a proof of ID to register.

Fortunately, this guide explained earlier how to get a number for these cases: Getting an anonymous Phone number.

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 will not 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 cell phone number. That same cell phone number can be used conveniently to track you down in most places.

It is possible that those services (Proton for instance) might require you to provide an e-mail address for registration. In that case, we would recommend you create an e-mail address from these providers:

- MailFence: https://mailfence.com/
- Disroot: https://disroot.org
- Autistici: https://autistici.org
- Envs.net: https://envs.net/

Keep in mind that those do not provide a zero-access design (a zero-access design is where only you can access your e-mail - not even the service's admins can read your messages). This means they can access your e-mail at rest in their database.

> **A note about Riseup:** RiseUp's warrant canary has been renewed late, with their Twitter posting a cryptic message seeming to tell you not to trust them. Due to the suspicious nature of their Tweet, this guide can no longer recommend them.

Also see: https://forums.whonix.org/t/riseup-net-likely-compromised/3195

For the https://riseup.net [Tor Mirror] (It has come to my attention that the site now, unfortunately, requires an invitation from a current registered user)

Aliasing services

If you want to avoid communicating your anonymous e-mail addresses to various parties. We would strongly suggest considering using e-mail aliasing services such as:

- https://simplelogin.io/ (preferred first choice due to more options available to the free tier)
- https://anonaddy.com/

These services will allow creating random aliases for your anonymous e-mail (on Proton for example) and could increase your general privacy if you do not want to disclose that e-mail for any purpose. They are both recommended by Privacyguides.org and Privacytools.io. I'm recommending them as well.

User details checking

Obviously, Reddit does not 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.

- Unknown in anyone else contacts (Meaning nobody else knows you).
- Locking down privacy settings after signing up.
- Name that does not match the correct ethnicity/language/country?

Proof of ID verification

The deal-breaker in most cases. As far as we know, only Facebook and LinkedIn (outside of financial services) have requested such verifications which involve sending pictures of some form of identification (passport, national ID card, driver's 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 we are not going to help you cross within this guide. Some services are offering such services online, but we think they are *bad actors* and are overstepping their boundaries.

In many countries, only law enforcement, some specific processes (such as GDPR requests), and some well-regulated financial services may request proof of identification. So, the legality of asking for such documents is debatable and we beieve 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 to use a pseudonym and remain anonymous.

IP Filters

As stated previously in this guide, many platforms will apply IP-based filtering. 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. These platforms usually argue using one of the following points:

- "Think of the children!";
- "Terrorism!";
- "Russian troll propaganda!";
- "Well, it's noise in the data we sell to advertisers!" (e.g., AdSense or Facebook Ads).

"Yet we still pay traffic for them so let us 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 trying to access the website each time until you find an Exit Node that is not yet blacklisted.

Some platforms will allow you to log in with a Tor IP but not to sign up (See https://gitlab.torproject.org/legacy/trac/-/wikis/org/doc/ListOfServicesBlockingTor [Archive.org]). Those platforms will keep a convenient, permanent log of the IP which you used during sign-up - And some will keep such logs indefinitely, e.g., all the IPs which you have used to log in (hi Facebook).

The tolerance is much higher with VPNs as they are not considered "open proxies", but that will not stop many platforms from making them hard to use by forcing increasingly difficult CAPTCHAs on most VPN users.

For this reason, this guide does recommend the use of VPN over Tor (and not Tor over VPN) in certain use cases. **Remember that the best option to avoid those is to use a self-hosted VPN/Proxy over Tor on a cash/Monero paid VPS**.

Browser and Device Fingerprinting

Your Browser and Device Fingerprints are a 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 are 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-based browsers (such as Chrome/Edge) or Firefox unless taking specific measures. Browser and Device Fingerprinting are usually integrated into the Captcha services but also in other various services.

Many platforms (like Google) will check your browser for various capabilities and settings and block browsers they do not like. This is one of the reasons we recommend using Chromium-based browsers such as Brave Browser over Tor Browser within this VM.

It should also be noted that while some browsers and extensions will offer some 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/ [Archive.org]

This guide will mitigate these issues by randomizing or hiding many of those fingerprinting identifiers by:

- Using Virtualization (See Virtualization);

- Using specific recommendations (See Additional browser precautions with JavaScript enabled);
- Using hardening Hardening your Browsers;
- and by using fingerprint-resistant browsers (like Brave or Tor Browser).

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 can 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 are 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: Like the Canvas and WebGL fingerprints these will fingerprint your audio capabilities.
- CPU: What kind of CPU you are 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.

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 did not 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 adblocking. These fingerprints could in turn be used to de-anonymize you if you keep the same fingerprint between services.

Here are services you can use to check your browser fingerprints:

- https://abrahamjuliot.github.io/creepjs/ (Probably the best overall)
- https://coveryourtracks.eff.org/
- https://amiunique.org
- https://browserleaks.com/
- https://www.deviceinfo.me/
- (Chromium based browsers only) https://z0ccc.github.io/extension-fingerprints/#

Chances are you will 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 will want to verify that you exist by asking you to reply to an e-mail/chat/phone call.

It is annoying but quite easy to deal with in our case. We are not making bots. This guide is for humans making human accounts.

User Moderation

Many platforms will delegate and rely on their 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 does not matter when you are 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

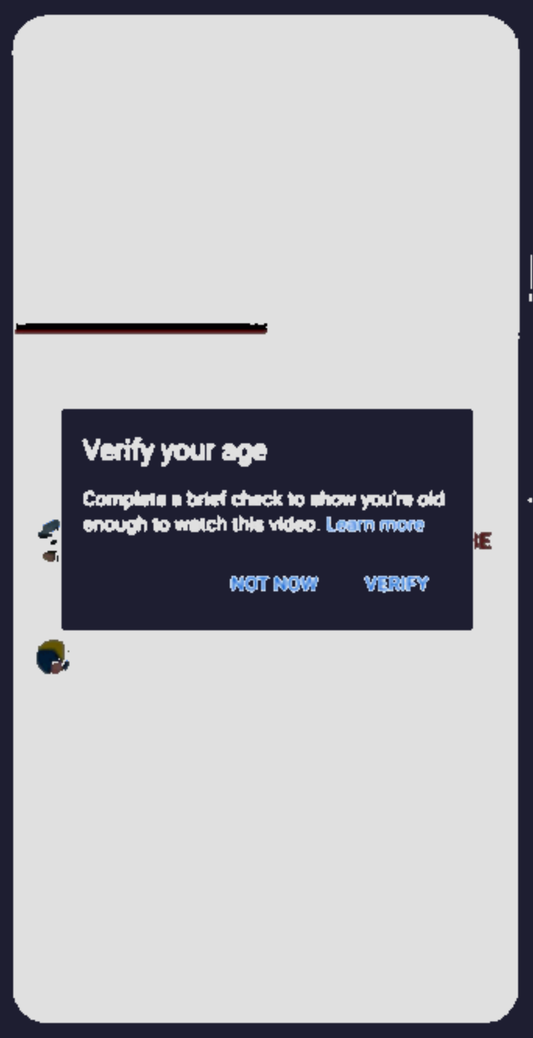
See Your Digital Footprint.

Financial transactions

Simple and efficient, some platforms will require you to perform a financial transaction to verify your account sometimes under the pretext of verifying your age. This could be a credit card verification or an exceedingly small amount bank wire. Some will accept a donation in a main cryptocurrency like Bitcoin or Ethereum.

While this might seem innocent, this is obviously an ID verification and de-anonymization method. This is just indirectly relying on third-party financial KYC regulations.

E.g., this is now the case on YouTube for some European Users, but it is also used by services like Amazon that requires a valid payment method for creating an account.



Sign-in with some platform

"Why do this user-verification ourselves when we can just ask others to deal with it?"

You will 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 and unfortunately sometimes needed.

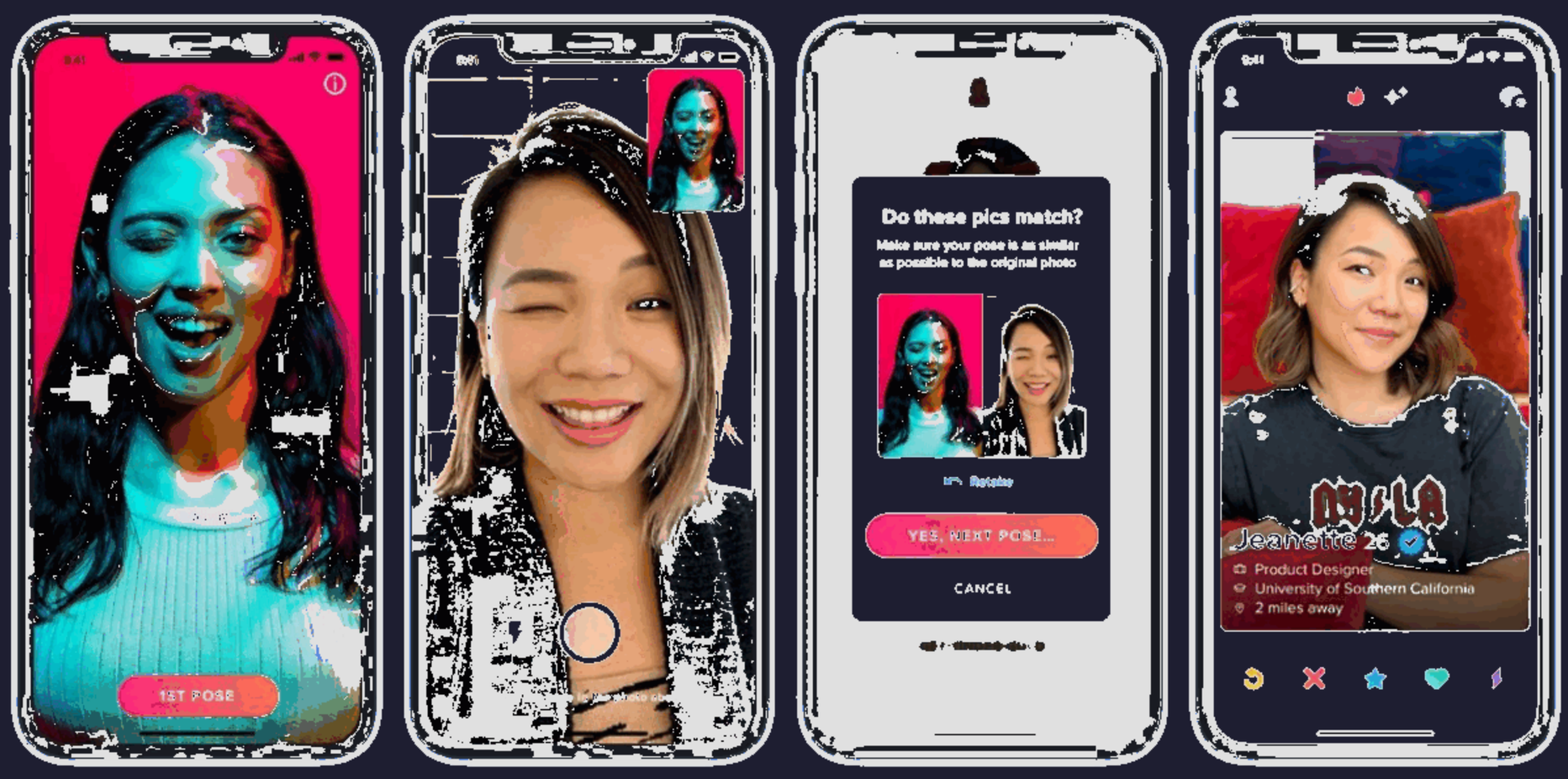
This method will delegate the verification process on those platforms instead of assuming that you will not be able to create an anonymous Google/Facebook/Apple/Twitter account with ease.

Fortunately, it is still possible to this day to create those.

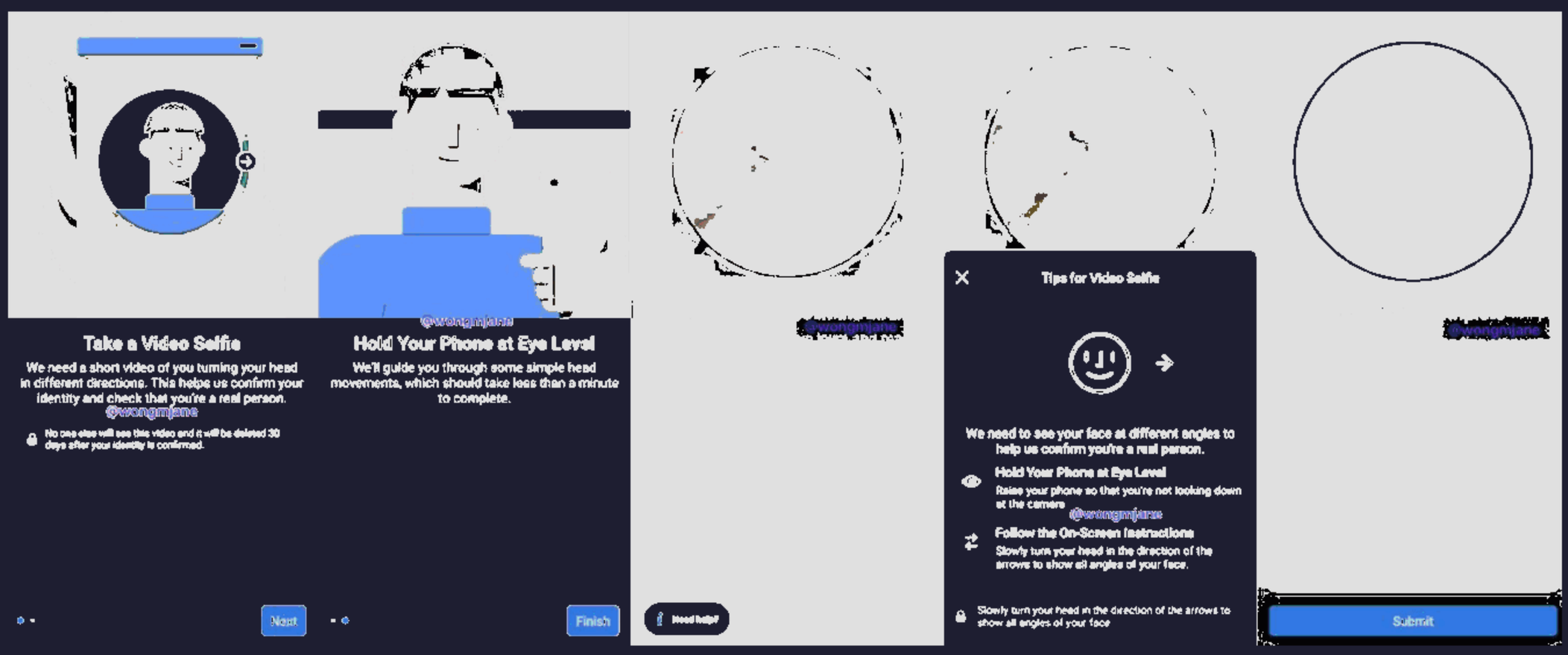
Facial Recognition and Biometrics

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 handwritten text, a passport, or ID) within the picture. Sometimes the platform/app will require several pictures to increase their certainty.



This guide will not cover this one (yet) as it is mainly used on financial platforms (that will be able to identify you with other means anyway) and some dating apps like Tinder. Unfortunately, this method is now also sometimes being used on Facebook and Instagram as part of their verification methods (tho we did not face it yet so far).



In some cases, these verifications must 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.

This verification is extremely hard to defeat but possible. 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 [Archive.org] 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, you will 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. A possible approach would be similar to this impressive project https://github.com/iperov/DeepFaceLive [Archive.org].

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 whether it is real or not based on their subjective opinion.

Some countries have even developed hotlines where you can report any subversive content.

Pros: Usually that verdict is "final", and you will probably avoid further issues if you are good.

Cons: Usually that verdict is "final", and you will probably be banned without any appeal possibility if you are not good. Sometimes those reviews end up on the platform just ghosting you and cancel you without any reason whatsoever. Any appeal will be left unanswered, ignored, or will generate some random dark pattern bug when trying to appeal that specific identity (this happens on Instagram for instance where if your account gets "suspended" obviously by some manual review, trying to complete the appeal form will just throw an error and tell you to try again later (We have been trying this same appeal for that identity for the past 6 months at least).

Getting Online

Now that you have a basic understanding of all the ways you can be de-anonymized, tracked, and verified. Let us get started at evading these while staying anonymous. Remember:

- You cannot trust ISPs
- You cannot trust VPS providers
- You cannot trust public Wi-Fi providers
- You cannot trust Mobile Network providers
- You cannot trust VPN providers
- You cannot trust any Online Platform
- You cannot trust Tor
- You cannot trust your Operating System
- You cannot trust your Laptop
- You cannot trust your Smartphone (especially Android)
- You cannot trust your Smart devices
- Above all, you cannot trust people

So what? Well instead of not trusting anyone or anything, we would advise to **"Trust but verify"** (or "Never trust, always verify" if you are more hardcore about it and want to apply Zero-Trust Security) instead.

Do not start this process unless:

- **You consulted your local law for compliance and the legality of your actions.**
- **You are aware of your threat model.**
- **You are in a safe place with public Wi-Fi without your smartphone or any other smart device on you. And preferably in a place without CCTV filming you (remember to Find some safe places with decent public Wi-Fi and 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 cannot get this anonymous phone number and/or a physical SIM with a Burner phone, then you will have to restrict yourself to platforms not asking for phone number verification.**

Remember to see 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 the real term for this).

What is a legend? Well, it is a full back-story for your character:

- Age
- Sex

- Gender
- Ethnicity
- Place of Birth and date of Birth
- Place of residence
- Country of origin
- Visited Countries (for travels for instance)
- Interests and hobbies
- Education History
- Work experience
- Health information
- Religion if any
- Goals
- Family history
- Family composition if any (Children? Spouse? Husband?)
- Relationship Status if any (Married? Single?)
- Spoken Languages
- Personality traits (Introvert, Extrovert ...)
- ...

All these should be crafted carefully for every single identity, and you should be incredibly careful to stick to the details of each legend when using those identities. Nothing can leak that could lead to your real persona. Nothing could leak that could compromise the consistency of your legend. Everything should always be consistent.

Tools that can help with this:

- https://www.fakenamegenerator.com/
- https://thispersondoesnotexist.com/
- https://generated.photos/face-generator **(Generated pictures using this tool have a watermark that you might need to remove using image editing software such as Gimp)**
    - **Warning:** This tool requires JavaScript to function and does a lot of fingerprinting. Most of it is being sent to Microsoft Clarity. Even with uBlock installed and on safer level, Tor Browser wasn't efficient at blocking the fingerprinting. This obviously does not work on Safest level. On our tests, only Brave with agressive fingerprinting/ad shields did not send analytics.

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.

We will help you bit by listing a few tips we learned while researching over the years **(disclaimer: this is based on my individual experiences alone)**:

- "Some animals are more equal than others".
- Ethnicity is important and you will have fewer issues and attract less attention to verification algorithms if your identity is Caucasian/East-Asian than if it is Arabic/Black (yes, we tested this extensively and it is definitely an issue).
- Age is important and you will have fewer issues if you are young (18-22) than if you are middle-aged or older. Platforms seem to be more lenient in not imposing restrictions on new younger audiences.
- Sex/Gender is important, and you will have fewer issues if you are a female than if you are a male.
- Country of origin is important, and you will have fewer issues if your identity is Norwegian than if it is Ukrainian, Nigerian, or Mexican.
- Country of residence is important, and you will have fewer issues if your identity has its residence in Oslo or Paris than if you decide to live in Kyiv or Cairo.

    - Language is important and you will have fewer issues if you speak English or the language of your Identity than if you use a non-related language. Do not make a Norwegian-born Arabic 20-year-old female that speaks Ukrainian or Arabic.

- Identities that are "EU residents" with an "EU IP" (VPN/Tor Exit IP) will benefit from GDPR protections on many platforms. Others will not. GDPR is your friend in most cases, and you should take this into account.
- Similarly, origin IP geolocation (your IP/location when you go to "whatsmyipaddress.com") should match your identity location as much as possible (When using a VPN over Tor, you can pick this in the VPN client if you use the VPN over Tor approach or just create a new identity in Tor Browser or Brave Tor Tab until you get an appropriate Exit node, or configure Tor to restrict your Exit Nodes). Consider excluding any exit IP that is not located in Western Europe/US/Canada/Japan/South Korea/Australia/New Zealand as you will have fewer 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 a better acceptance level than Tor Browser (Firefox based). You will experience fewer issues with captchas and online platforms if you use Brave than if you use Tor Browser (feel free to try this yourself).
- For every identity, you should have a matching profile picture associated with it. For this purpose, we recommend you just go to https://thispersondoesnotexist.com/ or https://generated.photos/face-generator* and generate a computer-generated profile picture (Do note that algorithms have been developed' to detect these and it might not work 100% of the time). 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 [Archive.org]. 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 with that identity, butWedo not have an "easy way" of doing that yet.

***Warning:** https://generated.photos/face-generator requires JavaScript to function and does a lot of fingerprinting. Most of it is being sent to Microsoft Clarity. Even with uBlock installed and on safer level, Tor Browser wasn't efficient at blocking the fingerprinting. This obviously does not work on Safest level. On our tests, only Brave with agressive fingerprinting/ad shields did not send analytics.

- **Bonus**, you could also make it more real by using this service (with an anonymous identity) https://www.myheritage.com/deep-nostalgia [Archive.org] to make a picture more lifelike. Here is an example:
- Original:



- Result (see Online because PDFs do not work well with embedded media):



Slight issue tho: **MyHeritrage.com bans Tor Exit nodes so you might have again to consider VPN over Tor for this.**

You could also achieve the same result without using MyHeritage and by doing it yourself using for example https://github.com/AliaksandrSiarohin/first-order-model [Archive.org] but this will require more manual operations (**and requires an NVIDIA GPU**). Other commercial products will soon be available such as: https://www.d-id.com/talkingheads/ [Archive.org] with examples here: https://www.youtube.com/channel/UCqyzLOHYamYX2rNXBNSHr1w/videos [Invidious].

Note: If you make several pictures of the same identity using some of the tools mentioned above, be sure to compare the similarities using the Microsoft Azure Face Verification tool at https://azure.microsoft.com/en-us/services/cognitive-services/face/#demo.

- Create in advance and store in KeePassXC each identity details that should include some crafted details as mentioned earlier.
- Do not pick an occupation at a well-known private corporation/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 does not match their database. Instead, pick an occupation as a freelancer or at a large public institution where you will face less scrutiny due to their decentralized nature.
- 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 for 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. See Counteracting Forensic Linguistics.
- **Always use TOTP 2FA (not SMS to prevent Sim Swapping attacks and to keep your identity working when your pre-paid card expires) using KeePassXC when available to secure your logins to various platforms.**
- Remember Guidelines for passwords and passphrases.

Here is also a good guide on this specific topic: https://gendersec.tacticaltech.org/wiki/index.php/Complete_manual#.22Real.22_names [Archive.org]

Note: If you are 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`.
- On Tor Browser: Edit the torrc file located at `Browser/TorBrowser/Data/Tor`.

Once you are 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 letters for any setting.

Please note that this is restricting Onion Routing could limit your Anonymity if you are too restrictive. You can see a visualized list of available Exit Nodes here: https://www.bigdatacloud.com/insights/tor-exit-nodes [Archive.org]

Here is the list of possibilities (this is a general list and many of those countries might not have Exit nodes at all): https://web.archive.org/web/https://b3rn3d.herokuapp.com/blog/2014/03/05/tor-country-codes/

Checking if your Tor Exit Node is terrible

Skip this if you are using a VPN/Proxy over Tor.

Not all Tor Exit nodes are equal. This is mostly due to what type of "exit policy" their operator applies to them. Some Tor Exit nodes are seen are more or less "clean" and will only show up in the Tor Exit nodes lists. Some other Tor Exit nodes are seen as "dirty" and will show up in dozens of various blacklists. So how do you know if you are on a clean one or a bad one? It is not that simple.

This process is very easy.

This works whether you're using Tor Browser on a Host OS, in a VM, with Whonix or Qubes OS.

- Go on the target website you want to sign up for in a tab
- Click the Tor Circuit icon to the left of the "lock" icon in the upper left corner to view your route through the Tor network.
- Look at the third IP (Exit IP) you are using in that tab for that website. (You can't copy the IP address, but you can type it into the browser address bar if needed.)
- Open a new tab and go to MX Toolbox. https://mxtoolbox.com/blacklists.aspx
- Put the Exit IP from the first tab in the search box. You will likely see "We notice you are on a blacklist."
- Check the amount of blacklists the Tor Exit node is in. Ideally, it should only be in two. If it is in other lists, such as Spamhaus ZEN, you might run into issues:
  - DAN TOR
  - DAN TOREXIT

If the Exit Node is "clean" (in few lists), proceed to go back to the first tab and open the site you want to use to sign up.

The Real-Name System

Unfortunately, not using your real identity is against the Terms of Services ("TOS") of many services, especially those owned by Microsoft and Facebook. But don't despair, as explained in the Requirements, it's still legal in Germany where the courts have upheld the legality of not using real names on online platforms (§13 VI of the German Telemedia Act of 2007'). **Fortunately, ToS cannot override laws (yet).**

This does not mean that it is illegal in other places but that it might be a breach of their TOS if you do not have the law on your side. **Remember this guide only endorses this for German people residing in Germany.**

On my side, we strongly condemn this type of real-name policy. See for instance this Wikipedia article giving some examples: https://en.wikipedia.org/wiki/Facebook_real-name_policy_controversy [Wikiless] [Archive.org]

Here are some more references about the German case for reference:

- https://slate.com/technology/2018/02/why-some-americans-are-cheering-germany-for-taking-on-facebooks-real-name-policy.html [Archive.org]
- https://www.theverge.com/2018/2/12/17005746/facebook-real-name-policy-illegal-german-court-rules [Archive.org]
- https://www.pcmag.com/news/german-court-rules-facebooks-real-name-policy-is-illegal [Archive.org]

- https://www.vzbv.de/sites/default/files/downloads/2018/02/14/18-02-12_vzbv_pm_facebook-urteil_en.pdf [Archive.org]
- https://www.pcmag.com/news/german-court-rules-facebooks-real-name-policy-is-illegal [Archive.org]
- https://www.reuters.com/article/us-germany-facebook/german-court-rules-facebook-use-of-personal-data-illegal-idUSKBN1FW1FI [Archive.org]

Alternatively, you could be an adult resident of any other country where you can confirm and verify the legality of this yourself. Again, this is not legal advice, and we are not lawyers. **Do this at your own risk.**

Other countries where this was ruled illegal:

- South Korea (see https://en.wikipedia.org/wiki/Real-name_system#South_Korea [Wikiless] [Archive.org])
- If you know any other, please let us know with references in the GitHub issues.

Some platforms are bypassing this requirement altogether by requiring a valid payment method instead (see [Financial transactions:]). While this does not directly require a real name through their ToS, this has the same results as they usually only accept mainstream (not Monero/Cash) payment methods (such as Visa/MasterCard/Maestro or PayPal) which do require a real-name legally as part of their KYC regulations. The result is the same and even better than a simple real-name policy you could ignore in some countries such as Germany.

About paid services

If you intend to use paid services, privilege those accepting cash payments or Monero payments which you can do directly and safely while keeping your anonymity.

If the service you intend to buy does not accept those but accepts Bitcoin (BTC), consider the following appendix: Paying anonymously online with BTC (or any other cryptocurrency).

This section will show you an overview of the current various requirements on some platforms:

- **Consider using the recommended tools on https://privacyguides.org [Archive.org] for better privacy instead of the usual mainstream ones.**
- **Consider using the recommended tools on https://www.whonix.org/wiki/Documentation [Archive.org] as well instead of the usual mainstream ones such as E-mail providers: https://www.whonix.org/wiki/E-Mail#Anonymity_Friendly_Email_Provider_List [Archive.org]**

The following overview does not mention the privacy practices of those platforms but only their requirements for registering an account. If you want to use privacy-aware tools and platforms, head on to https://privacyguides.org [Archive.org].

Legend:

- "Unclear": Unclear due to lack of information or confusing information.
- "Maybe": It did happen in a minority of my tests.
- "Likely": It did happen in most of my tests.
- "Yes" or "No": This either happened or never happened systematically in all my tests.
- "Easy": The overall experience was straightforward with little to no obstacles.
- "Medium": The overall experience has some obstacles, but it is still doable without too much hassle.
- "Hard": The overall experience is a painful struggle with many obstacles.
- "N/A": Not Applicable because it was not possible to test within the context of this guide
- "Indirectly": This means they do require something but indirectly through a third-party system (Financial KYC for example).
- **See The Real-Name System for essential information. See below for details.**

Below you'll find a list of "problematic services". If they're not below, it means there are no issues at all with anything (like Briar for example)

Amazon

- Is this against their ToS? No, but yes https://www.amazon.com/gp/help/customer/display.html?nodeId=202140280 [Archive.org]

"1. Amazon Services, Amazon Software

A. Use of Amazon Services on a Product. To use certain Amazon Services on a Product, you must have your own Amazon.com account, be logged in to your account on the Product, **and have a valid payment method associated with your account.** "

While it does not technically require a real name. It does require a valid payment method. Unfortunately, it will not accept "cash" or "Monero" as a payment method. So instead, they are relying on financial KYC (where a real-name policy is pretty much enforced everywhere).

- Will they require a phone number? Yes, but see below
- Can you create accounts through Tor? Yes, but see below

Because of this valid payment method requirement, we could not test this. While this is seemingly not against their ToS, it is not possible within the context of this guide unless you manage to obtain a valid KYC payment method anonymously which AFAIK is pretty much impossible or extremely difficult.

So, AFAIK, it is not possible to create an anonymous Amazon account.

Apple

- Is this against their ToS? Yes https://www.apple.com/legal/internet-services/icloud/en/terms.html [Archive.org]

"IV. Your Use of the Service

A. Your Account

In order to use the Service, you must enter your Apple ID and password to authenticate your Account. **You agree to provide accurate and complete information when you register with, and as you use, the Service ("Service Registration Data"), and you agree to update your Service Registration Data to keep it accurate and complete".**

- Will they require a phone number? Yes
- Can you create accounts through Tor? Yes

Note that this account will not allow you to set up an Apple mail account. For that, you will need an Apple device.

Binance

- Is this against their ToS? Yes https://www.binance.com/en/terms [Archive.org]
- Will they require a phone number? No, they do require an e-mail
- Can you create accounts through Tor? No

Discord

- Is this against their ToS? No https://discord.com/terms [Archive.org]
- Will they require a phone number? No, but they do require an e-mail
- Can you create accounts through Tor? We 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/Proxy over Tor to mitigate such issues.

Element

- Is this against their ToS? No https://element.io/terms-of-service [Archive.org]
- Will they require a phone number? No, they do not even require an e-mail
- Can you create accounts through Tor? Yes

Expect some Captchas during account creation on some homeservers.

Facebook

- Is this against their ToS? Yes https://www.facebook.com/terms.php [Archive.org]

"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 is very difficult and their onion address will not help. In most cases, you'll just have a random error at sign-up and your account suspended after sign-in."

But this clause of their ToS is illegal in Germany (see Requirements).

Facebook is one of the most aggressive platforms with identity verification and is pushing hard their "real name policy". It is why this guide is only advised to German residents.

Over our tests tho we were able to pinpoint a few tips:

- It will be easier if you have an Instagram account first.
- Signing up through Tor is almost impossible (even using their .onion address which is a joke) and will only succeed if you are " very lucky" (I assume if you are using an exit node that is not yet known by Facebook verification systems). In most cases, 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 must be ready for a lot of trial and error here.
- Signing up through a Self-Hosted VPN/Proxy is your best bet but make sure your profile/identity matches the IP geolocation.
- My earlier 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 fewer issues if you are 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 trigger an account suspension for violating Community Guidelines or Terms of Services (despite you not using the account at all for anything else than signing-up/logging-in). Ideally, you should log-in back with the same IP from a self-hosted VPN/Proxy.

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 quite 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 to send various documents which require far less Photoshop skills.

It is also possible that they ask you to take a selfie video or picture-making certain gestures to prove your identity. If that is the case, we are afraid it is a dead-end for now unless you use a deepfake face swapping technique.

If you do file an appeal, you will have to wait for Facebook to review it (I do not know whether 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 [Archive.org]
- 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.

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.

GitLab

- Is this against their ToS? No https://about.gitlab.com/handbook/legal/subscription-agreement/ [Archive.org]

- Will they require a phone number? Nope, all good
- Can you create accounts through Tor? Yes, but expect captchas

GitLab is straightforward and requires no phone number.

Google

- Is this against their ToS? No https://policies.google.com/terms [Archive.org]
- 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 be required

Proton is good ... but to appear less suspicious, it is simply better to also have a mainstream Google Mail account.

As Proton, Google will also most likely require a phone number during sign-up as part of their verification process. However contrary to Proton, Google will store that phone number during the sign-up process and will also limit the number of accounts that can be created during the sign-up'.

From my experience during my research, this count is limited to three accounts/phone numbers. 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. Do not 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:

- **(Trick)** Log into Google Mail on desktop and go into the Gmail Quick Settings > See all Setting > Forwarding and POP/IMAP > Add a forwarding address > Verify (using Proton) > 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 Proton 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 Proton instead. This trick will allow you to receive all the e-mails from your Gmail addresses on your Proton (or other) address without needing to login into your Google accounts (reducing risks of it being suspended, especially if you use Tor).
- Enable 2FA within the Google account settings. First, you will have to enable 2FA using the phone number. Then you will 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 is done, remove the phone 2FA from the Google account. This will prevent someone from using that phone number in the future (when you do not have it anymore) to recover/gain access to that account.
- Add Proton 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 do not touch it unless needed (as mentioned, you will use Proton 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 Proton) 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 (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 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 the 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 will not be able to use it to sign-up on a different account. Be careful when using those to avoid losing them. They are precious.

It is also possible that Google will require an ID check through indirect financial KYC or ID picture check if you try to access/publish mature content on their platform.

Instagram

- Is this against their ToS? **Maybe?** We are not sure https://help.instagram.com/581066165581870?ref=dp [Archive.org]

"You can't impersonate others or provide inaccurate information. You do not 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 are not, and you can't create an account for someone else unless you have their express permission".

This one is a bit of an Oxymoron don't you think? So, we are not sure whether it is allowed or not.

- 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 be required

It is 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 is the case, we are afraid it is a dead-end for now.

It is no secret that Instagram is part of Facebook however it is more lenient than Facebook when it comes to user verification. It is quite unlikely you will get suspended or banned after signing up. But it could help.

For instance, we noticed that you will face fewer issues creating a Facebook account if you already have a valid Instagram account. You should always create an Instagram account before trying Facebook.

Unfortunately, there are some limitations when using the web version of Instagram. For instance, you will not be able to enable Authenticator 2FA from the web for a reason we do not know.

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

Jami

- Is this against their ToS? No https://jami.net/privacy-policy/ [Archive.org]
- Will they require a phone number? No, they do not even require an e-mail
- Can you create accounts through Tor? Nope it does not work for some technical reason

Kraken

- Is this against their ToS? Yes https://www.kraken.com/legal [Archive.org]
- Will they require a phone number? No, they do require an e-mail
- Can you create accounts through Tor? Yes

LinkedIn

- Is this against their ToS? Yes https://www.linkedin.com/legal/user-agreement [Archive.org]

"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 sixteen. "

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 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 corporation 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 to go through a verification process that will, unfortunately, require you to send an ID proof (identity card, passport, driver's license). This ID verification is processed by a company called Jumio 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 are a student/intern/freelance) or pretend you work for a large public institution that is too large for anyone to care or 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

MailFence

- Is this against their ToS? No
- Will they require a phone number? No, but they require an e-mail
- Can you create accounts through Tor? Maybe. From my tests, the signing-up verification e-mails are not sent when using Tor to sign-up. No issues however when using a VPN over Tor or a Proxy over Tor.

Medium

- Is this against their ToS? No, unless it is about crypto https://policy.medium.com/medium-terms-of-service-9db0094a1e0f [Archive.org]
- Will they require a phone number? No, but they require an e-mail
- Can you create accounts through Tor? No issues with that so far

Signing-in does require an e-mail every time.

Microsoft

- Is this against their ToS? Yes https://www.microsoft.com/en/servicesagreement/ [Archive.org]

"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? Likely but not always. Depending on your luck with your Tor exit node, they may only require e-mail verification. If you use a VPN over Tor, they will likely only ask for an e-mail.
- Can you create accounts through Tor? Yes, you can but expect captchas, at least e-mail verification, **and likely phone verification.**

So yes, it is still possible to create an MS account without a phone number and using Tor or VPN, but you might have to cycle through a few exit nodes to achieve this.

After signing up you should set up 2FA authentication within the security options and using KeePassXC TOTP.

OnlyFans

- Is this against their ToS? No, it looks fine https://onlyfans.com/terms [Archive.org]
- Will they require a phone number? No, they do require an e-mail
- Can you create accounts through Tor? Yes, you can

Unfortunately, you will be extremely limited with that account and to do anything you will need dot complete their verification process which requires a KYC type financial transaction check. So, not very useful.

Proton

- Is this against their ToS? No https://proton.me/legal/terms [Archive.org]
- Will they require a phone number? Maybe. This depends on the IP you are coming from. If you come from Tor, it is likely. From a VPN, it is less likely.
- Can you create accounts through Tor? Yes, but highly likely that a phone number will be required when only an e-mail or a captcha will be required over a VPN. They even have a ".onion" address at http://protonmailrmez3lotccipshtkleegetolb73fuirgj7r4o4vfu7ozyd.onion/.

You obviously need an e-mail for your online identity and disposable e-mails are pretty much banned everywhere.

Proton is a free e-mail provider based in Switzerland that advocates security and privacy.

They are recommended by Privacyguides.org. 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. If their claim is true and the hash is not linked to your account, and that you followed my guide about the phone number, you should be reasonably safe from tracking.

This e-mail account can be used for creating a Google/Gmail account.

Reddit

- Is this against their ToS? No https://www.redditinc.com/policies [Archive.org]
- Will they require a phone number? No, they will not.
- Can you create accounts through Tor? Yes

Reddit is simple. All you need to register is a valid username and a password. Normally they do not even require an e-mail (you can skip the e-mail when registering, leaving it blank).

No issues whatsoever signing up over Tor or VPN besides the occasional Captchas.

Consider reading this reddit post: https://old.reddit.com/r/ShadowBan/comments/8a2gpk/an_unofficial_guide_on_how_to_avoid_being/ [Archive.org]

Slashdot

- Is this against their ToS? Yes https://slashdotmedia.com/terms-of-use/ [Archive.org]

"8. Registration; Use of Secure Areas and Passwords

Some areas of the Sites may require you to register with us. When and if you register, you agree to (a) provide accurate, current, and complete information about yourself as prompted by our registration form (including your e-mail address) and (b) to maintain and update your information (including your e-mail address) to keep it accurate, current, and complete. You acknowledge that should any information provided by you be found to be untrue, inaccurate, not current, or incomplete, we reserve the right to terminate this Agreement with you and your current or future use of the Sites (or any portion thereof)".

- Will they require a phone number? No
- Can you create accounts through Tor? Yes

Telegram

- Is this against their ToS? No https://telegram.org/tos [Archive.org]
- 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 log in.

It will require a phone number (that can only be used once) and nothing else.

In most cases, we had no issues whether it was over Tor or VPN, but we had a few cases where our 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 we had no success with getting any answer.

Their appeal process is just sending an e-mail to recover@telegram.org [Archive.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

Tutanota

- Is this against their ToS? No https://tutanota.com/terms/ [Archive.org]
- Will they require a phone number? No, but they do require an e-mail.
- Can you create accounts through Tor? Not really, almost all Tor Exit nodes are banned AFAIK

Twitter

- Is this against their ToS? No https://twitter.com/en/tos
- Will they require a phone number? Extremely likely, possibly now a requirement in all cases.
- Can you create accounts through Tor? Yes, but expect some captchas and your phone number will be required after a while.

Twitter is extremely aggressive in preventing anonymity on its 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, the Twitter verification system will call the phone no matter what. In that case, you will have to pick up and hear the verification code. We suspect this is another method of preventing automated systems and malicious companies or entities 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
- **Caution:** Remove the phone number from the account (at your own risk, this often leads to suspension of the account)
- Follow some people based
- Log out and leave it be.

After about a week, you should check 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. We call this the "Double-tap".

This time you will need to submit an appeal using a form, provide a good reason and wait for the appeal to be processed by Twitter. During that process, you may receive an e-mail (on Proton) asking you to reply to a customer service ticket to prove that you do have access to your e-mail and that it is you. This will be directed toward your Gmail address but will arrive on your Proton.

Do not reply from Proton as this will raise suspicions, you must sign in to Gmail (unfortunately) and compose a new mail from there copy-pasting the E-Mail, Subject, and Content from Proton. As well as a reply confirming you have access to that e-mail.

After a few days, your account should get unsuspended "for good". 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. Do not use this account for trolling.

Twitch

- Is this against their ToS? No https://www.twitch.tv/p/en/legal/terms-of-service/ [Archive.org]
- Will they require a phone number? No, but they do require an e-mail.
- Can you create accounts through Tor? Yes

Note that you will not be able to enable 2FA on Twitch using only e-mail. This feature requires a phone number to enable.

WhatsApp

- Is this against their ToS? **Yes** https://www.whatsapp.com/legal/updates/terms-of-service-eea [Archive.org]

"**Registration**. You must register for our Services **using accurate information**, provide your current mobile phone number, and, if you change it, update your mobile phone number using our in-app change number feature. You agree to receive text messages and phone calls (from us or our third-party providers) with codes to register for our Services".

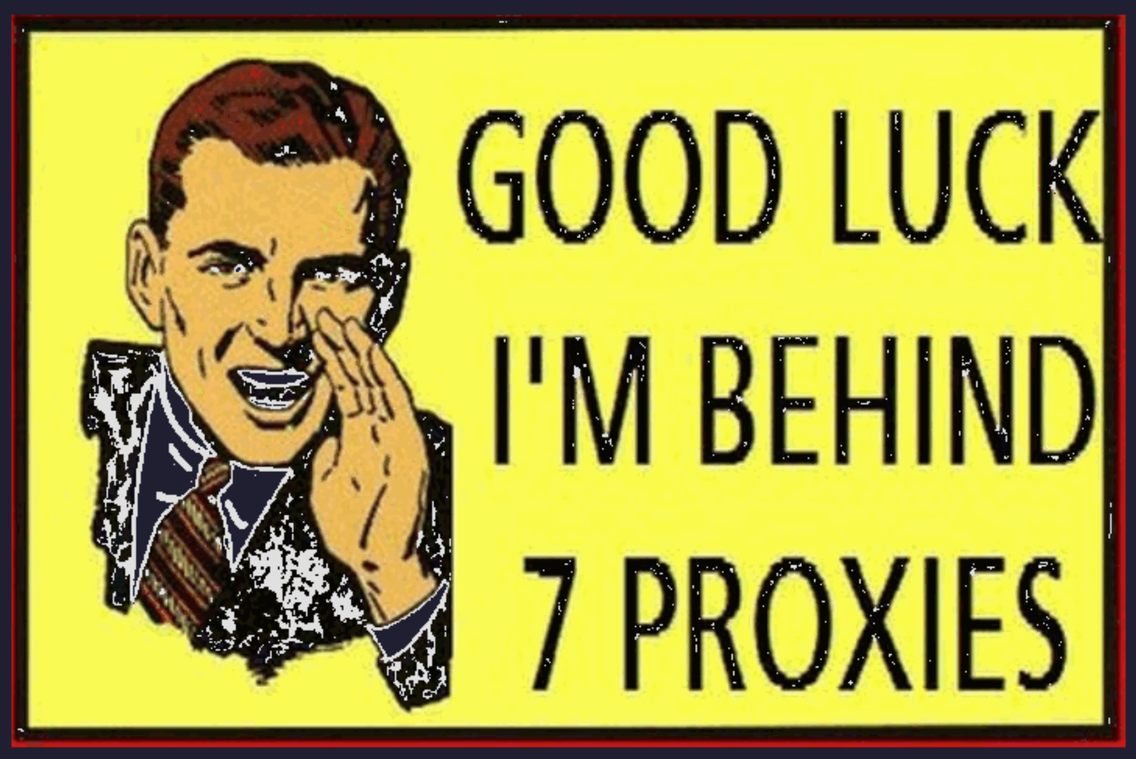
- Will they require a phone number? Yes, they do.
- Can you create accounts through Tor? No issues with that so far.

4chan

- Is this against their ToS? No
- Will they require a phone number? No, they will not.
- Can you post there with Tor or VPN? Not likely.

4chan is 4chan ... This guide will not explain 4chan to you. They block Tor exit nodes and known VPN IP ranges.

You are going to have to find a separate way to post there using at least seven proxies 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 cases NOT anonymous and can be traced back to you when you buy/sell any (remember the Your Crypto 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 we would recommend). Using a VPN on your phone is not one of those ways. All of those are, unfortunately, "tedious" to say the least.

- Use an Android Emulator within the Windows VM and run the App through your multi-layer of Tor/VPN. The 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/ [Archive.org]. As for myself, we will recommend the use of:
  - Android-x86 on Virtualbox (see https://www.android-x86.org/documentation/virtualbox.html [Archive.org]) that you can also set up easily.
  - AnBox (https://anbox.io [Archive.org]) that you can also set up rather easily including on the Whonix Workstation, see https://www.whonix.org/wiki/Anbox [Archive.org]
- **Not recommended:** Using a non-official app (such as Wassapp for WhatsApp) to connect from the Windows VM to the app. Use at your own risk as you could get banned for violating the terms of services by using a non-official App.
- **Not 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. We will not detail this here, but it is an option.

There is no way to reliably set a decent multi-layered connectivity approach easily on an Android phone (it is not even possible on IOS as far as we know). By reliable, we mean being sure that the smartphone will not 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.

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 will not help you there as this would be illegal in most places.

How to Chat & Share Files Anonymously

There are plenty of messaging apps everywhere. Some have excellent UI and UX and terrible Security/Privacy. Some have excellent Security/Privacy but terrible UI and UX. It is not easy to pick the ones that you should use for sensitive activities. So, this section will help you do that.

Before going further, there are also some key basic concepts you should understand:

E2E Encryption

End-to-end Encryption (aka e2ee) is a rather simple concept. It just means only you and your destination know each-others public encryption keys and no one in between that would be eavesdropping would be able to decrypt the communication.

However, the term is often used differently depending on the provider:

- Some providers will claim e2ee but forget to mention what is covered by their protocols. For instance, is metadata also protected within their e2ee protocol? Or is it just the content of the messages?
- Some providers do provide e2ee but only as an opt-in option (disabled by default).
- Some providers do offer e2ee with 1 to 1 messaging but not with group messaging.
- Some providers will claim the use of e2ee, but their proprietary apps are closed source where no one can verify the claim and the strength of the encryption used.

For these reasons, it is always important to check the claims of various apps. Open-Source apps should always be preferred to verify what kind of encryption they are using and if their claims are true. If not open source, such apps should have an openly available independent (made by a reputable third party) report confirming their claims.

Roll your own crypto

See the Bad Cryptography section at the start of this guide.

Always be cautious of apps rolling their own crypto until it has been reviewed by many in the crypto community (or even better published and peer-reviewed academically). Again, this is harder to verify with closed-source proprietary apps.

It is not that rolling your own crypto is bad in essence, it is that good cryptography needs real peer-reviewing, auditing, testing... And since you are probably not a cryptanalyst (and we are not either), chances are high we are not competent to assess the cryptography of some apps.

Forward Secrecy

Forward Secrecy (FS aka PFS for Perfect Forward Secrecy) is a property of the key agreement protocol of some of those messaging apps and is a companion feature of e2ee. This happens before you establish communication with the destination. The "Forward" refers to the future in time and means that every time you establish a new e2ee communication, a new set of keys will be generated for that specific session. The goal of forward secrecy is to maintain the secrecy of past communications (sessions) even if the current one is compromised. If an adversary manages to get hold of your current e2ee keys, that adversary will then be limited to the content of the single session and will not be able to easily decrypt past ones.

This has some user experience drawbacks like for instance, a new device could not be able to conveniently access the remotely stored chat history without additional steps.

So, in short, Forward Secrecy protects past sessions against future compromises of keys or passwords.

More on this topic on this YouTube video: https://www.youtube.com/watch?v=zSQtyW_ywZc [Invidious]

Some providers and apps claiming to offer e2ee do not offer FS/PFS sometimes for usability reasons (group messaging for instance is more complex with PFS). It is therefore important to prefer open-source apps providing forward secrecy to those that do not.

Zero-Access Encryption at rest

Zero-Access Encryption at rest is used when you store data at some provider (let us say your chat history or chat backups) but this history or backup is encrypted on your side and cannot be read or decrypted by the provider hosting it.

Zero-Access encryption is an added feature/companion to e2ee but is applied mainly to data at rest and not communications.

Examples of this issue would be iMessage and WhatsApp, see the Your Cloud Backup & Sync Services at the start of this guide.

So again, it is best to prefer Apps/Providers that do offer Zero-Access Encryption at rest and cannot read/access any of your data/metadata even at rest and not only limited to communications.

Such a feature would have prevented important hacks such as the Cambridge Analytica scandal if it were implemented.

Metadata Protection

Remember the Your Metadata section (including geo-location). End-to-end Encryption is one thing, but it does not necessarily protect your metadata.

For Instance, WhatsApp might not know what you are saying but they might know who you are talking to, how long and when you have been talking to someone, who else is in groups with you, and if you transferred data with them (such as large files).

End-to-end Encryption does not in itself protect an eavesdropper from harvesting your metadata.

This data can also be protected/obfuscated by some protocols to make metadata harvesting substantially harder for eavesdroppers. This is the case for instance with the Signal Protocol which does offer some added protection with features like:

- The Sealed Sender option.
- The Private Contact Discovery.
- The Private Group System.

Other Apps like Briar or OnionShare will protect metadata by using the Tor Network as a shield and storing everything locally on-device. Nothing is stored remotely, and all communications are either direct using proximity wi-fi/Bluetooth or remotely through the Tor network.

Most apps however and especially closed-source proprietary commercial apps will collect and retain your metadata for various purposes. And such metadata alone is enough to figure out a lot of things about your communications.

Again, it is important to prefer open-source apps with privacy in mind and various methods in place to protect not only the content of communications but all the associated metadata.

Open-Source

Finally, Open-Source apps should always be preferred because they allow third parties to check actual capabilities and weaknesses vs claims of marketing departments. Open-Source does not mean the app should be free or non-commercial. It just means transparency.

App[0]	e2ee[1]	Roll Your Own Crypto	Perfect Forward Secrecy	Zero-Access Encryption at-rest[5]	Metadata Protection (obfuscation, encryption…)	Open-Source	Default Privacy Settings	Native Anonymous Sign-up (no e-mail or phone)
Berty (avoid)	Yes	No	Yes	Yes	Yes	Yes [10]	Good	Yes
Briar (preferred)	Yes	No [1]	Yes	Yes	Yes (strong)	Yes	Good	Yes
Cwtch (preferred)	Yes	No	Yes	Yes	Yes (strong)	Yes	Good	Yes
Discord (avoid)	No	Closed-source[7]	No	No	No	No	Bad	E-Mail Required
Element / Matrix.org (preferred)	Yes (opt-in)	No	Yes	Yes	Poor[2]	Yes	Good	Yes
Facebook Messenger (avoid)	Partial (Only 1to1 / opt-in)	Closed-source[7]	Yes	No	No	No	Bad	E-Mail and Phone required
OnionShare (preferred)	Yes	No	TBD[8]	TBD[8]	Yes (strong)	Yes	Good	Yes
Apple Messages (aka iMessage)	Yes	Closed-source[7]	No	Partial	No	No	Good	Apple device Required
IRC	Yes (OTR plugins)	No	No	No	No	Yes	Bad	Yes
Jami (preferred)	Yes	No[3]	Yes	Yes	Partial	Yes	Good	Yes
KakaoTalk (avoid)	Yes	Closed-source[7]	No[4]	No	No	No	Bad	No (but possible)
Keybase	Yes	No	Partial (exploding message)	No	No	Yes	Good	E-Mail Required
Kik (avoid)	No	Closed-source[7]	No	No	No	No	Bad	No (but possible)
Line (avoid)	Partial (opt-in)	Closed-source[7]	No	No	No	No	Bad	No (but possible)
Pidgin with OTR (avoid)	Yes (OTR[3])	No	Yes	No	No	Yes	Bad	Yes
Tox (avoid)	Yes	No	No	No	No	Yes	Good	Yes

App[0]	e2ee[1]	Roll Your Own Crypto	Perfect Forward Secrecy	Zero-Access Encryption at-rest[5]	Metadata Protection (obfuscation, encryption…)	Open-Source	Default Privacy Settings	Native Anonymous Sign-up (no e-mail or phone)
Session (Preferred only on iOS)	Yes	No	No	Yes	Yes	Yes	Good	Yes
Signal	Yes	No	Yes	Yes	Yes (moderate)	Yes	Good	Phone Required
Skype (avoid)	Partial (Only 1to1 / opt-in)	Closed-source[7]	No	No	No	No	Bad	No (but possible)
SnapChat (avoid)	No	Closed-source[7]	No	No	No	No	Bad	No (but possible)
Teams (avoid)	Yes	Closed-source[7]	No	No	No	No	Bad	No (but possible)
Telegram	Partial (Only 1to1 / opt-in)	Yes (MTProto[8])	Partial (secret chats only)	Yes	No	Partial[5]	Medium (e2ee off by default)	Phone Required
Viber (avoid)	Partial (Only 1to1)	Closed-source[7]	Yes	No	No	No	Bad	No (but possible)
WeChat (avoid)	No	Closed-source[7]	No	No	No	No	Bad	No
WhatsApp (avoid)	Yes	Closed-source[7]	Yes	No	No	No	Bad	Phone Required
Wickr Me	Partial (Only 1to1)	No	Yes	No	Yes (moderate)	No	Good	Yes
Gajim (XMPP) (preferred)	Yes	No	Yes	No	No	Yes	Good	Yes
Zoom (avoid[10])	Disputed[11]	No	TBD[8]	No	No	No	Bad	E-Mail Required
Molly	Yes	No	Yes	Yes	Yes (moderate)	Yes	Good	Phone Required

Legend:

1. The mention "preferred" or "avoid" refers to the use of those apps for sensitive communications. This is just my opinion, and you can make your own using the resources above and others. Remember "Trust but verify".
2. e2ee refers to "end-to-end encryption"
3. Additional steps might be needed for securing Tor Connectivity
4. Their ability and willingness to fight for privacy and not cooperate with various adversaries
5. Only the client apps are open-source, not the server-side apps
6. This means the data is fully encrypted at rest (and not only during transit) and unreadable by any third party without a key you only know (including backups)
7. Unverifiable because it is proprietary closed source.
8. To Be Determined, unknown at the time of this writing
9. Jami will require you to enable DHTProxy in their options to work and it will be limited to text only.
10. Session also uses their own Onion Routing solution called LokiNet

Some apps like Threema and Wire were excluded from this comparison due to not being free and not accepting anonymous cash methods such as Cash/Monero.

Conclusion

Remember: Checklist of things to verify before sharing information.

We will recommend these options in that order (as also recommend by Privacyguides.org' except for Session and Cwtch):

- macOS:
  - Native Tor Onion Routing Support (**preferred**):
    - OnionShare version >2.3 (https://onionshare.org/ [Tor Mirror] [Archive.org])**
    - Cwtch (https://cwtch.im [Archive.org] **warning, this is at the alpha/beta stage**)**
  - Non-Native Tor Support (needs additional steps for ideal anonymity to proxy it through Tor through Virtualization or Proxying):
    - Element/Matrix.org (https://element.io/ [Archive.org])
    - Jami (https://jami.net/ [Archive.org])*
    - Gajim/XMPP (https://gajim.org/ [Archive.org])
- Windows:
  - Native Tor Onion Routing Support (**preferred**):
    - OnionShare version >2.3 (https://onionshare.org/ [Tor Mirror] [Archive.org])**
    - Cwtch (https://cwtch.im [Archive.org] **warning, this is at the alpha/beta stage**)**
  - Non-Native Tor Support (needs additional steps for ideal anonymity to proxy it through Tor through Virtualization or Proxying):
    - Element/Matrix.org (https://element.io/ [Archive.org])
    - Jami (https://jami.net/ [Archive.org])*
    - Gajim/XMPP (https://gajim.org/ [Archive.org])
- Linux:
  - Native Tor Onion Routing Support (**preferred**):
    - Briar (https://briarproject.org/ [Archive.org])*
    - OnionShare version >2.3 (https://onionshare.org/ [Tor Mirror] [Archive.org])**
    - Cwtch (https://cwtch.im [Archive.org] **warning, this is at the alpha/beta stage**)**

- Non-Native Tor Support (needs additional steps for ideal anonymity to proxy it through Tor through Virtualization or Proxying):
  - Element/Matrix.org (https://element.io/ [Archive.org])
  - Jami (https://jami.net/ [Archive.org])*
  - Gajim/XMPP (https://gajim.org/ [Archive.org])

- Note that for Jami to work over Tor, you will have to enable the local DHTProxy option within Jami Settings. This will only work for text messages and not for calls/videos)

** Note that these options (Briar, Cwtch, and OnionShare) do not support multi-devices yet. Your information is strictly stored on the device/OS where you are setting it up. Do not use those on a non-persistent OS unless you want ephemeral use.

Any safe options for mobile devices? **Yes, but these are not endorsed/recommended except Briar on Android. Remember also that this guide discourages the use of smartphones for sensitive activities in general.**

- Android:
  - Briar (https://briarproject.org/ [Archive.org])
  - Cwtch (https://cwtch.im [Archive.org] **warning, this is at the alpha/beta stage)**
- iOS:
  - Due to the lack of any better option and while it is **normally not recommended**: Session Messenger: https://getsession.org/ [Archive.org]. Why is it not recommended these days within the privacy community? **See:** Caution about Session messenger **to find out why we are cautious about Session Messenger**.

Note that all the non-native Tor options must be used over Tor for safety (from Tails or a guest OS running behind the Whonix Gateway such as the Whonix Workstation or an Android-x86 VM).

WhileWedo not recommend most of the messaging platforms for the various reasons outlined above (phone number and e-mail requirements), this does not mean it is not possible to use them anonymously if you know what you are doing. You can use even Facebook Messenger anonymously by taking the necessary precautions outlined in this guide (virtualization behind a Tor Gateway on a non-persistent OS).

The ones that are preferred are recommended due to their stance on privacy, their default settings, their crypto choices but also because they allow convenient anonymous sign-up without going through the many hassles of having a phone number/e-mail verification method and are open source. Those should be privileged in most cases.

You can also consult the following external resources for more comparisons (**we do not necessarily endorse their opinions**):

- SecuChart, https://bkil.gitlab.io/secuchart/ [Archive.org] [Repository] (Maintained open-source project)
- Wikipedia, https://en.wikipedia.org/wiki/Comparison_of_cross-platform_instant_messaging_clients [Wikiless] [Archive.org]
  - Wikipedia, https://en.wikipedia.org/wiki/Comparison_of_instant_messaging_protocols [Wikiless] [Archive.org]
- Whonix Documentation, Instant Messenger Chat https://www.whonix.org/wiki/Chat [Archive.org] (Outdated, Unmaintained but contains insightful information)
- **Outdated, or unmaintained, or abandoned resources scheduled for removal from our guide in next release:**
  - ~~Secure Messaging Apps https://www.securemessagingapps.com/ [Archive.org]~~
  - ~~Proton Blog, https://proton.me/blog/whatsapp-alternatives/ [Archive.org]~~
  - ~~SecureChart.org, https://securechatguide.org/featuresmatrix.html [Archive.org]~~
  - ~~Messenger-Matrix.de at https://www.messenger-matrix.de/messenger-matrix-en.html [Archive.org]~~

We do not endorse or recommend some mainstream platforms for anonymity including the much-praised Signal which to this date still requires a phone number to register and contact others. In the context of this guide, we strongly recommend against using Signal if possible. The same recommendation applies to popular forks of Signal such as Molly (https://molly.im [Archive.org])

How to share files publicly

Warning: before sharing anything publicly, make sure your files are curated of any information that could compromise your identity. See Checklist of things to verify before sharing information.

Consider the following platforms:

- Cryptpad.fr (https://cryptpad.fr/): Free tier limited to 1GB total and recommended by PrivacyGuides.org at https://privacyguides.org/cloud/ [Archive.org]
- Proton Drive (https://proton.me/drive/): Paid. Requires you to have "Proton Unlimited" or "Mail Plus". Proton Drive is E2EE and recommended by PrivacyGuides.org
  - Like Proton and Proton VPN, it's not easy to sign up anonymously. When you try to register through Tor, they request verification either by phone number, or by providing a donation
- Filen (https://filen.io/): free tier limited to 10GB total

Consider the use of IPFS:

- Pinata (https://www.pinata.cloud/): Free tier limited to 1GB total

Redacting Documents Safely

You might want to self-publish some information safely and anonymously in the form of writing, pictures, videos, ...

For all these purposes here are a few recommendations:

- Ideally, you should not use proprietary software such as Adobe Photoshop, Microsoft Office...
- Preferably, you should use open-source software instead such as LibreOffice, Gimp...

While the commercial alternatives are feature-rich, they are also proprietary closed-source and often have various issues such as:

- Sending telemetry information back to the company.
- Adding unnecessary metadata and sometimes watermarks to your documents.
- These apps are not free, and any leak of any metadata could be traced back to you since you had to buy these somewhere.

It is possible to use commercial software for making sensitive documents, but you should be extra careful with all the options in the various Apps (commercial or free) to prevent any data leak from revealing information about you.

Here is a comparative table of recommended/included software compiled from various sources (PrivacyGuides.org, Whonix, Tails, Prism-Break.org, and me). Keep in mind my recommendation considers the context of this guide with only sporadic online presence on a need basis.

Type	Whonix	Prism-Break.org	PrivacyGuides.org	Tails	This guide
Offline Document Editing	LibreOffice	N/A	LibreOffice*	LibreOffice	LibreOffice, Notepad++
Online Document Editing (collaboration)	N/A	Cryptpad.fr	Cryptpad.fr, Etherpad.org, Privatebin.net	N/A	Cryptpad.fr, Etherpad.org, Privatebin.net
Pictures Editing	Flameshot (L)	N/A	N/A	GIMP	GIMP
Audio Editing	Audacity	N/A	N/A	Audacity	Audacity
Video Editing	Flowblade (L)	N/A	N/A	N/A	Flowblade (L) Olive (?) OpenShot (?) ShotCut (?)
Screen Recorder	Vokoscreen	N/A	N/A	N/A	Vokoscreen
Media Player	VLC	N/A	N/A	VLC	VLC
PDF Viewer	Ristretto (L)	N/A	N/A	N/A	Browser

Type	Whonix	Prism-Break.org	PrivacyGuides.org	Tails	This guide
PDF Redaction	PDF-Redact Tools (L)	N/A	N/A	PDF-Redact Tools (L)	LibreOffice, PDF-Redact Tools (L)

Legend: * Not recommended but mentioned. N/A = Not Included or absence of recommendation for that software type. (L)= Linux Only but can maybe be used on Windows/macOS through other means (HomeBrew, Virtualization, Cygwin). (?)= Not tested but open-source and could be considered.

In all cases, we strongly recommend only using such applications from within a VM or Tails to prevent as much leaking as possible. If you do not, you will have to sanitize those documents carefully before publishing (See Metadata auditing).

Communicating sensitive information

You might be interested in communicating information to some organization such as the press anonymously.

If you must do so, you should take some steps because you cannot trust any organization to protect your anonymity. See Checklist of things to verify before sharing information.

For this, we strongly recommend the use of SecureDrop (https://securedrop.org/ [Archive.org]) which is an open-source project from the Freedom of the Press Foundation.

- Do take a moment to their read their "source guide" here: https://docs.securedrop.org/en/stable/source.html [Archive.org]
- Ideally, you should use SecureDrop over Tor and you will find a curated list of those here https://github.com/alecmuffett/real-world-onion-sites#securedrop [Archive.org]

If not SecureDrop is not available, you could consider any other means 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 recipient has published a GPG key somewhere. You can look this up here:
  - On their verified Social Media accounts (Twitter) if they provided it.
  - On https://keybase.io (Tor address http://keybase5wmilwokqirssclfnsqrjdsi7jdir5wy7y7iu3tanwmtp6oid.onion)
  - On OpenPGP directories such as: **(be careful as those are public directories and anyone can upload any key for any e-mail address, you will have to cross-check the signature with other platforms to be sure it is theirs).**
    - https://pgp.mit.edu/
    - https://keyserver.ubuntu.com/
    - https://keys.openpgp.org
- Using any other platform (even Twitter DMs) but again using GPG to encrypt the message for the recipient.

What you should avoid:

- 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 option).

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.

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.
- Check regularly the eventual appearance of compromise of any of your identities using https://haveibeenpwned.com/ [Archive.org] (obviously from a safe environment).

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 a USB key. Here are the various possibilities.

Selected Files Backups

Requirements

For these back-ups, you will need a USB key or an external hard drive with enough storage capacity to store the files you want to back up.

Veracrypt

For this purpose, we 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 will need is to follow Veracrypt tutorial here: https://www.veracrypt.fr/en/Beginner%27s%20Tutorial.html [Archive.org]

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 safely.

Hidden File containers with plausible deniability

The process is also fairly simple and similar to the earlier tutorial except for this time you will 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 us 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 not what you 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 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 [Archive.org]

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 are using. If you do not 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 [Archive.org]. Remember the Local Data Leaks and Forensics section.**
- We 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 is absolutely identical on each key and your computer. See the How to securely delete specific files/folders/data on your HDD/SSD and Thumb drives section of this guide for help on doing this.**
- If you have time, **We will even recommend that you delete wipe the keys completely before making any modification on such containers on your computer (if you do not 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 are 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 before 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. Therefore, we do not recommend using these from your host OS.
- Do not store these on external SSD drives if you are not sure you can 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 are using Native macOS encryption. When using Windows, do not back up 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 backup 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 will have to trust and cannot really control for most). Alternatively, you could encrypt the backup media yourself separately (for instance with Veracrypt). We are 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 will require more work to restore your system in an encrypted state (re-encryption after restoring).
- (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 is 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 will have to re-do a full backup every time).
  - This method is the easiest to manage.

We made extensive testing using live backups utilities (Macrium Reflect, EaseUS Todo Reflect, Déjà Dup...) and personally we do not think it is worth it. Instead, we would recommend that you periodically back up your system with a simple Clonezilla image. It is much easier to perform, much easier to restore, and usually works reliably without issues in all cases. And contrary to many beliefs, it is 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, we recommend using file containers or encrypted media directly and manually as explained in the earlier 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 the 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

We 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 Clonezilla

QUBESOS

Qubes OS recommends using their own utility for backups as documented here https://www.qubes-os.org/doc/backup-restore/ [Archive.org]. But it is just a hassle and provides limited added value unless you just want to back up a single Qube. So instead, we are 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 simple steps.

So, you should follow the steps in Clonezilla

Windows

We 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 we do not see any advantage in using any of them vs Clonezilla.

Some warnings:

- If you use Bitlocker for encryption with TPM 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/SSD 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, we do not recommend using Bitlocker at all.
- You should always have a backup of your Veracrypt rescue disk at hand somewhere to be 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/SSD after a failure, Windows 10/11 may refuse to boot if your hard drive ID is changed. You should also save this ID before backing up as you might need to change the ID of the new drive as Windows 10/11 might require a matching ID before booting. See Diskpart.
- **In case you are 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 is okay to back up the Decoy OS partition without issues, but you should never back up the partition containing the Hidden OS.**

Follow the steps in Clonezilla

macOS

we 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 [Archive.org] and https://support.apple.com/en-ie/guide/mac-help/mh11421/11.0/mac/11.0 [Archive.org].

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 so that it is 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 [Archive.org]

It is just simpler and will work online while you work. You will be able to recover your data on any other Mac from the recovery options and you will be also able to use this disk for backing up other devices.

It is possible to also use Clonezilla to clone your Mac Hard Drive, but it could bring hardware compatibility issues and probably will not add much in terms of security. So, for macOS, We are not specifically recommending Clonezilla.

Online Backups

Files

This is a tricky one. The problem is that it depends on your threat model.

- **TLDR: Do not store file containers with plausible deniability (Veracrypt) online.** If you use containers with plausible deniability, you should never store them on any platform where you do not 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 normally encrypted backups without plausible deniability, you could store them pretty much anywhere if they are properly encrypted locally before uploading (for example with Veracrypt, using strong passphrases and encryption). **Do not ever trust the 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...) if they are strongly encrypted locally before uploading. But you could also prefer privacy caring services such as Cryptpad.fr (1GB).

Obviously do not ever do/access those backups from unsecured/unsafe devices but only from the secure environments, you picked before.

Self-hosting

Self-hosting (using Nextcloud for instance) is also a possibility provided you do have an anonymous hosting

Please see Recommended VPS hosting providers.

Please also consider Monero Disclaimer.

Cloud-hosting

For smaller files, consider:

- Cryptpad.fr (https://cryptpad.fr/): Free tier limited to 1GB total and recommended by PrivacyGuides.org at https://privacyguides.org/cloud/ [Archive.org]
- Filen (https://filen.io/): free tier limited to 10GB total

We are currently not aware of any online storage/hosting platform accepting cash payments unlike providers mentioned before.

If you do intend to store sensitive data on "mainstream platforms" (Dropbox, Google Drive, OneDrive...), **remember not to ever store plausible deniability containers on those and remember to encrypt and check (for metadata...) anything locally before uploading there**. Either with software like Veracrypt or with a software like Cryptomator (https://cryptomator.org/). Do not ever upload non-encrypted files on those platforms and repeating myself, only access them from a secure shielded VM.

Information

If you just want to save information (text), we will recommend the use of secure and private pastebins. Mostly we will stick to the ones recommended by PrivacyGuides.org (https://www.privacyguides.org/productivity/#paste-services [Archive.org]) :

- https://privatebin.info/
- https://cryptpad.fr/pad/

On these providers, you can just create a password-protected pad with the information you want to store.

Just create a pad, protect it with a password and write your info in it. Remember the address of the pad.

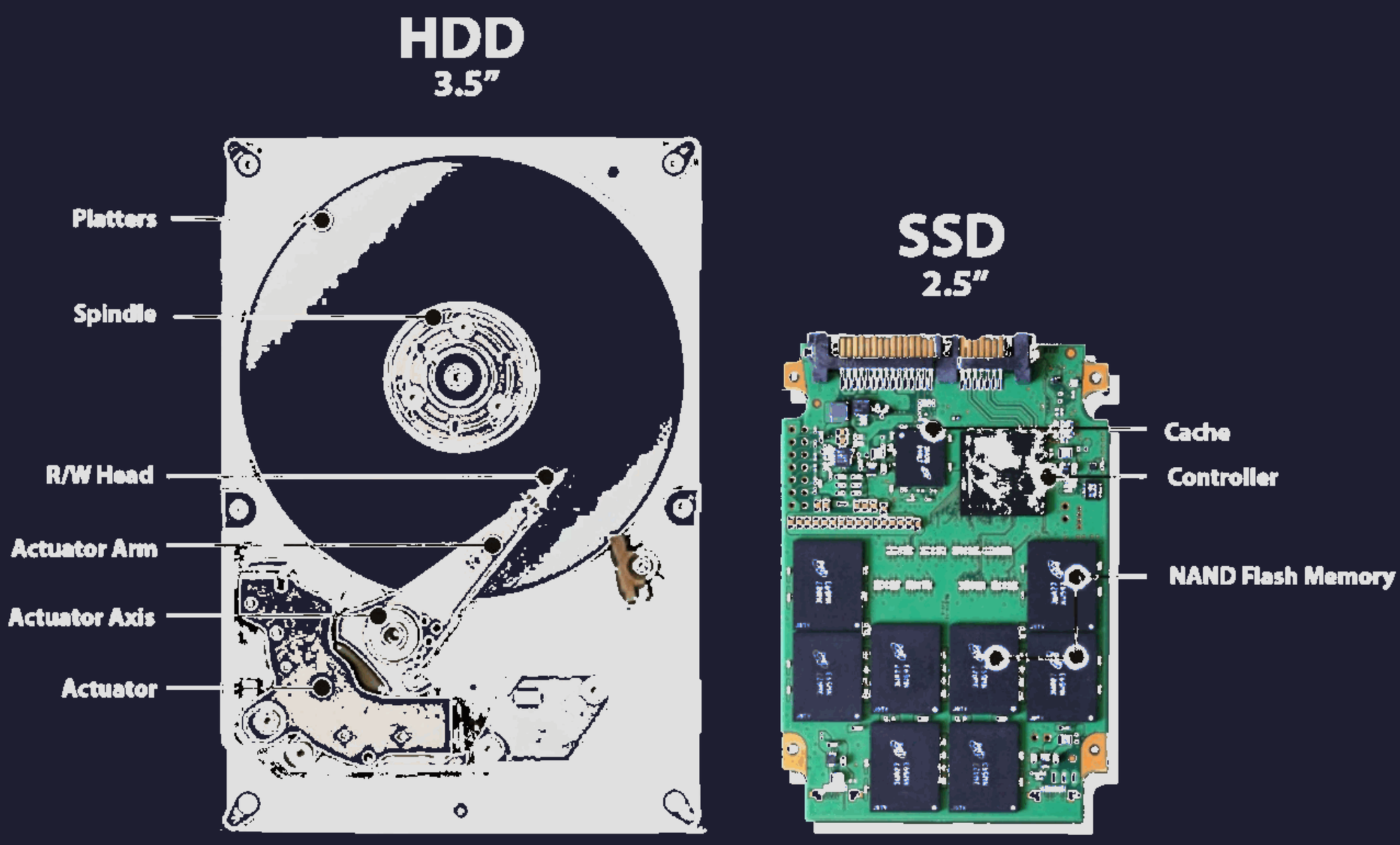
Synchronizing your files between devices Online

To that, the answer is very simple and a clear consensus for everyone: https://syncthing.net/ [Archive.org]

Just use SyncThing, it is the safest and most secure way to synchronize between devices, it is free and open-source, and it can easily be used in a portable way without install from a container that needs syncing.

Covering your tracks

Understanding HDD vs SSD

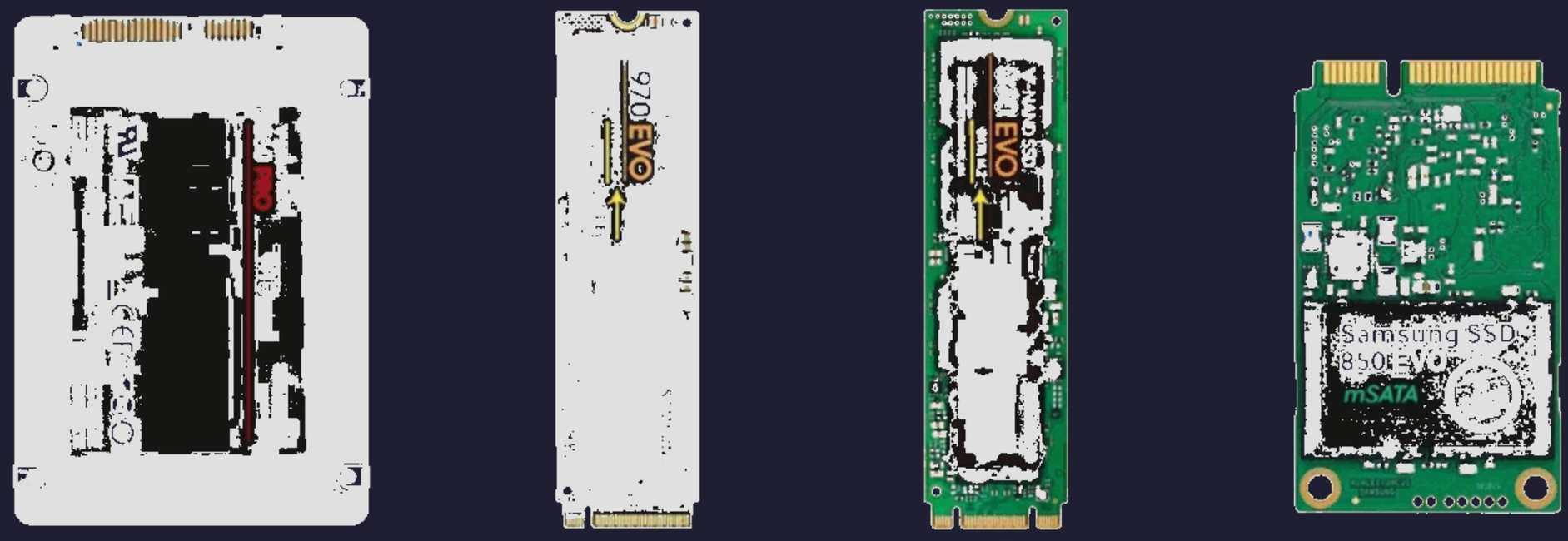


If you intend to wipe your whole HDD laptop, the process is rather straightforward. The data is written at a precise location on a magnetic (hard) platter (why it is called a hard drive) and your OS knows precisely where it is on the platter, where to delete it, and where to overwrite it for secure deletion using simple processes (like just overwriting that location over and over until no traces are left).

On the other hand, if you are using an SSD drive, the process is not as simple as the drive uses several internal mechanisms to extend 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 two 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:



2.5" SATA NVMe M.2 SATA mSATA

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 are using. So, it is important you know which one you have inside 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. At a high level, wear leveling works as follows. The space on every disk is divided into blocks that are themselves divided into pages, like the chapters in a book are made of pages. When a file is written to disk, it is 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 will not 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 number 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 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 does not even realize it has happened. This means, however, that even if you try to overwrite a file, there is 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 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 (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 SSD. Avoid them on an SSD drive.

Trim Operations

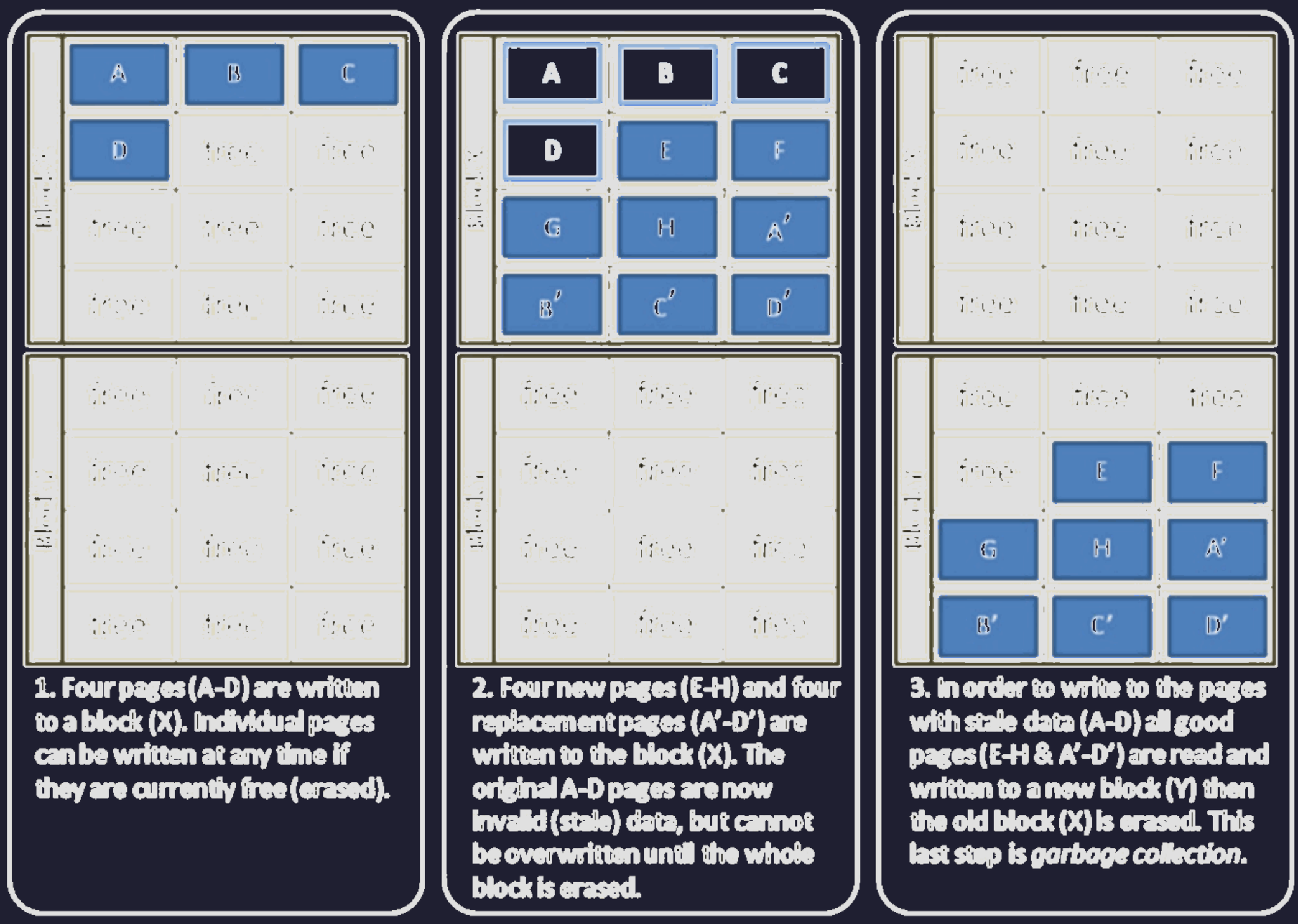
So, what now? Well here comes the Trim operation. When you delete data on your SSD, your OS should support what is called a Trim operation command and **could (should)** issue this Trim command to the SSD drive periodically (daily, weekly, monthly...). This Trim command will then let know the SSD drive controller that there are pages within blocks containing data that are now free to be 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 10/11, Ubuntu, Qubes OS 4.1.x ...).

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 will not see this and will just see free space as you delete files, but your SSD controller will not (this is called Write Amplification). 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 of your 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 four 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) [Wikiless] [Archive.org]; 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.

Garbage Collection

Garbage collection 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.

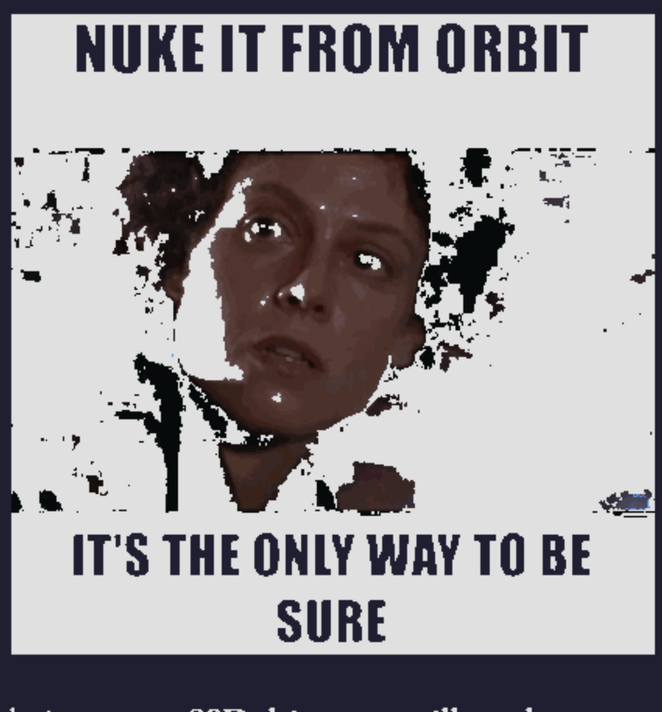
Garbage collection in itself does NOT require Trim to function, but it will be much faster and more efficient if Trim is performed. Garbage collection is one of the processes that will actually erase data from your SSD drive permanently.

Conclusion

So, the fact is that it is very unlikely' and difficult for a forensic examiner to be able to recover data from a Trimmed SSD but it is not completely impossible either'' if they are 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 secure enough on any SSD drive and will be recommended as the standard method of deletion.

How to securely wipe your whole Laptop/Drives



So, you want to be sure. To achieve 100% secure deletion on an SSD drive, you will need to use specific SSD techniques (If you are using an HDD drive, skip this part and go to your OS of choice):

- Easy options for less experienced people:
  - If available, just use the Secure Erase option available from your BIOS/UEFI (ATA/NVME Secure Erase or Sanitize).
    - It's worth noting that this relies on your drive's firmware. Some drive manufacturers have messed up the implementation, causing data to still be recoverable.
  - 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 for 11$ and use it to erase any disk.
- Technical options for more advanced people:
  - Overwrite the entire drive's contents
    - HDDs:
      - Overwrite the drive's contents using a tool like srm, wipe, shred, etc.. Ideally you want to use the Gutmann method, which was created for most effective data erasure on all drives. This method also works on SSDs, although it is overkill.
      - Simply overwriting the drive's contents is not always enough. Dedicated secure deletion tools are designed to perform multiple passes to more effectively wipe data. This is expecially important on older drives. we recommend using either `wipe` or `srm`.
        - If using `wipe`, just use its default options (`wipe /dev/sdX`), as the defaults are tuned to most effectively wipe data on HDDs.
        - If using `srm`, make sure to manually specify that it should perform a Gutmann wipe (`srm -G /dev/sdX`).
    - SSDs:
      - Overwrite the drive's contents. Tools like wipe or shred are often overkill, as they perform up to 35 passes. While they work, most SSDs require no more than a couple passes.
      - Use `wipe` with only a couple passes: `wipe -qQ2 /dev/sdX`.
        - `-qQ2` means 2 passes. Replace `2` with the desired number of passes.
      - Use `srm` with a 3-pass overwrite: `srm -P /dev/sdX`.
      - Use `dd`: `dd if=/dev/urandom of=/dev/sdX bs=8M status=progress conv=fsync`. This command will overwrite the drive with random data. To perform multiple passes (I recommend at least 2), simply run the command again until you're satisfied.
        - The reason you run it twice is because SSDs have hidden ("overprovisioned") storage which can contain remnants of deleted data. Wiping twice forces the drive to wipe its overprovisioned storage. This is only guaranteed to work if each pass writes different data (which is why we wipe with random data on each pass).
        - `bs=8M` writes 8MiB blocks at a time. This doesn't affect the quality of the data deletion, but adjusting it could affect how long it takes to wipe the drive.

    - 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 SSD 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 destroy the data and change the encryption key if present.
    - ATA/NVMe Sanitize Overwrite **(terribly slow, could be dangerous and not recommended)**: This method performs a block erase and then overwrite every storage block (it is the same as Block Erase but will overwrite data in addition). This method is overkill and not necessary.
- Physical Destruction:
    - HDDs:

        a. Open the drive (with a screwdriver, usually Torx T8)

        b. Remove platters (with a screwdriver, usually Torx T6)

        c. Rub the platters with a rare earth magnet

        d. Break/Deform/Crush the platters

        e. Burn the platters or cook them in an oven (**do not** skip this step)

        f. Separate the debris

        g. Throw away in separate places

    - SSDs:
        - Ideally you should wipe the drive through other means first, as this method alone is not known to be secure against all attackers
        - Open the drive
        - Break/Crush the board and memory cells
        - Burn them
        - Separate the debris
        - Throw away in separate places
    - Bonus: See https://www.youtube.com/watch?v=-bpX8YvNg6Y [Invidious]

For maximum overkill paranoia security, Sanitize Block Erase option should be preferred but Secure Erase is probably more than enough when considering your drive is already encrypted. Unfortunately, are no **free** easy (bootable with a graphical menu) all-in-one tools available and you will be left with either going with drive manufacturers provided tools, the free manual hdparm , and nvme-cli 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'.

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 will not be necessary. Check this before going ahead to avoid the hassle, see 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 Using System Rescue to securely wipe an SSD drive

- Option C: Wipe your disk and re-install Linux with new full disk encryption to overwrite all sectors with new encrypted data. **This method will be terribly slow compared to Option A and B as it will slowly overwrite your whole SSD. Also, note that this might not be the default behavior when using LUKS. You might have to check the option to also encrypt the empty space for this effectively wipe the drive.**

Keep in mind all these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, wear-leveling mechanisms might prevent this from working properly.

External SSD

First please see 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 [Archive.org]

If your USB controller and USB SSD disk support Trim and ATA/NVMe secure erase, you could wipe them cautiously using hdparm using the same method as the System Disk above except you will not install Linux on it obviously. Keep in mind tho that this is not recommended (see Considerations above).

If it does not 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 [Archive.org] or from the command line using secure-delete using this tutorial https://superuser.com/questions/19326/how-to-wipe-free-disk-space-in-linux [Archive.org]).

Keep in mind all these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, wear-leveling mechanisms might prevent this from working properly.

Internal/System HDD

- Option A: Check if your BIOS/UEFI has a built-in option and use them and if it does, use the correct option (Wipe + Passes in the case of an HDD).
- Option B: See Using ShredOS to securely wipe an HDD drive
- Option C: Wipe your disk and re-install Linux with new full disk encryption to overwrite all sectors with new encrypted data. **This method will be terribly 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/ [Archive.org]
  - https://linoxide.com/linux-command/commands-wipe-disk-linux/ [Archive.org]
  - https://wiki.archlinux.org/index.php/Securely_wipe_disk [Archive.org]

I recommend using dd or shred for this purpose.

- Option B: Install and use BleachBit https://www.bleachbit.org/download/linux [Archive.org] or follow this EFF tutorial https://ssd.eff.org/en/module/how-delete-your-data-securely-linux [Archive.org]
- Option C: See Using ShredOS to securely wipe an HDD drive

Windows

Unfortunately, you will not 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 Manufacturer tools for Wiping HDD and SSD drives.

- Option C: See Using System Rescue to securely wipe an SSD drive
- Option D: Wipe your disk and re-install Windows before performing 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 these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, wear-leveling mechanisms might prevent this from working properly.

External SSD

First please see 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 Manufacturer tools for Wiping HDD and SSD drives.

If you are 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 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 these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, 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 Manufacturer tools for Wiping HDD and SSD drives
- Option C: See Using ShredOS to securely wipe an HDD drive

External/Secondary HDD and Thumb Drives

- Option A: Check Manufacturer tools for Wiping HDD and SSD drives
- Option B: Use external tools such as:
  - Eraser (open-source): https://eraser.heidi.ie/download/ [Archive.org]
  - KillDisk Free: http://killdisk.com/killdisk-freeware.htm [Archive.org]
- Option C: See 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/dskut14079/mac [Archive.org].

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 is 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 earlier 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 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 Considerations for using external SSD drives

If your USB controller and USB SSD disk support 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 are 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-dskut14079/mac [Archive.org]

- https://support.apple.com/guide/disk-utility/encrypt-protect-a-storage-device-password-dskutl35612/mac [Archive.org] 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 these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, 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 [Archive.org] 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/SSD and Thumb drives

The same principles from the earlier chapters apply to this one. The same issues arise too.

With an HDD drive, you can securely delete files by just deleting them and then apply one or 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 are 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 SSD (whether it is LUKS, Filevault 2, Veracrypt, or Bitlocker) could unlock your drive and then attempt a recovery using classic recovery utilities and could succeed if the data were not trimmed properly. But this is again highly unlikely.

Since the Trim operation is not continuous on most recent hard drives 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 SSD is to again overwrite all the free space after deletion of the files in question or to decrypt/re-encrypt the drive. But 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, SSD, 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 are using Plausible Deniability on an SSD drive against all recommendations.

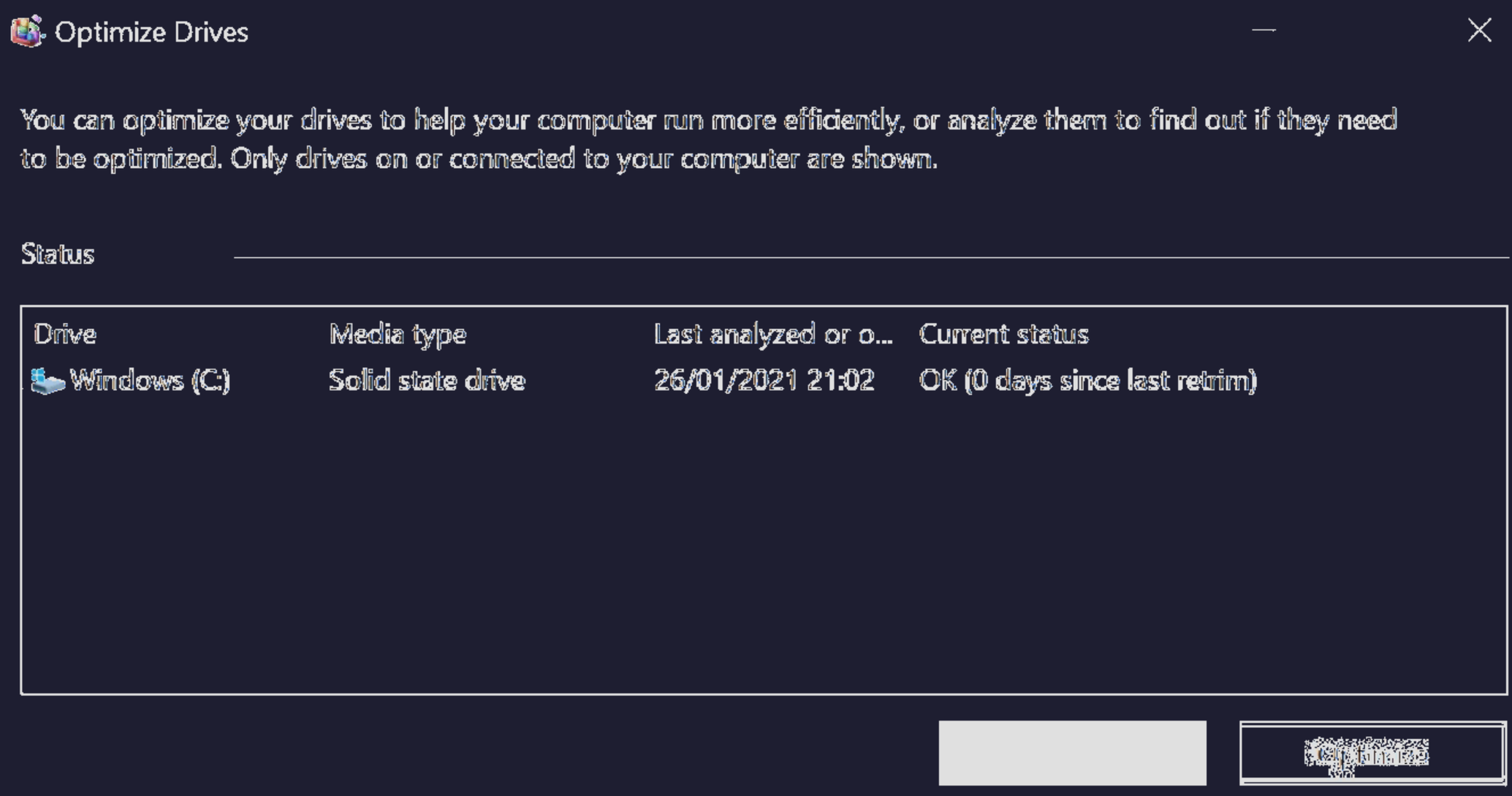
System/Internal SSD drive

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 do not want to wait for the periodic Trim (set to Weekly by default in Windows 10/11), you could also force a disk-wide Trim using the Windows native Optimize tool (see Windows Cleaning Tools).

If data were 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 are done. That is probably enough in my opinion.



If you want more security and do not trust the Trim operation, then you will 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 these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, wear-leveling mechanisms might prevent this from working properly.

Internal/External HDD or a USB Thumb Drive

Please refer to Windows Cleaning Tools and pick a utility before going ahead.

The process is quite 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 [Archive.org])

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 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 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 are done. That is probably enough in my opinion.

If Trim is not supported or you are 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 these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, 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 do not 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 do not trust the Trim operation, then you will 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 [Archive.org]) 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 a utility such as:
  - BleachBit https://www.bleachbit.org/download/linux [Archive.org]
  - 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 [Archive.org]) 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 these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, 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 [Archive.org]
- Or you can do this from the command line following this tutorial: https://linuxhint.com/completely_wipe_hard_drive_ubuntu/ [Archive.org] (For this purpose we recommend wipe and shred).

External SSD drive

First please see 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 the 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 [Archive.org] 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 [Archive.org]
  - 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 [Archive.org]) 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 these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, 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 [Archive.org]

As with other Linux Systems, if you want more security and do not trust the Trim operation then you will 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. We didn't find a reliable tutorial on how to do this safely on Qubes OS but it is possible this tutorial could work: https://wiki.archlinux.org/index.php/dm-crypt/Device_encryption#Re-encrypting_devices [Archive.org] (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 [Archive.org]) and then trim using "fstrim --all" and then fill up the entire free space of the disk using a utility such as:
  - BleachBit https://www.bleachbit.org/download/linux [Archive.org]
  - 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 [Archive.org]) 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 these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, 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 Qube 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 [Archive.org]
- Or you can do this from the command line following this tutorial: https://linuxhint.com/completely_wipe_hard_drive_ubuntu/ [Archive.org] (For this purpose we recommend wipe and shred).

External SSD drive

First please see 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 a "fstrim --all" from the terminal to trim the drive. Refer to this Documentation (https://github.com/Qubes-Community/Contents/blob/master/docs/configuration/disk-trim.md [Archive.org]) 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 Qube 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 [Archive.org] 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 [Archive.org]
  - 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 [Archive.org]) 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 are 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 are separate partitions on the same SSD drive.

Keep in mind all these options need to be applied on the entire physical drive and not on a specific partition/volume. If you do not, wear-leveling mechanisms might prevent this from working properly.

macOS

System/Internal SSD drive

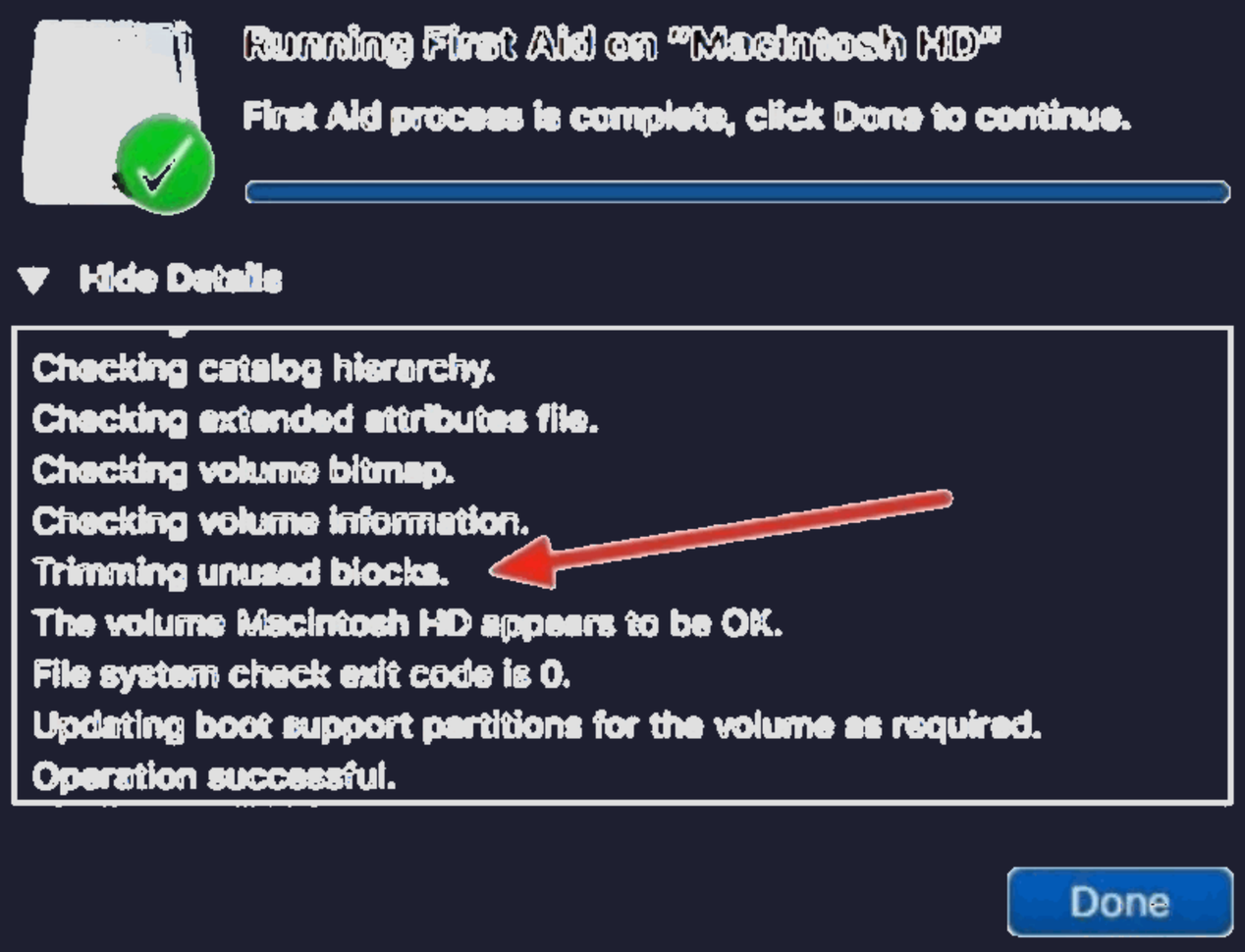
Just permanently delete the file (and empty recycle bin) and it should be unrecoverable due to trim operations and garbage collection.

- If your file system is APFS, you do not need to worry about Trim, it happens asynchronously as the OS writes data according to their 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 [Archive.org])



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. So, you are left with using other tools:

- Permanent Eraser http://www.edenwaith.com/products/permanent%20eraser/ [Archive.org]
- 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 [Archive.org].

In the case of USB thumb drives, consider wiping them completely using Disk Utility as instructed previously.

External SSD drive

First please see 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 earlier section will arise here. You can never really be 100% sure your SSD data is deleted when you ask it to do so unless you wipe the whole drive using specific methods above.

We are not aware of any 100% reliable method to delete single files selectively and securely 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 will have to trust the SSD Trim operation **which in my opinion is enough. It is 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 [Archive.org] as well as their general hardening tutorial for all platforms here https://www.whonix.org/wiki/System_Hardening_Checklist [Archive.org]

Metadata auditing

Format conversion does not reliably strip metadata. Converting a DOCX to PDF using Word embeds the Word document metadata into the PDF. Printing to PDF is somewhat cleaner but still retains producer information. The safe path is to use a purpose-built tool.

File type	Tool	What to check and remove
JPEG / TIFF / HEIC	`exiftool -all= file.jpg`	EXIF, XMP, IPTC - all containers
PNG	`exiftool -all= file.png`	tEXt/iTXt chunks, XMP
DOCX / XLSX / PPTX	MAT2 or Word's Document Inspector	Author, revision history, tracked changes, template path, comments
PDF	`exiftool -all= file.pdf` or MAT2	XMP packet, producer string, author, keywords
Any untrusted file	Dangerzone	Convert to safe PDF via isolated container, stripping all metadata
Video (MP4 / MOV)	`exiftool -all= file.mp4`	GPS, device info, creation time, encoder version

The safest workflow for sharing sensitive documents: open the original in Dangerzone, which renders it in an isolated container and exports a clean PDF with metadata stripped. For images, run `exiftool -all= -overwrite_original filename` and verify with `exiftool filename` that the output is clean before sharing.

Pictures and videos

EXIF is the metadata standard used by digital cameras and smartphones. A typical smartphone photo contains: GPS coordinates at the moment of capture (latitude, longitude, altitude, and sometimes bearing); camera make, model, and serial number; lens focal length; timestamp including timezone offset; and software version used to process the image.

The GPS data alone is frequently sufficient to identify a specific room in a specific building. Several high-profile source identification cases have turned on EXIF geolocation in images sent to journalists or posted online.

Why stripping EXIF is not always enough: EXIF is one metadata container. JPEG and TIFF files also support **XMP** (Extensible Metadata Platform), an Adobe-developed metadata format embedded as an XML packet inside the file. Tools that strip EXIF do not necessarily strip XMP. XMP can carry the same geolocation, authorship, and device information - sometimes more. ExifTool reads and writes both; many simpler "EXIF removers" do not touch XMP at all. Additionally, some camera manufacturers embed proprietary metadata in their own namespace inside XMP that persists even after standard EXIF removal.

A further subtlety: some platforms (notably older versions of Twitter and Facebook) strip EXIF server-side before serving images - but they store the original. Do not rely on platform stripping as a privacy control.

On Windows, macOS, and Linux we would recommend ExifTool (https://exiftool.org/ [Archive.org]) and/or ExifCleaner (https://exifcleaner.com/ [Archive.org]) that allows viewing and/or removing those properties.

ExifTool is natively available on Tails and Whonix Workstation.

PDF metadata

PDFs carry their own metadata layer. The **XMP packet** in a PDF can contain author, creator application (including version number), producer (the software that generated the PDF, e.g. "Microsoft Word 16.0.1" or a specific version of LibreOffice), creation and modification timestamps, and document title. The **producer string** is particularly useful to investigators because specific software versions are associated with specific time windows and installations.

Less obviously, **font embedding** in PDFs can fingerprint a document to a specific installation. Font subsets - the specific character outlines embedded in the PDF - vary slightly depending on which fonts are installed, which renderer is used, and which version of the software generated the file. Comparing embedded font data across multiple documents can link them to the same author even with no other identifying metadata present.

PDFPARANOIA (LINUX/WINDOWS/MACOS/QUBESOS)

Consider using https://github.com/kanzure/pdfparanoia [Archive.org] which will remove metadata and watermarks on any PDF.

EXIFCLEANER (LINUX/WINDOWS/MACOS/QUBESOS)

Just install it from https://exifcleaner.com/ [Archive.org], run and drag and drop the files into the GUI.

EXIFTOOL (LINUX/WINDOWS/MACOS/QUBESOS)

It is actually simple, just install exiftool and run:

- To display metadata: `exiftool filename.pdf`
- To remove all metadata: `exiftool -All= filename.pdf`

ExifCleaner

Just install it from https://exifcleaner.com/ [Archive.org], run and drag and drop the files into the GUI.

ExifTool

It is actually simple, just install exiftool and run:

- To display metadata: `exiftool filename.jpg`
- To remove all metadata: `exiftool -All= filename.jpg`

Remember that ExifTool is natively available on Tails and Whonix Workstation.

Windows Native tool

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 [Archive.org]

Cloaking/Obfuscating to prevent picture recognition

Consider the use of Fawkes https://sandlab.cs.uchicago.edu/fawkes/ [Archive.org] (https://github.com/Shawn-Shan/fawkes [Archive.org]) to cloak the images from picture recognition tech on various platforms.

Or if you want online versions, consider:

- https://lowkey.umiacs.umd.edu/ [Archive.org]
- https://adversarial.io/ [Archive.org]

DOCX and Office Documents

Microsoft Office documents are ZIP archives containing XML files, and those XML files contain extensive metadata. This includes: author name and initials (from the Office profile at creation time); last-modified-by name; creation and modification timestamps; revision count; total editing time; company name from the Office installation; the path of the document template used at creation (which can include a username or network path); and, critically, **revision history and tracked changes** - deletions and edits that the author thought were removed may be stored in the document and recoverable by anyone who opens it in a sufficiently capable viewer.

Several cases of documents leaked to journalists have resulted in source identification because the author's name or network username was embedded in the XML. John Doe metadata has identified real people. The fix is to use **File → Inspect Document** in Word before sharing (it will show hidden data and offer to remove it) or to use MAT2 to strip metadata entirely. Converting to PDF does not reliably remove this information - see below.

Alternatively, on Windows, macOS, Qubes OS, and Linux we would recommend ExifTool (https://exiftool.org/ [Archive.org]) and/or ExifCleaner (https://exifcleaner.com/ [Archive.org]) that allows viewing and/or removing those properties

EXIFCLEANER

Just install it from https://exifcleaner.com/ [Archive.org], run and drag and drop the files into the GUI.

EXIFTOOL

It is actually simple, just install exiftool and run:

- To display metadata: `exiftool filename.docx`
- To remove all metadata: `exiftool -All= filename.docx`

LIBREOFFICE DOCUMENTS

- select Files in the upper menu
    - Select Properties

    - Uncheck "Apply User Data"
    - Uncheck "Save Preview image with the Document"
    - Click "Reset Properties"
    - Make sure there is nothing on the Description and Custom Properties tabs
- Select Tools in the upper menu
    - Select Options
    - Select Security
    - Click "Security Options and Warning"
    - Check:
        - "When printing"
        - "When saving or sending"
        - "When creating PDF files"
        - "Remove personal information on saving"

In addition, on Windows, macOS, Qubes OS, and Linux we would recommend ExifTool (https://exiftool.org/ [Archive.org]) and/or ExifCleaner (https://exifcleaner.com/ [Archive.org]) that allows viewing and/or removing additional properties

EXIFCLEANER

Just install it from https://exifcleaner.com/ [Archive.org], run and drag and drop the files into the GUI.

EXIFTOOL

It is actually simple, jut install exiftool and run:

- To display metadata: `exiftool filename.odt`
- To remove all metadata: `exiftool -All= filename.odt`

All-in-one Tool

Another option good tool to remove metadata from various documents is the open-source mat2 recommended by privacyguides.org (https://0xacab.org/jvoisin/mat2 [Archive.org]) which you can use on Linux quite easily. I never managed to make it work properly within Windows due to various dependencies issues despite the provided instructions. It is however very straightforward to install and use on Linux.

So, we 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 Creating a mat2-web guest VM for removing metadata from files



Remove metadata

The file you see is just the tip of the iceberg. Remove the hidden metadata.

Browse... PRISM.pptx

UPLOAD

?

© jvoisin - source - ♥

Mat2 is also pre-installed on the Whonix Workstation VM and available on Tails by default.

Tails

Tails is great for this; you have nothing to worry about even if you use an SSD drive. Shut it down and it is 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. For more information, see their VM Live Mode [Archive.org] and Warning [Archive.org].

macOS

Guest OS

Revert to an earlier snapshot on Virtualbox (or any other VM software you are 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 [Archive.org]

QUARANTINE DATABASE (USED BY GATEKEEPER AND XPROTECT)

macOS (up to and including 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: `sqlite3 ~/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2 "select * from LSQuarantineEvent"`

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`

Lastly, you can also disable Gatekeeper altogether by issuing the following command in the terminal:

- `sudo spctl --master-disable`

Refer to this section of this guide for further information https://github.com/drduh/macOS-Security-and-Privacy-Guide#gatekeeper-and-xprotect [Archive.org]

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 will 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 will 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 we are 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, we tested various browsers (On macOS Catalina, Big Sur, and Monterey), and here are the results as of the date of this guide:

Browser	Quarantine DB Entry	Quarantine File Attribute	Origin File 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

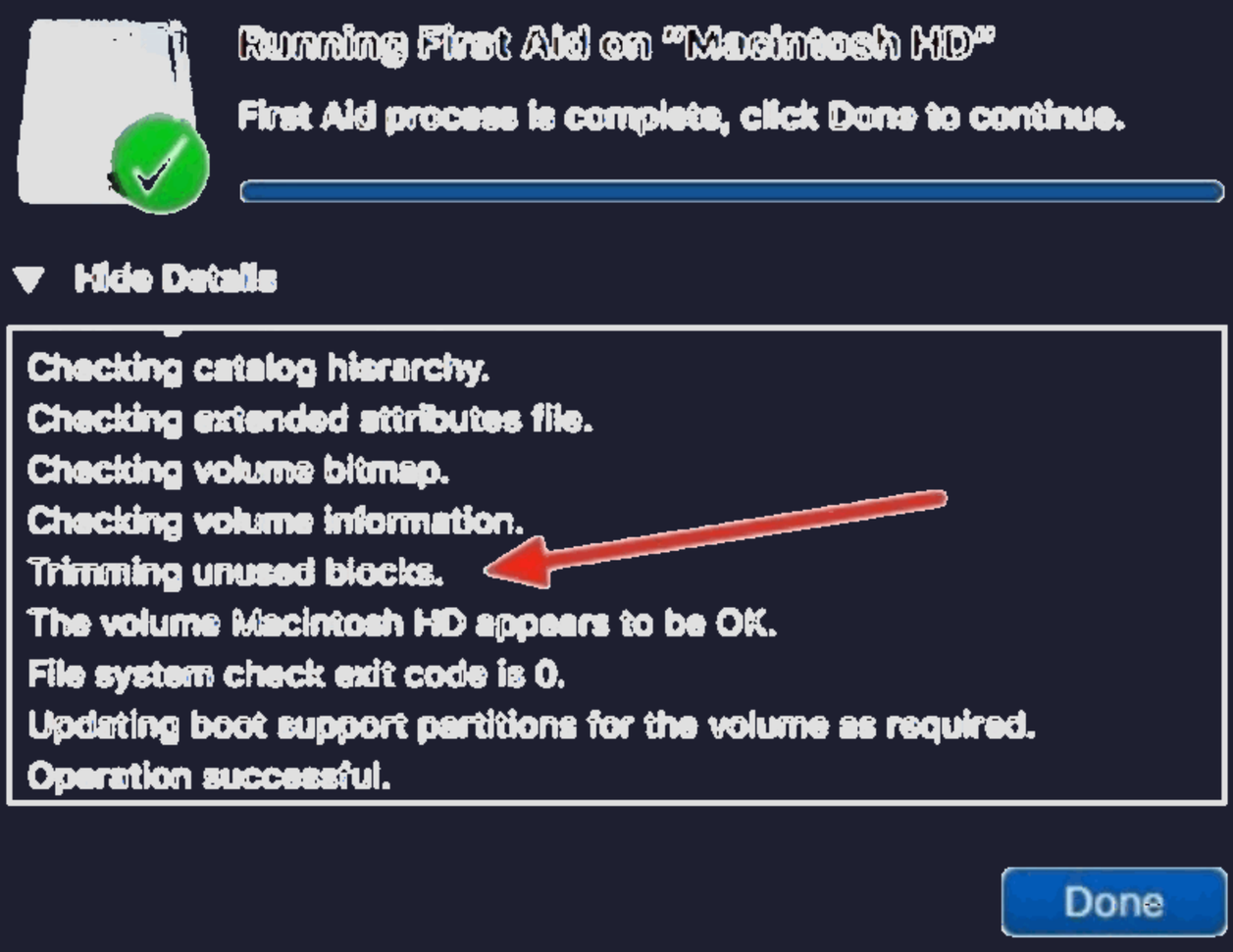
In addition, macOS keeps various logs of mounted devices, connected devices, known networks, analytics, documents revisions...

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 [Archive.org]

Many of those can be deleted using various commercial third-party tools but we would personally recommend using the free and well-known Onyx which you can find here: https://www.titanium-software.fr/en/onyx.html [Archive.org]. Unfortunately, it is closed-source, but it is notarized, signed, and has been trusted for many years.

FORCE A TRIM OPERATION AFTER CLEANING

- If your file system is APFS, you do not 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 [Archive.org]).



Linux (Qubes OS)

Please consider their guidelines https://github.com/Qubes-Community/Contents/blob/master/docs/security/security-guidelines.md [Archive.org]

If you are using Whonix on Qubes OS, please consider following some of their guides:

- Whonix System Hardening guide https://www.whonix.org/wiki/System_Hardening_Checklist [Archive.org]
- Enabling App Armor on Qubes https://www.whonix.org/wiki/Qubes/AppArmor [Archive.org]
- Also, consider the use of Linux Kernel Guard https://www.whonix.org/wiki/Linux_Kernel_Runtime_Guard_LKRG [Archive.org]

Linux (non-Qubes)

Guest OS

Revert to an earlier snapshot of the Guest VM on Virtualbox (or any other VM software you are 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 should not have traces to clean within the Host OS since you are doing everything from a VM if you follow this guide.

Nevertheless, you might want to clean some logs. Consider having a look this convenient (but unfortunately unmaintained) tool: https://github.com/sundowndev/covermyass [Archive.org]

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 [Archive.org]

Windows

Guest OS

Revert to an earlier snapshot on Virtualbox (or any other VM software you are using) and perform a trim command on your Windows using the Optimize as explained at 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 the 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 us 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 us 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 should not have Bluetooth at all), Go into the Device Manager, Select Bluetooth, Disable the Device and Re-Enable the 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), your network connections, app crash information, and various errors. It is 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 five logs using a 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 are 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.

BROWSER HISTORY

- Brave (in case you did not 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 is time 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 we know, no utilities clean those yet (BleachBit or PrivaZer for instance) so you will 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 [Archive.org]
- Within Regedit, enter this to the address bar: `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles`

- There you will 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 exceptionally useful sources of information for forensics and you need to clean those. Especially if you mounted any "hidden volume" anywhere. Again, you should not do this on the Decoy OS:

- Download Shellbag Analyzer & Cleaner from https://privazer.com/en/download-shellbag-analyzer-shellbag-cleaner.php [Archive.org]
    - 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 (do not backup)
        - Select SSD cleanup (if you have an SSD)
        - Select one pass (All zero)
        - Clean

EXTRA TOOLS CLEANING

After cleaning those earlier traces, you should also use third-party utilities that can be used to clean various traces. These include the traces of the files/folders you deleted.

Please refer to Windows Cleaning Tools before continuing.

PrivaZer

Here are the steps for PrivaZer:

- Download and install PrivaZer from https://privazer.com/en/download.php [Archive.org]
    - Run PrivaZer after install
    - Do not 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 to be 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. We 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 is 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: We would just pick Normal Deletion + Trim (Trim itself should be enough). Secure Deletion with Trim (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", we would select Normal Cleanup which will fill the entire free space with Data. We do not really trust Smart Cleanup as it does not actually fill all the free space of the SSD with Data. But again, this is probably not needed and overkill in most cases.
- If you have an HDD drive:
    - Secure Overwriting Tab: We would just pick Secure Deletion (1 pass).
    - Free Space: We 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/SSD) and will not clean free space. It is not secure on an HDD nor an SSD.
    - Quick Cleanup will do secure deletion (on HDD) and normal deletion + trim (on SSD) but will not clean free space. This is secure enough for SSD but not for HDD.
    - Normal Cleanup will do secure deletion (on HDD) and normal deletion + trim (on SSD) and will then clean the whole free space (Smart Cleanup on HDD and Full Cleanup on SSD) and should be secure. This option is the best for HDD but completely overkill for SSD.
- Click Clean and wait for cleaning to finish. Could take a while and will fill your whole free space with data.

BleachBit

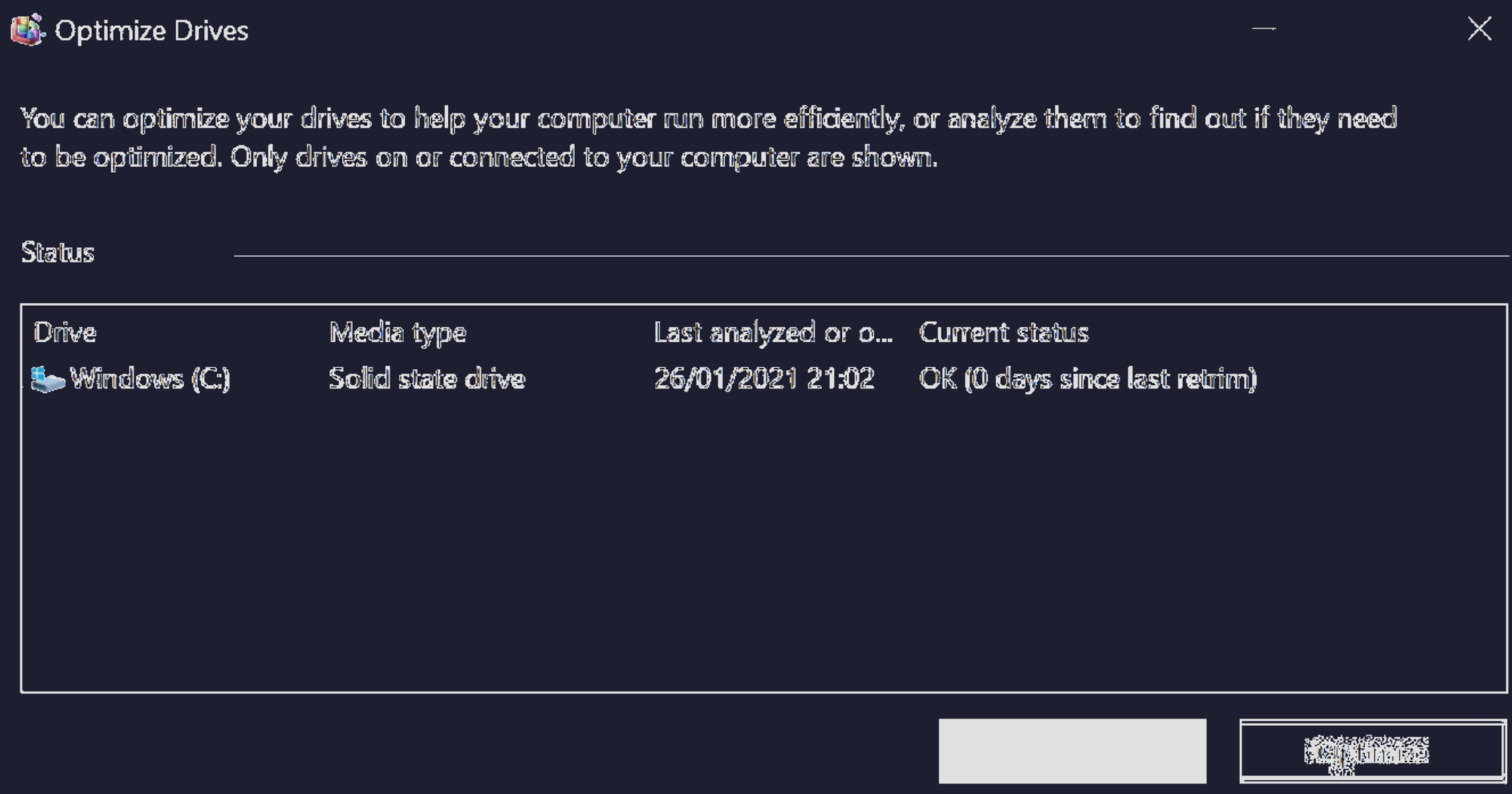
Here are the steps for BleachBit:

- Get and install the latest version from BleachBit here https://www.bleachbit.org/download [Archive.org]
- 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 earlier utility, we would not clean the free space on an SSD drive because we 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 SSD drives.

FORCE A TRIM WITH WINDOWS OPTIMIZE (FOR SSD DRIVES)

With this Native Windows 10/11 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 Defragment. You are done as this will not defragment but only optimize. Meaning it will initiate a Trim operation (https://en.wikipedia.org/wiki/Trim_(computing) [Wikiless] [Archive.org]).



Removing your identities from search engines and other platforms

Chances are your actions (such as posts on various platforms, your profiles) will be indexed (and cached) by many search engines.

Contrary to widespread belief, it is possible to have some but not all 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 will 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 the 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/ [Archive.org]
- https://justgetmydata.com/ [Archive.org]

When you are 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 [Archive.org] 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 [Archive.org] 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 a cached version of pages 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) 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 [Archive.org] and then select Tools and Delete URL https://webmaster.yandex.com/tools/del-url/ [Archive.org]

There you could input the URL that does 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 the 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 we know, there is no readily available tool to force this, and you will have to wait for the results to get updated if there is any. If you know a way, please report this to us through the GitHub issues.

Yahoo Search

Yes, Yahoo Search still exists but as per their help page https://help.yahoo.com/kb/SLN4530.html [Archive.org], there is no way to remove information or refresh information besides waiting. This could take 6 to 8 weeks.

Baidu

As far asWeknow, there is no readily available tool to force this unless you control the website (and do it through their webmaster tools). Therefore, you will 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 asWeknow, 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 [Wikiless] [Archive.org]

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.

Archive.today

Some information can sometimes be removed on demand (sensitive information for example) as you can see many examples here: https://blog.archive.today/archive

This is done through their "ask" page here: https://blog.archive.today/ask

Internet Archive

You can remove pages from internet archives but **only if you 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 we exclude or remove my site's pages from the Wayback Machine?

You can send an e-mail request for us to review to info@archive.org with the URL (web address) in the text of your message".

Others

Have a look at those websites:

- https://justdeleteme.xyz/
- https://inteltechniques.com/workbook.html [Archive.org]

Some low-tech old-school tricks

Hide Communications in Plain Sight

You must 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 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 that combines the idea of a Dead Drop and Secure Communication Obfuscation through Steganography and/or Kleptography and has many names such as Koalang or "Talking Around" or even "Social Steganography". This technique is very old and still widely used nowadays by teenagers to bypass parental control. It is 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 ...) know that you are in trouble, and they should look out for you?

All this without revealing the identity of the person you are 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 emojis 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 mean a lot to a knowledgeable reader who could then take appropriate agreed-upon actions. You could also hide the message using Steganography using for instance https://stegcloak.surge.sh/.

You do not 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 this 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 quite simple and just requires a wire/cable. Simply lay objects on your desk/night table or in your drawers following a straight line. You can use a simple USB cable as a tool to align them.

Make a line with your cable and place objects along the line. When you are 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 without taking pictures.

Fortunately, modern technology has made this even simpler. If you suspect someone might be looking through your stuff while you are away, you can just take a picture of the area with your phone before leaving. When you are 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 extremely 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 is 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 is a simple old-school trick that could tell you someone searched a document you had.

Some last OPSEC thoughts

Wait, what is OPSEC? Well, OPSEC means Operations Security. The basic definition is: "OPSEC is the process of protecting individual pieces of data that could be grouped together to give the bigger picture."

The important step here, and probably the easiest one, is a lesson you can take from the movie Fight Club: the first rule is that you **do not** talk about Fight Club. This applies to many aspects of your online operational security or OPSEC. Taking your time to go through this guide will reward you with the tools and knowledge to embrace a fuller, more secure experience on the internet. Rest assured that this guide will reveal things to you that will frustrate your enemy. You will learn how to protect your operating systems and lockdown your critical information and ensure mission success. But the one thing you must adhere to is this rule of thumb - do not talk about operation details. The biggest adversarial threat to you is OSINT (discussed below and throughout the document). The enemy will gather information on you based on what they observe about you and your activities online and in real life.

Adversaries take many forms. To some, they are actors of a foreign government, while to others they may be simply a rival company's employee looking to find disgruntled workers to target for further pressuring. To most, the general task of OPSEC is that this is your ship - you must not do anything or say anything to sink your own ship. Simply expressing your frustration with your boss or your work conditions or your equipment, might be enough to generate not only a behavior profile but also a vector of attack. A disgruntled employee, in this example, is what generally provides enough information to warrant pressuring of that employee for further information and possibly even extortion, blackmail, or worse. Failure to implement basic OPSEC can lead to failure at various points. It can lead to serious injury or even death if your threat model is a determined attacker, foreign actor, and so on.

You must live by the simple rule that "loose lips sink ships" - but also that they are usually your lips which will do the sinking. OPSEC is often just applying common sense and being cautious about your activities including in the physical world:

Digital and Online OPSEC

- **Remember to use passphrases or suits of words instead of short passwords and use a different one for each service. See Guidelines for passwords and passphrases.**
- Make sure you are not keeping a copy of this guide anywhere unsafe after. The sole presence of this guide will most likely defeat all your plausible deniability possibilities.
- OSINT "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 [Archive.org].
- Do not ever use biometrics alone to safeguard your secrets. Biometrics can be used without your consent.
- Do check the signatures and hashes of software and documents you download before installing/viewing them.
- Do not have the same behavior such as visiting the same links on the clearnet then visit the same with the your anoynous online identity. Watch this DEF CON 25 presentation if you didn't before: DEF CON 25 - Svea Eckert, Andreas Dewes - Dark Data [Invidious].
- Encrypt everything but do not take it for granted. Remember the $5 wrench.

Physical and IRL OPSEC

- Remember to "Physically Tamper protect your laptop".
- See Important notes about evil-maid and tampering.
- Check out How to spot if someone has been searching your stuff.
- Consider the use of Haven https://guardianproject.github.io/haven/ [Archive.org] on some old android phone to keep watch on your home/room while you are away.
- Remember Warning about smartphones and smart devices. Do not forget your smart devices can compromise your anonymity.
- Do not ever travel with those devices if you must 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 a USB data blocker for charging.
- Remember the first rule of fight club and do not talk to anyone about your sensitive activities using your real identity.
- Keep a normal life and do not be weird. If you spend all your online time using Tor to access the internet and have no social network accounts at all ... You are already suspicious and attracting unnecessary attention.
- Keep plausible deniability as an option but remember it will not help against the 5$ wrench either.
- 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 [Wikiless] [Archive.org].
- Check for tampering regularly (not only your devices but also your home/room).
- If you can, do not talk to the police/authorities (at least if you are in the US) https://www.youtube.com/watch?v=d-7o9xYp7eE [Invidious] without a lawyer. Remain silent.
- 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.
- Keep your situation awareness high but not too high as to appear suspicious.
- Consider using a physical security key (e.g., YubiCo YubiKey) for various protections against account compromise. **(Not covered in this version of the guide but is a work in progress for later versions.)**
- Read the tips here https://www.whonix.org/wiki/DoNot [Archive.org]
- **Have common sense, do not be dumb, look and learn from others' mistakes, watch/read these:**
  - Medium.com, Darkweb Vendors and the Basic Opsec Mistakes They Keep Making https://medium.com/@c5/darkweb-vendors-and-the-basic-opsec-mistakes-they-keep-making-e54c285a488c [Scribe.rip] [Archive.org]
  - 2020, Sinwindie, OSINT, and Dark Web Markets, Why OPSEC Still Matters https://www.youtube.com/watch?v=IqZZU9lFlF4 [Invidious]
  - 2020, RSA Conference 2020, When Cybercriminals with Good OpSec Attack https://www.youtube.com/watch?v=zXmZnU2GdVk [Invidious]
  - 2015, DEF CON 22, Adrian Crenshaw, Dropping Docs on Darknets: How People Got Caught https://www.youtube.com/watch?v=cQ2OZKitRwc [Invidious] (Slides [Archive.org])
  - 2017, Ochko123 - How the Feds Caught Russian Mega-Carder Roman Seleznev https://www.youtube.com/watch?v=6Chp12sEnWk [Invidious]
  - 2017, DEF CON 25 - Svea Eckert, Andreas Dewes - Dark Data [Invidious]
  - 2015, DEF CON 22, Zoz, Don't Fuck It Up! https://www.youtube.com/watch?v=J1q4Ir2J8P8 [Invidious]
  - 2020, Bad Opsec, How Tor Users Got Caught, https://www.youtube.com/watch?v=GR_U0G-QGA0 [Invidious]
  - 2022, Master of OpSec Masters: A View Through the Prism of Time, https://officercia.mirror.xyz/4x2-M4R2cSnID1wpsTO4CQNrMQ5JUFouR-rZ_N4xO-Q [Archive.org]
  - 2022, How can you become a one-man-army OSINT specialist? https://officercia.mirror.xyz/5KSkJOTgMtvgC36v1GqZ987N-_Oj_zwvGatOk0A47Ws [Archive.org]

It is recommended that you learn about the common ways people mess up OPSEC https://dan-kir.github.io/2022/05/26/OPSEC-notes.html [Archive.org]. Whatever you do, take OPSEC seriously, and Don't Fuck It Up!

FINAL OPSEC DISCLAIMER: KEEP YOUR ANONYMOUS IDENTITIES COMPLETELY SANDBOXED FROM YOUR NORMAL ENVIRONMENT AND REAL IDENTITY. DO NOT SHARE ANYTHING BETWEEN THE ANONYMOUS ENVIRONMENTS AND THE REAL IDENTITY ENVIRONMENT. KEEP THEM COMPLETELY COMPARTMENTALIZED ON EVERY LEVEL. MOST OPSEC FAILURES ARE DUE TO SOMEONE ACCIDENTALLY LEAKING INFORMATION RATHER THAN TECHNICAL FAILURES.

What to do if you detected tampering or searching

- In the case of a laptop, they likely placed a key-logger, and possible network and gps capabilities. We recommend to open your laptop take the drive (which should be fully encrypted) and leave for a safe place and abandonning the laptop. Do not try to remove the "bug" as this could put you in physical danger.

- If you detected searching of your room, home... Again we recommend leaving for a safe place while abandoning everything in the room that could also be "bugged".
- Do your best to not let your adversary suspect or know you detected the search and/or the tampering. Be creative. Call a friend for example just to tell you're gonna go to the supermarket to buy food.

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/SSD/Laptop and trash it somewhere.
- Do the same with your backups.
- Keep the details of your lawyer nearby or if needed, call them 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 are 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 cannot 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 [Wikiless] [Archive.org] and this other visual resource with law references https://www.gp-digital.org/world-map-of-encryption/ [Archive.org]. Australia in particular has broad privacy laws and passed the Surveillance Legislation Amendment (Identify and Disrupt) Act 2021, which grants authorities powers to modify, add, copy, and delete data on a suspect's devices and accounts.

A small final editorial note

After reading this whole guide, we 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 a 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, we 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 length in this guide can be forged or tampered with by a motivated adversary for any purpose. Even if you do manage to keep secrets from prying eyes, anyone can 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.

"A lie can travel halfway around the world while the truth is putting on its shoes"

Please keep thinking for yourself, use critical thinking, and keep an open mind. "Sapere Aude" (Dare to know!).

"In the end the Party would announce that two and two made five, and you would have to believe it" -- George Orwell, 1984, Book One, Chapter Seven.

Consider helping others (see Helping others staying anonymous)

Donations

This project has no funding or sponsoring, and donations are more than welcome.

See: Contribute at the top of the website.

(Please do verify the checksum and GPG signature of this file for authenticity, this is explained in the README of the repository if you do not know how to do that).

Helping others staying anonymous

If you want to give a hand to users facing censorship and oppression, please consider helping them by helping the Tor Network. You can do so in several ways:

- The Easiest:
  - Using the Snowflake addon on your browser (https://snowflake.torproject.org/ [Archive.org])
- Slightly more work:
  - Running a Tor relay node (https://community.torproject.org/relay/ [Archive.org])
    - See [Recommended VPS hosting providers]
    - Additional Tutorial: https://torrelay.ca/ [Archive.org]

If you want a bit more challenge, you can also run a Tor Exit node anonymously using the recommended VPS providers above.

For this, see https://blog.torproject.org/tips-running-exit-node [Archive.org]

This project for instance is running several Tor Exit nodes using donations to fund. You can see them here: https://metrics.torproject.org/rs.html#search/family:970814F267BF3DE9DFF2A0F8D4019F80C68AEE26

Windows Installation

This is the Windows 10/11 installation process that should be valid for any Windows 10/11 install within this guide.

Windows 10 (See below for Windows 11)

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).

- (Only for VirtualBox VM Install) Go into the VirtualBox Machine Settings menu. Select network. Unplug the cable.
- Click "Install Now"
- Select "I don't have a product key"
- Select the flavor you want:
  - You intend to use Plausible Deniability: Windows Home
  - You do not intend to use Plausible Deniability: Windows Pro
  - 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 the installation (skip the next step).
- If this is part of a plausible deniability encryption set up on the Host OS:
    - If you are installing Windows for the first time (Hidden OS):
        - Delete the current partitions
        - Create the 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 States"
- 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 three security questions and answer whatever you want (not real data).
- Do not use Online Speech Recognition
- Do not let the app use your location
- Do not 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 > Go on the top left menu icon and sekect Privacy and Security
    - 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)
- Automatic file download disabled
- 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 11 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.

Windows 11

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).

- (Only for VirtualBox VM Install) Go into the VirtualBox Machine Settings menu. Select network. Unplug the cable. For this task, you can also follow this excellent tutorial by Oracle https://blogs.oracle.com/virtualization/post/install-microsoft-windows-11-on-virtualbox [Archive.org]
- Select your language, currency and keyboard layout
- Click "Install Now"
- (Only for VirtualBox VM Install) Push Shift and F10 at the same time
- (Only for VirtualBox VM Install) Launch "regedit" in the command prompt
- (Only for VirtualBox VM Install) When the Registry Editor opens, navigate to `HKEY_LOCAL_MACHINE\SYSTEM\Setup`, right-click on the "Setup" key and select "New => Key". When prompted to name the key, enter "LabConfig" and press enter.

- (Only for VirtualBox VM Install) Now right-click on the "LabConfig" key and select "New => DWORD (32-bit)" value and create a value named "BypassTPMCheck", and set its data to "1". With the same steps create the "BypassRAMCheck" and "BypassSecureBootCheck"
- Select "I don't have a product key"
- Accept the agreement
- Select the flavor you want:
    - You intend to use Plausible Deniability: Windows Home
    - You do not intend to use Plausible Deniability: Windows Pro
    - VM OS: Use Windows Pro or Windows Pro N
- Select Custom Install
- Storage:
    - If this is a simple OS installation (Host OS with Simple Encryption) or VM without encryption, **select the whole disk** and proceed with the installation (skip the next step).
    - If this is part of a plausible deniability encryption set up on the Host OS:
        - If you are installing Windows for the first time (Hidden OS):
            - Delete the current partitions
            - Create the 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 States"
- Select the Keyboard Layout and skip a second 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 three security questions and answer whatever you want (not real data).
- Ddisable Location
- Disable find my device
- Disable optional diagnostic data
- Only send "required diagnostic data"
- Do not improve Inking and Typing
- Disable the tailored experience.
- Disable the Advertising ID
- Click Accept

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)
- Automatic file download disabled
- Documents: Disable it (change button)
- Music Library: 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 11 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.

Windows Additional Privacy Settings

As written earlier in this guide and as noted by PrivacyGuides.org, Windows 10/11 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 of utilities to (hopefully) force Windows 10/11 into not sending data back to Microsoft.

Here are the steps in detail:

- **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. Do not connect Windows 10/11 to the internet before those settings are applied. You can download and copy those to the USB key (for transfer onto a Windows 10/11 fresh installation) or if it is a VM, you can transfer them to the VM within Virtualbox (VM Settings > General > Advanced > Drag n Drop > Enable Host to Guest).
- (For more advanced users) Download and install W10Privacy from https://www.w10privacy.de/english-home/ [Archive.org]
  - 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/ [Archive.org]
  - Type 1 and go into Telemetry
  - Type 1 and go into Firewall
  - Type 2 and add Spy Rules
  - Reboot
- Also, consider using ShutUp10++ from https://www.oo-software.com/en/shutup10 [Archive.org]
  - Enable at least all the recommended settings
- Finally, again for users with moderate skills, consider installing Safing Portmaster from https://safing.io/portmaster/ [Archive.org] **(Warning: there might be issues with some VPN clients. See:** https://docs.safing.io/portmaster/install/status/vpn-compatibility [Archive.org]
- Go back one last time to the settings to delete 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 those utilities frequently and after any Windows major update as they tend to silently re-enable telemetry using those updates.

As a bonus, it could be interesting to also consider Hardening your Windows Host OS somewhat. See https://github.com/beerisgood/windows10_hardening [Archive.org] (This is a security guide, not a privacy guide. If you use this guide, do not enable Hyper-V as it does not play well with Virtualbox, and do not enable features that were specifically disabled for privacy reasons earlier. Such as SmartScreen, cloud protection...)

Windows Installation Media Creation

Windows 10

These are the steps to create a Windows 10 (21H1) Installation Media using this tool and instructions:

https://www.microsoft.com/en-us/software-download/windows10 [Archive.org]

- Download the tool and execute it from your Download folder.
- Agree to the terms
- Select the process to Create an 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

Windows 11

- Go to https://www.microsoft.com/software-download/windows11 and download the ISO.

Secure Wiping an SSD with System Rescue

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/ [Archive.org] (download the ISO and write to a 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 will not 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 [Archive.org]
            - https://ata.wiki.kernel.org/index.php/ATA_Secure_Erase [Archive.org]
            - https://tinyapps.org/docs/wipe_drives_hdparm.html [Archive.org]
        - ATA Sanitize:
            - Follow this tutorial https://tinyapps.org/docs/ata_sanitize_hdparm.html [Archive.org]
- 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 [Archive.org]
        - https://tinyapps.org/docs/nvme-secure-erase.html [Archive.org]
        - https://tinyapps.org/docs/nvme-sanitize.html [Archive.org]

Clonezilla

- Get Clonezilla by just following these instructions: https://clonezilla.org/liveusb.php [Archive.org] (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 [Archive.org]
    - **If you are backing up a disk with simple Encryption, encryption of the backup is not required since you are 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, we strongly recommend 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 are 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 [Archive.org]

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 SSDs on a USB 3.0 7200rpm HDD).

Diskpart

Diskpart is a Windows utility that can be used to perform various operations on your hard drive. In this case, You will use Diskpart to show the Disk ID but also change it if necessary.

This could be needed if you restore a backup on a new HDD/SSD 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)

Safe Browser on the Host OS

If you can use Tor

This guide will **only recommend** using Tor Browser within the host OS because it has the best protection 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** [Archive.org]**: "If your personal safety depends on remaining anonymous, we highly recommend using Tor Browser instead of Brave Tor windows. ".**

This Browser on the host OS will only be used to download various utilities and will never be used for actual sensitive activities.

Refer to Installing and using desktop Tor Browser.

If you are experiencing issues connecting to Tor due to Censorship or Blocking, you might consider using Tor bridges as explained here: https://bridges.torproject.org/ [Archive.org]

Use this browser for all the next steps within the host OS unless instructed otherwise.

If you cannot use Tor

Because it is too dangerous/risky/suspicious. We would recommend as a last resort using Firefox, or Brave only using Private Windows for now.

See What about when Tor and VPNs aren't possible? before continuing.

Only do this from a different safe public Wi-Fi every time (See Find some safe places with decent public Wi-Fi) and using a long-range connection (See 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.

Windows Cleaning Tools

In this guide we will recommend two-third native tools and two third-party tools:

- Native Tools:
  - Windows 10/11 Disk Cleanup Utility: https://support.microsoft.com/en-us/windows/disk-cleanup-in-windows-10-8a96ff42-5751-39ad-23d6-434b4d5b9a68 [Archive.org]

> This tool will clean up a bunch of things natively. It is not enough, and we instead recommend using the 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 10/11 Optimize Utility (Defrag on HDD Drives): https://support.microsoft.com/en-us/windows/defragment-your-windows-10-pc-048aefac-7f1f-4632-d48a-9700c4ec702a [Archive.org] (yes the tutorial is for Windows 10 but should work on 11 too)

> For security, this tool is particularly 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/ [Archive.org]
  - The closed-source utility PrivaZer https://privazer.com/ [Archive.org]

I prefer PrivaZer because it has more customization and smarter features, but we would understand if you do not trust them and prefer open-source software in which case we 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.
- The Windows System Resource Usage Monitor (SRUM).
- Various histories of various programs (such as the recent lists).
- Various logs
- The free (unallocated) space of your hard drive.
- 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.

Secure Wiping an HDD with ShredOS

Several utilities are recommended (like the old unmaintained DBAN or System Rescue CD (https://www.system-rescue.org/ [Archive.org])) for this but we will recommend the use of ShredOS.

Feel free to go with DBAN instead if you want (using this tutorial: https://www.lifewire.com/how-to-erase-a-hard-drive-using-dban-2619148 [Archive.org]), 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.x86_64 [Archive.org]

- Unzip the ISO file
- Download Rufus from https://rufus.ie/ [Archive.org]
- Launch Rufus
- Select the ShredOS IMG file
- Write it to a 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 the screen

Linux

- Follow instructions on https://github.com/PartialVolume/shredos.2020.02 [Archive.org]
- Reboot and boot the USB key
- Follow the instructions on the screen

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 right disk. Wipe and Passes are for HDD drives. There are specific options for SSD 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 [Archive.org]
- Seagate SeaTools: https://www.seagate.com/support/downloads/seatools/ [Archive.org]
- Samsung Magican: https://www.samsung.com/semiconductor/minisite/ssd/download/tools/ [Archive.org]
- Kingston SSD Manager: https://www.kingston.com/unitedstates/en/support/technical/ssdmanager [Archive.org]
- 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 [Archive.org]
- Crucial Storage Executive: https://www.crucial.com/support/storage-executive [Archive.org]
- Western Digital Dashboard: https://support.wdc.com/downloads.aspx?p=279 [Archive.org]
- HP: Follow instructions on https://store.hp.com/us/en/tech-takes/how-to-secure-erase-ssd [Archive.org]
- Transcend SSD Scope: https://www.transcend-info.com/Support/Software-10/ [Archive.org]
- 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 [Archive.org]

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

- Toshiba Storage Tools: https://www.toshiba-storage.com/downloads/ [Archive.org]

External SSD Drive Considerations

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, we 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/ [Archive.org]

Some might be very efficient but many are gimmicky gadgets.

If you 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 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 is 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 is pretty much the only way to be sure it is not only supported but actually works. Some utilities can tell you whether it is supported or not like CrystalDiskInfo [Archive.org] but will not actually check if it is working. See Manufacturer tools for Wiping HDD and SSD drives.

If it does not 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 [Archive.org]

ATA/NVMe Operations (Secure Erase/Sanitize)

It is not "recommended". Please read the disclaimers here https://ata.wiki.kernel.org/index.php/ATA_Secure_Erase [Archive.org] and here https://wiki.archlinux.org/index.php/Solid_state_drive/Memory_cell_clearing [Archive.org]

But this seems to be based on anecdotal experiences. So, if you are 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/ [Archive.org]

Your mileage may vary. Use at your own risk.

macOS

Trim Support

According to Apple Documentation, Trim is supported on APFS (asynchronously) and HFS+ (through period trim or first-aid).

So, if it is 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 is 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 is not APFS (see this Tutorial for info https://www.lifewire.com/enable-trim-for-ssd-in-os-x-yosemite-2260789 [Archive.org])

If it does not work, we are not aware of any reliable method to enable TRIM besides the commercial utility Trim Enabler here https://cindori.org/trimenabler/ [Archive.org] which claims support for external drives.

ATA/NVMe Operations (Secure Erase/Sanitize)

We are not aware of any method of doing so reliably and safely on macOS. So, you will 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)

Removing Metadata with mat2-web

Download the latest Debian testing amd64 netinst ISO from https://www.debian.org/CD/netinst/ [Archive.org]

(Get testing to get the latest mat2 release, stable is a few versions back)

This is very lightweight, and we recommend doing it from a VM (VM inside a VM) to benefit from Whonix Tor Gateway. While it is possible to put this VM directly behind a Whonix Gateway, Whonix will not easily allow communications between VMs on its network by default. You could also just leave it on Clearnet during the install process and then leave it on the Host-Only network later, or install it from a VM within a VM then move it to host OS for Host-Only usage like we show below:

1. Create a new machine with any name like **Mat2**.
2. Select **Linux** for the Type.
3. Select **Debian (64-bit)** as the Version.
4. Leave the default options and click **Create**.
5. Select the VM and click **Settings**.
6. Select **System** and disable the **Floppy disk** on the Motherboard tab.
7. Select the Processor tab and **enable PAE/NX.**
8. Select **Audio** and **disable Audio**.
9. Select **USB** and **disable the USB controller**.
10. Select **Storage** and select the CD drive to mount the Debian Netinst ISO.
11. Select **Network** and **Attach to NAT**.
12. Launch the VM.
13. Select **Install** (not Graphical install).
14. Select **Language**, **Location**, and **Keyboard layout** as you wish.
15. Wait for the network to configure (automatic DHCP). This takes a few seconds.
16. Pick a name like **Mat2.**
17. Leave the **domain** empty.
18. Set a **root** password as you wish (preferably a good one).
19. Create a new **user** and **password** as you wish (preferably a good one).
20. Select the **Time Zone** of your choice.
21. Select **Guided - Use the entire disk**.
22. Select the only disk available (**/dev/sda** in our case).
23. Select **All files in one partition**.
24. Confirm and write changes to the disk.
25. Select **No** to scan any other CD or DVD.
26. Select any region and any mirror of your choice and leave **proxy** blank.
27. Select **No** to take part in any survey.
28. Select **only System Standard Utilities.** Uncheck everything else using **space.**
29. Select **Yes** to install GRUB bootloader.
30. Select **/dev/sda** and continue.
31. Complete the install and reboot.

32. Log in with your **user** or **root**. You should never use root directly as a best security practice but in this case, it is okay.
33. Update your install by running `apt upgrade`. It should be upgraded since it is a net install, but we're double checking.
34. Install the necessary packages for mat2 by running `apt install ffmpeg uwsgi python3-pip uwsgi-plugin-python3 lib35rsvg2-dev git mat2 apache2 libapache2-mod-proxy-uwsgi`.
35. Go to the **/var/www** directory by running `cd /var/www/`.
36. **Clone mat2-web** from the mat2-web repository by issuing `git clone https://0x-acab.org/jvoisin/mat2-web.git`.
37. **Create a directory for uploads** by running `mkdir ./mat2-web/uploads/`.
38. **Give permissions to Apache2** to read the files by running `chown -R www-data:www-data ./mat2-web`.
39. **Enable apache2 uwsgi proxy** by running `/usr/sbin/a2enmod proxy_uwsgi`.
40. **Upgrade pip** by running `python3 -m pip install pip --upgrade`.
41. **Install these Python modules** by running `python3 -m pip install flasgger pyyaml flask-restful flask cerberus flask-cors jinja2`.
42. **Move to the config directory** of mat2 by running `cd /var/www/mat2-web/config/`.
43. **Copy the apache2 config file** to **/etc** by running `cp apache2.config /etc/apache2/sites-enabled/apache2.conf`.
44. **Remove the default config file** by running `rm /etc/apache2/sites-enabled/000-default.conf`.
45. **Edit the apache2 config file** provided by mat2-web by running `nano /etc/apache2/sites-enabled/apache2.conf`.
46. **Remove the first line** `Listen 80` by typing **Ctrl+K** to cut the line.
47. **Change the uwsgi path** from `/var/www/mat2-web/mat2-web.sock` to `/run/uwsgi/uwsgi.sock` and type **Ctrl+X** to exit, followed by **Y** then **Enter**.
48. **Copy the uwsgi config file** to **/etc** by running `cp uwsgi.config /etc/uwsgi/apps-enabled/uwsgi.ini`.
49. **Edit the uwsgi config file** by typing `nano /etc/uwsgi/apps-enabled/uwsgi.ini` and change **uid** and **guid** to `nobody` and `nogroup` respectively. Save and exit with **Ctrl+X**, followed by **Y**, then **Enter**.
50. Run `chown -R 777 /var/www/mat2-web` to change ownership to **mat2-web.**
51. **Restart uwsgi** by running `systemctl restart uwsgi`. There should be no errors.
52. **Restart apache2** by running `systemctl restart apache2`. There should be no errors.
53. Now navigate to **Settings** > **Network** > **Attached to** and **select Host-only Adapter**. Click **OK** to save.
54. Reboot the VM via **Machine** > **Reset**. Confirm the reset.
55. Log into the VM as the **user** from **Step 19** and type `ip a`. Note the IP address it was assigned under link/ether, the one that has **192.168.*.***.
56. From the VM Host OS, **open a Browser** and navigate to the IP of your Debian VM. It will be something like: **http://192.168.1.55**.
57. You should now see a Mat2-Web website running smoothly.
58. **Shutdown the Mat2 guest VM** by running `shutdown -h now` to halt the machine.
59. **Take a snapshot of the VM** within Virtualbox while the guest VM is shutdown.

__Restart the Mat2 VM_ and you are ready to use Mat2-web to remove metadata from most files!_*

After use, shut down the VM and revert to the snapshot to remove traces of the uploaded files. This VM does not require any internet access unless you want to update it, in which case, you need to place it back on the **NAT network** and do the next steps.

For updates of Debian, **start the VM** and run `apt update` followed by `apt upgrade`.

For updates of mat2-web, type `cd /var/www/mat2-web` and run `git pull`.

After updates, shutdown, change to the **Host-only Adapter**, take a new snapshot, remove the earlier one.

You are 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.

Do not ever expose this VM to any network unless temporarily for updates. This web interface is not suitable for any direct external access.

BIOS/UEFI Disk Wipe Options by Brand

Here are some links on how to securely wipe your drive (HDD/SSD) from the BIOS for various brands:

- Lenovo ThinkPads: https://support.lenovo.com/be/en/solutions/migr-68369 [Archive.org]
- HP (all): https://support.hp.com/gb-en/document/c06204100 [Archive.org]

- Dell (all): https://www.dell.com/support/kbdoc/en-us/000146892/dell-data-wipe [Archive.org]
- Acer (Travelmate only): https://us.answers.acer.com/app/answers/detail/a_id/41567/~/how-to-use-disk-sanitizer-on-acer-travelmate-notebooks [Archive.org]
- Asus: no option AFAIK except maybe for some ROG models.
- Gigabyte: no option AFAIK
- Honor: no option AFAIK
- Huawei: no option AFAIK

Smartphones and Smart Devices Warning

When conducting sensitive activities, remember that:

- **You should not bring your real smartphone or smart devices with you (even turned off).** Correlation attacks are possible on the Cell Networks to find which phone "turned off" before your burner phone "turned on". While this might not work the first time, after a few times, the net will tighten, and you will get compromised. It is better to leave your main smartphone at home online (see this article (Russian, use Google Translate link): https://biboroda.livejournal.com/4894724.html [Google Translate] [Archive.org])
- **Again, do not take them with you unless it is absolutely necessary. If you really must,** you could consider powering it off and removing the battery or, if not possible, the use of a faraday cage bag to store your devices. There are many such faraday "signal blocking" bags available for sale and some of these have been studied for their effectiveness. If you cannot 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).
    - Warning: consider that sensor data itself can also be reliably used to track you'.
    - 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.
    - **This could also include your car which could for example have a cell network device (including at least an IMEI) and a functionality to call emergency services**

Additionally, if using a smartphone as a burner, know that they send a lot of diagnostics by default. Enough to potentially identify you based on your device usage patterns (a technique known as biometric profiling). You should avoid using your burner unless absolutely necessary, to minimize the information that can be collected and used to identify you.

Lastly, you should also consider this useful sheet from the NSA about Smartphone security: https://web.archive.org/web/20210728204533/https://s3.documentcloud.org/documents/21018353/nsa-mobile-device-best-practices.pdf.

Note: Please do not consider commercial gimmicky all-in devices for anonymity. The only way to achieve proper OPSEC is by doing it yourself. See those examples to see why it is not a clever idea:

- **AN0M:** https://www.theguardian.com/australia-news/2021/sep/11/inside-story-most-daring-surveillance-sting-in-history [Archive.org]
- **Encrochat:** https://en.wikipedia.org/wiki/EncroChat [Wikiless] [Archive.org]
- **Sky ECC:** https://en.wikipedia.org/wiki/Sky_ECC [Wikiless] [Archive.org]

You should never rely on an external commercial service to ensure your first line of anonymity. But you will see that paid services can still be used later from an already anonymous identity if bought anonymously while observing good operational security.

Getting an Anonymous VPN or Proxy

If you follow our advice, you will also need a VPN subscription but this time you will need an anonymous one that cannot be tied to you by the financial system. Meaning you will need to buy a VPN subscription with cash or a reasonably private cryptocurrency (for example Monero). You will later be able to use this VPN to connect to various services anonymously but **never directly from your IP**. This VPN can never be used in any other non-anonymous context without jeopardzing your anonymity.

There are, two viable options:

Cash/Monero-Paid VPN

There are three VPN companies recommended by PrivacyGuides.org (https://www.privacyguides.org/vpn/ [Archive.org]) that accept cash payments: Mullvad, iVPN, and Proton VPN.

Here are their logging policies:

- Mullvad: https://mullvad.net/en/help/no-logging-data-policy/ [Archive.org]
- iVPN: https://www.ivpn.net/privacy/ [Archive.org]
- ProtonVPN: https://protonvpn.com/support/no-logs-vpn/ [Archive.org]

In addition, we will also mention a newcomer to watch: Safing SPN https://safing.io/ [Archive.org]) which (while still in the alpha stage at the time of this writing) which also accepts cash and has a very distinct new concept for a VPN which provides benefits similar to Tor Stream isolation with their "SPN"). Note that Safing SPN is not available on macOS at the moment. This possibility is "provisional" and at your own risk, but we think was worth mentioning.

Personally, for now, we would recommend Mullvad due to personal experience.

We would not recommend Proton VPN as much because they do require an e-mail for registration unlike Mullvad, iVPN, and Safing. Proton also has a tendency to require phone number verification for users who register over Tor.

How does this work?

- Access the VPN website with a Safe Browser (see Safe Browser.
- Go to iVPN, Mullvad, or Safing 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 of 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, or pay with Monero if available. If they do not accept Monero but do accept BTC, consider Paying anonymously online with BTC.
- 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 and touch DNA.
- A less-obvious alternative could be to put super glue on your fingertips, to avoid making it obvious you're wearing gloves. However, this can prevent effective use of touchscreens, as well as failing to as effectively prevent you from touch DNA. Also, if spotted, it can be quite suspicious to be caught with super glue on your fingers.
- Do not use any material/currency that was manipulated by someone that can be related to you in any way.
- Do not use the currency you just got from an ATM that could record dispensed bills serial numbers.
- Be careful if you print anything that it is not watermarked by your printer (See Printer Watermarking).
- Do not lick the envelope or the stamps 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 of residual DNA traces from your environment or you contaminating the materials.
- The more people frequent a space, the lower the risk, as your DNA will be obscured by the DNA of other people as they pass through
- Security cameras can be a risk. Try to cover your face. Also, gait recognition may be a concern. See [Gait Recognition and Other Long-Range Biometrics]

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 do not 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/Proxy on a Monero/Cash-paid VPS (for users more familiar with Linux)

The other alternative is setting up your own VPN/Proxy 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 block-listed somewhere are lower than known VPN providers.

This does also offer some disadvantages 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 [Archive.org]) or from a local reseller using cash from https://localmonero.co.

Do not in any circumstance use this new VPS/VPN/Proxy 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 do not trust VPN providers' "no-logging policies". This VPN provider should never know your real origin IP.

Please see Recommended VPS hosting providers

VPN VPS

There are plenty of tutorials on how to do this like this one https://proprivacy.com/vpn/guides/create-your-own-vpn-server [Archive.org]

Socks Proxy VPS

This is also an option obviously if you prefer to skip the VPN part.

It is probably the easiest thing to set up since you will just use the SSH connection you have to your VPS and no further configuration should be required besides setting the browser of your guest VM to use the proxy in question.

Here are a few tutorials on how to do this very quickly:

- (Windows/Linux/macOS) https://linuxize.com/post/how-to-setup-ssh-socks-tunnel-for-private-browsing/ [Archive.org]
- (Windows/Linux/macOS) https://www.digitalocean.com/community/tutorials/how-to-route-web-traffic-securely-without-a-vpn-using-a-socks-tunnel [Archive.org]
- (Windows) https://www.forwardproxy.com/2018/12/using-putty-to-setup-a-quick-socks-proxy/ [Archive.org]
- (Linux/macOS) https://ma.trias.be/socks-proxy-linux-ssh-bypass-content-filters/ [Archive.org]

Here is my basic tutorial:

Linux/macOS

Here are the steps:

- Get your anonymous VPS set-up
- From a terminal, SSH to your server by running: `ssh -i ~/.ssh/id_rsa -D 8080 -f -C -q -N username@ip_of_your_server`
- Configure your browser to use localhost:8080 as a Socks Proxy for Browsing
- Done!

Explanation of arguments:

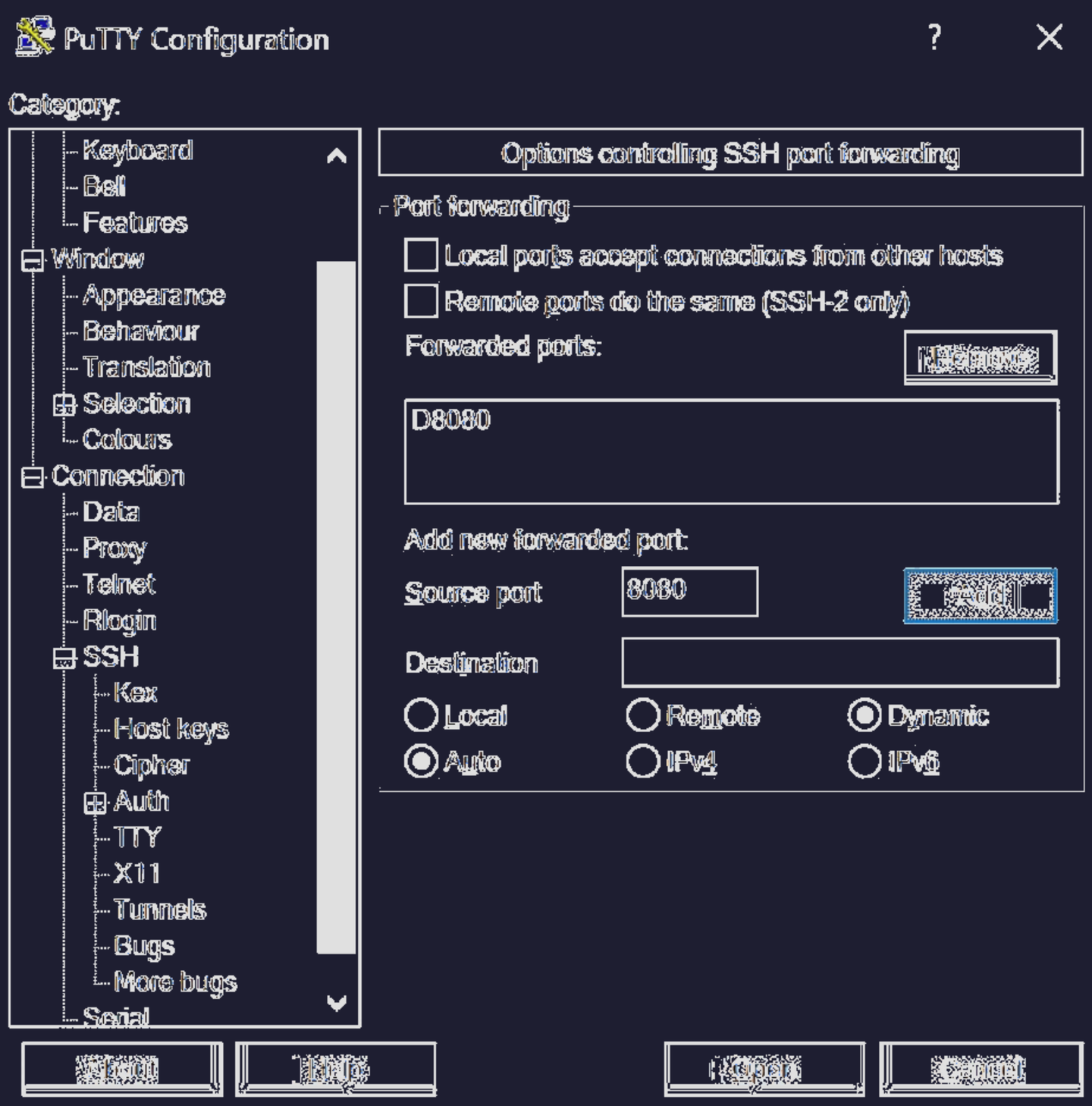
- -i: The path to the SSH key to be used to connect to the host
- -D: Tells SSH that we want a SOCKS tunnel on the specified port number (you can choose a number between 1025 and 65536)
- -f: Forks the process to the background
- -C: Compresses the data before sending it
- -q: Uses quiet mode
- -N: Tells SSH that no command will be sent once the tunnel is up

Windows

Here are the steps:

- Get your anonymous VPS set-up
- Download and install Putty from https://www.putty.org/ [Archive.org]

- Set the following options in Putty and connect to your server



- Connect to your VPS using those settings
- Configure your Browser to use localhost:8080 as a Socks Proxy
- Done!

When Tor and VPNs Aren't Possible

USE EXTREME CAUTION: THIS IS HIGHLY RISKY.

There might be worst-case situations where using Tor and VPNs are not possible due to extensive active censorship or blocking. Even when using Tor Bridges (see Using Tor bridges in hostile environments)

Now, there might also be situations where simply using Tor or a VPN alone could be suspicious and could be dangerous for your safety. If this is the case, you could be in a very hostile environment where surveillance and control are high.

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 Wi-Fi) using your laptop and Tails "unsafe browser". See https://tails.boum.org/contribute/design/Unsafe_Browser/ [Archive.org].

If Tor usage alone is suspicious or risky, you should NOT allow Tails to try establishing a Tor connection at start-up by doing the following:

- At startup open the Additional Settings.
- Enable Unsafe Browser.
- Change the Connection from Direct to "Configure a Tor Bridge or Local Proxy"
- After Start-up, Connect to a safe Network
- When prompted, just quit the Tor Connection Wizard (to not establish a Tor connection)
- Start and use the Unsafe Browser

We would strongly recommend the use of a long-range "Yagi" type directional Antenna with a suitable 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 right equipment. So, this option should not be used during an extended period (minutes at best). See 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 without using Tor/VPN) instead of Tails in such hostile environments if you want data persistence but this might be riskier. We would not 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.

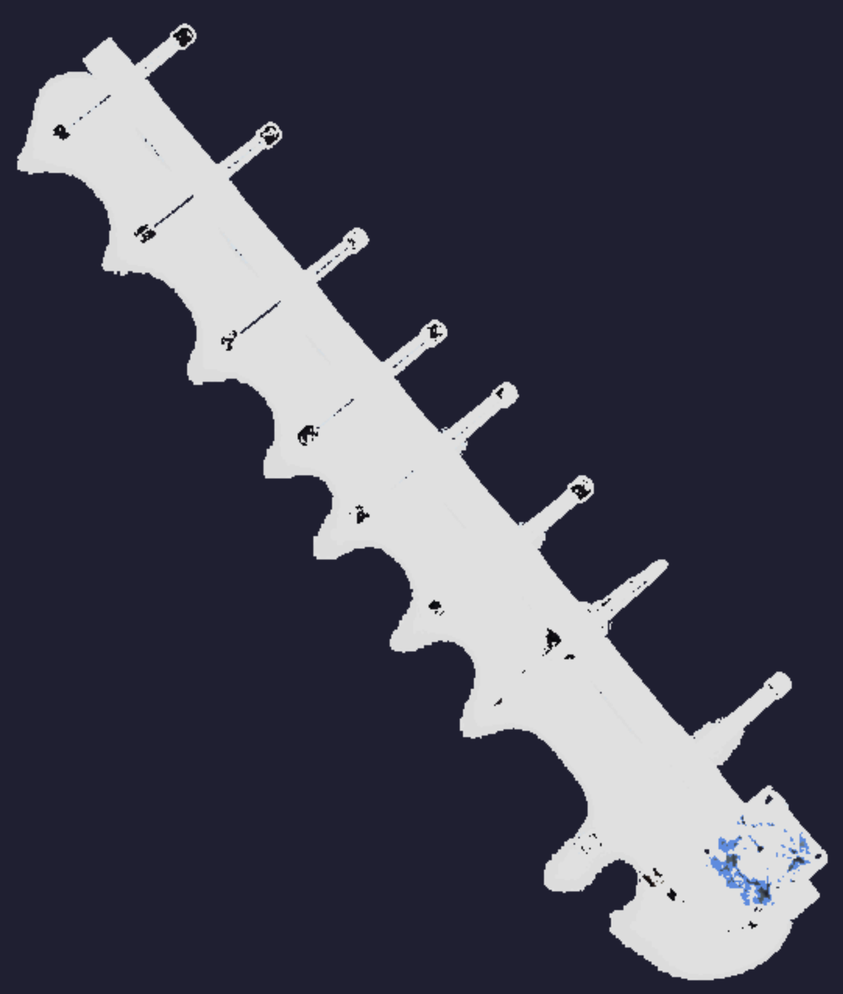
If you resort to this, please keep your online time as short as possible (minutes and not hours).

Be safe and extremely cautious. This is entirely at your own risk.

Consider reading this older but still relevant guide https://archive.flossmanuals.net/bypassing-censorship/index.html [Archive.org]

Connecting to Public Wi-Fi from a Safe Distance

It is 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 are not 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 antennas need to be combined with specific USB adapters that have an external Antenna plug and sufficiently high power to use them.

We would recommend the AWUS036 series in the Alfa brand of adapters (see https://www.alfa.com.tw/ [Archive.org]). 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/ [Archive.org]).

See this post for a comparison of various adapters: https://www.wirelesshack.org/best-kali-linux-compatible-usb-adapter-dongles.html [Archive.org] (Usually those antennas are used by Penetration Testers to probe Wi-Fis from a distance and are often discussed within the scope of the Kali Linux distribution).

The process is simple:

- Plugin and install your USB adapter on your Host OS.
- **Do not 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 of a place with 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 [Invidious], also see their other products here https://www.netally.com/products/ [Archive.org]). 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 a situation where VPN and Tor are not an option, then this could be your only security.

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 [Archive.org] (use the Linux configurations below to get the necessary configuration files)
- Windows Tutorials:
  - Mullvad: https://mullvad.net/en/help/install-mullvad-app-windows/ [Archive.org]
  - iVPN: https://www.ivpn.net/apps-windows [Archive.org]
  - Safing: https://docs.safing.io/portmaster/install/windows [Archive.org]
  - Proton VPN: https://protonvpn.com/support/protonvpn-windows-vpn-application/ [Archive.org]
- macOS:
  - Mullvad: https://mullvad.net/en/help/install-and-use-mullvad-app-macos/ [Archive.org]
  - IVPN: https://www.ivpn.net/apps-macos/ [Archive.org]
  - Safing: Not available on macOS
  - Proton VPN: https://protonvpn.com/support/protonvpn-mac-vpn-application/ [Archive.org]
- Linux:
  - Mullvad: https://mullvad.net/en/help/install-mullvad-app-linux/ [Archive.org]
  - iVPN: https://www.ivpn.net/apps-linux/ [Archive.org]
  - Safing: https://docs.safing.io/portmaster/install/linux [Archive.org]
  - Proton VPN: https://protonvpn.com/support/linux-vpn-setup/ [Archive.org]

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 the Tor network in case of VPN failure.

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/ [Archive.org]
  - Proton VPN: https://protonvpn.com/support/what-is-kill-switch/ [Archive.org]

    - Mullvad: https://mullvad.net/en/help/using-mullvad-vpn-app/#killswitch [Archive.org]

- Whonix Workstation: Coming Soon, it is certainly possible, but we did not find a suitable and easy tutorial yet. It is also worth remembering that if your VPN stops on Whonix, you will 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
    - Proton VPN same as Windows with the client, the option should be in the provided VPN client https://protonvpn.com/blog/macos-vpn-kill-switch/ [Archive.org]

- Linux:

    - Mullvad:

        - https://mullvad.net/en/help/wireguard-and-mullvad-vpn/ [Archive.org]
        - https://mullvad.net/en/help/linux-openvpn-installation/ [Archive.org]

    - Proton VPN: https://github.com/ProtonVPN/linux-cli/blob/master/USAGE.md#kill-switch [Archive.org]
    - iVPN:

        - https://www.ivpn.net/knowledgebase/linux/linux-wireguard-kill-switch/ [Archive.org]
        - https://www.ivpn.net/knowledgebase/linux/linux-kill-switch-using-the-uncomplicated-firewall-ufw/ [Archive.org]

Network Surveillance and Censorship Detection with OONI

So, what is OONI? OONI stands for Open Observatory of Network Interference and is a sub-project of the Tor Project.

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 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.

While this might not be important 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.**
- **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 What about when Tor and VPNs aren't possible?).**

The probe can be found here: https://ooni.org/install/ [Archive.org] for various platforms (iOS, Android, Windows, macOS, and Linux).

Checking Files for Malware

Integrity (if available)

Usually, integrity checks are done using hashes of files (usually stored within checksum files). Older files could use CRC, more recently MD5 but those present several weaknesses (CRC, MD5 that make them unreliable for file integrity checks (which does not mean they are not still widely used in other contexts).

This is because they do not prevent Collision well enough and could allow an adversary to create a similar but malicious file that would still produce in the same CRC or MD5 hash despite having different content.

For this reason, it is usually recommended to use SHA-based hashes and the most used is probably the SHA-2 based SHA-256 for verifying file integrity. SHA is much more resistant to collisions than CRC and MD5. And collisions with SHA-256 or SHA-512 are rare and hard to compute for an adversary.

If a SHA-256 checksum is available from the source of the file, you should not hesitate to use it to confirm the integrity of the file. Note that SHA-1 is not recommended, but is better than not having a hash to compare.

This checksum should itself be authenticated/trusted and should be available from an authenticated/trusted source (obviously you should not trust a file just because it has a checksum attached to it alone).

In the case of this guide, the SHA-256 checksums are available for each file including the PDFs but are also authenticated using a GPG signature allowing you to verify the authenticity of the checksum. This will bring us to the next section about authenticity.

So how to check checksums? (In this case SHA-256 but you could change to SHA-512

- Windows:
  - Open a Command Prompt
  - Run `certutil -hashfile filename.txt sha256` (replace sha256 by sha1 or sha512 or md5)
  - Compare your result to one from a source you trust for that file
- macOS :
  - Open a Terminal
  - SHA: Run `shasum -a 256 /full/path/to/your/file` (replace 256 by 512 or 1 for SHA-1)
  - MD5: Run `md5 /full/path/to/your/file`
  - Compare your result to one from a source you trust for that file
- Linux:
  - Open a Terminal
  - Run `shasum /full/path/to/your/file` (replace shasum by sha256sum, sha512sum or md5sum)
  - Compare your result to one from a source you trust for that file

Remember that checksums are just checksums. Having a matching checksum does not mean the file is safe.

Authenticity (if available)

Integrity is one thing. Authenticity is another thing. This is a process where you can verify some information is authentic and from the expected source. This is usually done by signing information (using GPG for instance) using public-key cryptography.

Signing can serve both purposes and allow you to check for both integrity and authenticity.

If available, you should always verify the signatures of files to confirm their authenticity.

In essence:

- Install GPG for your OS:
  - Windows: gpg4win (https://www.gpg4win.org/ [Archive.org])
  - macOS: GPGTools (https://gpgtools.org/ [Archive.org])

    - Linux: It should be pre-installed in most distributions
- Download the Signature key from a trusted source. If someone is not giving you a key directly, you should check for multiple versions on other websites to confirm you are using the right key (GitHub, GitLab, Twitter, Keybase, Public Keys Servers...).
- Import the trusted key (replace keyfile.asc by the filename of the trusted key):
    - Windows:
        - From a Command Prompt, Run `gpg --import keyfile.asc`
    - macOS:
        - From a Terminal, Run `gpg --import keyfile.asc`
    - Linux:
        - From a Terminal, Run `gpg --import keyfile.asc`
- Verify the file signature against the imported (trusted) signature (replace filetoverify.asc by the signature file that was associated with the file, replace filetoverify.txt by the actual file to verify):
    - Windows:
        - Run `gpg --verify-options show-notations --verify filetoverify.asc filetoverify.txt`
        - The result should show the signature is good and match the trusted signature you imported earlier.
    - macOS:
        - Run `gpg --verify-options show-notations --verify filetoverify.asc filetoverify.txt`
        - The result should show the signature is good and match the trusted signature you imported earlier.
    - Linux:
        - Run `gpg --verify-options show-notations --verify filetoverify.asc filetoverify.txt`
        - The result should show the signature is good and match the trusted signature you imported earlier.

For some other tutorials, please see:

- https://support.torproject.org/tbb/how-to-verify-signature/ [Archive.org]
- https://tails.boum.org/install/vm/index.en.html [Archive.org] (See Basic OpenPGP verification).
- https://www.whonix.org/wiki/Verify_the_Whonix_images [Archive.org]

All these guides should also apply to any other file with any other key.

Security (checking for actual malware)

Every check should ideally happen in sandboxed/hardened Virtual Machines. This is to mitigate the possibilities for malware to access your Host computer.

Anti-Virus Software

You might be asking yourself, what about Anti-Virus solutions? Well, no ... these are not perfect solutions against many modern malware and viruses using polymorphic code. But it does not mean they cannot help against less sophisticated and known attacks. It depends on how to use them as AV software can become an attack vector in itself.

Again, this is all a matter of threat modeling. Can AV software help you against the NSA? Probably not. Can it help you against less resourceful adversaries using known malware? Probably.

Some will just argue against them broadly like Whonix but this topic is being discussed and disputed even at Whonix by other members of their community.

Contrary to popular myths perpetuating the idea that only Windows is subject to malware and that detection tools are useless on Linux and macOS:

- Yes, there are viruses and malware for Linux''''
- Yes, there are viruses and malware for macOS''

My take on the matter is on the pragmatic side. There is still room for some AV software for some selective and limited use. But it depends on which one and how you use them:

- Do not use AV software with real-time protection as they often run with administrator privileges and can become an attack vector.
- Do not use Commercial AV software that uses any "cloud protection" or sends extensive telemetry and samples to their company.
- Do use Open-Source non-real-time offline Anti-Virus/Anti-Malware tools as an added measure to scan some files such as:
  - Windows/Linux/macOS/Qubes OS: ClamAV (https://www.clamav.net/ [Archive.org])
  - Linux/Qubes OS: RFXN Linux Malware Detect (https://github.com/rfxn/linux-malware-detect [Archive.org])
  - Linux/Qubes OS: Chkrootkit (http://www.chkrootkit.org/ [Archive.org])
- You could also use online services for **non-sensitive files*** such as VirusTotal (https://www.virustotal.com/gui/) or Hybrid-analysis (https://hybrid-analysis.com/).
  - You could also just check the VirusTotal database for the hash of your file if you don't want to send it over (see https://developers.virustotal.com/v3.0/docs/search-by-hash [Archive.org] (See the [Integrity (if available):] section again for guidance on how to generate hashes).
  - Other tools are also available for non-sensitive files and a convenient list is right here: https://github.com/rshipp/awesome-malware-analysis#online-scanners-and-sandboxes [Archive.org]
- **Please be aware that while VirusTotal might seem very practical for scanning various files, their "privacy policy" is problematic (see https://support.virustotal.com/hc/en-us/articles/115002168385-Privacy-Policy [Archive.org]) and states:**

"When you submit Samples to the Services, if you submit Samples to the Services, You will collect all of the information in the Sample itself and information about the act of submitting it".

So, remember 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 (that do not send samples online).

So, if you are in doubt:

- For non-sensitive files, we do encourage you to check any documents/images/videos/archives/programs you intend to open with VirusTotal (or other similar tools) because ... Why not? (Either by uploading or checking hashes).
- For sensitive files, we would recommend at least an offline unprivileged ClamAV scan of the files.

For instance, this guide's PDF files were submitted to VirusTotal because it is meant to be public knowledge and we see no valid argument against it. It does not guarantee the absence of malware, but it does not hurt to add this check.

Manual Reviews

You can also try to check various files for malware using various tools. This can be done as an extra measure and is especially useful with documents rather than apps and various executables.

These methods require more tinkering but can be useful if you want to go the extra length.

PDF files

Again, regarding the PDFs of this guide and as explained in the README of my repository, you could check for anomalies using PDFID which you can download at https://blog.didierstevens.com/programs/pdf-tools/ [Archive.org]:

- Install Python 3 (on Windows/Linux/macOS/Qubes OS)
- 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 #This indicates the presence of Javascript
/JavaScript 0 #This indicates the presence of Javascript
/AA 0 #This indicates the presence of automatic action on opening
```

```
/OpenAction 0 #This indicates the presence of automatic action on opening
/AcroForm 0 #This indicates the presence of AcroForm which could contain JavaScript
/JBIG2Decode 0 #This indicates the use of JBIG2 compression which could be used for obfuscating content
/RichMedia 0 #This indicates the presence of rich media within the PDF such as Flash
/Launch 0 #This counts the launch actions
/EmbeddedFile 0 #This indicates there are embedded files within the PDF
/XFA 0 #This indicates the presence of XML Forms within the PDF
```

Now, what if you think the PDF is still suspicious? Fear not ... there are more things you can do to ensure it is not malicious:

- **Qubes OS:** Consider using https://github.com/QubesOS/qubes-app-linux-pdf-converter [Archive.org] which will convert your PDF into a flattened image file. This should theoretically remove any malicious code in it. Note that this will also render the PDF formatting useless (such as links, headings, bookmarks, and references).
- **(Deprecated) Linux/Qubes OS** (or possibly macOS through Homebrew or Windows through Cygwin): This should *theoretically* remove any malicious code in it, but will also render the PDF formatting useless (such as links, headings, bookmarks, and references). Something similar to how rasterizing this website into a PDF works when swtiching to dark-mode. **Note that this tool is deprecated and relies on a library called "ImageMagick" which is known for several security issues. You should not use this tool even if it is recommended in some other guides.**

Other types of files

Here are some various resources for this purpose where you will find what tool to use for what type:

- **For Documents/Pictures:** Consider using https://github.com/freedomofpress/dangerzone [Archive.org] which was inspired by Qubes PDF Converted above and does the same but is well maintained and works on all OSes. This tool also works with Images, ODF files, and Office files (Warning: On Windows, this tool requires Docker-Desktop installed and this might (will) interfere with Virtualbox and other Virtualization software because it requires enabling Hyper-V. VirtualBox and Hyper-V do not play nice together. Consider installing this within a Linux VM for convenience instead of a Windows OS).
- **For Videos:** Be extremely careful, use an up-to-date player in a sandboxed environment. Remember https://www.vice.com/en/article/v7gd9b/facebook-helped-fbi-hack-child-predator-buster-hernandez [Archive.org]
- This practical cheat sheet from SANS: https://digital-forensics.sans.org/../media/analyzing-malicious-document-files.pdf [Archive.org] (warning, many of those tools might be harder to use on 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 that might be of use).
- This GitHub repository with various resources on malware analysis: https://github.com/rshipp/awesome-malware-analysis [Archive.org]
- This interesting PDF detailing which tool to use for which file type https://www.winitor.com/pdf/Malware-Analysis-Fundamentals-Files-Tools.pdf [Archive.org]

Even with all those resources, keep in mind you might still get advanced malware if those are not detected by those various tools. Be careful and remember to handle these files within isolated Virtual Machines, if possible, to limit the attack surface and vectors.

Bypassing Local Restrictions on Supervised Computers

There might be situations where the only device you have at your disposal is not really yours such as:

- Using a Work computer with restrictions in place on what you can do/run.
- Misuse of Parental control features to monitor your computer usage (despite you being a non-consenting Adult).
- Misuse of various monitoring apps to monitor your computer usage against your will.

The situation might look desperate, but it is not necessarily the case as there are some safe ways to bypass these depending on how well your adversaries did their job securing your computer.

Portable Apps

There are plenty of methods you could use to bypass those restrictions locally. One of them would be to use portable apps. Those apps do not require installation on your system and can be run from a USB key or anywhere else.

But this is not a method we would recommend.

This is because those portable apps will not necessarily hide themselves (or be able to hide themselves) from the usage reports and forensic examination. This method is just too risky and will probably arise issues if noticed if you are in such a hostile environment.

Even the most basic controls (supervision or parental) will send out detailed app usage to your adversary.

Bootable Live Systems

This method is the one we would recommend in those cases.

It is relatively easy for your adversary to prevent this by setting up firmware BIOS/UEFI (see Bios/UEFI/Firmware Settings of your laptop) controls but usually most adversaries will overlook this possibility which requires more technical knowledge than just relying on Software.

This method could even decrease suspicion and increase your plausible deniability as your adversaries think they have things under control and that everything appears normal in their reports.

This method only depends on one security feature (that they probably did not turn on in most cases): Boot Security.

Boot Security is divided into several types:

- Simple BIOS/UEFI password preventing the change of the boot order. This means you cannot start such a live system in place of your supervised OS without providing the BIOS/UEFI password.
- Secure Boot. This is a "standard" feature preventing you from starting unsigned systems from your computer. While this feature could be configured to only allow your supervised system, usually by default it will allow running an entire range of signed systems (signed by Microsoft or the Manufacturer for instance).

Secure Boot is relatively easy to bypass as there are plenty of Live Systems that are now Secure Boot compliant (meaning they are signed) and will be allowed by your laptop.

The BIOS/UEFI password on the other hand is much harder to bypass without risks. In that case, you are left with two options:

- Guess/Know the password so that you can change the boot order of your laptop without raising suspicions
- Reset the password using various methods to remove the password. **we would not recommend doing this because if your adversaries went the extra length of enabling this security feature, they probably will be suspicious if it were disabled, and this might increase suspicion and decrease your plausible deniability considerably.**

Again, this feature is usually overlooked by most unskilled/lazy adversaries and in my experience left disabled.

This is your best chance into bypassing local controls without traces.

The reason is that most of the controls are within your main Operating System software and only monitor what happens within the Operating System. Those measures will not be able to monitor what happened at the Hardware/Firmware level before the Operating System loads.

Precautions

While you might be able to bypass local restrictions easily using a Live System such as Tails, remember that your network might also be monitored for unusual activities.

Unusual network activities showing up from a computer at the same time your computer is seemingly powered off might raise suspicions.

If you are to resort to this, you should never do so from a monitored/known network but only from a safe different network. Ideally a safe public wi-fi (See Find some safe places with decent public Wi-Fi).

Do not use a live system on a Software supervised/monitored device on a known network.

Refer to the Tails route to achieve this. See the Tails route and What about when Tor and VPNs aren't possible? sections.

Choosing a Browser for Your Guest VM

Temporary Important Warning: Please see [Microarchitectural Side-channel Deanonymization Attacks:] for all browsers except Tor Browser.

There are 6 possibilities of browser to use on your guest/disposable VM:

- Brave (Chromium-based)
- Edge (Chromium-based, Windows Only)
- Firefox
- Safari (macOS VM only)
- Tor Browser

Here is a comparison table of one fingerprinting test of various browsers with their native settings (**but Javascript enabled for usability, except for Tor Safest mode**).

Disclaimer: these tests while nice are not conclusive of the real fingerprinting resistance. But they can help compare browsers between each other.

Browser	https://coveryourtracks.eff.org/ Fingerprinting Test with real Ad
Safari (Normal)*	Fail (Unique)
Safari (Private Window) *	Fail (Unique)
Edge (Normal)**	Fail (Unique)
Edge (Private Window) **	Fail (Unique)
Firefox (Normal)	Fail (Unique)
Firefox (Private Window)	Fail (Unique)
Chrome (Normal)	Fail (Unique)
Chrome (Private Window)	Fail (Unique)
Brave (Normal)	Passed (Randomized)
Brave (Private Window)	Passed (Randomized)
Brave (Tor Window)	Passed (Randomized)
Tor Browser (Normal mode)	Partial
Tor Browser (Safer mode)	Partial
Tor Browser (Safest mode)	Unknown (Result did not load)

- *: macOS only. **: Windows only.

Another useful resource to be considered for comparing browsers is: https://privacytests.org/ [Archive.org]

Brave

This is my recommended/preferred choice for a Browser within your guest VMs. This is not my recommended choice for a Browser within your Host OS where we strictly recommend Tor Browser as they recommend it themselves.

Why Brave despite the controversies?

- You will encounter fewer issues later with account creations (captchas ...). This is based on our experiences trying to create plenty of online identities using various browsers. You will have to trust us on that.
- You will enjoy native ad-blocking where none is available in others by default without adding extensions.
- Performance is arguably better than Firefox.
- Brave is arguably better at fingerprinting resistance than others.
- Security of Chromium-based Browser is arguably better and more secure than Firefox'. Within the context of this guide, security should be privileged to prevent any vulnerability or exploit from gaining access to the VM.
- Comparison of both by Mozilla: https://www.mozilla.org/en-US/firefox/browsers/compare/brave/ [Archive.org]
- Comparison of both by Techlore: https://www.youtube.com/watch?v=qkJGF3syQy4 [Invidious]
- The whole traffic will be routed over a VPN over Tor anyway. So even if you mistakenly opt-in for some telemetry, it is not so important. Remember that in this anonymity threat model, we are mostly after anonymity and security. The privacy of our online identities does not matter that much unless the privacy issue is also a security issue that could help deanonymize you.
- Brave was found to be sending no identifiable telemetry compared to other browsers.

Ungoogled-Chromium

This browser is considered a security liability due to their systemic lagging on security patches.

It is strongly advised not to use Ungoogled-Chromium.

Edge

This is for Windows users only. Edge is a solid choice too.

- You will encounter fewer issues later with account creations (captchas ...). This is based on our experiences trying to create plenty of online identities using various browsers. You will have to trust us on that.
- Better Security than Firefox as it is Chromium-based'.
- Better Performance than Firefox.
- The whole traffic will be router through Tor anyway.
- Can benefit from additional security using Microsoft Defender Application Guard (MDAG). Note that this feature cannot be enabled in a Virtualbox VM unfortunately.
- Native tracker blocking (Similar to Brave Shields).

Cons:

- You will have to disable some telemetry within the Browser

Safari

The macOS default browser.

Pros:

- It is a Browser with decent security and sandboxing capabilities.

Cons:

- It is macOS only (obviously)
- It requires signing-in into the App Store to install extensions (impossible within the scope of this guide since it is a VM)
- Even if you could, it lacks the best Extensions available for Firefox and Chrome.

Overall, we would not recommend using Safari on a macOS VM but instead, go for another Browser such as Brave or Firefox.

Firefox

And of course, lastly, you could go with Firefox,

Pros:

- Well, it is out of the "Chromium" world and not taking part in expanding Chromium market share
- In addition to being out of the Chromium world, it is also completely out of the Google world (despite the Mozilla Foundation being almost entirely funded by Google).
- An impressive amount of customization through extensions for every possible need.
- Firefox can be severely hardened to almost match the security of Chromium-based browsers.

Cons:

- Poorer performance compared to Chromium.

Security (especially sandboxing) of Firefox is arguably weaker than Chromium-based browsers.

- You will experience more captchas (this is based on my tests).

Tor Browser

If you are extra paranoid and want to use Tor Browser and have "Tor over VPN over Tor", you could go with Tor Browser within the VM as well. This is completely pointless/useless.

We would not recommend this option. It is just silly.

Hardening Your Browsers

In this section, we'll discuss hardening your browsers. This has a heavy focus on the difference between Tracking Reduction and Tracking Evasion, and the pros and cons of either. First, let's define what they are as described by Rohan Kumar:

- Tracking reduction (TR)
  - TR aims to reduce the amount of data collected about an exposed user. It reduces a footprint's spread primarily by blocking trackers. Sometimes this can increase the size of a footprint.
- Tracking evasion (TE)
  - TE reduces the amount of data exposed by a user. Rather than eliminating data collection itself, TE prevents useful data from being made available in the first place. In other words, it reduces a footprint's size.

Browsers that provide Tracking Reduction are to be used for a more casual [Threat Model]Threat modeling resources whereas Tracking Evasion is more complex. But both need to be explored. Tracking Reduction focuses on browsing with less tracking. It involves things like content-blocking, firewalls, opt-outs, flipping telemetry buttons, etc. If you're this far into the guide, you likely have a very good understanding of this already. Tracking Evasion, however, involves techniques like using the portable Tor Browser Bundle to anonymize your footprint and online identity, avoiding identifiable extensions, and using randomized keystroke delays. It's more about minimizing your online footprint, to give you a less fingerprintable browsing environment and internet usage.

A brief mention of this is necessary in determining operation needs for both. You need a certain level of understanding in both to achieve good standards and develop better browsing habits. This can and will overall provide you with a more viable solution to public trackers, government organizations looking to trace/track your browsing habits back to you, even just trolls attempting to doxx you.

The following are the recommended safest routes for each browser according to the current versions of their respective software and the ability each one has to become more secure. In the guide we will provide both Tracking Reduction & Evasion and it will not require you to write even a single line of code.

Brave

- Download and install Brave browser from https://brave.com/download/ [Archive.org]
- **Open** Brave Browser
- Go into **Settings** > **Appearances** (`brave://settings/appearance`)
  - **Disable** "Show Top Sites"
  - **Disable** "Show Brave Suggested Sites"
  - **Disable** "Show Brave Rewards icon in address bar"
  - **Enable** "Always show full URLs"
- Go into **Settings** > **Shields** (`brave://settings/shields`)
  - Set Shields to **Advanced**
  - Set "Trackers and Ads blocking" to **Aggressive**
  - Set Upgrade connections to HTTPS to **Enabled**
  - Set Cookie blocking to **Only cross-site**
  - Set Fingerprinting blocking to **Standard** or **Strict**
- Go into **Settings** > **Social media blocking** (`brave://settings/socialBlocking`)
  - **Uncheck** everything unless needed
- Go to **Settings** > **Search engine** (`brave://settings/search`)
  - See Search Engines
- Go into **Settings** > **Extensions** (`brave://settings/extensions`)
  - **Disable** everything except "Private Window with Tor"
  - Set both **Resolve** methods to "Ask"

- Go into **Settings** > **Wallet** (`brave://settings/wallet`)
    - **Disable** "Show Brave Wallet icon on toolbar"
    - Set **Default Ethereum wallet** to "None"
    - Set **Default Solana wallet** to "None"
- Go into **Settings** > **Privacy and Security** (`brave://settings/privacy`)
    - Leave **WebRTC** to "Default"
    - **Disable** "Allow privacy-preserving product analytics (P3A)"
    - **Disable** "Automatically send daily usage ping to Brave"
    - Go into "Clear Browsing Data"
        - Select **On Exit**
        - Check all options
        - **Click** "Save"
- Open a new Tab
- **Click** "Customize" in the lower right corner
    - **Disable** everything in Customize Dashboard except maybe the clock
- Go into **Settings** > **Shields** > **Content filters** (`brave://settings/shields/filters`)
    - Select any additional adblocking filter you want
        - Recommended: **CJX's Annoyance List**, **Easylist-Cookie List**, **Fanboy Annoyances List**, **Fanboy Social List**, **Fanboy's Mobile Notifications List**, and **uBlock Annoyances List**
    - Add custom filter lists
        - Add the Actually Legitimate URL Shortener Tool which uses the rules found in ClearURLs below
        - Add the AdGuard URL Tracking Protection which enables generic `$removeparam` rules
    - To keep all applied filters, **click** "Save"
- Do not ever enable Brave Rewards (button should be hidden on all sites)

Addons to consider on Brave if you want additional protections:

- LocalCDN (https://chrome.google.com/webstore/detail/localcdn/njdfdhgcmkocbgbhcioffdbicglldapd)
    - Alternatively, DecentralEyes (https://chrome.google.com/webstore/detail/decentraleyes/ldpochfccmkkmhdbclfhpagapcfdljkj)
- PrivacyBadger (https://chrome.google.com/webstore/detail/privacy-badger/pkehgijcmpdhfbdbbnkijodmdjhbjlgp)
- NoScript (https://chrome.google.com/webstore/detail/noscript/doojmbjmlfjjnbmnoijecmcbfeoakpjm)
- Either ClearURLs (https://chrome.google.com/webstore/detail/clearurls/lckanjgmijmafbedllaakclkaicjfmnk) **OR** the custom list above
- LibRedirect (https://libredirect.github.io/)

That's it and you should be pretty much covered. For full paranoia, you can also just "Block Scripts" to disable Javascript. Note that even disabling Javascript might not protect you fully.

Ungoogled-Chromium

This browser is considered a security liability due to their systemic lagging on security patches.

It is strongly advised not to use Ungoogled-Chromium.

Edge

Windows only:

- Open Edge
- Go into Settings
- Go to Profiles and make sure everything is unchecked in every section (Personal Info, Passwords, Payment info, Profile preferences)
- Go to Privacy, search, and services:
    - Go to Tracking Prevention:
        - Set to Strict or at least Balanced
        - Set to always use Strict with InPrivate Windows
- Go to Privacy:
    - Enable send Do Not Track
    - Disable the options for the website to check your payment methods
- Go to Optional Diagnostic Data:
    - Disable it
- Go to Personalize your Web Experience:
    - Disable it
- Go to Security
    - Disable everything
- Go to Services
    - Disable everything
    - In Address Bar and Search:
        - Disable everything and change the search engine (see Search Engines)
- Go to Cookies and Sites Permissions:
    - Within All Permissions:
        - Within Cookies, make sure "Block Third-Party Cookies" is checked
        - Block everything except:
            - Javascript
            - Images

Enable Application Guard for Edge (only on Host OS, not possible within a VirtualBox VM):

Skip if this is a VM

- Open Control Panel.
- Click on Programs
- Click on Turn Windows features on or off link
- Check the Windows Defender Application Guard option
- Click OK.
- Click Restart.
- Now you can open Edge and open a new "Application Guard" Window.

That's about it for Edge but you are also free to add extensions from the Chrome Store such as:

- uBlock Origin (https://chrome.google.com/webstore/detail/ublock-origin/cjpalhdlnbpafiamejdnhcphjbkeiagm)
- LocalCDN (https://chrome.google.com/webstore/detail/localcdn/njdfdhgcmkocbgbhcioffdbicglldapd)
    - Alternatively, DecentralEyes (https://chrome.google.com/webstore/detail/decentraleyes/ldpochfccmkkmhdbclfhpagapcfdljkj)
- PrivacyBadger (https://chrome.google.com/webstore/detail/privacy-badger/pkehgijcmpdhfbdbbnkijodmdjhbjlgp)
- HTTPS Everywhere (https://chrome.google.com/webstore/detail/https-everywhere/gcbommkclmclpchllfjekcdonpmejbdp)
- NoScript (https://chrome.google.com/webstore/detail/noscript/doojmbjmlfjjnbmnoijecmcbfeoakpjm)
- ClearURLs (https://chrome.google.com/webstore/detail/clearurls/lckanjgmijmafbedllaakclkaicjfmnk)
- LibRedirect (https://libredirect.github.io/)

Safari

macOS Only:

- Open Safari
- Click the Safari top left Menu
- Click Preferences
    - On the General Tab:
        - Change New Windows to "Empty Page"
        - Change New Tabs to "Empty page"
        - Change the Remove History after to "1 day"
        - Change the Remove Download list items to "When Safari Quits" or "When Successful Download"
        - Uncheck "Open Safe Files After Downloading"
    - On the Security Tab:
        - Disable "Warn when visiting a Fraudulent Website" (this sends the URLs your visit to Google for screening)
    - On the Privacy Tab:
        - Uncheck "Web Advertising"
    - On the Advanced Tab:
        - Check the "Show full website address"

Consider Additional browser precautions with JavaScript enabled

That's about it. Unfortunately, you will not be able to add extensions as those will require you to sign in into the App Store which you cannot do from a macOS VM. Again, we would not recommend sticking to Safari in a macOS VM but instead switching to Brave or Firefox.

Firefox

Normal settings

- Open Firefox
- On the Firefox Home Page:
    - Click Personalize
    - Uncheck/Disable Everything
- Open Settings:
    - Go into Search

        - Change the search engine (See Search Engines)
    - Go into Privacy & Security
        - Set to Custom
            - Cookies: Select All Third-Party Cookies
            - Tracking Content: In all Windows
            - Check Cryptominers
            - Check Fingerprinters
        - Set always send "Do Not Track"
    - Go to Logins and Passwords
        - Uncheck "Ask to save logins and passwords for websites"
    - Go to Permissions
        - Location: check block new requests
        - Camera: check block new requests
        - Microphone: check block new requests
        - Notifications: check block new requests
        - Autoplay: select Disable Audio and Video
        - Virtual Reality: check block new requests
        - Check Block Pop-ups
        - Check Warn when websites try to install add-ons
    - Go to Firefox Data Collection and Use
        - Disable everything
    - Go to HTTPS-Only Mode
        - Enable it on all Windows

Advanced settings

Consider Arkenfox/user.js, a heavily maintained and very easy to use browser config which uses a "user.js" to set all the privacy settings and disk avoidance values. Below we recommend that if you are not setting the Arkenfox config, at least setting the **about:config** values below. Arkenfox applies many others but these are the bare minimum for your protection while browsing. Remember: doing nothing and using a browser with its defaults will already be leaking many identifiable and trackable characteristics which are unique to you. See Browser and Device Fingerprinting for more details on why default settings in browsers are unsafe.

Those settings are explained on the following resources in order of recommendation if you want more details about what each setting does:

1. https://wiki.archlinux.org/title/Firefox/Privacy [Archive.org] **(most recommended)**
2. https://proprivacy.com/privacy-service/guides/firefox-privacy-security-guide [Archive.org]

Here are most of the steps combined from the sources above (some have been omitted due to the extensions recommended later below):

- Navigate to "about:config" in the URL bar
- Click Accept the Risk and Continue
    - Safe Settings (should not break anything)
        - Disable Firefox Pocket
            - Set "extensions.pocket.enabled" to false
        - Disable All Telemetry

- Set "browser.newtabpage.activity-stream.feeds.telemetry" to false
- Set "browser.ping-centre.telemetry" to false
- Set "browser.tabs.crashReporting.sendReport" to false
- Set "devtools.onboarding.telemetry.logged" to false
- Set "toolkit.telemetry.enabled" to false
- Search for "toolkit.telemetry.server" and clear it
- Set "toolkit.telemetry.unified" to false
- Set "beacon.enabled" to false
- Disable Pre-Fetching
  - Set "network.dns.disablePrefetch" to true
  - Set "network.dns.disablePrefetchFromHTTPS" to true
  - Set "network.predictor.enabled" to false
  - Set "network.predictor.enable-prefetch" to false
  - Set "network.prefetch-next" to false
  - Set "browser.urlbar.speculativeConnect.enabled" to false
- Disable Javascript in PDFs
  - Set "pdfjs.enableScripting" to false
- Disable obsolete SSL encryption
  - Set "security.ssl3.rsa_des_ede3_sha" to false
  - Set "security.ssl.require_safe_negotiation" to true
- Disable Firefox Accounts
  - Set "identity.fxaccounts.enabled" to false
- Disable Geolocation
  - Set "geo.enabled" to false
- Disable Web Notifications
  - Set "dom.webnotifications.enabled" to false
- Disable Copy/Paste Notifications
  - Set "dom.event.clipboardevents.enabled" to false
- Disable Microphone/Camera status fetching
  - Set "media.navigator.enabled" to false
- Enable "Do Not Track"
  - Set "privacy.donottrackheader.enabled" to true
- Disable SafeBrowsing
  - Set "browser.safebrowsing.malware.enabled" to false
  - Set "browser.safebrowsing.phishing.enabled" to false
  - Set "browser.safebrowsing.downloads.remote.enabled" to false
- Moderate Settings (could break some websites)
  - Disable WebRTC (this will break all websites with video/audio communications)

        - Set "media.peerconnection.enabled" to false
        - Set "media.navigator.enabled" to false
    - Disable WebGL (this will break some media intensive websites)
        - Set "webgl.disabled" to true
    - Disable DRM
        - Set "media.eme.enabled" to false
        - Set "media.gmp-widevinecdm.enabled" to false
    - Set Cookiies Behavior
        - Set "network.cookie.cookieBehavior" to 1
        - Set "network.http.referer.XOriginPolicy" to 2
    - Change referer policy
        - Set "network.http.referer.XOriginTrimmingPolicy" to 2
    - Change Session Storage behavior
        - Set "browser.sessionstore.privacy_level" to 2
    - Disable Connection Tests for Captive Portals
        - Set "network.captive-portal-service.enabled" to false
    - Disable "Trusted Recursive Resolver"
        - Set/Create "network.trr.mode" and set it to 5
- Advanced (this will break some websites)
    - Set "privacy.resistFingerprinting" to true
    - Set "privacy.trackingprotection.fingerprinting.enabled" to true
    - Set "privacy.trackingprotection.cryptomining.enabled" to true
    - Set "privacy.trackingprotection.enabled" to true
    - Set "browser.send_pings" to false
    - Set "change privacy.firstparty.isolate" to true
    - Set "network.http.referer.XOriginPolicy" to "2" or use **Smart Referer** below
    - Set "change network.cookie.lifetimePolicy" to 2 (this deletes all cookies after each session)

Addons to install/consider

- uBlock Origin (https://addons.mozilla.org/en-US/firefox/addon/ublock-origin/)
- Smart Referer (https://addons.mozilla.org/firefox/addon/smart-referer/)
    - Set "network.http.referer.XOriginPolicy" value of "2" to "0" (so the extension works). **Disable** the whitelist (uncheck the **Use default whitelist** box) and set **Domain name matching** to **Strict**.
- NoScript (https://addons.mozilla.org/en-US/firefox/addon/noscript/)
    - Blocks **all** scripts by default, no exceptions. Necessary in regular browser if you want to block all script executions. Not necessary in Tor Browser.
    - Within the options, change **Default** options to check everything except "ping", "unrestricted CSS", and "LAN". This will re-enable JavaScript and other web features, to prevent many websites from breaking
- LibRedirect (https://libredirect.github.io/)

    - Redirect less privacy friendly websites like YouTube and Wikipedia to more privacy friendly open-source alternatives
- Skip Redirect (https://github.com/sblask/webextension-skip-redirect)

Bonus resources

Here are also two recent guides to harden Firefox:

- https://chrisx.xyz/blog/yet-another-firefox-hardening-guide/ [Archive.org]
- https://ebin.city/~werwolf/posts/firefox-hardening-guide/ [Archive.org]

Virtualization

So, you might ask yourself, what is Virtualization?

Basically, it is like the Inception movie with computers. You have emulated software computers called Virtual Machines running on a physical computer. And you can even have Virtual Machines running within Virtual machines if you want to (but this will require a more powerful laptop in some cases).

Here is a little basic illustration of what Virtualization is:



Each Virtual Machine is a sandbox. Remember the reasons for using them are to prevent the following risks:

- Mitigate local data leaks and easier clean-up in case something gets messed up or it is suspected to be compromised.
- Reduce malware/exploit attack surfaces (if your VM is compromised, the adversary still must figure out he is in a VM and then gain access to the Host OS which is not so trivial).
- Mitigate online data leaks by being able to enforce strict network rules on Virtual Machines for accessing the network (such as passing through the Tor Network).

Nested virtualization risks

There is an inherently larger attack surface when nesting virtualization.

Here's some host information that can be leaked through the Virtual Machine:

- Organizationally unique identifier or OUI - the unique identifier assigned to VMWare Guest VMs;
- Virtual Windows registry keys like `ProductID` might show the Host Machine's environment: `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProductId XXXXX-123-1234567-12345`
- HDD, GPU, and mouse drivers can be exposed through: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\`
- Registry entries will show that this is a virtual mouse: `%WINDIR%\system32\drivers\vmmouse.sys`
- Descriptor Table Registers: https://stackoverflow.com/questions/52505313/what-are-descriptor-registers/52505743#52505743
    - Since it's a Virtual Machine using the same CPU cores, the descriptor values are relocated due to there only being space for one of each identifier per CPU. This is a dead giveaway and is used in detection by advanced malware. It's employed by malware architects to tell when the program is being ran in a forensics environment (e.g., Remnux or Flare VM) - popular tools/OS that are used by experts to analyze malware.
- Guest VMs also indirectly access the same hardware as the Host OS.

See https://www.malwarebytes.com/blog/news/2014/02/a-look-at-malware-with-virtual-machine-detection for more techniques used by malware to detect virtualization. These techniques are mostly prevented by appending some settings to your VM config file (.vmx). https://blog.talosintelligence.com/2009/10/how-does-malware-know-difference.html

Using Tor Bridges in Hostile Environments

In some environments, your ISPs might be trying to prevent you from accessing Tor. Or accessing Tor openly might be a safety risk.

In those cases, it might be necessary to use Tor bridges to connect to the Tor network (see Tor Documentation https://2019.www.torproject.org/docs/bridges [Archive.org] and Whonix Documentation https://www.whonix.org/wiki/Bridges [Archive.org]). Optionally, if you are able, you should (seriously!) consider running a bridge https://blog.torproject.org/run-tor-bridges-defend-open-internet/ [Archive.org] yourself, as this would greatly help reduce the amount of censorship in the world.

Bridges are special Tor entry nodes that are not listed on the Tor public directory. Some of those are running on people running the Snowflake Browser extension while others are running on various servers around the world. Most of those bridges are running some type of obfuscation method called obfs4.

*Only available for Desktop Tor users: Recently, the Tor Project has made it incredibly simple to access Bridges with **Connection Assist**, and it is now automatically done in hostile or censored regions. Simply open the Tor Browser and the connection will be configured based on your needs on any hostile network. Previously, we had a list of options below this paragraph which were necessary to enable and configure bridges, but now that this is done automatically using moat.* [Archive.org]

Here is the definition from the Tor Browser Manual: "obfs4 makes Tor traffic look random and prevents censors from finding bridges by Internet scanning. obfs4 bridges are less likely to be blocked than its predecessor, obfs3 bridges".

Some of those are called "Meek" bridges and are using a technique called "Domain Fronting" where your Tor client (Tails, Tor Browser, Whonix Gateway) will connect to a common CDN used by other services. To a censor, it would appear you are connecting to a normal website such as Microsoft.com. See https://gitlab.torproject.org/legacy/trac/-/wikis/doc/meek for more information.

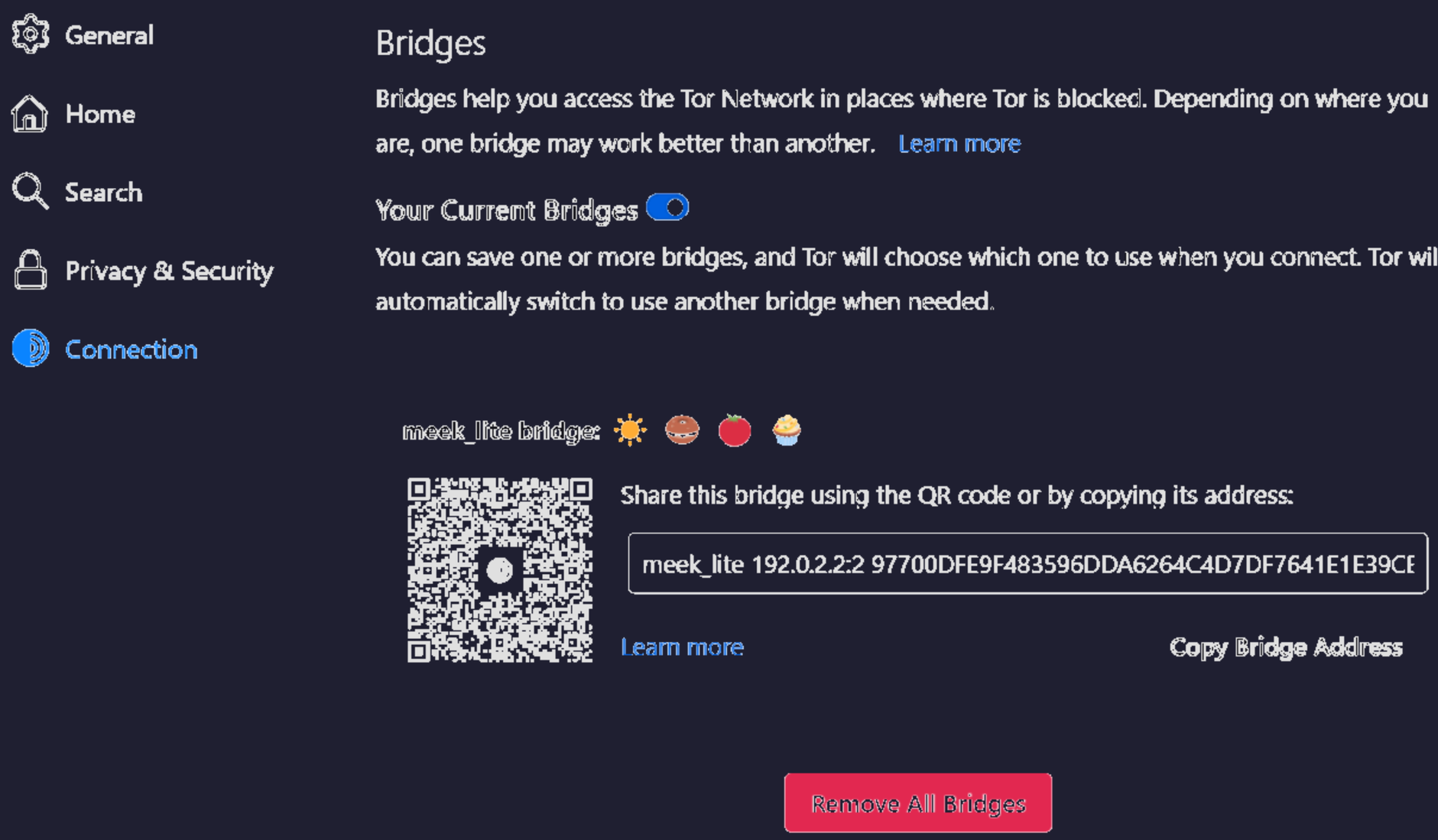
As per their definition from their manual: "meek transports make it look like you are browsing a major web site instead of using Tor. meek-azure makes it look like you are using a Microsoft web site". Snowflake bridges make it appear like your connections are phone calls to random internet users. This is a type of "domain fronting" . See "domain fronting" from the link in the previous paragraph for a detailed explanation of these types of secret "bridges".

Lastly, there are also bridges called Snowflake bridges that rely on users running the snowflake extension in their browser to become themselves entry nodes. See https://snowflake.torproject.org/ [Archive.org].

First, you should proceed with the following checklist to make sure you cannot circumvent Tor Blocking (double-check) and try to use Tor Bridges (https://bridges.torproject.org/ [Archive.org]):

- (Recommended if blocked but **safe**) Try to get an obfs4 bridge in the Tor connection options.
- (Recommended if blocked but **safe**) Try to get a snowflake bridge in the Tor connection options.

- **(Recommended if hostile/risky environment)** Try to get a meek bridge in the Tor connection options (might be your only option if you are for instance in China).



(Illustration from Tor Browser Bridge Configuration)

If none of those build-in methods are working, you could try getting a manual bridge either from:

- https://bridges.torproject.org/bridges?transport=meek (for a meek bridge)
- https://bridges.torproject.org/bridges?transport=obfs4 (for an obfs4 bridge)

This website obviously could be blocked/monitored too so you could instead (if you have the ability) ask someone to do this for you if you have a trusted contact and some e2e encrypted messaging app.

Finally, you could also request a bridge request by e-mail to bridges@torproject.org with the subject empty and the body being: "get transport obfs4" or "get transport meek". There is some limitation with this method tho as it is only available from a Gmail e-mail address or Riseup.

- See: [A note about Riseup:] Riseup has potentially been compromised. Use it at your own risk.

Hopefully, these bridges should be enough to get you connected even in a hostile environment.

If not, consider What about when Tor and VPNs aren't possible?

Installing and Using Desktop Tor Browser

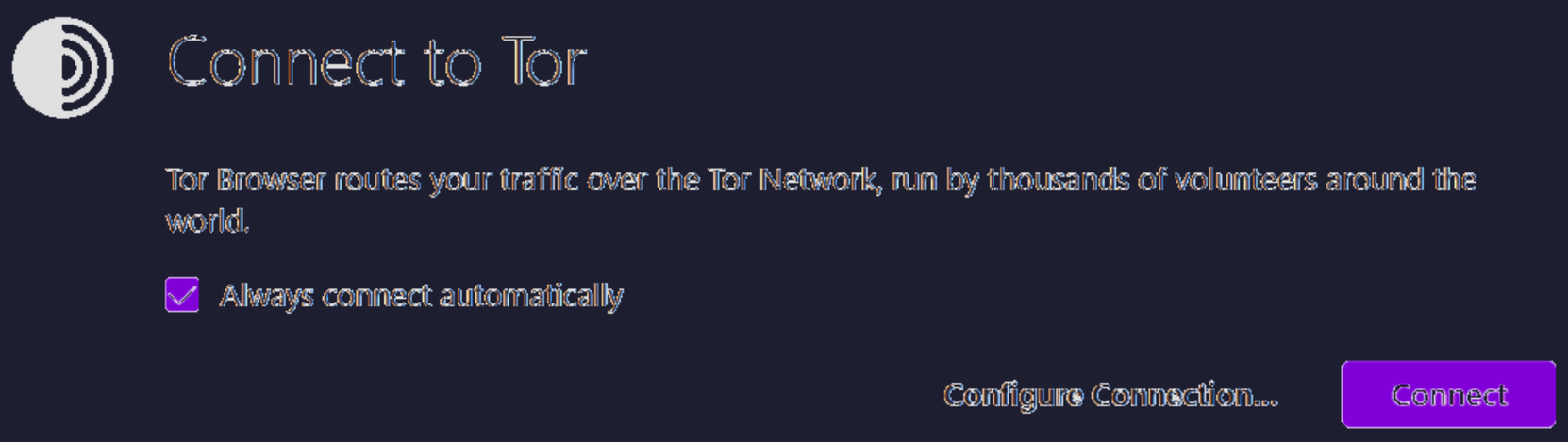
Installation

This is valid for Windows, Linux, and macOS.

- Download and install Tor Browser according to the instructions from https://www.torproject.org/download/ [Archive.org]
- Open Tor Browser

Usage and Precautions

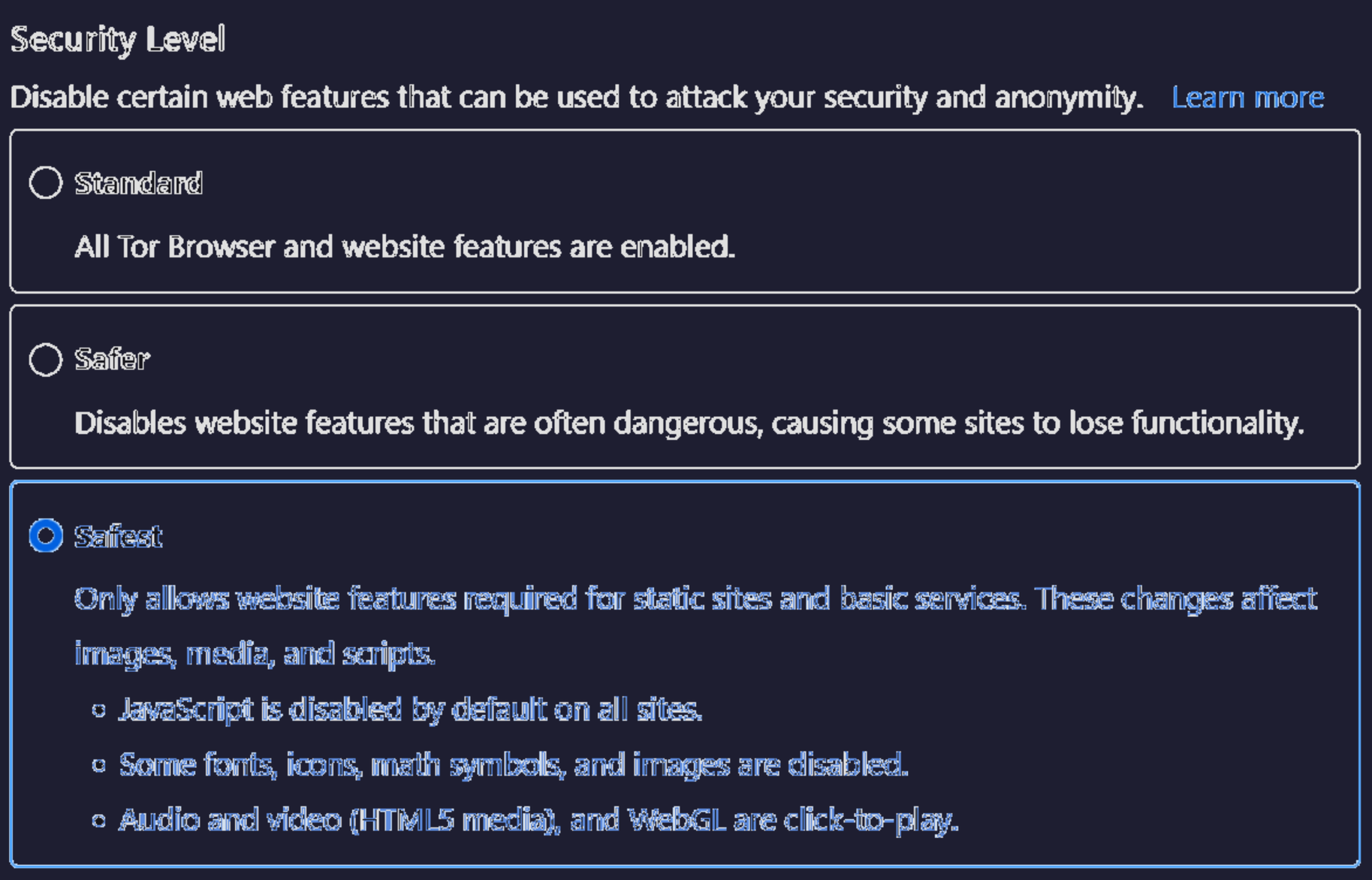
- After opening Tor Browser, you will see an option to **Connect**, a checkbox to **Always connect automatically** and a button to **Configure connection**. The Tor Network settings are there for you to possibly configure Bridges to connect to Tor if you are experiencing issues connecting to Tor due to Censorship or Blocking. As explained here: Using Tor bridges in hostile environments, this is now done automatically by the Tor Browser on Desktop.



- Personally, in the case of censorship or blocking, we would recommend using Meek-Azure bridges if needed. And Snowflake bridges as a second option.



At this point, still before connecting, you should click the little shield Icon (upper right, next to the Address bar) and select your Security level (see https://tb-manual.torproject.org/security-settings/ [Archive.org] for details). Basically, there are three.



- Standard (the default):
  - All features are enabled (including JavaScript)
- Safer:
  - JavaScript is disabled on non-HTTPS websites
  - Some fonts and symbols are disabled
  - Any media playback is "click to play" (disabled by default)
- Safest:
  - Javascript is disabled everywhere

- Some fonts and symbols are disabled
- Any media playback is "click to play" (disabled by default)

We would recommend the "Safest" level by default. The "Safer" level should be enabled if you think you need access to a website not working without JavaScript. The Safest mode will most likely break many websites that rely actively on JavaScript.

If you are extra paranoid, use the "Safest" level by default and consider downgrading to Safer is the website is unusable because of Javascript blocking.

Optional and not recommended by the Tor Project: If you are not using the "Safest" level, we will diverge from some but agree with others (for instance the Tails project and others) and will actually recommend some modifications of the default Tor Browser in the addition of two extensions:

- uBlock Origin (as it is the case on Tails) while leaving the extension on the default settings:
  - Head over to https://addons.mozilla.org/en-US/firefox/addon/ublock-origin/ within Tor Browser and install the extension.
- LibRedirect: This is very practical if you use the "Safest" mode as Invidious instances require no JavaScript.
  - Head over to https://libredirect.github.io/ within Tor Browser and install the extension.

Let's keep in mind that even 3 letters agencies recommend blocking ads for their internal users in order to improve security.

If you did not go for the above **personal and not officially recommended options**, the Safer level should still be used with some extra precautions while using some websites: see Additional browser precautions with JavaScript enabled.

Now, you are really done, and you can now surf the web anonymously from your desktop device.

Anonymous Cryptocurrency Payments

There are many services that you might want to use (VPS hosting, mail hosting, domain names...) but require payment of some kind.

As mentioned before in this guide multiple times, we strongly recommend the use of services accepting cash (that you could send anonymously through the postal services) or Monero which you can buy and use directly and safely.

- But what if the service you want does not accept Monero but does accept a more mainstream cryptocurrency such as Bitcoin (BTC) or Ethereum (ETH)?

Bitcoin and other "mainstream cryptocurrencies" are not anonymous at all (Remember Your Crypto Transactions) and you should never ever purchase, for example, Bitcoin from an exchange and then use these directly for purchasing services anonymously. This will not work, and the transaction can be traced easily.

- **Stay away from so-called "private" mixers, tumblers and coinjoiners.** You might think this is a good idea, but not only are they useless with cryptocurrencies such as BTC/ETH/LTC, they are also dangerous. They take custody of your coins. Use Monero to anonymize your crypto. Do not use a normal KYC-enabled exchange to buy/sell your Monero (such as Kraken), since this information on your purchases and withdrawals (for intended use) are retained in the exchange. Instead, use a P2P exchange that doesn't require KYC such as what can be found on https://kycnot.me/.
- **See Warning about special tumbling, mixing, coinjoining privacy wallets and services.**

Using Bitcoin anonymously option

Despite this, it is possible to safely anonymize Bitcoin through the use of non-custodial collaborative transactions and privacy-preserving spending tools. This is possible with a protocol called ZeroLink and an implementation called Whirlpool which as two clients that utilize it and provide the necessary spending tools, detailed below. So, you might be wondering how? Well, it is actually pretty simple:

1. Purchase Bitcoin at a non-KYC exchange (such as one found on https://kycnot.me/)
2. Create a wallet with Samourai Wallet (Android) or Sparrow Wallet (Desktop). Both of these use the Whirlpool protocol to gain the user forward-facing on-chain privacy on Bitcoin.
3. Deposit coins into the wallet and follow the relevant instructions (Samourai, Sparrow) to remove their historic links.
4. Funds should only be spent from the Postmix account, as that is the account with the coins that have gained anonymity through Whirlpool.
5. **You should run your own node when using Bitcoin and always use that for connecting from your wallet. You do not need to purchase separate hardware to do so, and it's simple to do so by using the Tor Network as well.**

Using Monero anonymously option

1. Purchase Monero at a non-KYC exchange (such as one found on https://kycnot.me/)
2. Create a Monero wallet on one of your anonymized VMs (for example, on the Whonix Workstation which includes a Monero GUI wallet natively or using the Monero GUI wallet from https://www.getmonero.org/downloads/ on other OSes)
3. Transfer your Monero from the wallet from which you bought it to the wallet on your VM. We cannot stress enough how import it is to have two separate wallets for this process, even for handling Monero.
4. On the same VM (for instance again the Whonix Workstation), create a Bitcoin Wallet (again this is provided natively within the Whonix Workstation)
5. From an anonymized browser (such as Tor Browser), use a non-KYC (Know Your Customer) service swapping service (see Crypto Swapping Services without Registration and KYC) and convert your Monero to BTC and transfer those to the BTC Wallet you have on your anonymized VM
6. You should now have an anonymized Bitcoin wallet that can be used for purchasing services that do not accept Monero.

You should never access this wallet from a non-anonymized environment. Always use well-thought OPSEC with your BTC transactions. Remember those can be traced back to you.

The origin of those BTC cannot be traced back to your real identity due to the use of Monero **unless Monero is broken** or if you consolidate outputs from spending at separate merchants. It is recommended to use privacy preserving wallets in the Bitcoin section. Please do read Monero Disclaimer.

Regarding Zcash: this section previously included use of Zcash but it has been removed in light of newer, more accurate information.

Warning about special tumbling, mixing, coinjoining privacy wallets and services

Centralized "private" tumblers, mixers and coinjoiners are not recommended since they do not provide anonymity in a way that truly unlinks an output from its history. Here are some references about this issue:

- Mixing detection on Bitcoin transactions using statistical patterns. Archive.org
- An Analysis Of Bitcoin Laundry Services Archive.org
- Mixing Strategies in Cryptocurrencies and An Alternative Implementation Archive.org

Mixing BTC in this way should prevent any chain analysis on future transactions. This will *not* however hide any past transactions or the fact you purchased BTC from a KYC exchange. Instead we recommend to use Bitcoin wallets that utilize Whirlpool or Monero (preferred).

When converting from BTC to Monero

Now, as part of any process above, if you want to convert BTC back to Monero, we recommend not using a swapping service but instead recommend using the new Monero Atomic Swap Tool: https://unstoppableswap.net/. This will prevent unnecessary fees and intermediates when using a commercial swapping service. The website is self-explanatory with detailed instructions for all OSes.

Recommended VPS Hosting Providers

We will only recommend providers that accept Monero as payment and here is my personal shortlist:

- **Njalla https://njal.la/ (my personal favorite but quite expensive, recommended by PrivacyGuides.org.**
- **1984.is (my second favorite, much less expensive) https://www.1984.is.**
- To be considered at your own risk (untested):
  - https://cryptoho.st/ (warning, this might be against their ToS as they require personal identification on registration)
  - https://www.privex.io/
  - https://cockbox.org/ (warning, this provider is rather "edgy" and could offend some people)

Also consider these lists:

- Tor Project: https://community.torproject.org/relay/community-resources/good-bad-isps/ [Archive.org]
- PrivacyGuides.org: https://privacyguides.org/providers/hosting/ [Archive.org]

Lastly, you could pick one (at your own risk) from the list here that does accept Monero: https://www.getmonero.org/community/merchants/#hosting [Archive.org]

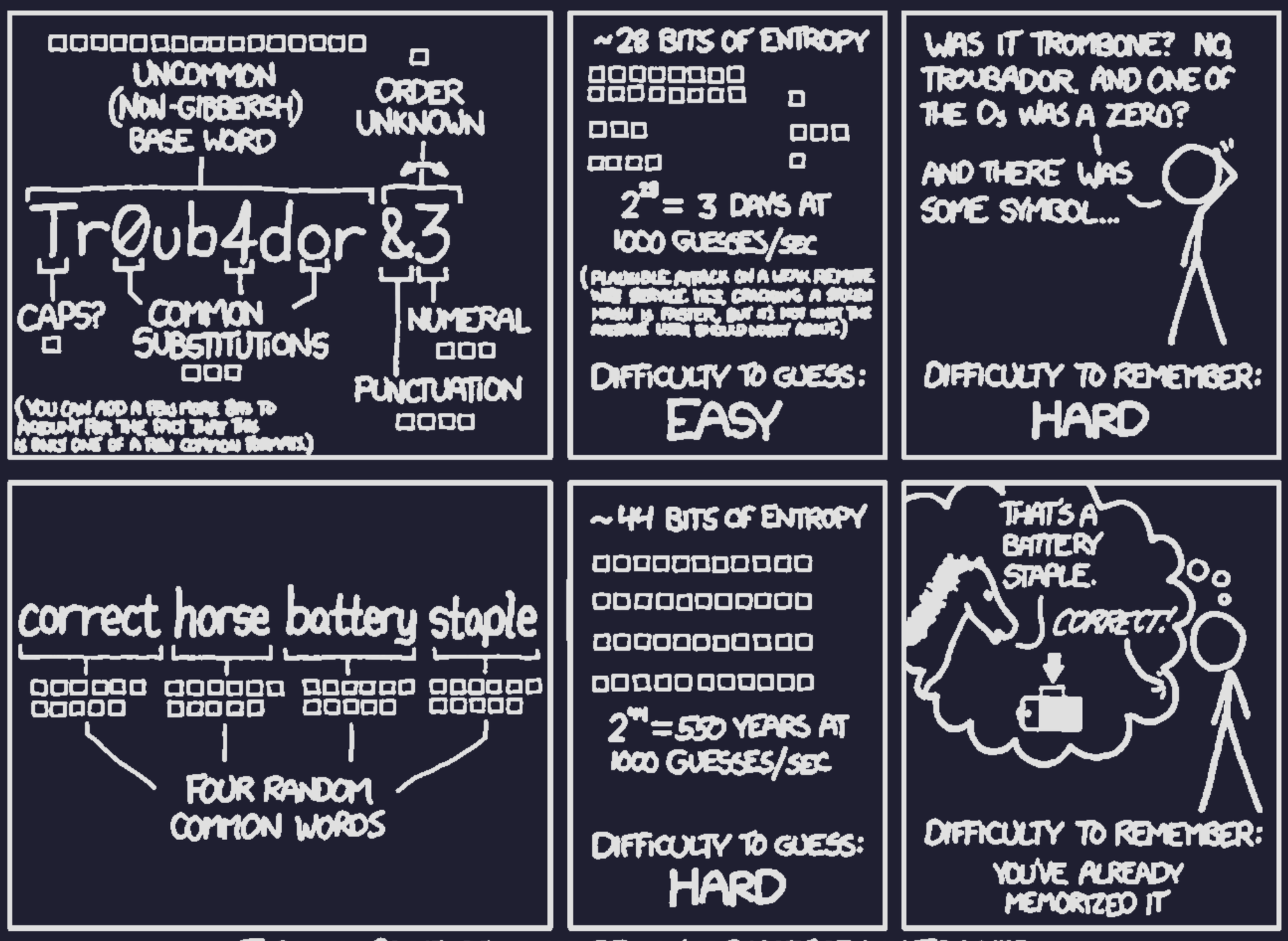
Please do read Monero Disclaimer.

If the service does not accept Monero but does accept BTC, consider the following appendix: Paying anonymously online with BTC.

Password and Passphrase Guidelines

My opinion (and the one of many'''') is that passphrases are generally better than passwords. So instead of thinking of better passwords, forget them altogether and use passphrases instead (when possible). Or just use a password manager with very long passwords (such as KeePassXC, the preferred password manager in this guide).

The well-known shown-below XKCD https://xkcd.com/936/ [Archive.org] is still valid despite some people disputing it (See https://www.explainxkcd.com/wiki/index.php/936:_Password_Strength [Archive.org]). Yes, it is quite old now and is a little bit outdated and might be misinterpreted. But generally, it is still valid and a good argument for using passphrases instead of passwords.



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

Here are some recommendations (based on Wikipedia):

- Long enough to be hard to guess (typically four words is a minimum, five or more is better).
- Not a famous quotation from literature, holy books, et cetera.
- Hard to guess by intuition---even by someone who knows the user well.
- Easy to remember and type accurately.
- For better security, any easily memorable encoding at the user's own level can be applied.
- Not reused between sites, applications, and other different sources.

- Do not use only "common words" (like "horse" or "correct")

Here is a nice website showing you some examples and guidelines: https://www.useapassphrase.com/

Watch this insightful video by Computerphile: https://www.youtube.com/watch?v=3NjQ9b3pgIg [Invidious]

Use a different one for each service/device if possible. Do not make it easy for an adversary to access all your information because you used the same passphrase everywhere.

You might ask how? Simple: use a password manager such as the recommended KeePassXC. Only remember the passphrase to unlock the database and then store everything else in the KeePassXC database. Within KeePassXC you can then create extremely long passwords (30+ random characters) for each different service.

Search Engines

Which search engine to pick in your VMs?

We will not go into too many details. Just pick one from PrivacyGuides.org (https://www.privacyguides.org/search-engines/ [Archive.org]).

Personally, my favorites are:

- https://duckduckgo.com/ (because you can easily use operators such as "!g" to google or "!b" to Bing)
- https://www.startpage.com/
- SearX (https://searx.me/) instances listed here: https://searx.space/

Note that some of those have a convenient ".onion" address:

- DuckDuckGo: http://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/

In the end, we were often not satisfied with the results of both those search engines and still ended up on Bing or Google.

Counteracting Forensic Linguistics

Note that this information is taken and adapted from a Dread Post available here:
http://dreadytofatroptsdj6io7l3xptbet6onoyno2yv7jicoxknyazubrad.onion/post/aad54fe83b33a8a45920/

No plagiarism is intended but some important adaptations and modifications have been made to improve the source post in various ways.

Introduction

Stylometry is our personal and unique writing style. No matter who you are, you have a unique finger printable, and traceable writing style. This has been understood for a while now, and a branch of forensics is built off of this principle: forensic linguistics. In this field, the particular name for forensic linguistics applied to internet crime is called "Writeprint". Writeprint primarily aims to determine author identification over the internet by comparing a suspect's text to a known collection of writer invariant (normally written) texts, and even without comparison texts, this forensic technique can yield personal information about an author such as gender, age, and personality.

What does an adversary look for when examining your writing?

1. Lexical features: analysis of word choice.
2. Syntactic features: analysis of writing style, sentence structure, punctuation, and hyphenation.
3. Structural features: analysis of structure and organization of writing.
4. Content-specific words: analysis of contextually significant writing such as acronyms.
5. Idiosyncratic features: analysis of grammatical errors, this is the most important factor to consider because it provides relatively high accuracy in author identification

Examples

You might think that this is not something that an adversary pays attention to? Think again! There have been multiple cases where adversaries such as law enforcement have used Writeprint techniques to help catch and sentence people. Here are some examples:

- The OxyMonster case (https://arstechnica.com/tech-policy/2018/06/dark-web-vendor-oxymonster-turns-out-to-be-a-frenchman-with-luscious-beard/ [Archive.org]):

- Public data revealed that Vallerius (a.k.a OxyMonster) has Instagram and Twitter accounts. Agents compared the writing style of "OxyMonster" on the Dream Market forum while in a senior Moderator role to the writing style of Vallerius on his public Instagram and Twitter accounts. Agents discovered many similarities in the use of words and punctuation to including the word "cheers;" double exclamation marks; frequent use of quotation marks; and intermittent French post.

Do not use the same writing style for your sensitive activities as for your normal activities. In particular, pay close attention to your use of common phrases, and punctuations. Also, as a side note: limit the amount of reference material that an adversary can use as comparison text, you do not want to find yourself in trouble because of your political Twitter post, or that Reddit post you made years ago, do you?

- Here is another example from the book American Kingpin, about how a DEA agent investigated the writing style of DPR (Dread Pirate Roberts a.k.a Ross Ulbricht, founder of the Silk Road Dark Market) from a unique perspective: For one, Ross Ulbricht used the word "epic" a lot, which showed that he was likely young. He also used emoji smiley faces in his writing, though he never used a hyphen as the nose, writing them as ":)" rather than the old-fashioned ":-)". Yet the one attribute about Ulbricht that stood out was that rather than writing "yes" or "yeah" on the site's forums, Ulbricht instead always typed "yea".

Pay attention to the little things that might add up. If you usually reply with "ok" to people, maybe try to reply with "okay" for your sensitive activities. You should NEVER use words or phrases from your sensitive activities (even if they are not in a public post) for normal purposes, and vice versa. Ross Ulbricht used "frosty" as the name for his Silk Road servers, and for his YouTube account, which helped convince law enforcement that Dread Pirate Roberts was in fact, Ross Ulbricht.

How to counteract the efforts of your adversary

1. Reduce the amount of comparison text for adversaries to compare you with. This goes with having a small online footprint for your normal activities.
2. Use a word processor (such as LibreWriter) to fix any grammatical/spelling errors that you regularly encounter.
3. Reduce or change the idioms that you use while conducting sensitive activities.
4. Understand how your identity affects your writing style: Is your alias younger? Older? More educated? Or less educated? If your identity is older, maybe speak in a more JRR Tolkien style of writing.
5. Pay attention to how your slang and spelling might identify you. If you are from the UK, you should say "maths", but if you are from the US you say "math". It does not matter how you say "maths", all that matters is that it can be used to profile you. This also applies to slang as many regions each have different and extremely particular slang. You do not ask someone from the USA for a "rubber" and expect them to give you an "eraser" as an example.
6. Pay attention to your use of emoticons and emojis. In the previous example, the DEA agent was able to make a correct assumption that Ulbricht was likely young because he did not use a hyphen when making a smiley emoticon.
7. Pay attention to how you structure your writing. Do you use two spaces after a period? Do you constantly use parenthesis in your writing? Do you use the oxford comma?
8. Consider what symbols you use in your writing. Do you use €, £ or $? Do you use "dd-mm-yyyy" or "mm-dd-yyyy" for dates? Do you use "08:00 pm" or "20:00" for time?

What different linguistic choices could say about you

Emoticons

1. Russians for example use ")" instead of ":-)" or ":)" to express a smiley face.
2. Scandinavians use "=)" instead of ":-)" or ":)" for a smiley face.
3. Younger people generally do not use a hyphen in their smiley faces and just use ":)".

Structural features

1. Two spaces after a period give off the impression that you are quite older because this is how typing was taught to people learning to type with typewriters.
2. In the US people write numbers out with commas between numbers to the left of the starting number and with periods between numbers to the right of the starting number. This is in contrast to how people write out numbers on the rest of the planet.

US: 1,000.00$

> Europe: 1.000,00€

Spelling slang and symbols

1. Obviously, people in different nations use different slang. This is even more pronounced when you use slang that is not as well known in other places such as someone from the UK mentioning a "headmaster" when in other nations it is referred to as a "principal".
2. Spelling is another important factor that is similar to slang, except it is harder to control. If you want to pretend that you are from the USA, but you actually live in Australia, it only takes one time of spelling "colour" as color to let people understand that something is up.
3. Some people also spell words in a particular way that is not regional for example you might spell "ax" as "axe" or vice versa.
4. Of course, the symbols you use on your keyboard can give a lot of information away, such as £'s or $'s.

Techniques to prevent writeprinting

Here are some techniques in order of use:

Spelling and grammar checking

This helps prevent some fingerprinting done using your spelling and grammar mistakes

Offline using a word processor

Use a word processor such as LibreWriter and use the spelling and grammar checks features to fix mistakes you might have typed.

Online using an online service

If you do nothave a word processor available or don't want to use one, you can also use an online spelling and grammar checker such as Grammarly (this requires an e-mail and an account creation).

Translation technique

Disclaimer: a study archived here: https://web.archive.org/web/20181125133942/https://www.cs.drexel.edu/~sa499/papers/adversarial_stylometry.pdf seems to indicate the translation technique is inefficient to prevent stylometry. This step might be useless.

After being done with spelling and grammar fixes. Use a website or software such as Google Translate (or for a more privacy-friendly version, https://simplytranslate.org/) to translate between several different languages before translating back to your original language. These translations back and forth will alter your messages and make fingerprinting more difficult.

Search and replace

Finally, and optionally, add some salt by purposefully adding some mistakes to your messages.

First decide upon a list of words that you frequently do not misspell, maybe the words "grammatical", "symbol", and "pronounced" (this list should include more words). **Do not use an AutoCorrect automatic replace option for this as it might correct when it does not make sense.** Instead, use Search and Replace and do this manually for each word. **Do not use "Replace All" either and review each change.** This is just the first step, for providing misinformation against linguistic fingerprinting.

Next, find a list of words that you commonly use in your writing. Let us say that we love to use contractions when wew rite, maybe we always use words such as: "can't", "don't", "shouldn't", "won't", or "let's". Well, maybe go into LibreWriter and use "Search and Replace" to replace all contractions with the full versions of the words ("can't" > "cannot", "don't" > "do not", "shouldn't" > "should not", "won't" > "will not", "let's" > "let us"). This can make a large difference in your writing and give a difference in how people and most importantly your adversaries perceive you. You can change most words to be different, as an example you can change "huge" to "large". Just make sure these words fit with your identity.

Now, consider changing your words choices to fit a geographic location. Maybe you live in the US, and you want to give the impression that your identity is from the UK. For example, you can make use of location-based spelling and lexicon. This is risky, and one mistake can give it away.

First off, you need to decide where you want to give the impression of your location. Here is an example to give off the impression that you are from the US, or the UK. First, you will need to understand a thing or two about where your identity is "from", do not pretend that you are from the UK, yet have no idea about it other than it exists.

After you have decided upon a good location that your identity is from, research the differences in language between the two languages (in this case between UK English and US English). Thanks to the internet, this is quite easy, and you can find Wikipedia pages conveniently highlighting the regional differences of a language between two nations. Pay attention to how certain words are spelled ("metre" > "meter") and what words are exchanged with each other ("boot" > "trunk"). Now that you have a list of words that can be exchanged with each other, and a list of spelling that are different, use the "Search and Replace" in your editor and change the words such as "colour" into "color", and "lorry" into "truck". **Again, do**

not use an AutoCorrect feature or "Replace All" as some changes might not make sense. Review each proposed change. As an example, if you were to use AutoCorrect or "Replace all" on the word "boot" to change into "trunk", this would make perfect sense in the context of cars. But it would not make any sense in the context of shoes.

Final advice

Understand that you have to constantly think of what you type and how you type while conducting sensitive activities.

Understand that altering your writing style for such purposes can ultimately change your baseline writing style, ironically making your writing traceable over longer periods.

Proofread yourself at least one time after you are done writing anything to verify you made no mistakes in your process. Trust (yourself) but verify anyway.

You might also consider the use of something like AnonyMouth https://web.archive.org/web/https://github.com/psal/anonymouth [Archive.org] which is a tool that you can use to anonymize your documents, developed by PSAL, Drexel University's Privacy, Security, and Automation Laboratory https://psal.cs.drexel.edu/index.php/Main_Page [Archive.org]. Such tools can prove invaluable.

Bonus links

- https://seirdy.one/posts/2022/07/09/stylometric-fingerprinting-redux/ [Archive.org]: Stylometric fingerprinting redux
- https://www.whonix.org/wiki/Surfing_Posting_Blogging#Stylometry [Archive.org]: Whonix documentation about stylometry.
- https://wikipedia.org/wiki/Forensic_linguistics [Wikiless] [Archive.org]: Gives a brief rundown of the basics of forensic linguistics, not too informative.
- https://wikipedia.org/wiki/Writeprint [Wikiless] [Archive.org]: Gives a brief and informative rundown of forensic linguistics applied to internet investigations.
- https://wikipedia.org/wiki/Stylometry [Wikiless] [Archive.org]: Gives a brief overview of Stylometry.
- https://wikipedia.org/wiki/Content_similarity_detection [Wikiless] [Archive.org]: We would recommend reading this, quite informative.
- https://wikipedia.org/wiki/Author_profiling [Wikiless] [Archive.org]: Read through this as well if you are interested in this topic.
- https://wikipedia.org/wiki/Native-language_identification [Wikiless] [Archive.org]: This is less important if you use a translator, but if you do not use a translator to communicate on forums that are not in your native language, consider giving this a quick read through.
- https://wikipedia.org/wiki/Computational_linguistics [Wikiless] [Archive.org]: Only read through this if this topic is interesting to you.
- https://regmedia.co.uk/2017/09/27/gal_vallerius.pdf [Archive.org]: Explains how authorities used forensic linguistics to help arrest OxyMonster (pages 13 -- 14).
- https://wikipedia.org/wiki/Ted_Kaczynski#After_publication [Wikiless] [Archive.org]: May have an IQ of 167, but he was caught primarily based on forensic linguistics.
- https://i.blackhat.com/USA-19/Wednesday/us-19-Wixey-Im-Unique-Just-Like-You-Human-Side-Channels-And-Their-Implications-For-Security-And-Privacy.pdf [Archive.org]: Explains how your writing style can be used to track you, we highly recommend reading through these slides, or watching the accompanying presentation on YouTube.
- https://media.defcon.org/DEF%20CON%2026/DEF%20CON%2026%20presentations/DEFCON-26-Matt-Wixey-Betrayed-by-the-Keyboard-Updated.pdf [Archive.org]: Explains how your writing style can be used to track you, we highly recommend reading through these slides, or watching the accompanying presentation on YouTube, this is quite similar to the last presentation.
- https://i.blackhat.com/us-18/Wed-August-8/us-18-Wixey-Every-ROSE-Has-Its-Thorn-The-Dark-Art-Of-Remote-Online-Social-Engineering.pdf [Archive.org]: This goes over how to potentially spot deception through the internet, and presents a checklist to see how trustworthy someone is. We would advise reading the slides or watching the presentation on YouTube.

Additional Browser Precautions with JavaScript Enabled

To avoid Browser and User Fingerprinting through JavaScript but while keeping JavaScript enabled, some additional safety measures should be observed at least on some websites:

These recommendations are similar to the ones at the beginning of the guide and especially valid for certain websites. Mostly, the recommendation is to use privacy-friendly front-end instances and alternative services for a variety of services:

- For YouTube links, use an Invidious instance (https://github.com/iv-org/invidious [Archive.org])

    - We recommend [https://yewtu.be]
- For Twitter links, use a Nitter instance (https://github.com/zedeus/nitter [Archive.org])
    - We recommend [https://nitter.net]
- For Wikipedia links, use a Wikiless instance (https://codeberg.org/orenom/wikiless [Archive.org])
- For Reddit, use a LibReddit instance (https://github.com/spikecodes/libreddit [Archive.org])
- For Maps, consider using https://www.openstreetmap.org
- For Translation, consider using SimplyTranslate at https://simplytranslate.org/
- For Search Engines use privacy-focused search engines such as:
    - StartPage: https://www.startpage.com/
    - DuckDuckGo: https://duckduckgo.com/
    - SearX (https://searx.me/) instances: list available here: https://searx.space/

(Optional) Consider the use of the https://libredirect.github.io/ [Archive.org] extension to automate the use of the above services.

Comparing Guide Versions

If you want to compare an older version of the PDF with a newer version, consider these online tools (note that we do not endorse those tools in relation to their privacy policies, but it should not matter since these PDFs are public):

- https://tools.pdf24.org/en/compare-pdf
- https://products.aspose.app/pdf/comparison
- https://draftable.com/compare

Crypto Swapping Without Registration or KYC

General Crypto Swapping

Skip to next section for BTC to Monero. Do not use swapping services for BTC to Monero.

Here is a small list of non-KYC crypto swapping services, remember they all have a cost and fees:

- https://sideshift.ai
- https://bisq.network/
- Kilo Swap (Onion Hidden Service): http://mlyusr6htlxsyc7t2f4z53wdxh3win7q3qpxctbam6jf3dmua7tnzuyd.onion/coinswap

Consider having a look at https://kycnot.me/ which is an open-source project listing non-KYC exchanges/swapping services (repository at https://codeberg.org/pluja/kycnot.me).

BTC to Monero only

Do not use any swapping service, use their Atomic Swap feature. See this Monero Atomic Swap Tool: https://unstoppableswap.net/.

This will prevent unnecessary fees and intermediates when using a commercial swapping service. The website is self-explanatory with detailed instructions for all OSes.

Installing a Zcash Wallet

Remember this should only be done on a secure environment such as VM behind the Whonix Gateway.

Debian 11 VM

- Load the Debian VM
- Open a browser

- Go to https://packages.debian.org/buster/amd64/libindicator3-7/download and download from a listed mirror.
- Go to https://packages.debian.org/buster/amd64/libappindicator3-1/download and download from a listed mirror.
- Go to the ZecWallet Lite Website to download the latest DEB package https://www.zecwallet.co/#download (change the download directory to /home/user for convenience)
- Open a Terminal window and run the following commands (with the updated downloaded version if needed):
  - `sudo dpkg -i ./libindicator3-7_0.5.0-4_amd64.deb`
  - `sudo dpkg -i ./libappindicator3-1_0.4.92-7_amd64.deb`
  - `sudo dpkg -i ./Zecwallet_Lite_1.7.5_amd64.deb`
- Click the upper left menu, find then launch ZecWallet Lite

Ubuntu 20.04/21.04/21.10 VM

- Load the Ubuntu VM
- Open a browser
- Go to the ZecWallet Lite Website to download the latest DEB package https://www.zecwallet.co/#download
- Open a Terminal window
- Go to your download directory and run the following command (with the updated downloaded version if needed), for example: `sudo apt install ./Zecwallet_Lite_1.7.5_amd64.deb`
- Click the upper left menu, find then launch ZecWallet Lite

Windows 10/11 VM

- Load the Windows VM
- Open a browser
- Go to https://www.zecwallet.co/#download
- Download and install the latest Windows installer
- Launch ZecWallet Lite

Whonix Workstation 16 VM

- Load the Whonix Workstation VM
- Open Tor Browser
- Go to https://packages.debian.org/buster/amd64/libindicator3-7/download and download from a listed mirror.
- Go to https://packages.debian.org/buster/amd64/libappindicator3-1/download and download from a listed mirror.
- Go to the ZecWallet Lite Website to download the latest DEB package https://www.zecwallet.co/#download (change the download directory to /home/user for convenience)
- Open a Terminal window and run the following commands (with the updated downloaded version if needed):
  - `sudo dpkg -i ./libindicator3-7_0.5.0-4_amd64.deb`
  - `sudo dpkg -i ./libappindicator3-1_0.4.92-7_amd64.deb`
  - `sudo dpkg -i ./Zecwallet_Lite_1.7.5_amd64.deb`
- Click the upper left menu and go to Development, then launch ZecWallet Lite

Pre-Sharing Information Checklist

Here is a checklist of things to verify before sharing information to anyone:

- Check the files for any metadata: see Metadata auditing
- Check the files for anything malicious: see Checking files for malware
- Check the files for any watermarking: see Watermarking
- Check any writing for possible forensics analysis: see Counteracting Forensic Linguistics
- Have a look at this part of the Whonix documentation: https://www.whonix.org/wiki/Surfing_Posting_Blogging#Anonymous_File_Sharing [Archive.org]
- Carefully assess the potential consequences and risks of communicating any sensitive information for you and others (legally, ethically, and morally). Remember ... Do not 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.

Monero Disclaimer

First, please read this small introduction video to Monero: https://www.youtube.com/watch?v=H33ggs7bh8M [Invidious]

The anonymity of Monero depends on its crypto algorithms. If you do use Monero from a KYC Exchange. You can be almost certain that you are safe today. But you might not be in the long-term future if Monero algorithms are ever broken (think Quantum Computing). Do keep in mind that KYC regulations might force operators (such as Crypto Exchanges) to keep your financial records for up to 10 years and that you, therefore, need Monero algorithms to not be broken for the next 10 years as well.

You may want to watch this insightful video for more details: https://www.youtube.com/watch?v=j02QoI4ZInU [Invidious]

Also please consider reading: Privacy Limitations in Anonymity Networks with Monero [Archive.org]

Use these at your own risk. Sending cash payments to providers through the postal service is always a better solution if/when possible.

Threat Modeling Resources

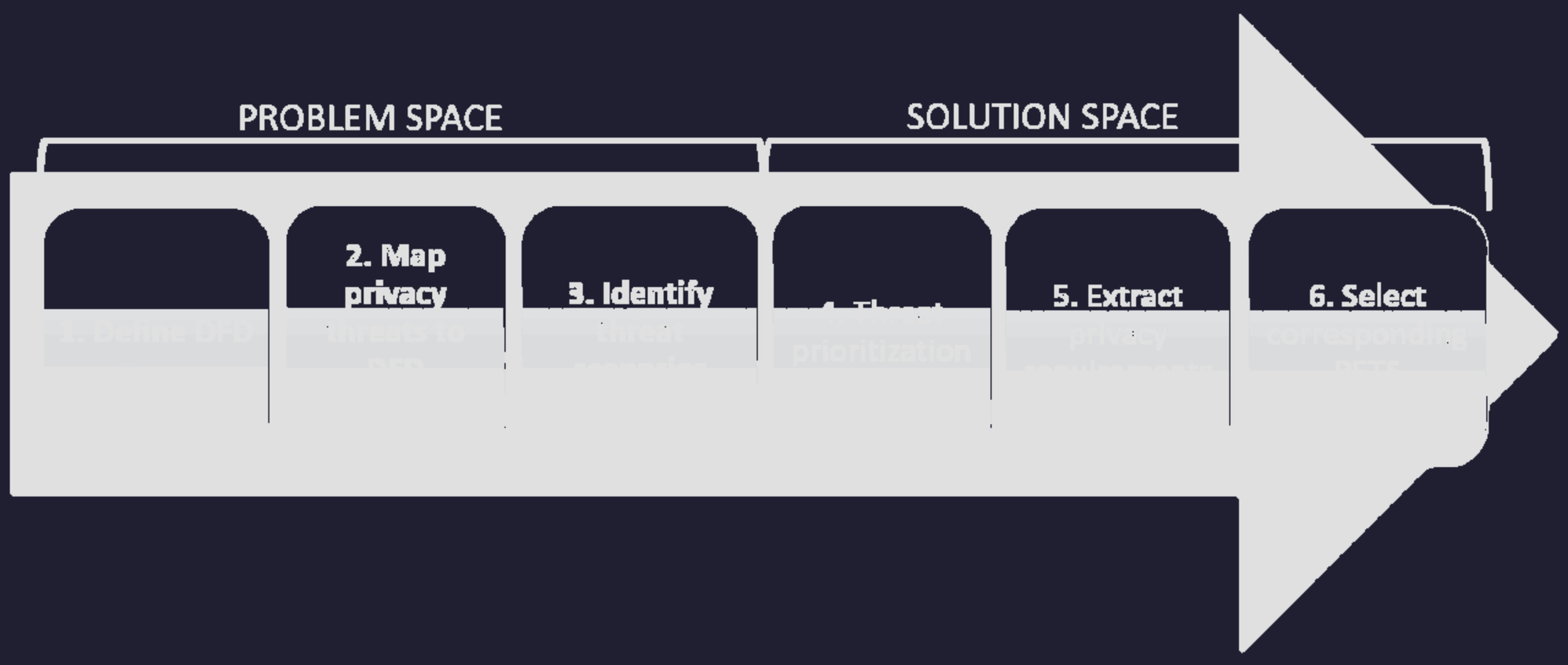
Here are various threat modeling resources if you want to go deeper in threat modeling.

We recommend the LINDDUN https://www.linddun.org threat modeling method [Archive.org]:

- Researchers created an online tool to help make your threat model at https://www.linddun.org/go [Archive.org].
- It is synergistic with STRIDE below.
- It is focused on privacy but is clearly perfectly suitable for anonymity.
- It is accessible to all skill levels including beginners (providing many tutorials) but also suitable for highly skilled readers.
- It is used in the making of the Threat Modeling Manifesto: https://www.threatmodelingmanifesto.org/ [Archive.org]

LINDDUN threat modeling tutorials and resources:

- **We recommend the following quick tutorial video from "The Hated One" YouTube channel with the approval and review from LINDDUN designers:** https://www.youtube.com/watch?v=6AXkJ3dot2s [Invidious] to get started.
- More resources for deeper understanding and usage:
  - You can read more here: A Lightweight Approach to Privacy Threat Modeling
  - Here are two videos from Dr. K. Wuyts (imec-DistriNet, KU Leuven) explaining the process:
    - Privacy & prejudice: on privacy threat modeling misconceptions [Invidious]
    - Privacy Threat Model Using LINDDUN [Invidious]



(Illustration from LINDDUN2015)

Here are alternative resources and methodologies if LINDDUN doesn't suit you:

- Online Operations Security: https://github.com/devbret/online-OPSEC
- Microsoft's STRIDE: https://en.wikipedia.org/wiki/STRIDE_%28security%29 [Wikiless] [Archive.org]
- PASTA: https://versprite.com/tag/pasta-threat-modeling/ [Archive.org]
- Threat Modeling: 12 Available Methods: https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods [Archive.org]
- Threat Modelling: https://www.geeksforgeeks.org/threat-modelling/ [Archive.org]

Evil-Maid and Tampering Notes

Your context needs to be taken into account.

Preventing an evil-maid attack attack or tampering might lead to bad consequences. Your adversary might then resort to other means to obtain the key.

On the other hand, allowing the attack but detecting it will not let your adversary know that you are aware of the tampering. You can then take steps safely to not reveal information and possibly leave.

See the Some last OPSEC thoughts section for some tips.

Types of CPU Attacks

Select security issues plague many Intel CPUs, such as transient execution attacks (formerly called speculative execution side channel methods). Here you can check your CPU against affected micro-processors with known bugs https://www.intel.com/content/www/us/en/developer/topic-technology/software-security-guidance/processors-affected-consolidated-product-cpu-model.html [Archive.org].

The Advanced Programmable Interrupt Controller (APIC) is an integrated CPU component responsible for accepting, prioritizing, and dispatching interrupts to logical processors (LPs). The APIC can operate in xAPIC mode, also known as legacy mode, in which APIC configuration registers are exposed through a memory-mapped I/O (MMIO) page.

Enter AEPIC (stylized ÆPIC), the first architectural CPU bug that leaks stale data from the microarchitecture without using a side channel. It architecturally leaks stale data incorrectly returned by reading undefined APIC-register ranges. This novel method was revealed in the paper *ÆPIC Leak: Architecturally Leaking Uninitialized Data from the Microarchitecture* which you can read here: Borrello2022AEPIC [Archive.org]

Model-specific registers (MSRs) and their configuration bits can also be detected automatically on Intel and AMD CPUs: Kogler2022 [Archive.org]. This allows an attacker (with heavy knowledge of CPU functionality) to view information about the MSRs, which are essentially special CPU registers allowing interaction with low-level CPU features and advanced configuration of the CPU's behavior. Modern x86 CPUs have hundreds of these, which are usually documented very little and in increasingly less verbosity over the past few years.

Orbot on Android Warning

While this is often misunderstood, Orbot on Android does not make your Tor-enabled apps go through Tor if you add them to the list. Orbot is acting as a device-wide VPN (also known as a "transparent proxy"). The list of apps using Orbot is a whitelist. This list will not make some apps magically use Tor and unchecked ones use the clear-net. This only ensures the device-wide VPN is using Tor to route traffic. This means that Orbot can only control what app can access the VPN it creates. Other apps will lose connectivity.

What is important to know is that, if you launch an app (or Android does it automatically) while Orbot is not running, the app will just use the normal network, without involving Orbot (with the exception of some apps supporting a proxy Orbot).

Additionally, you should not be surprised by Tor Browser not working when using Orbot in VPN mode, as the Tor design does not allow "Tor over Tor" (you cannot re-enter the Tor network from a Tor exit node).

This is explained rather well by Alexander Færøy, who is a core developer at the Tor Project, in their TorifyHOWTO: Tor over Tor.

"When using a transparent proxy, it is possible to start a Tor session from the client as well as from the transparent proxy (read the warning!), creating a "Tor over Tor" scenario. Doing so produces undefined and potentially unsafe behavior. In theory, however, you can get six hops instead of three, but it is not guaranteed that you'll get three different hops - you could end up with the same hops, maybe in reverse or mixed order. It is not clear if this is safe. It has never been discussed. You can choose an entry/exit point, but you get the best security that Tor can provide when you leave the route selection to Tor; overriding the entry / exit nodes can mess up your anonymity in ways we don't understand. Therefore Tor over Tor usage is highly discouraged."

And from a post on the Tor Stack Exchange:

"The danger (beyond the performance hit) which keeps me from running Tor over Tor has to do with timing and congestion measurements. Adversaries watching your traffic at the exit(s) of your circuits have a better chance of linking your Whonix activity with your [Tor Browser Bundle] activity when those shared circuits slow down or drop packets at the same time. This can happen without Tor over Tor when your instances use a common upstream link. The linkage will be made tighter and more explicit if you run the Whonix Tor traffic through your TBB SOCKS5 Tor circuits. This tighter linkage raises the danger of successful correlation."

Session Messenger Caution

- They use LokiNet, which requires Oxen to run nodes to route Session traffic, and it costs 15,000 $OXEN or 3,750 $OXEN for a shared node, which is about ~$1,800 US dollars or ~$500 US dollars, respectively. This high cost creates a barrier to entry for running nodes, potentially centralizing the network and limiting its accessibility.
    - The price of running nodes essentially puts their network behind a paywall if you want to run a node, even just to contribute bandwidth to the network like you might with Tor. But there is a stakeless fork of Lokinet.
    - Session's developers claim this to be an attempt to prevent sybil attacks, but many have argued that this only encourages such attacks; by doing so, guaranteeing only governments and other well-funded organizations (the people these networks normally try to protect against) will ever have the financial resources to run nodes. (Eh, it's all pretty debatable. But $OXEN is privacy-focused.)
- They dropped critical security features of their protocol (perfect forward secrecy (PFS) and deniability) in favor of long-term message keys and self-deleting cryptographic signatures, which provide much weaker security guarantees.
    - This *might* not be as bad, if the nodes are free to run, but they're not.
- April 2021: Session was audited with satisfactory results, but that audit does not mention these changes. We also currently lack sufficient information on LokiNet (the onion routing network used by Session) to endorse it. Session is still recommended by some, for example Techlore.
- Their funding is completely opaque.

In short, our opinion is that you may use Session Messenger on iOS due to the absence of a better alternative (such as Briar). But if Briar or another app (maybe Cwtch in the future) becomes available, we will recommend going away from Session messenger as soon as possible. It is a last resort.

OPSEC Failure Case Studies

The following cases are drawn from public court records, journalism, and post-mortems. They are included not to gloat over people who were caught, but because each illustrates a specific, repeatable failure mode that is directly relevant to the guidance elsewhere in this guide. In every case, the technical anonymity tools available were sufficient - the failures were human.

Hector Monsegur (Sabu) - LulzSec, 2011

Context: Monsegur was a core member and de facto leader of LulzSec, a high-profile hacking group responsible for breaches of Sony, the FBI, and others. He was arrested in June 2011 and subsequently became an FBI informant when they threatened to take his kids away.

The failure: On a single occasion, Monsegur logged into an IRC channel associated with Anonymous/LulzSec without routing his connection through Tor. His real IP address - assigned to his home internet connection in New York - was logged by the IRC server. The FBI subpoenaed those logs.

What it demonstrates: A single lapse in a single session is sufficient to de-anonymize an otherwise disciplined operator. Tor is only effective if it is used *every time* without exception. There is no "just this once" at the operational level. The value of anonymity is destroyed the moment it is broken, even once.

What he should have done: Used Tor or a trusted VPN for every IRC connection without exception, ideally from hardware and a network not associated with his identity. A single dedicated device used exclusively for sensitive activities would have prevented cross-contamination.

Ross Ulbricht - Silk Road, 2013

Context: Ulbricht operated Silk Road, the pseudonymous darknet market, under the name "Dread Pirate Roberts" for approximately two years before his arrest in October 2013.

The failure: Multiple compounding errors over time, not a single incident. Before Silk Road existed, Ulbricht posted to a Bitcoin forum under the username "altoid" advertising the site - and separately used the same "altoid" username to post a job listing that included his personal Gmail address. Investigators matched the username across posts. Additionally, early posts on Stack Overflow linked to his real identity. His laptop, when seized at arrest, was open and unlocked - meaning his encrypted drives were fully accessible at the moment of capture.

What it demonstrates: Username reuse across contexts is one of the most reliable de-anonymization vectors available to investigators. A pseudonym used for sensitive activity must never appear in any context connected to your real identity - not a job post, not a forum question, not a throwaway comment years earlier. Compartmentalization must be total and must precede the activity, not follow it.

What he should have done: Used entirely separate identities, devices, and communication channels for Silk Road administration and personal activity, with no shared usernames, email addresses, or writing contexts. Pre-arrest, encrypted his working drive with a passphrase so that seizure of an open laptop would not yield plaintext access.

"Defcon" (Blake Benthall) - Silk Road 2.0, 2014

Context: Silk Road 2.0 launched shortly after the original Silk Road was seized. Its administrator, operating as "Defcon," was arrested in November 2014.

The failure: According to the criminal complaint, an FBI undercover agent had achieved a position inside the site's staff. More technically, the complaint describes the use of login timing correlation: investigators observed that "Defcon" logged into the site at times that correlated with other identifiable online activity. Additionally, the server infrastructure was identified through misconfigured hidden service configurations that leaked real IP addresses - a recurring operational failure in darknet markets.

What it demonstrates: Two distinct lessons. First, human infiltration of trusted circles is often more effective than technical attacks - no cryptography protects against a trusted insider. Second, server-side operational security (correctly configuring Tor hidden services so they do not leak their real IP under any condition) is as important as client-side anonymity. A perfectly anonymous administrator is irrelevant if the server itself is identifiable.

What he should have done: Audited all server configurations for IP leakage before launch and regularly thereafter. Treated all staff as potential informants at the operational level, compartmentalizing information accordingly.

Jeremy Hammond - AntiSec, 2012

Context: Hammond was a member of AntiSec, an offshoot of Anonymous, responsible for the Stratfor breach. He was arrested in March 2012.

The failure: Monsegur (Sabu), by this point an FBI informant, directed Hammond toward targets chosen by the FBI while gathering evidence against him. Hammond used strong technical practices but was socially engineered through a trusted relationship. He also reused a password pattern that investigators were able to identify across accounts.

What it demonstrates: A complementary lesson to the Monsegur case: even technically disciplined operators are vulnerable to compromise through trusted human relationships. Password reuse or patterned passwords across identities provides a correlation vector even when no single credential is directly compromised.

What he should have done: Used unique, randomly generated credentials for every identity and service with no shared patterns. The more important lesson - that operational trust in individuals cannot be verified cryptographically - has no clean technical solution, but compartmentalizing what each collaborator knows limits the damage any single compromise can cause.

Peter Stokes - Scattered Spider, 2026

Context: Stokes was accused, alongside other members of Scattered Spider, of over 100 corporate intrusions and more than $100 million in ransom payments (including the originally failed $8 million ransom). He used ngrok for tunneling and a commercial VPN service (Tzulo) to obscure network-layer traffic while conducting operations (See: Traffic Anonymization), on the assumption that IP obfuscation was sufficient to prevent correlation (See: Correlation vs. Attribution). Microsoft had already flagged him to the FBI once before, in an October 2024 criminal referral citing internal telemetry that researchers believed linked him to other members of the group.

The failure: Stokes ran his operations from a single Windows installation, signed into a Microsoft Account, used across all his activity: legitimate use, ngrok configuration, and communications tied to the group. Every Windows install signed into an MSA is assigned a persistent Global Device Identifier (GDID), a server-side identifier that does not change with IP address, VPN provider, or exit node, and is not affected by disabling standard telemetry settings. Investigators obtained IP records from ngrok and the VPN provider, then obtained Microsoft records showing the same GDID accessing ngrok at the exact time the ngrok account was created, and showing that GDID recurring at matching times across accounts already known to be his, across four countries, over roughly eight months. The VPN changed his IP on every connection. It did nothing to change the GDID.

What it demonstrates: Network-layer anonymization (VPN, tunneling) and identity-layer persistence (a single OS install tied to a personal Microsoft Account) are separate problems, and solving one does nothing for the other. A device-level identifier that survives IP rotation defeats the entire premise of "anonymous because the IP changed." It also shows that this class of identifier can pre-date public awareness of it by years: the GDID is described in exactly one line of Microsoft's public documentation, and had no meaningful public technical writeup until this case surfaced it. Assuming a mitigation is unnecessary because it isn't widely known is not a safe assumption. It also shows that using the same physical machine and OS install across both mundane, identified activity and sensitive activity is sufficient to link the two, independent of every precaution taken at the network layer.

What he should have done: Never run sensitive operations from a Windows install signed into a personal or otherwise identifiable Microsoft Account. Use a dedicated, minimal, non-MSA-linked environment, ideally Linux-based (Whonix, Tails, Qubes-Whonix) rather than Windows, for anything requiring real separation between personas. Never mix identified daily-driver use and sensitive operational use on the same install, account, or physical device, regardless of network-layer precautions. Treat any identifier that persists independent of IP, cookies, or browser state (device IDs, hardware serials, account-tied telemetry) as the primary threat to compartmentalization, since VPNs and tunnels only address the network layer and do nothing to sever an identity-layer link that already exists. See: Windows Global Device Identifier.

Common threads

Reading across these cases, several patterns repeat:

- **Single-session lapses break long-term anonymity.** Consistency is not optional.
- **Cross-context identity linkage is the most common investigative vector.** Usernames, writing style, email addresses, and posting history are all searchable and correlated.
- **Server-side and client-side security are both required.** Strong client anonymity does not compensate for a leaking server.
- **Human relationships are the most reliable attack surface.** Infiltration and informants feature in the majority of significant darknet takedowns.
- **Physical capture with an unlocked device undoes everything.** Disk encryption only helps if the device is locked at the moment of seizure.

Post-Quantum Cryptography

Note: This section deals with a threat that is not immediate for most users. If your threat model involves a nation-state adversary or communications whose sensitivity extends years into the future, read carefully. If you are an average user, you can skim this section for now but should revisit it as the technology matures.

Most of the encryption protecting your communications today - including the key exchanges inside Signal, HTTPS connections, VPNs, and PGP - relies on mathematical problems that are computationally infeasible for any classical computer to solve. A sufficiently powerful quantum computer running Shor's algorithm would break these problems efficiently. No such computer exists yet. The largest current machines are still far from the scale needed. But that is not a reason to ignore the problem.

The threat is called **"harvest now, decrypt later"** (HNDL). It works like this: an adversary - a government intelligence agency is the realistic candidate here - records and stores your encrypted traffic today, at scale. They cannot read it now. But if a capable quantum computer is built in the next 10-20 years, they decrypt that archived traffic retroactively. For most people reading this, that is not a pressing concern. For a journalist protecting a source whose identity would still be dangerous to expose in 2035, or an activist living under a government that keeps very long institutional memories, it is worth taking seriously.

The good news is that the cryptographic community has been aware of this problem for over a decade, and the tooling is arriving. In 2024, NIST finalized the first post-quantum cryptographic standards:

- **ML-KEM** (Module-Lattice-Based Key-Encapsulation Mechanism, formerly known as Kyber) - replaces the classical key exchange step in protocols such as TLS and Signal's X3DH. It is based on the hardness of the Module Learning With Errors (MLWE) problem, which is believed to resist both classical and quantum attacks.
- **ML-DSA** (Module-Lattice-Based Digital Signature Algorithm, formerly Dilithium) - a post-quantum replacement for RSA and ECDSA signatures used to authenticate identities and sign software.
- **SPHINCS+** (now standardized as SLH-DSA) - a hash-based signature scheme. Slower and larger than ML-DSA, but it relies only on the security of hash functions rather than lattice assumptions, making it a conservative fallback if lattice-based cryptography is ever weakened.

NIST explicitly recommends deploying these in **hybrid mode** during the transition period - meaning alongside classical algorithms rather than replacing them outright. This way, an attacker would need to break both simultaneously.

Signal's PQXDH

In September 2023, Signal deployed **PQXDH** (Post-Quantum Extended Diffie-Hellman), upgrading its X3DH key agreement protocol to combine ML-KEM-1024 with the classical X25519 Diffie-Hellman exchange. The result is hybrid: security holds as long as either component remains unbroken.

For Signal users, **this was automatic and transparent**. No configuration is required. New conversations started after the rollout use PQXDH by default. This directly addresses the HNDL threat for forward secrecy - an adversary who recorded your Signal traffic cannot use a future quantum computer to derive your session keys.

Note that this only covers the key exchange layer. The authentication layer (identity keys) is not yet post-quantum hardened in Signal, though this is an active area of development.

What you should do now

For most threat models, the practical steps are straightforward:

- **Use Signal.** PQXDH is already deployed, no action required.
- **Keep your browser updated.** Chrome and Firefox have had hybrid post-quantum key exchange (X25519 + ML-KEM) in TLS enabled by default since 2024. This protects your HTTPS connections against HNDL at the transport layer.
- **Do not rely on PGP/GPG for long-term confidentiality of highly sensitive material.** PGP key exchanges (RSA, ECDH) are not post-quantum hardened. Messages encrypted to a PGP key today could be decrypted retroactively by a quantum-capable adversary who has stored them. If you must use PGP, treat it as protection against present-day adversaries only.
- **Check your VPN provider.** Most commercial VPNs have not yet deployed post-quantum key exchange. Some (Mullvad, ProtonVPN) have added it. If HNDL is in your threat model, check your provider's documentation or switch to one that supports it.

For those with genuinely high-risk profiles: communications whose exposure would still be dangerous in ten or more years deserve attention now. Switching to Signal and establishing fresh sessions (rather than relying on long-running session state from before the PQXDH rollout) is the most practical near-term step.

A note on Monero

Monero's cryptographic primitives - specifically its use of Ed25519 and Curve25519 - are vulnerable to Shor's algorithm on a sufficiently powerful quantum computer. The Monero Research Lab has studied this problem and no post-quantum upgrade has been deployed. The community regards this as a medium-term concern, not an immediate one, given the current state of quantum hardware. For operational anonymity today, Monero remains appropriate. Do not assume long-term financial privacy against a quantum-capable adversary.

Stylometric Analysis and Writing Style

Note: Stylometric de-anonymization is a real but narrow threat. It is relevant to people who publish substantial amounts of text under a pseudonym over time, or who are suspected leakers being compared against a known corpus of their writing. It is not a realistic threat for most users of this guide. Read this section if you write publicly under a pseudonym, communicate repeatedly with the same adversary, or are a potential whistleblower whose writing may be compared against internal documents.

Stylometry is the statistical analysis of writing style for authorship attribution. The core insight is that people write in consistent, measurable ways that persist across topics and contexts - and that these patterns are difficult to suppress consciously. An author who habitually uses the Oxford comma, prefers "however" to "but," writes sentences averaging 22 words, and rarely uses exclamation marks will tend to do so whether writing a

forum post, an email, or a leaked document. Even one time that you might forget to anonymize your writing could be your downfall. In 2026 Vitalik Buterin, Ethereum co-founder, is attempting to have AI de-anonymize himself *intentionally* to show this is highly probable in the near future.

What features are measured

Modern stylometric systems analyse dozens to hundreds of features simultaneously. The most discriminating are **function words** - articles, prepositions, conjunctions - which are used largely unconsciously and are highly consistent per author. Content words (nouns, verbs, topic-specific vocabulary) are poor stylometric features because they vary with subject matter; function words do not. Other features include: sentence length distribution; punctuation habits (comma frequency, semicolon use, dash preference); paragraph length; vocabulary richness (type-token ratio); character-level n-grams; and syntactic patterns such as passive voice frequency.

In controlled academic evaluations, state-of-the-art systems achieve attribution accuracy above 80% across corpora of 50 or more candidate authors when each author has contributed several thousand words. Accuracy degrades significantly with shorter texts (under 500 words), larger candidate sets, or when the candidate corpus and the anonymous text are from different genres or contexts.

Deployed tools and real cases

JGAAP (Java Graphical Authorship Attribution Program), developed at Duquesne University, is the most widely used open academic tool and has been applied in legal proceedings. **Burner** is a more recent system designed specifically for adversarial de-anonymization of online pseudonyms.

The most documented real-world case is the 2013 identification of J.K. Rowling as the author of *The Cuckoo's Calling*, published under the pseudonym Robert Galbraith. Stylometric analysis by Peter Millican and Patrick Juola comparing the novel against Rowling's known work and a set of candidate authors produced a strong match before the identification was confirmed through other means. The corpus in this case was large - full novels - which is the condition under which stylometry works best.

In national security contexts, stylometric analysis has been used or attempted in leak investigations to compare anonymous documents against the known writing of suspected sources, though specific cases are rarely publicly confirmed.

What works and what does not

Naive countermeasures fail. Synonym substitution - replacing words with alternatives of similar meaning - does not affect function word patterns, sentence structure, or punctuation habits, which are the most discriminating features. Simply trying to "write differently" without a specific method is ineffective because the unconscious habits that stylometry measures are, by definition, ones the author does not notice.

What has some effectiveness: writing in a register genuinely unlike your natural style (e.g., formal legal prose when you normally write casually) does degrade attribution accuracy, because register shift affects multiple feature classes simultaneously. Keeping texts short - under 300-400 words - meaningfully reduces attribution confidence. Collaborative writing, where multiple authors contribute to a single document, degrades single-author attribution significantly.

AI rewriting as a countermeasure is an active area of research with mixed results. Large language models do alter function word distributions and sentence structure when rewriting text, and early studies suggest this degrades stylometric attribution accuracy to some degree. However, LLM rewriting does not reliably remove all stylistic signal - some author-specific patterns survive paraphrasing - and introduces a new signal: the statistical fingerprint of the specific model and prompt used. Whether this trade is favourable depends on the adversary's capabilities. LLM rewriting is probably useful as one layer in a defence-in-depth approach for high-risk writers, but should not be relied upon as a complete solution.

Honest threat model

Stylometry requires a reasonably large text sample from the anonymous author (ideally 1,000+ words), a candidate set of known authors to compare against, and a known writing corpus for those candidates. This limits realistic deployment to: leak investigations where investigators have a short list of suspects with known writing samples; de-anonymization of long-running pseudonymous authors with substantial published output; and academic or forensic authorship disputes.

It is not a practical threat for: one-off anonymous communications; users whose adversary does not have a comparison corpus of their writing; or short messages where the text sample is insufficient for reliable analysis. For most people reading this guide, the other threats documented here - metadata, network-layer identification, device fingerprinting - are far more likely vectors than stylometry. Address those first.

Acknowledgments

- **Very Special Thanks to Edward Snowden and who inspired us to write this guide (buy and read his book please https://en.wikipedia.org/wiki/Permanent_Record_(autobiography) [Wikiless] [Archive.org])**
- **Huge thanks to the people who donated to this project anonymously**
- **Special Thanks to LiJu09 for helping with the Light theme of the website (https://github.com/LiJu09)**

- **Special Thanks to Simplelogin.io people for providing a free lifetime premium access to their service**
- Thanks to GitHub for hosting this project and the many people who starred it
- Thanks to Njal.la for providing a domain name and VPS hosting anonymously
- Thanks to 1984.is for providing VPS hosting anonymously
- Thanks to all the people who contributed and shared this guide with others
- Thanks to the people at the Internet Archive and Archive.today projects
- Thanks to the people at the Monero project
- Thanks to the people at the Zcash project
- Thanks to the people at the Wikipedia project
- Thanks to the people at the Tails project
- Thanks to the people at the HiddenVM 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 and OONI Projects
- Thanks to the people at the Briar project
- Thanks to the people at the OnionShare project
- Thanks to the people at the Element/Matrix project
- Thanks to the people at the Jami project
- Thanks to the people at the KeePass and KeePassXC projects
- Thanks to the people at the Fawkes project
- Thanks to the people at the VirtualBox project
- Thanks to the people at the ExifCleaner, Mat2, and ExifTool projects
- Thanks to the people at the Go Incognito Project from Techlore
- Thanks to Didier Stevens for his pdf-tools
- Thanks to the people at the EFF
- Thanks to the people at the SANS
- Thanks to the people at the OWASP Project
- Thanks to the people at the Privacyguides.org project
- Thanks to the people at BlackHat, DEF CON, and CCC
- Thanks to the people at Bellingcat and other OSINT/Forensics researchers **(and sorry for making their life more difficult with this guide)**
- Thanks to the makers of the Social Dilemma documentary **(go watch it if you did not yet)**
- Thanks to Michael Bazzell and his great OSINT books which we recommend you **buy** at https://inteltechniques.com
- Thanks to Randall Munroe at XKCD for his great and insightful 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

References