// CYBERSECURITY

Protecting yourself

Because of the nature of this work, **we are targeted constantly** — phishing messages built to look like genuine logins, and brute-force attempts against every account and server we run. It is relentless, and it reaches anyone connected to this kind of documentation.

If you share, research, report or repost this material, some of that attention reaches you too. What follows is what works, **in the order we would do it**.

ON THIS PAGE +

WHAT THAT LOOKS LIKE, MEASURED — 14 TO 21 SEPTEMBER 2026

21.2 million requests served in seven days

22,872
asked for something that does not exist

7,111
matched known attack-tool signatures, about 1,000 a day

291
separate machines doing it

CONTEXT

This is not hypothetical

Four things that are established fact, each investigated and published by people who had to prove it.

— **Palestinian human rights workers had military spyware on their phones.** In 2021 Front Line Defenders found NSO Group's Pegasus on the devices of six Palestinian human rights defenders, people documenting exactly the kind of thing this archive documents. Citizen Lab and Amnesty's Security Lab independently confirmed the findings. Pegasus takes everything: messages, microphone, camera, location, often with no click required.

Front Line Defenders, the original report

- **The same spyware was sold to dozens of governments and used on journalists worldwide.** The Pegasus Project examined a leaked list of 50,000 phone numbers selected for targeting: reporters, lawyers, opposition politicians, and the family of a murdered journalist. Sixteen news organisations spent months verifying it.

  Forbidden Stories — The Pegasus Project

- **Mass collection of ordinary people's data is real, and was denied until it was proven.** The 2013 Snowden documents showed bulk interception of communications belonging to people suspected of nothing. One document showed the NSA handing raw, unfiltered intercepts, including Americans' data, to Israeli intelligence.

  The Guardian, the NSA/ISNU memorandum

- **Censorship of this subject has been measured, not just alleged.** Meta commissioned an independent human rights assessment of its handling of Palestinian content in May 2021. The report concluded Meta's actions had an adverse impact on Palestinian users' rights to free expression and political participation — findings Meta published itself.

  BSR, *Human Rights Due Diligence of Meta's Impacts in Israel and Palestine*, 2022 — commissioned and published by Meta

Where this subject is concerned, **assume you are being targeted and work from there**. The tools above are sold commercially, the threshold for using them keeps falling, and the people who have been hit by them almost never saw it coming.

That is not a reason for anxiety. It is simply the footing to work from: treat your accounts, your phone and your footage as though someone is interested in them. The cost of being wrong in that direction is an afternoon of setup. The cost of being wrong in the other direction is infinitely more painful. Work down this page in order, the further you get the better, and every step stands on its own.

LEVEL 1

A security key, the single most effective thing you can do

Start here. Not because it is easiest, but because it shuts down the attack we see most: a convincing fake login page. Even if an attacker is able to get your password, they will not be able to gain access to your accounts.

IF YOU HAVE NOT MET ONE BEFORE

A security key is **a small physical object, about the size of a house key or a short USB stick**. You plug it into your computer, or tap it against your phone, and touch it with a finger to confirm you are there. That is the whole interaction. It costs around €25 to €60, lasts for years, needs no battery and no charging, and works with Google, Microsoft, Apple, Proton, GitHub and most things that matter.

For phishing resistance it is stronger than an authenticator code, and much stronger than a code sent to your phone. A number can be moved onto someone else's SIM with one call to your provider, and a text can be made to look like it came from your bank. Those send you a number to type in, and a number can be typed into the wrong place, which is exactly what a fake login page is built to make you do. A physical key does not give you anything to type. It checks the real address of the site itself, silently, and simply refuses to answer a fake one.

THE ONLY PART OF AN ADDRESS THAT DECIDES ANYTHING



Both have a padlock. Both look right at a glance. Only one of them is the real domain; the other is a phishing link.

> A key registers against the real domain and will only answer to it. You are being asked to spot the difference in the address bar; the key is not asked, it just checks. That is the whole reason it cannot be phished.

Buy two hardware security keys

DO THIS FIRST ~€25–60 EACH

A phishing page takes your password *and* your six-digit code and uses both instantly on the real site. The code is valid, you just typed it into the wrong place. An authenticator app cannot detect this. A security key can, because it checks the domain itself.

- `YubiKey` — the standard, and the one services are most likely to have tested against. Works on a phone over USB-C or NFC.

- `Nitrokey` — open-source hardware and firmware, made in Germany, if you would rather not take a closed device on trust.

Your phone can act as a key too. Android and iPhone can both approve a sign-in on another device as a passkey, and the Bluetooth proximity check that involves is what stops someone relaying the prompt to you from another country. So it is real phishing protection, and it costs nothing. The difference is where the secret sits: on a hardware key it never leaves the key, while a phone passkey is synced by Apple or Google, which makes it only as strong as that account. Worth using if buying keys is not realistic right now, or as cover while they are in the post.

> #### WHY A CODE CAN BE STOLEN AND A KEY CANNOT
>
> When you register a key it stores a secret tied to **the website's real domain**. At login the site sends a challenge and the key signs it, but only after checking that the domain asking matches the one it registered with.
>
> On a lookalike domain the key simply refuses. Not because it noticed something suspicious, but because the maths does not produce a valid signature for the wrong domain. There is nothing for the attacker to steal and replay.
>
> **This protection is structural.** It does not depend on you spotting a misspelled address while tired. That is exactly why it is first on this page.

> #### BUY TWO, THE MISTAKE PEOPLE REGRET
>
> Register **both** keys on every account and keep the second somewhere else — a drawer at home, a trusted person, a safe. An account secured with one key and no backup is an account you can permanently lose.
>
> Save each service's recovery codes into your password manager as you go, and turn off SMS as a fallback afterwards — otherwise you have left the weak door open beside the strong one.
>
> Where a site does not support keys, use Aegis (Android, open source, encrypted backups) rather than SMS.

GO DEEPER

Security keys, every feature, and which model to buy Full guide →

What each of the eight things on a YubiKey actually does, which model suits you, the setup order that avoids locking yourself out, and the attacks it does not stop.

40 MIN ~€25–60 BUY TWO

LEVEL 2

Secure your email before anything else

Your email is the master key. Nearly every other account resets its password to it, so an attacker who owns your inbox owns everything downstream.

Lock the inbox down

HIGHEST PRIORITY | 30 MIN

- [x] **Register both security keys here first**, before any other account.
- [x] **Then review every recovery route.** Turn off SMS wherever the service lets you, because a SIM swap can bypass the key. Keep an authenticator code only where you need it as a fallback, protect its seed and recovery codes in your password manager, and confirm both physical keys work before removing anything.
- [ ] **Check for forwarding rules and filters you did not create.** This is how a quiet intruder keeps reading after you change the password, and almost nobody checks.
- [ ] **Review connected apps and third-party access**; revoke anything you do not recognise or no longer use.
- [ ] **Check active sessions and devices**, and sign out everything unfamiliar.
- [ ] **Set a unique password** from your manager, never one reused anywhere else.

> **THE FORWARDING-RULE TRICK, AND WHY IT MATTERS**
>
> A common pattern after a successful phish: the attacker changes nothing you would notice. They add a filter that silently forwards a copy of every message, or only those matching "password", "invoice", "verify", to an address of theirs, and leave.
>
> You change your password. You turn on 2FA. The rule is still there, still running. **After any suspected compromise, check the rules before anything else.**

> **CONSIDER MOVING PROVIDERS**
>
> Proton Mail and Tuta encrypt your mailbox so the provider cannot read it, and both have free tiers. Neither is magic — mail to and from ordinary addresses is still

> readable at the other end, but it removes your provider as a single place where everything can be demanded or breached.
>
> SimpleLogin gives a different alias to every service, so a leak cannot be correlated with the others, and you can see which company leaked it.

LEVEL 3

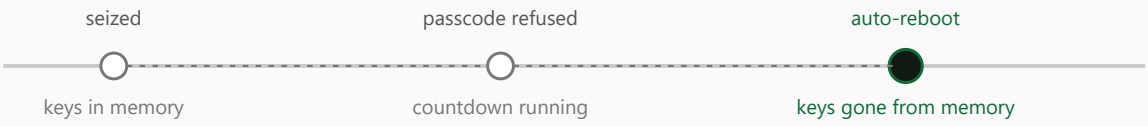
Switch your phone to GrapheneOS

Your phone is the most revealing object you own: it knows where you sleep, who you are near and what you read. It is also the device targeted spyware is aimed at.

TESTED IN THE REAL WORLD

A journalist's phone was seized. They could not open it.

Richard Medhurst was arrested at Heathrow on **15 August 2024** under section 12 of the UK Terrorism Act 2000, and his devices were taken. He reported that repeated requests for his passcode were refused, and that his GrapheneOS phone protected its contents through the **auto-reboot** feature.



A device left without the correct passcode for a set period restarts itself, returning to the state before first unlock, where the encryption keys are not held in memory. Reporting on the case in *Computer Weekly*, Professor Steven Murdoch of University College London noted that even leading commercial forensic tools have at times failed to extract data from GrapheneOS devices without the password, especially where the phone had recently rebooted.

`Computer Weekly, the full account · archived copy`

GrapheneOS, your phone, without the tracking built in FREE PIXEL ONLY

Your phone is the most revealing object you own: it knows where you sleep, who you are near and what you read. A normal Android reports a great deal of that to Google and to every app with a location permission. GrapheneOS removes the privileged Google integration and adds

substantial exploit hardening. That raises the cost of attacking the phone; it does not make the device immune to Pegasus-class spyware or to a newly discovered exploit.

`grapheneos.org` · `install guide` · `FAQ`

> **WHAT YOU GET, AND WHAT IT COSTS YOU**
>
> **What you get:**
>
> - No Google services unless you add them, and when you do, they run sandboxed as ordinary apps with no special privileges.
> - Permissions Android doesn't offer: deny an app *network access* entirely, block sensors.
> - A hardened memory allocator and other defences aimed squarely at the exploit tools sold to governments.
> - Auto-reboot after a set idle period, returning a seized phone to the state where its encryption keys aren't in memory.
> - A duress PIN that wipes the device.
>
> **What it costs:** it only installs on Google Pixel phones, an irony worth naming, and it is because Pixels ship the hardware security features and permit a verified custom OS. Older Pixels stop getting hardware updates, so buy one still in support. Some banking and government apps refuse to run.
>
> Installing is done from a computer over USB and takes about half an hour. The web installer is beginner-friendly.

"But I have heard things about the people behind it" WORTH ADDRESSING

You may run into this, so it is better you hear it here. GrapheneOS has had public friction with parts of the security community — disputes about the lead developer's conduct toward critics, an earlier falling-out with the company behind his previous project, and in 2023 his decision to step down as lead developer and foundation director, citing harassment. None of it is secret and none of it is flattering.

It is also, for this decision, close to irrelevant, and the reason why is the whole argument for open source.

- **You are not trusting the people, you are trusting the code.** All of it is published. Anyone can read what it does, and security researchers who dislike the project intensely have every incentive to find something wrong in it.

- **It is reproducible.** Builds can be checked against the published source, so what runs on your phone can be shown to match what you can read.

- **A closed operating system offers you none of that**, and asks for the same trust on nothing but a brand.

- **And it has been tested where it counts**, not in a lab, but against a state trying to get in. See below.

> **SEIZED, AND THEY COULD NOT OPEN IT**
>
> The journalist **Richard Medhurst** was arrested at Heathrow on **15 August 2024** under section 12 of the UK Terrorism Act 2000, and his devices were taken. He reported that repeated requests for his passcode were refused, and that his GrapheneOS phone protected its contents because of the **auto-reboot** feature: a device left without the correct passcode for a set period restarts itself, returning to the state before first unlock, where the encryption keys are not in memory.
>
> Reporting on the case in *Computer Weekly*, Professor Steven Murdoch of University College London noted that even leading commercial forensic tools have at times failed to extract data from GrapheneOS devices without the password, especially where the phone had recently rebooted.
>
> That is what "battle tested" means here. Not a marketing claim, a documented case of a journalist's phone in the hands of a state that wanted what was on it, and did not get it.
>
> `Computer Weekly, the full account · archived copy`

> **JUDGE IT ON THE THREAT YOU ACTUALLY FACE**
>
> The people who argue about this project online are mostly arguing about personalities. The question in front of you is narrower: does this software make a phone harder to get into than the one you are carrying now.
>
> On the published evidence, and on at least one case where it was tried for real, it does.

If you do not want GrapheneOS LINUX PHONES

Reasonable objections: GrapheneOS runs only on Pixel hardware, so it means buying a Google phone; it is still Android underneath; and you cannot physically verify that the camera and

microphone are off. A **Linux phone** answers all three, and asks for something substantial in return.

- **FuriPhone FLX1**, the most practical one to actually live with right now. Modern hardware by the standards of this category, ships with Plasma Mobile, and will also run Ubuntu Touch or postmarketOS. `furilabs.com`

- **Volla Phone with Ubuntu Touch**, the other genuinely usable daily driver, and the friendliest of these to someone who does not want a project. `ubuntu-touch.io`

- **PinePhone Pro** — **discontinued in 2025**, but images are still actively maintained and there is a large community. Only worth it secondhand, and only if tinkering is part of the appeal. `pine64.org`

- **Librem 5**, the hardware kill switches are the real draw: physical switches that **disconnect the modem, the wifi, the microphone and the camera**. Official support has reached end of life, so treat it as a single-purpose privacy device rather than your only phone. `puri.sm`

- **postmarketOS**, can give some older Android handsets a second life running real Linux. Support varies enormously by model; check yours before buying anything. `postmarketos.org`

> **BE CLEAR ABOUT WHICH THING YOU ARE BUYING**
>
> Most people assume Linux means more secure. On phones that is **not** true, and it is worth knowing before you spend money.
>
> **GrapheneOS is better at resisting attack.** Verified boot, strong per-app sandboxing, a hardened memory allocator, and defences aimed squarely at the exploit tools sold to governments. A typical Linux phone has none of that: no verified boot, desktop-style software with a much larger attack surface, and app isolation that is weak by comparison.
>
> **A Linux phone is better at control.** No Google anywhere, a system you can actually inspect and modify, and on some models **physical switches** — which is the only way to be certain a microphone is off, and stronger than any software promise.
>
> Different threat models. If you are worried about targeted spyware, take GrapheneOS. If you are worried about what the device does routinely, and want to be able to prove the camera is disconnected, a Linux phone answers that better.

WHAT IT COSTS YOU, PLAINLY

The apps. Banking, ride-hailing and most everyday apps can work normally on GrapheneOS. Install sandboxed Google Play where an app needs it, then allow only the permissions and settings it genuinely requires, such as location or notifications. A minority of banking, government or DRM-dependent apps still refuse device-integrity checks, so test the specific app before relying on it.

The camera is markedly worse than any modern phone, which matters a great deal if you film.

Battery and polish. Software improves monthly and basic functions can still be unreliable. Pine64's own wording about their hardware is that functionality "can be spotty".

And the modem is still proprietary. A Linux phone does not remove the baseband you cannot inspect, it is a separate processor on every phone ever made. The kill-switch models deal with it by cutting its power, which is the honest solution.

THE REALISTIC MIDDLE GROUND

If what you actually want is Android without Google, rather than Linux specifically, **CalyxOS** and **/e/OS** sit between the two: de-Googled, far easier to live with than a Linux phone, and weaker than GrapheneOS on the hardening that matters most.

For most people objecting to GrapheneOS on principle rather than on threat model, that is the pragmatic answer.

GO DEEPER

GrapheneOS, a phone that does not testify against you Full guide →

The full install, which devices are supported, the duress PIN and its legal risk, and a straight account of what it does not protect you from.

30 MIN | FREE | PIXEL ONLY

LEVEL 4

KeePassXC , one password instead of two hundred

The reason people reuse passwords is that remembering unique ones is impossible. A manager removes the need to remember anything but one passphrase, and generates passwords no human

would invent. KeePassXC is free, open source, and stores everything in a single encrypted file that you keep, nothing in anyone's cloud, no account, nothing to subpoena or breach.

Set it up properly FREE DO THIS EARLY

`keepassxc.org/download` · `documentation` · Windows, macOS, Linux

- **Create one database** (a `.kdbx` file) and put it somewhere you back up.
- **Choose a master passphrase, not a password** — see below. This is the only one you will ever memorise.
- **Add a key file as a second factor**, stored somewhere other than the database. Both are then needed to open it.
- **Better still, add your YubiKey.** KeePassXC supports hardware-key challenge–response, so the database cannot be opened without the physical key.
- **Install the browser extension** so it fills passwords for you, which also defeats phishing, because it will not fill on the wrong domain. `KeePassXC-Browser`
- **Back the file up.** Lose it with no copy and everything in it is gone, there is no reset link and no company to ask.

> **GENERATING PASSWORDS, THE PART MOST PEOPLE GET WRONG**
>
> Use the built-in generator (the dice icon) for **every site**, and never invent one yourself. Humans produce predictable passwords even when trying not to.
>
> - **For websites:** 20+ characters, letters, numbers and symbols. You will never type it, so length costs you nothing. If a site rejects it, shorten to what it allows, still random.
> - **For your master passphrase:** switch the generator to **passphrase** mode and take **six or more words**. Something like `correct-battery-staple-ocean-tiger-mint` is both easy to remember and enormously harder to crack than a short password with symbol substitutions.
> - **Never reuse the master passphrase anywhere else.** It protects everything.
>
> If you would rather generate the master passphrase with physical dice instead of trusting any software, the EFF publishes a word list for exactly that: EFF dice-

generated passphrases.

WHAT ELSE IT DOES, THIS IS WHY IT IS WORTH LEARNING

- **Two-factor codes.** It can store TOTP secrets and generate the codes, replacing an authenticator app. Convenient, but understand the trade-off: password and second factor then live in the same file. For your most critical accounts, keep the second factor on a separate device or a hardware key.
- **Attachments.** Store recovery codes, backup key files and licence keys inside the encrypted database rather than loose on your disk.
- **Password health.** It flags weak, reused and expired entries, and can check them against known breaches.
- **Auto-type.** Fills credentials into desktop applications, not just browsers.
- **Custom fields.** Security questions, account numbers, anything that should not sit in a notes app.

SYNCING IT BETWEEN DEVICES, WITHOUT A CLOUD COMPANY

The database is one file, so anything that moves files can sync it. The file is already encrypted, so even a cloud provider holding it cannot read the contents, but you can avoid handing it over at all:

- Syncthing — syncs directly between your own devices, with no server in the middle.
- Cryptomator, if you do use a cloud folder, encrypt it again first.

Do not sync it to two devices and edit both offline, you will create a conflicted copy and it is tedious to merge. Let one device be where you add entries.

If all of this sounds like more than you want to manage, Bitwarden handles syncing for you and is a perfectly good choice. A manager you use beats a better one you abandon.

GO DEEPER

KeePassXC, the full guide Full guide →

Choosing a master passphrase, generating everything else, the browser extension, syncing without losing entries, and pairing it with a YubiKey so the database needs the physical key.

1 HOUR FREE OPEN SOURCE

LEVEL 5

Get your work off Google

Not only on privacy grounds, but on the grounds that you can lose everything without warning and without recourse. We know because it happened to us.

WHAT HAPPENED TO THIS ARCHIVE

We kept footage on Google Drive, as almost everyone does. **More than half of it was removed, over 50% of what we had stored there was gone**, and there was no meaningful way to get it back. No warning before, no explanation after, and an appeal process that answers with the same automated response however you phrase it.

We are not claiming someone at Google went looking for us. We have no way of knowing whether a human was ever involved, and we are not going to invent a motive we cannot prove. The point does not need one: **documentation of atrocities trips every automated filter there is**, graphic violence, weapons, injury, death, and at that scale the systems are automatic, the thresholds are not published, and there is no appeal that reaches a person who understands what they are looking at.

The lesson is structural, not political. Evidence held on a platform that can delete it is not archived. It is borrowed.

The problem is not only deletion

WHY THIS MATTERS

Losing files is the sharpest version, and it is not the only cost of keeping this work inside one company's account.

- **It is not private.** Files in Drive are not end-to-end encrypted, the provider can read them, scan them, and hand them over when compelled. That is not a rumour, it is how the product works.
- **One account is one point of failure.** Drive, Gmail, Photos, YouTube, your phone backup and your logins are the same account. Lose it and you lose all of them at once, including the recovery route for everything else.
- **Removal is permanent in practice.** There is no undo, no escalation, and no obligation on anyone to explain.
- **Nothing that exists in only one place is archived.** Whatever you use after this, the rule stands: two

copies in two places you control, and one of them offline.

> **WHY THIS SUBJECT IN PARTICULAR**
>
> Automated moderation cannot distinguish a war crime from the celebration of one. The footage that matters most, a strike, a body, an injury, a weapon — is exactly the footage most likely to be classified as graphic violence and removed.
>
> This has been measured elsewhere. An independent human rights assessment commissioned and published by Meta found its own handling of Palestinian content had an adverse impact on users' rights to free expression. The same dynamic applies to storage and video platforms; it is simply less visible there, because a deleted file does not trend.

Where to put it instead STORAGE

Two we can speak to directly, and what each is good at. The decisive property is **end-to-end encryption**: if the provider cannot read the file, the file cannot be classified, and it cannot be removed for what is in it.

- **Proton Drive** — Swiss, end-to-end encrypted, and the one we can vouch for from our own experience: **nothing we stored there was deleted, including at the height of the genocide, when the same material was being removed elsewhere.** Paid plans; the free tier is small. `proton.me/drive`

- **MEGA** — New Zealand, end-to-end encrypted, and generous with space, which matters when you are storing video. Good for bulk. `mega.io`

- **Your own disks** — unglamorous, and still the only storage nobody can revoke. Two external drives, one kept somewhere else.

- **A torrent, for anything public**, once material is seeded by other people it cannot be withdrawn by anyone, including us. That is why this archive publishes one.

> **WHAT ENCRYPTION HERE DOES AND DOES NOT BUY**
>
> End-to-end encrypted storage means the provider holds bytes it cannot read. It cannot scan them, so it cannot flag them, so the deletion problem above largely disappears. That is the whole reason to move.

> It does **not** put you beyond legal process. Proton is a Swiss company and answers to Swiss courts: in 2021 it complied with an order to log a user's IP address in a French case. It could not hand over message contents, because it does not have them, but the distinction matters and you should know it rather than find it out later.
>
> And if you forget the password on an end-to-end encrypted account, **the data is gone**. Nobody can reset it for you. Put it in your password manager.

The rest of the account SWAP THESE OVER TIME

Storage is the urgent one. The rest can be done gradually, and the point of doing it at all is to stop one company being the single thing that, if lost, takes everything with it.

- **Email → Proton Mail or Tuta.** Encrypted, and not the same account as your storage and your phone. `proton.me/mail` · `tuta.com`
- **Browser → Mullvad Browser or Firefox.** Mullvad Browser is Tor Browser's anti-fingerprinting work without the Tor network — covered properly two sections down. `mullvad.net/browser`
- **Search → Yandex or DuckDuckGo.** A one-click change with no cost to you. `yandex.com` · `duckduckgo.com`
- **Maps → Organic Maps or OsmAnd**, both built on OpenStreetMap and both able to work offline, which is worth more than the privacy in a place with degraded networks. `organicmaps.app`
- **Phone → GrapheneOS**, which is the largest single step and has its own section above.
- **VPN → Mullvad**, if you want one, no account, no email, a random number. Its limits are covered later on this page. `mullvad.net`

> **DO IT IN THIS ORDER**
>
> **1. Get the irreplaceable material out first.** Footage, documents, anything that cannot be recreated. Onto encrypted storage and onto a disk you own, before anything else changes.
>
> **2. Then move your email**, because it is the recovery route for every other account and the thing you least want to lose access to.

> **3. Then the small things** — search, browser, maps, which cost nothing and can happen whenever.
>
> **4. Do not delete the Google account.** Leave it, empty, as a forwarding address and a recovery option. Deleting it in one motion is how people lose the account that other services still point at.

LEVEL 6

Opening files without risking your computer

A document, PDF or image can carry an attack, and a convincing attachment is the other half of a phishing attempt. Open anything you did not expect somewhere disposable.

Dangerzone, the easy answer for documents FREE START HERE

Drop in a suspicious PDF or Office file and get back a clean PDF. It does the dangerous part inside a sealed container and rebuilds the file from images, so anything hidden inside is discarded rather than run. Made by the Freedom of the Press Foundation.

`dangerzone.rocks`

A disposable virtual machine FREE SOME SETUP

For anything beyond a document, an unknown program, a risky site, a file you must actually work with — run it in a VM. If it turns out to be malicious, delete the VM; your real system was never touched.

`VirtualBox` — free, all platforms, the safe default · `VMware Workstation / Fusion` — faster and smoother, and **free for personal use** since Broadcom dropped the fee · `GNOME Boxes`, on Linux, the simplest front end for the kernel's KVM virtualisation; isolation still depends on the guest settings and on keeping the host patched · `UTM` — macOS, including Apple Silicon · **Windows Sandbox** — built into Windows Pro, resets every time

> TWO SETTINGS THAT MATTER
>
> **No shared folders.** The convenient feature letting the VM see your real documents is exactly the bridge malware uses. Move files in deliberately, one at a time.
>
> **Network off by default.** Most of what you're protecting against wants to call home or fetch a second stage.

> Take a **snapshot** of the clean VM once set up, and roll back to it rather than trusting that you cleaned up. A VM is a strong boundary, not a perfect one — escapes exist, and are rare and expensive. For ordinary risks it is more than enough.

GO DEEPER

Dangerzone and virtual machines, the full guide Full guide →

Converting a hostile document into a safe one in a click, when a throwaway virtual machine is the right answer instead, and the settings that quietly undo the isolation.

20 MIN · FREE · OPEN SOURCE

LEVEL 7

Browser, fingerprinting and VPN

Clearing cookies no longer helps much. Sites identify you from the characteristics of your browser itself, a combination usually unique, which survives private mode, a new IP address and a VPN.

Browser fingerprinting, and why blending in beats hiding FREE · READ THIS ONE

Clearing cookies no longer helps much. Sites identify you from the characteristics of your browser itself, a combination usually unique, which survives private mode, a new IP address and a VPN.

Test yourself: coveryourtracks.eff.org · amiunique.org

> **HOW IT WORKS**
>
> When you load a page, your browser answers many small questions without being asked: screen size and colour depth, timezone, language, operating system, CPU core count, installed fonts, graphics card.
>
> Two techniques go further. **Canvas fingerprinting** asks your browser to draw an invisible image; tiny differences in your graphics hardware and drivers make the result measurably yours. **Audio fingerprinting** does the same with a sound never played.
>
> None is identifying alone. Together they are: "Firefox 141 on Linux, 2560×1440, Europe/Athens, these 94 fonts, this GPU" may describe one person in several million. No cookie is needed, so clearing them changes nothing.

The counterintuitive part: making your browser unusual makes this worse. A rare extension, a custom font, a resized window all make you *more* distinguishable. The defence is not to hide, it is to look identical to a large crowd.

WHICH BROWSER TO USE

Mullvad Browser, the best default for most people. Built by the Tor Project with Mullvad: Tor Browser with the Tor network removed, so every user presents the same fingerprint. Free, and no Mullvad subscription needed.

Tor Browser, when you need anonymity, not just privacy. Slower; some sites block it.

Firefox with arkenfox, a good everyday browser, hardened. More work.

Brave — randomises the signals instead of standardising them. A reasonable, different bet.

Add uBlock Origin to Firefox or Brave. Do *not* add extensions to Mullvad or Tor Browser — you'd be stepping out of the crowd that protects you.

Chrome is the thing to move away from: made by the largest advertising company in the world, and the browser these techniques are most at home in.

Mullvad VPN — hide your traffic from your provider €5/MONTH

Without a VPN, whoever provides your internet sees every site you visit and can log it, sell it, or hand it over. A VPN moves that view to the VPN company, so the only question that matters is whether you trust them more, and whether they hold anything to hand over.

`mullvad.net` · flat €5/month · no email required · accepts cash in an envelope

WHY THIS ONE, AND WHAT A VPN WILL NOT DO

Why Mullvad: your account is a random 16-digit number. No email, no name, no password. You can pay in cash. There is nothing in their records connecting an account to a person, a stronger guarantee than a promise not to keep logs.

What it does not do: a VPN is not anonymity. It does not stop a site fingerprinting your browser, does not stop tracking while you're logged into your own accounts, and does not hide you from a service you sign into. Almost every VPN advertised online sells exactly that false impression.

If you need anonymity rather than privacy from your ISP, you want Tor, not a VPN.

GO DEEPER

Mullvad Browser · Mullvad VPN Full guide →

What a site reads about you, why looking identical to thousands of others is the only defence that works, and the things a VPN does nothing about.

15 MIN EACH | FREE / €5 MO | NO AFFILIATE LINK

GO DEEPER

Mullvad VPN, the only one we recommend, and its limits Full guide →

What a VPN does and does not hide, why almost all of them are worse than useless, what makes this one different, and how to set it up without leaks.

15 MIN | €5 A MONTH | NO AFFILIATE LINK

LEVEL 8

Your computer, and why we suggest Linux

We run Linux on anything that matters, and we think it is worth considering for sensitive work. That is our own call, not settled fact, so here is what it rests on and where the evidence stops.

Why we prefer a system that can be inspected OUR POSITION

We do not have evidence of a deliberate operating-system backdoor in Windows or macOS, and we should not present suspicion as fact. The narrower, defensible point is that most of their code cannot be independently inspected. Linux can be audited, changed and built by people outside the vendor, which matters more when the work is actively targeted.

For most people that trade is reasonable. For work that is actively targeted, we think it is not. Here is what that judgement rests on.

> **WHAT IS DOCUMENTED**
>
> - **Both companies were named as PRISM participants** in the 2013 Snowden documents — Microsoft first, in 2007; Apple in 2012. Guardian, June 2013
> - **Microsoft worked with the NSA on access to its services**, including helping circumvent its own encryption on Outlook.com. Guardian, July 2013
> - **Secret access arrangements have existed elsewhere in the security industry.** The CIA and the German BND secretly owned Crypto AG, whose encryption machines were sold to governments while the owners could read the traffic. That history explains caution; it

> does not prove anything about a modern desktop operating system. National Security Archive

> **WHAT FOLLOWS FROM THAT EVIDENCE**
>
> The Snowden documents establish government access to company services and data. They do not establish a hidden access mechanism in Windows or macOS. Closed source also does not prove one exists; it means outsiders have fewer ways to check the claim either way.
>
> That uncertainty is one reason we prefer software whose source can be examined. It is a risk-management choice, not a finding that every reader needs to make.
>
> Open source is not automatically safe; plenty of it is unaudited, and serious flaws have gone unnoticed for years. But it *can* be examined, by you or by anyone, and reproducible builds let you verify that the program you received was built from the source you read. That possibility does not exist for Windows or macOS at all, and that, more than any single revelation, is why we moved.

Which Linux, if you have never used it FREE

You do not need to be technical. A modern desktop Linux installs in twenty minutes, updates itself, and runs a browser, office suite and video calls like anything else. Try it from a USB stick first, it runs without touching your existing system.

Start with `Ubuntu`. Not because it is the most private or the most elegant, because it has the largest community and the most documentation by a wide margin, and that is what decides whether someone switching stays switched. Search an error message and the answer that comes back is written for Ubuntu.

If you have a reason to look further: `Linux Mint` — Ubuntu underneath, without Canonical's additions, and a layout closer to Windows. · `Fedora` — newer security features sooner. · `Debian` — conservative, independent, and extremely stable.

> **BE REALISTIC ABOUT THE SWITCH**
>
> Some things genuinely do not work: Adobe's suite, certain games, some corporate conferencing tools, and hardware with Windows-only drivers. Check what you actually depend on before committing.
>
> **Linux is not automatically more secure.** A badly configured Linux machine is worse than a well-configured Windows one. What it gives you is the *ability to verify and control*, which the next section is about using.

> A reasonable middle path: keep your existing machine, put Linux on a spare laptop, and move the sensitive work there first.

Your computer's defaults FREE 30 MIN

Every desktop system ships configured for convenience and telemetry rather than for you. A handful of settings changes that, and none break normal use.

WINDOWS

- Turn on **BitLocker**; save the recovery key offline
- Use a **local account**, not a Microsoft account
- Privacy settings: off — diagnostics, advertising ID, activity history
- Turn off **Recall** if offered
- Defender is good — leave it on; skip paid antivirus

MACOS

- Turn on **FileVault**
- Turn on the **firewall** (it ships off)
- Privacy & Security → off: Analytics, personalised ads
- Review Location, Microphone and Camera permissions
- **Lockdown Mode** exists if you need it, it breaks things on purpose

---

LINUX

- **Full-disk encryption at install** — retrofitting is painful
- Firewall: `ufw` or `firewalld`, default deny inbound
- Turn on automatic security updates
- Keep **AppArmor**/**SELinux** on rather than disabling it to fix something
- Don't browse as root; don't pipe curl into bash from sites you don't trust

---

It used to be hard. That is the part that changed. THE ACTUAL REASON TO SWITCH NOW

The objection to Linux was never ideological. Something breaks at eleven at night, the error means nothing to you, and the best answer online is a forum post from 2014 about a different version. People avoided Linux because they were on their own with it.

Help is easier to get now. An assistant can translate an error, name the tool you need and explain a proposed command in the context of your version. That shortens the knowledge gap; it does not make generated commands safe or make the assistant an administrator.

- **Something will not install**, or broke after an update. Give it the exact error and ask for one diagnostic step at a time.

- **You do not know what the tool is called.** Describe what you want to do. Half of learning Linux was vocabulary.

- **Someone gave you a command.** Ask what it does before running it. Every time — see the note opposite.

> THE RULE THAT COMES WITH IT
>
> **Never paste a command you cannot explain back in your own words**, and never run a script that does twenty things at once. An assistant will confidently produce something that deletes what you meant to move, and `sudo` gives away everything.
>
> Ask what the command does first. It will tell you, happily, and that habit is worth more than any single fix it hands you.

> GAMERS: YOU DO NOT HAVE TO CHOOSE
>
> Steam runs a large share of the Windows catalogue on Linux through its compatibility layer, and most single-player games simply install and play. The real exception is competitive multiplayer with kernel-level anti-cheat, which mostly refuses to run and is unlikely to change.
>
> So **dual boot**. Keep Windows on the machine purely for those games, and do everything else, your accounts, your email, your files, your work, on the Linux side. The installer offers this during setup and it costs no extra effort.
>
> It is not the purist answer, since Windows is still on the disk. It is the one that gets people to actually switch, and someone doing their real work on Linux is worth far more than someone who read this and changed nothing.

GO DEEPER

Linux — switching, and the first week Full guide →

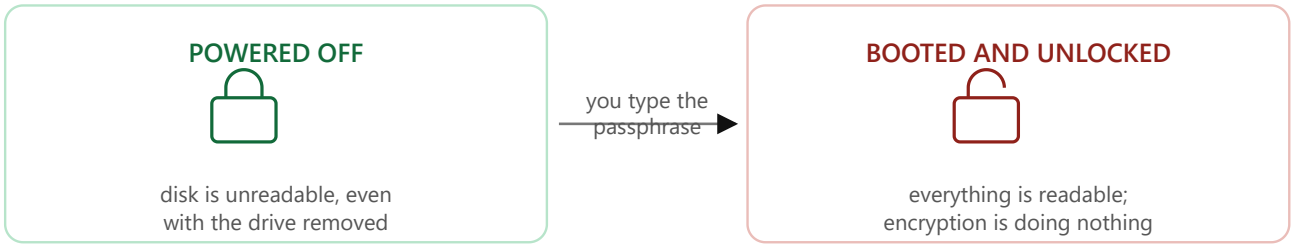
Which distribution, how to try it without touching your disk, installing with full-disk encryption, what replaces what, and the handful of commands worth knowing.

2 HOURS | FREE | REVERSIBLE

LEVEL 9

Encrypt the disk , and lock the boot

Your login password protects nothing. Anyone holding the machine can boot it from their own USB stick, or pull the drive out, and read every file on it without ever seeing your login screen. Full-disk encryption is the one measure that still works after the computer has left your hands, and on most systems it is a checkbox you have not ticked.



> This is why the habit matters as much as the setting: shut down rather than sleep whenever the machine is going somewhere it could be taken.

THE ONE THING TO UNDERSTAND BEFORE YOU START

Disk encryption protects a machine that is **powered off**. Once it has booted and you have typed the passphrase, the disk is unlocked and stays unlocked — encryption is doing nothing at all for you until the next shutdown.

So the habit matters as much as the setting: **shut down, do not sleep**, whenever the machine is going somewhere it could be taken from you, a border, a checkpoint, a search, a bag left in a car. A sleeping laptop is an unlocked laptop with the lid closed.

Windows — BitLocker CHECK THIS TODAY 20 MIN

Many Windows machines are already encrypted and their owners do not know it, and on those machines Windows has quietly uploaded the recovery key to a Microsoft account. That is convenient, and it means the key exists on someone else's server, where it can be handed over. Check what you have before you change anything.

- **See where you stand.** Open a Command Prompt as administrator and run `manage-bde -status`. It tells you which drives are protected and how.

- **Turn it on** from Settings → Privacy & security → Device encryption, or the BitLocker control panel on Pro editions.

- **Save the recovery key yourself, somewhere you control.** Print it, or put it in your password manager. Do not accept only the Microsoft account option.

- **Add a startup PIN.** By default BitLocker unlocks silently from the TPM, so the machine boots straight to the login screen and the encryption adds far less than you think. A PIN is what turns it into real protection.

- **Then remove the cloud copy** of the recovery key from your Microsoft account, once you are certain you have your own.

> #### WHY A TPM-ONLY SETUP IS WEAKER THAN IT SOUNDS
>
> With the default configuration, the TPM releases the key only when the measured boot state matches what BitLocker expects. This blocks reading the disk in another machine and normally blocks a modified boot chain. It does load the key before Windows sign-in, which gives an equipped attacker with prolonged physical access more avenues than a configuration that also requires a PIN.
>
> Adding a **pre-boot PIN** closes that. On Pro editions it is a Group Policy setting — **Require additional authentication at startup**, after which `manage-bde` can add the PIN to the drive. It is the single most valuable change in this section for a Windows user.

> #### HOME EDITIONS
>
> Full BitLocker is a Pro and Enterprise feature. Windows Home instead offers **Device Encryption** on hardware that supports it, which is the same underlying technology with fewer controls, and it escrows the recovery key to your Microsoft account automatically, with no local-only option.
>
> If that is unacceptable for what you do, the honest answers are to upgrade the edition, or to move the work to Linux, where the key is only ever yours.

Linux — LUKS FREE AT INSTALL TIME

The strongest and simplest of the three, with one catch: it is easiest by far to set up *during installation*. Every mainstream installer offers "Encrypt the new installation" as a checkbox on the disk step. Tick it. Retrofitting encryption onto a running system is possible and genuinely unpleasant.

- **Tick the encryption box in the installer**, and use a passphrase of four or more random words rather than a password.
- **Write the passphrase down until you have memorised it.** There is no recovery, no reset and no company to ask. That is the design.
- **Check it afterwards** with `lsblk -f`, your main partition should show as `crypto_LUKS`.
- **Add a second passphrase** as a backup key slot if you want one: `sudo cryptsetup luksAddKey /dev/sdaX`.
- **Back up the LUKS header.** It is small, it holds the key slots, and a corrupted header means the data is gone even with the right passphrase.

> **WHAT IS LEFT UNENCRYPTED, AND DOES IT MATTER**
>
> A standard install encrypts everything except a small boot partition and the EFI partition, which have to stay readable for the machine to start. No documents live there, but the kernel and initial boot files do, which is what makes the tampering attack below possible.
>
> For most people this is a fine trade. If it is not, some distributions can encrypt `/boot` as well, and Secure Boot plus a firmware password raises the bar considerably without that complexity.

> **UNLOCKING WITH A YUBIKEY**
>
> LUKS can be configured to unlock with a hardware key instead of, or in addition to, a typed passphrase, the same device from Level 1.
>
> **Always keep a passphrase in a second key slot** when you do this. A lost or dead key with no fallback means the disk is unrecoverable, and that failure is permanent.

macOS — FileVault 10 MIN

On a modern Mac the drive is always encrypted by the hardware, but without FileVault turned on, the key is available to the machine itself, so that encryption protects you against very little. FileVault is what ties the key to your password and makes a powered-off Mac genuinely closed.

- [ ] **Turn it on** in System Settings → Privacy & Security → FileVault.
- [ ] **Choose the local recovery key, not iCloud.** macOS offers to let Apple hold your recovery key. That is the same escrow problem as BitLocker: convenient, and it puts a copy somewhere that can be compelled.
- [ ] **Store that key in your password manager** or on paper somewhere secure.
- [ ] **Turn off automatic login**, and set the Mac to require a password immediately after sleep or screensaver.
- [ ] **Shut down rather than close the lid** when it matters, for the reason in the callout above.

> **APPLE SILICON AND INTEL DIFFER HERE**
>
> On Apple Silicon the security is built into the chip and there is no separate firmware password to set, the boot process is locked down by design, and startup security is managed through Recovery.
>
> On older Intel Macs a **firmware password** is a separate, worthwhile step: it blocks booting from external media and prevents an attacker from starting into Recovery to work around you.

The boot password, the step almost nobody takes 15 MIN

Encryption stops someone reading your disk. A firmware password stops them changing how the machine starts — booting from a USB stick, altering the boot order, or quietly replacing the software that asks you for your passphrase. The two protect different things and you want both.

- [ ] **Enter firmware setup** as the machine starts, usually F2, F10, Del or Esc, depending on the manufacturer.

- [ ] **Set the administrator or supervisor password**, not just the user password. The user one only stops the machine booting; the admin one stops the settings being changed.

- [ ] **Disable booting from USB and network** while you are in there, or move them below the internal drive in the boot order.

- [ ] **Leave Secure Boot enabled.** It means unsigned code cannot be loaded during startup, which is exactly the attack this is about.

- [ ] **Record the password in your password manager.** Forgetting a firmware password ranges from a service-centre visit to a replacement board.

> THE "EVIL MAID", AND WHAT A BOOT PASSWORD ACTUALLY BUYS
>
> The attack this defends against is simple: someone with brief physical access — a hotel room, an office overnight, a bag out of sight at a border — boots your machine from their own media and modifies the unencrypted boot files so that the next time you type your passphrase, it is captured.
>
> Your disk was encrypted the whole time. It did not help, because you unlocked it yourself afterwards.
>
> Be clear about the limits, though: **a firmware password is a speed bump, not a lock**. On many machines it can be cleared by opening the case, and some vendors have master overrides. It defeats opportunistic tampering and buys time; it does not defeat a determined, equipped adversary with your laptop in a back room.

What this does not protect you from READ THIS ONE

- [ ] **A running machine.** Once booted and unlocked, everything is readable. This is the most common way disk encryption turns out to have been pointless.

- [ ] **Sleep.** On most systems the key stays in memory while suspended. Hibernate or shut down instead.

- [ ] **Anything synced to the cloud.** The copies in your cloud drive, your email and your backups are not covered by this in any way.

- **Malware you are already running**, which operates as you, on an unlocked disk.

- **Being compelled to unlock it.** In several jurisdictions refusing to hand over a passphrase is itself an offence, and in many more the practical pressure is considerable. Encryption is a technical control, not a legal shield — know the law where you are before you rely on it.

LEVEL 10

Your wifi router

Everything in your home goes through it, and almost nobody ever opens its settings. It is usually the oldest, least-updated computer you own, and it is the one facing the street.

Move from WPA2 to WPA3 FREE 10 MIN

With **WPA2-Personal**, someone within range can capture a connection handshake and test password guesses offline. A long, random passphrase still resists that attack; the danger is a short or reused wifi password that can be guessed cheaply.

WPA3-Personal replaces that exchange with SAE, which resists passive offline password guessing and adds forward secrecy. It is a meaningful upgrade. It does not rescue a weak password, and transitional WPA2/WPA3 mode keeps some downgrade exposure for older devices.

Wi-Fi Alliance on WPA3 · KRACK, the 2017 WPA2 attack

> **IF SOME OF YOUR DEVICES ARE TOO OLD FOR WPA3**
>
> Most routers offer a **WPA2/WPA3 transitional** mode, letting new devices use WPA3 while older ones fall back. It is weaker than WPA3-only, an attacker can try to force the fallback, but far better than WPA2 alone.
>
> Check what needs it first. Usually it is one old printer or a smart plug, and those belong on the guest network anyway.

The rest of the router checklist FREE

- [ ] **Change the admin password.** Not the wifi password, the one for the router's own settings page. A great many are still the sticker default, and that is the first thing anything on your network tries.
- [ ] **Turn off WPS.** The convenience button with the PIN. The PIN is short enough to be guessed outright, and that defeats even a strong wifi password. `CISA guidance`
- [ ] **Turn off UPnP.** It lets any device on your network silently open a port to the internet without asking you.
- [ ] **Turn off remote administration** so the settings page is not reachable from outside your home.
- [ ] **Update the firmware**, and check whether updates still exist. Many routers are abandoned within a couple of years while still being sold.
- [ ] **Rename the network** so it does not announce the router model, that tells anyone nearby exactly which known flaws to try.
- [ ] **Put smart devices on the guest network.** It keeps them away from your laptop and phone.
- [ ] **Set your own DNS** on the router: `Quad9` or `NextDNS`, it covers every device at once.

> **THE ROUTER YOUR PROVIDER GAVE YOU**
>
> ISP-supplied routers are often the worst of the lot: remote management enabled for the provider, undocumented accounts, and firmware updated at the provider's convenience rather than when a flaw is found.
>
> You can usually buy your own and put theirs into modem-only or bridge mode. If you want to go further, OpenWrt is open-source router firmware with a long support life, and it runs on a lot of ordinary hardware.
>
> For a thorough, maintained checklist there is nothing better than routersecurity.org.

LEVEL 11

Operating systems built for this

Most people will never need these. Worth knowing they exist.

Tails, a computer that forgets FREE GUIDED SETUP

An operating system you boot from a USB stick. Everything goes through Tor, and when you shut down it remembers nothing, no history, no files, no trace on the machine you borrowed.

`tails.net` · `install guide`

> **IT USED TO BE TOO FIDDLY FOR MOST PEOPLE, LESS SO NOW**
>
> Tails had a reputation for being awkward: verifying the download, writing it to a USB stick correctly, getting a machine to boot from it, then working out persistent storage without losing anything. Plenty of people gave up somewhere in the middle.
>
> For someone non-technical, that used to make Tails impractical. That barrier is much lower now. An AI assistant can walk you through the whole thing on your own machine — checking the signature so you know the image was not tampered with, writing the stick, finding the boot key for your specific laptop, and explaining what persistent storage will and will not keep. It turns a fiddly, well-documented process into a sequence you can work through without already knowing the terminology.

> **WHEN IT IS THE RIGHT TOOL**
>
> Use it when you want a session to leave nothing behind: a shared or borrowed computer, or when the risk is someone later examining the machine. Runs on almost any laptop and costs the price of a USB stick.
>
> It does not protect you if you log into your own accounts inside it, that identifies you regardless of Tor. And amnesia cuts both ways: anything you want to keep must go into the optional encrypted Persistent Storage, deliberately.

Qubes OS — separation as the whole design FREE GUIDED SETUP

Instead of one computer running everything, Qubes runs many small virtual machines and keeps them apart: banking in one, email in another, anything risky in a disposable one that vanishes when closed. A compromise in one is contained rather than total.

`qubes-os.org` · related: `Whonix`, `Kicksecure`

> **THE HONEST ASSESSMENT**
>
> Qubes is the strongest practical desktop security available, and it asks a lot: specific hardware, real memory, and a change in how you work, you decide which compartment every task belongs to. Opening a file in a disposable VM is one click, which makes Level 6 almost automatic here.
>
> That used to put it beyond most non-technical people. An AI assistant can now help you check whether your hardware is suitable, follow the installer, create the first compartments and explain the choices as you make them. It does not remove Qubes' hardware requirements or its learning curve, but it makes the setup far more approachable than it used to be.
>
> It is aimed at people with a specific, serious reason. If you are reading this to reduce everyday tracking, Levels 1 and 2 will do far more for you.

LEVEL 12

Check if you are already in a breach

Before protecting anything, find out what is already out. Most people have credentials in several public breaches and have never been told.

Have I Been Pwned FREE START HERE 2 MIN

Type in an email address and it tells you which breaches it appears in, and what was taken in each: passwords, addresses, security questions, private messages. Run by a well-known security researcher, no account needed, and it does not store what you search for.

- [ ] **Check every address you use**, including old ones you no longer read. An abandoned mailbox is still a password-reset route.
- [ ] **Anywhere you reused that password, change it.** That is the actual risk: a breach at a forum you forgot about becomes a break-in at your email only because the password was the same.
- [ ] **Turn on notifications** so you are told about the next one rather than finding out years later. `Notify me`
- [ ] **If you own a domain, search the whole domain.** One check covers every address at it, including ones

staff or family created without telling you. `Domain search`

`haveibeenpwned.com`

> **WHAT A CLEAN RESULT DOES AND DOES NOT MEAN**
>
> It means your address is not in a breach that has become **public and known**. That is a narrower claim than it sounds. Plenty of breaches are never disclosed, are traded privately for years, or are disclosed without the data ever being published.
>
> So treat a clean result as good news about a specific list, not as a statement about your accounts. The measures elsewhere on this page are what actually protect them.

Your password manager already does this, for everything at once FREE THE ONE THAT SCALES

Checking addresses one at a time tells you which *sites* leaked. Your password manager can tell you which of *your passwords* are compromised, reused or weak, across every account you hold, in one pass. If you followed `Level 4`, this is already sitting in the software and most people never open it.

- **KeePassXC:** Database → Database Reports → *HIBP* and *Password Health*. Flags breached, reused, weak and ancient entries. `user guide`
- **Bitwarden:** Reports → Exposed Passwords, Reused Passwords, Weak Passwords, Data Breach Report. `reports`
- **Proton Pass and 1Password** have the same feature under a different name.
- **Work down the list by importance, not by order.** Email first, then anything you publish from, then anything holding money. You do not have to fix two hundred entries this evening.

> **IF YOUR PASSWORDS LIVE IN YOUR BROWSER INSTEAD**
>
> Chrome and Safari both check saved passwords against known breaches, and it is worth running once: Google Password Checkup, or Passwords → Security Recommendations on Apple devices.

> But note what you are being told. Browser password stores are the first thing an information stealer takes, so a breach report there is a reason to move them into a real password manager, not a reason to feel covered.

Check a password itself, safely FREE THE PART NOBODY KNOWS ABOUT

Separate from the email search, Have I Been Pwned lets you test **a single password** against more than a billion that have appeared in breaches. A hit does not mean your account was broken into; it means that password is on lists attackers work from, so it should not be in use anywhere.

`haveibeenpwned.com/Passwords`

- [ ] **Never type a password into any other "is my password leaked" site.** There is no way to tell a real checker from a harvester, and the incentive to build the second one is obvious.

- [ ] **This one is safe for a specific, checkable reason**, explained opposite. It is not a matter of trusting them.

> **WHY THIS ONE CAN BE TRUSTED WHEN OTHERS CANNOT**
>
> Your browser hashes the password locally and sends only the **first five characters** of that hash. The server returns every leaked hash starting with those five, often several hundred, and your own browser checks the list.
>
> The password never leaves your device, and the server cannot tell which of the answers it sent was the one you were asking about. That property is called k-anonymity, and it is why this is a different proposition from a site that simply asks you to paste a password in.

What to ignore SAVES MONEY

- [ ] **Paid "dark web monitoring".** Most of it resells the same public breach data you can search for nothing, wrapped in an alarming dashboard. Some is bundled free with a bank or an antivirus subscription; that is fine, just do not buy it on its own.

- [ ] **Services that want your full name, address and date of birth to "scan for you".** You would be handing over a fuller identity profile than most breaches contain.

- **Panic at a long list.** Ten old breaches of dead forums where you used a password you have since retired is not an emergency. One recent breach where you reused a current password is.

LEVEL 13

Messaging , three options, honestly compared

All three of these encrypt your messages properly. They differ in what they know about you, who could be compelled to hand it over, and how much friction you will accept. Read the first box before you pick, because the usual recommendation comes with a caveat we are not going to skip.

> ABOUT THE CONCERNS OVER SIGNAL
>
> Signal is the most recommended messenger in the world and its encryption is not in doubt: the protocol is open, audited, and good enough that WhatsApp and Google adopted it. You will also hear that it has a back door for the US government. **There is no evidence for that, and we are not going to claim it.**
>
> What is documented, and what the suspicion is built on, is this: **Signal is a US organisation on US infrastructure**, so it sits under US legal process; its early development was funded in part by the **Open Technology Fund**, a body funded by the US government; and until recently it required a phone number, which is an identity.
>
> Set against that: when they have been served with legal demands, what they were able to produce amounted to the date an account was created and when it last connected. That is a real, tested result, and it is the strongest argument in their favour. **You now have the same facts we do.** If that residual risk is unacceptable for what you do, the other two below answer it differently.

Signal FREE EASIEST NEEDS A PHONE NUMBER

The right answer for most people, most of the time, because the people you need to talk to already have it, and a secure messenger nobody you know uses protects nothing. Best encryption, best usability, and unusually little held about who talks to whom.

- **Turn on Registration Lock.** Without it, anyone who takes over your phone number can register your account on their phone. The single most important setting in the app.

- **Set a username** so you can be contacted without giving out your number, and turn off *Who can find me*

by number.

- [ ] **Turn on disappearing messages by default.** What is not on the phone cannot be read off it later.

- [ ] **Set a screen lock inside Signal itself**, separate from the phone's, and turn off message previews.

`signal.org`

> **GOOD FOR**
>
> Day-to-day conversation with people who are not going to install anything unusual. Voice and video calls. Anything where the alternative is the person defaulting back to SMS or WhatsApp, which is the real failure mode.
>
> **Weakest at:** hiding *that you use it*, and hiding your phone number from the organisation itself.

Session FREE NO PHONE NUMBER READ THE CAVEAT

Started as a fork of Signal and removes the two things people object to: **it needs no phone number and no email**, you get a random Session ID and that is your whole identity, and messages are **onion-routed through three hops**, so no single server knows both where a message came from and where it is going. The foundation behind it moved from Australia to **Switzerland in 2024**, citing the tightening of Australian surveillance law.

- [ ] **Nothing to sign up with.** No number, no address, no name. Losing your recovery phrase loses the account, so store it in your password manager.

- [ ] **Decentralised.** There is no company server holding the messages, which is the point.

- [ ] **Your Session ID is the thing to share**, and it is not linked to anything about you.

`getsession.org`

> **GOOD FOR**
>
> First contact with someone you do not want to exchange phone numbers with. Sources. Anyone who cannot or will not register an identity. Situations where the

> fact that you have an account at all should not be linkable to you.
>
> **Weakest at:** long-lived archives of sensitive conversation, until V2 ships. Turn on disappearing messages and the point largely goes away.

Element (Matrix) FREE YOU CAN HOST IT MOST SETUP

Element is the main app for **Matrix**, an open protocol rather than a product. The difference that matters: it is **federated**, like email — anyone can run a server, and people on different servers talk to each other. Which means you can run your own, and then no company holds your group's conversation at all. It is used this way by several European governments for exactly that reason.

- **No phone number needed.** A username on whichever server you choose.
- **Encryption is on for private rooms, but check it** — Matrix is a general protocol and not every room is encrypted. Verify the room is before saying anything that matters.
- **Built for groups**: rooms, threads, permissions, moderation. It behaves like a team chat, not a phone messenger.
- **You can host the server yourself**, which is the strongest privacy position available on this page, and also makes you responsible for keeping it patched.

`element.io · matrix.org`

> **THE CAVEAT**
>
> Message contents are encrypted; **a good deal of metadata is not**. Because rooms are replicated across every server whose users are in them, those server operators can see who is in a room and when they are active. On a public server that is a real exposure, and it is the argument for running your own.
>
> Managing encryption keys across several devices is also fiddlier than the other two, and losing them means losing access to old messages.

> **GOOD FOR**
>
> A team, a newsroom, or a group that works together over time and wants its own infrastructure. Bridges to other networks. Anything where you would otherwise

> reach for Slack or Discord and should not.
>
> **Weakest at:** one-to-one conversation with someone non-technical, where Signal wins on the only measure that counts, whether they will use it.

Which one, in one line each

IF YOU WANT THE SHORT ANSWER

- **Signal**, for everyone you already know, and for anything where usability decides whether it gets used at all.
- **Session**, when you must not hand over a phone number, or when the fact that you have an account should not be linkable to you.
- **Element**, for a group that works together, especially if you can host the server yourselves.
- **These are three good options, not the only defensible ones.** Briar and SimpleX answer other threat models well. Any end-to-end encrypted choice is stronger for sensitive conversation than SMS, ordinary email, Telegram group and cloud chats, or an ordinary social-platform direct message.

LEVEL 14

Home security cameras

A camera pointed at your home, reachable from the internet, is a serious thing to get wrong, and the industry has a poor record. Treat every one of them as hostile until you have configured it otherwise.

Why they are usually weak

READ FIRST

- **Default or shared passwords**, sometimes hard-coded and unchangeable.
- **Cloud by default.** The video leaves your house and is stored on a company's servers, reachable by their staff and by anyone who breaches them.
- **"P2P" remote viewing** that punches out through your router using UPnP, so the camera is reachable from outside even though you never forwarded a port.
- **Firmware that is rarely updated**, on hardware sold for years under a dozen different brand names.

This is not theoretical. In 2021 attackers compromised a support server used to administer cameras operated by Verkada. Verkada's own incident report says video or images belonging to 95 customer organisations were accessed. The affected sites included environments where camera footage was especially sensitive. `Verkada incident report`

How to run them safely FREE

- **Prefer local recording over cloud.** A camera that records to a box in your house cannot be handed over by a company or exposed in its breach.
- **Block the cameras from the internet entirely** if you can, a separate network or VLAN with no outbound access. If you need to see them from outside, use the manufacturer's own app over that separate network rather than opening a port on your router.
- **Turn off UPnP and P2P** on both the camera and the router.
- **Change every default password**, and give each camera its own from your password manager.
- **Turn on two-factor** if you must use a cloud account, and use a security key where offered.
- **Update firmware**, and treat a camera that has stopped receiving updates as end-of-life.
- **Think about where they point.** Indoor cameras covering living space are the ones that cause real harm when breached. Many people do not need them at all.

> **OPEN-SOURCE OPTIONS THAT KEEP THE VIDEO AT HOME**
>
> Frigate — records and does object detection locally, on your own machine, with no cloud account. Works with ordinary cameras that speak RTSP.
>
> Home Assistant — runs your cameras and smart devices locally and can keep them off the internet entirely.
>
> When buying, look for a camera that supports **RTSP** or **ONVIF**. That means it can feed a local recorder and does not depend on the manufacturer's cloud continuing to exist.

LEVEL 15

Software worth having, all free

Nothing here is sponsored and nothing here is paid for. Where a tool belongs to a section above, the level is named so you can go back for the reasoning rather than install something on the strength of one line.

MESSAGING

`Signal`, for the people you already know. Turn on Registration Lock and set a username so you need not give out your number.

`Session`, when handing over a phone number is the thing you cannot do.

`Element`, for a group that works together, and can be run on a server you control.

`Molly`, a hardened Signal client for Android that encrypts its local database at rest.

`Briar` — works with no internet at all, over Bluetooth or local wifi. Built for protests and shutdowns.

The three-way choice is argued out in Level 13.

EMAIL AND ALIASES

`Proton Mail` · `Tuta`, encrypted mail, free tiers.

`SimpleLogin`, a different alias for every site, so a leak cannot be linked back or correlated across services.

`Thunderbird`, a proper mail client you control.

PASSWORDS AND CODES

`KeePassXC`, an offline password manager that answers to nobody. Full guide in Level 4.

`Aegis`, two-factor codes on Android with encrypted backups, so losing the phone is not losing the accounts.

Where a site offers a hardware key, that beats both of these. Level 1.

OPENING FILES, AND BROWSING

`Dangerzone` — converts a document you do not trust into a PDF that cannot do anything. Level 6.

`Mullvad Browser`, Tor Browser's fingerprinting defences without the Tor network.

`Tor Browser`, when the network itself is the thing watching. Level 7.

FILES, BACKUP, AND STAYING OFF THE CLOUD

`VeraCrypt`, encrypted containers and drives.

`Cryptomator` — encrypt a folder before it syncs to anyone's cloud.

`Borg` · `restic` — encrypted, deduplicated backups.

`Syncthing` — sync between your own devices with no cloud in the middle.

`LibreOffice`, documents without an account, telemetry, or a company that can close them.

Encrypting the whole disk is built into your operating system already. Level 9.

PHONE

`F-Droid`, an app store of only free software.

`Exodus Privacy` — shows which trackers are inside an Android app before you install it. Genuinely eye-opening.

The operating system underneath matters more than any app on top of it. Level 3.

NETWORK

`Quad9`, a Swiss non-profit resolver that keeps no record tied to you and refuses to answer for known-malicious domains.

`NextDNS`, more control and much better filtering, but it is a company, and its log is a feature you switch on. Leave it off, or set the retention and jurisdiction deliberately.

`OnionShare` — send a file directly over Tor, no account, no service in the middle.

`OpenSnitch` (Linux) · `LuLu` (macOS) — nothing reaches the network without asking you first. Noisy for a week, then quiet, and the week is the point: you find out what has been talking.

LINUX

`Flatpak` — runs an application in a sandbox, so a compromise of it is not a compromise of you.

`Timeshift` — system snapshots; roll back a bad change in minutes.

`fwupd` — firmware updates, the layer almost everyone forgets.

`Lynis` — audits the system and hands you a prioritised list of what to fix.

Keep AppArmor or SELinux enabled. Disabling it to make something work is the most common self-inflicted hole on a Linux desktop. And be careful with sandbox tools that install themselves as root to work — Firejail is the common example, and its design has produced repeated privilege-escalation flaws.

WHOLE SYSTEMS BUILT FOR THIS

`Tails` — runs from a USB stick and forgets everything when you shut down.

`Qubes OS` — every task in its own isolated machine, so one bad file takes one compartment.

`Whonix` — forces all traffic through Tor at the network layer, so an application cannot leak around it.

When each of these is the right answer is Level 12.

LEVEL 16

Guides worth your time

There is a great deal of bad security writing. These are by people who support real users under real threat, and are kept current. They are ordered by depth: the first group assumes nothing, the later ones assume you want the engineering and the evidence.

START HERE

EFF — Surveillance Self-Defense

The best free course there is. Start with the threat-modelling module.

Data Detox Kit

Tactical Tech. Small daily steps, no jargon, the friendliest entry point here.

Privacy Guides

Current tool recommendations with the reasoning shown. Community-reviewed.

WRITTEN FOR PEOPLE UNDER THREAT

Security in a Box

Front Line Defenders and Tactical Tech. Written for activists, many languages.

Freedom of the Press Foundation

Practitioner-grade guidance from the people who build SecureDrop and train newsrooms. The closest thing here to a working reference that is updated as things break.

Totem

Free structured courses for journalists and activists, in several languages.

Me and My Shadow

Shows what your digital traces actually look like. Good for understanding, not just doing.

UNDERSTANDING FINGERPRINTING

Cover Your Tracks

EFF. Tests your browser and explains each signal in plain language. Start here.

BrowserLeaks

Every vector separately — canvas, WebGL, audio, fonts, WebRTC. The clearest way to see what a site reads.

Am I Unique?

Shows how rare your exact combination is, against a research dataset.

arkenfox wiki

Detailed, current, and unusually honest about which hardening actually helps and which is theatre.

The Design and Implementation of the Tor Browser

Why making everyone identical beats trying to hide, written by the people who built it. This is the archived draft, still the clearest statement of the argument; the living copy is maintained on the Tor Project's GitLab.

THE TECHNICAL DEEP END

Qubes OS documentation

Compartmentalisation followed to its conclusion. Worth reading even if you never install it, because it makes you notice how much your normal machine shares with itself.

Whonix documentation

Probably the largest body of free writing on anonymity engineering anywhere, including long, careful pages on what does not work and why.

GrapheneOS technical documentation

What hardening a phone means at the operating-system level — memory allocator, exploit mitigations, what the sandbox actually enforces. Specific and unusually free of marketing.

THE EVIDENCE THAT THIS IS DONE TO PEOPLE

Citizen Lab — targeted threats

The forensic record of journalists, lawyers and human-rights workers being hacked by states, case by case, with the technical proof attached. If you want to know whether the premise of this page is real, read this rather than us.

Amnesty International Security Lab

Publishes its methodology, not just its conclusions, so the work can be checked and repeated.

Mobile Verification Toolkit

The actual forensic tool used to find Pegasus on phones, with documentation you can follow. Demanding, and the honest answer to "how would I ever know".

EFF — Street-Level Surveillance

What the hardware on the ground actually is and does: IMSI catchers, plate readers, face recognition. Useful before filming anything in public.

OONI

Measures internet censorship from inside the countries doing it, with open data. How you show a block is real rather than a bad connection.

IF YOU HANDLE OTHER PEOPLE'S MATERIAL

Hacks, Leaks, and Revelations

Micah Lee's book on working with hacked and leaked datasets — handling, verifying and publishing without burning a source or yourself. Free to read in full online. The closest thing to a manual for what this archive does.

SecureDrop

How submissions are received when the sender's safety is the whole problem. The documentation is worth reading even if you never run one.

Holistic Security

Tactical Tech, on the part no tool fixes: how a group under pressure makes decisions, shares risk, and keeps working without burning out.

Digital First Aid Kit

Walks you through diagnosing it — account takeover, device seizure, harassment, suspected spyware, and tells you who to contact.

CPJ Digital Safety Kit

Committee to Protect Journalists. Concise and practical. The 2019 in the address is its first publication date — it is reviewed regularly and was last checked in February 2026.