Security, Privacy and Anonymity in Linux Mint

Michel NALLINO

Nice, France, 2025-06-28 - Revision 50

Document Internal Links

1. Introduction

Linux Mint security is based on Ubuntu one. But Ubuntu is not the first choice in terms of security:

- It lacks Security-Enhanced Linux (abbreviated in SELinux), a Linux security module, which makes it possible to define a policy of compulsory access control to the elements of a system derived from Linux. SELinux is considered more secure than Ubuntu enabled AppArmor.

[SELinux can be installed, but all its configuration needs to be done manually, a tough work, while it is installed and preconfigured in professional oriented distributions].

- Ubuntu includes several security tools (Flatpak, Snap, Firejail, Clamav, rootkit scanners, UFW/GUFW, GnuPG, Lynis, Tripwire…) but most (Flatpak, Firejail, Clamav, rootkit scanners, GnuPG, Lynis…) are not updated while they do need to be kept up-to-date.

- Among the tens of thousands of available Ubuntu packages, only 2300 are maintained for security.

- Linux Mint/Ubuntu default installations do allow weak security practices (GRUB password protection is not set by default, there are no requirements in terms of passwords length or passwords changes periodicity…).

However, we will see how to use and tailor Linux Mint/Ubuntu to an acceptable security level.

What are the Linux systems targeted by security attacks?

- At the moment, Linux Mint/Ubuntu and other Linux distributions still benefit of the fact that there are much fewer computers using them than using Windows or Mac-OS ones. The cost of developing an attack is not balanced by the expected benefit.

- So, the main targets are Linux servers, used by companies, with several possibilities of high benefits: ransomware, industrial or technical knowledge theft, control take of a network through web shells…

But single users are also targets:

- They share a large part of their operating systems with servers, have the same security breaches, and can be touched by the side effects of an attack targeting servers.

- There are now reports of attacks directly targeting users, with ransomware or with mail attachments such as a false campaign of job offer.

Internet privacy protection and anonymity concerns are also addressed in this document. All users are targeted by unwanted aggressive advertisements when browsing internet or when receiving mails. They are tracked, their data are gathered and sold. Their internet activity can be fully traced and consolidated.

Who are people interested in this document?

Home users and small companies users, using one of Linux Mint distributions (mainly Linux Mint 20.x, 21.x or 22.x), or using one of Ubuntu distributions (mainly Ubuntu 20.04 LTS, 22.04 LTS or 24.04 LTS).

[This document might be useful to users with other Ubuntu distributions, or with other Ubuntu derivatives than Linux Mint].

Not all users have the same security, privacy and anonymity requirements: physicians, lawyers, journalists, small companies, activists, whistle-blowers and cryptocurrencies owners will have higher requirements than "Joe User". Among the precautions mentioned in this document, each user will have to choose the ones most adapted to his/her computer and internet use.

Methodology: after a presentation of Ubuntu main security features, the approach is the usual "Threats List", "Prevention" (to prevent a threat or reduce its probability), "Detection" (to detect when a threat has become an effective attack), and "Pre-Established Arrangements" (to reduce the impact of a confirmed, effective attack and to recover system and user files).

Annexes include detailed, in-depth tutorials.

A "Threats / Prevention means correspondence matrix" allows reader feeling concerned by a threat to access the corresponding prevention mean(s).

Warning: from version 22.1 "Xia", Linux Mint no longer uses "apt" but "aptkit" and "captain" has replaced both "gdebi" and "apturl". In the code examples (after [code]) in the text, concerned users should adapt the syntax to those commands.

2. Ubuntu Main Security Features

The major security feature is the use of rights: rights to read, write and execute files. Files are separated in system owned files, found on "/" and user owned files, found on "/home/username/", abbreviated in "~/".

A user with standard rights can read, write and execute the files he owns, and read and execute the system owned ones. Using his "superuser" password, he can have a write access to system files.

A user with limited rights can read, write and execute the files he owns, and read and execute the system owned ones. He cannot get write access to system files.

This is a strong protection, implemented at the kernel level, and that can still be reinforced by the use of AppArmor and other restricting security features (Seccomp, Namespaces…).

See Annex 1: Launching Commands and GUI Applications with Superuser Rights.

Another security feature is that Linux Mint or Ubuntu distributions give access to trusted, secure packages:

- From Linux Mint repositories, distributed under Linux Mint team responsibility.

- From Ubuntu repositories, distributed under Canonical/Ubuntu team responsibility.

A third noticeable security disposition is that files are downloaded from internet without file attributes: they can be run only when user gives them an executable permission.

[Note that compressed files, in "tar.gz" or "zip" formats, can contain files with executable permission, which can be executed once uncompressed; this is a way malware are distributed].

This is completed by a firewall, running at the kernel level, using Netfilter (with Iptables, or UFW/GUFW interfaces).

Those dispositions, and others, make Ubuntu and Linux Mint secure by design.

However, operating systems and their applications are complex software, and they have bugs. Those bugs are the results of human errors at the specification, code writing, code compiling and testing stages. Some of those bugs induce security weaknesses that could be used by an attacker, and this affects Linux Mint/Ubuntu intrinsic security.

Moreover, default installation settings or user wrong ones may degrade security.

3. Threats List

3.1) Local security attacks, needing physical access to computer

[Mainly found in insecure environments such as company premises, hotel room…].

LSA1:
Local attacker without user rights connects to the computer.

LSA2:
Local attacker without user rights reads the disk contents.

LSA3:
Local attacker installs a hardware keylogger in the keyboard to steal passwords.

LSA4:
Computer is left unattended with a running session. A local attacker can use it and access files.

3.2) Distant security attacks

DSA1:
An unauthorized distant attacker connects to the computer, then uses a system uncorrected security breach and installs malware (viruses, ransomware, cryptocurrency miners, Trojans…) or accesses to user's files.

DSA2:
An authorized distant user, connected to the computer, gains privileges escalation.

DSA3:
A malformed web page exploits an uncorrected security breach in the browser, or in any internet connecting application, to put the operating system in an unstable condition, to gain privileges escalation and to install malware or to access user's files.

DSA4:
Distant attacker incites user to download and install an insecure program or malware. User can download this malware by browsing internet, or by receiving a mail attachment.

DSA5:
Distant attacker in your vicinity attempts to penetrate your LAN using wireless connection.

3.3) Various security threats

VST1:
You use some Windows programs under Linux Mint/Ubuntu with Wine or Mono and you expose your system to Windows security breaches.

VST2:
Your system security is not correctly set.

3.4) Privacy threats

PT1:
Your internet service provider (ISP) knows all what you do on internet (DNS requests, accessed and read web-pages) and logs all.

PT2:
Websites track you when you browse internet. Your browser displays unwanted advertisements, your personal data are collected, gathered and sold.

PT3:
Your mail service provider knows the content of all the mails you send or receive.

PT4:
You receive spams in your mailbox. Those spams track you, your personal data are collected, gathered and sold.

PT5:
The mails you send or receive can be read by anybody reading them on a server used to route them.

PT6:
You send or receive mails whose sender identity cannot be proven.

PT7:
You use LanguageTool, a free grammar and spell checker, and your documents contents are sent to an external server.

PT8:
You use a translation program, and your documents contents are sent to an external server.

PT9:
Your personal data (passwords, payment information, personal information) can be stolen.

PT10:
Applications can "phone home" and expose private information.

3.5) Anonymity threats

AT1:
Websites you visit know your real IP address and your connection date/hour. This, combined with tracking, allows them or official authorities to deduce your name, address and what was your internet activity at what time.
[That threatens users wanting to conceal what they do].

4. Prevention

4.1) Protect the access to your computer

This is prevention against LSA1, LSA2, LSA3 and LSA4 threats.

1) Physical protection

* If your computer is a desktop one, put your disks in removable caddies, remove the caddies after use and put them in a safe.

* If your computer is a desktop one, put the keyboard in a safe after use.

* If your computer is a laptop one, put it in a safe after use.

* Never let your computer unattended or unprotected when in an insecure environment (company premises at night, hotel room etc.)

2) Software protection

A user without any right can connect to your computer and gain root privileges by an attack under GRUB menu, using your computer to boot from your main disk or from a live CD / DVD / USB key.

Attacks are explained here:

Booting from live DVD:

https://www.cyberciti.biz/tips/howto-recovering-grub-boot-loader-password.html

Booting from computer disk:

http://3wymlmcsvxiaqzmbepsdawqpk6o2qsk65jhms72qqjulk5u4bgmvs3qd.onion/grub/boot_from_command_prompt (onion link, in Tor Network)

https://www.tecmint.com/how-to-hack-your-own-linux-system/

* The solution is to fix boot order in your UEFI/BIOS (disk first, before CD / DVD), to password protect the UEFI/BIOS access, and to password protect your computer boot with another password.

* You can also password protect your GRUB menu.

See Annex 2: Password Protect your GRUB Menu.

See Annex 3: Password Selection.

Boot password and GRUB password don't prevent to read the contents of your disks: the disks can be removed from your computer, and installed in another one to be read.

* To prevent to read the contents of your disks, use disk encryption: whole disk or "/home" partition (choices to be done during system installation); or create an encrypted container on your "/home" using Veracrypt (program download and documentation from https://veracrypt.eu/en/).

Some discussion:

full disk encryption may not be necessary, programs don't need to be encrypted, and will slow your disks accesses because of decryption time, will slow your SSDs by preventing trim and will render system recovery much more complicated after a crash;

full "/home" partition may not be necessary, not all the files it contains are confidential, encryption will slow partition access and will render user's files recovery more complicated after a crash;

most "reasonable" solution may be the use of Veracrypt, with a container where user stores his confidential files and passwords;

→ full disk or full "/home" partition encryption should be used only with very high security requirements, in a hostile environment.

[Note that disk encryption does not give an additional protection against distant attacks: once the computer is running, encrypted disks mounted and deciphered, the content of the disks is seen by the operating system in the same way as if disks were not encrypted].

See Annex 3: Password Selection.

See Annex 4: Encryption.

When encryption is not used, secure deleting files is a way to avoid further files recovering. The way to do it depends on the kind of disk used.

- With Hard Disk Drives:

The "shred" command can be used, it overwrites the file to delete (see its help or its man for syntax). However, user should check that the file system really overwrites the file on its location. Here is an extract of shred man:

> CAUTION: Note that shred relies on a very important assumption: that the file system overwrites data in place. This is the traditional way to do things, but many modern file system designs do not satisfy this assumption. The following are examples of file systems on which shred is not effective, or is not guaranteed to be effective in all file system modes:
>
> * log-structured or journaled file systems, such as those supplied with AIX and Solaris (and JFS, ReiserFS, XFS, Ext3, etc.),
>
> * file systems that write redundant data and carry on even if some writes fail, such as RAID-based file systems,
>
> * file systems that make snapshots, such as Network Appliance's NFS server,
>
> * file systems that cache in temporary locations, such as NFS version 3 clients,
>
> * compressed file systems,
>
> In the case of ext3 file systems, the above disclaimer applies (and shred is thus of limited effectiveness) only in data=journal mode, which journals file data in addition to just metadata. In both the data=ordered (default) and data=writeback modes, shred works as usual.

Since ext3 and ext4 journaled file systems are, by default, in the data=ordered mode, shred will work.

- With Solid State Drives:

SSDs cells have three states: erased, written to 0, and written to 1. Before to write on a cell, it should be erased. In order to limit write wear, the disk controller will not overwrite a file, but rather write the content in another place. Shred will not work...

To erase a cell before to write it slows down the SSDs write speed. Two mechanisms are there to avoid this, the garbage collector and trim. When a trim is done, the list of abandoned cells is sent by the file system to the disk controller. This can take a few minutes. But, once the disk controller has received the list of abandoned cells, it takes it a few seconds (for modern SSDs) to a few minutes (for older ones), to erase the abandoned cells, preventing any further file recovery.

In Linux Mint, a trim is, by default, done once a week.

It can also be done with the following code:

[code]

sudo fstrim -av

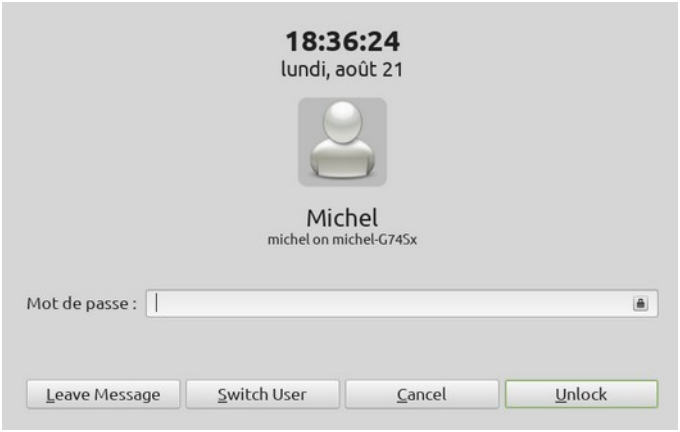
If the computer is left unattended with running session, a local attacker can use it and access files. Prevention is simple: lock the screen when you leave your computer. This can be done in two ways:

- In the Control Center, you can find Keyboard Shortcuts. Screen Lock shortcut is generally defined by default with the keys CTRL+ALT+L. Hitting these three keys at the same time locks the screen.

[You may want to personalize the Screen Lock shortcut, and replace CTRL+ALT+L by the keys combination of your choice].

You may add to your dashboard a Screen Lock icon:

Typing CTRL+ALT+L or clicking on the Screen Lock icon locks your screen and prevents the unwanted use of your computer, you then need to enter your user password to unlock the screen:



4.2) Update your system

This is generic prevention against LSA1, LSA2, DSA1, DSA2, DSA3, DSA4, DSA5.

As previously said, the protection mechanisms would be perfectly effective without programming flaws.

Linux kernel security is highly criticized:

- It is an aggregate of separate developments, made by different developers, at different times (some are very old), without development guidelines in terms of security.

- Google also criticizes the insufficient resources put into kernel security, in terms of the number of hours of development and testing.

The vulnerabilities are not only concentrated in the kernel, but also in system software, programs with internet access etc.

Because of those vulnerabilities, protections could be bypassed (for example, an attacker could acquire super-user rights without having to enter the password).

In Linux Mint and Ubuntu LTS distributions, the programs have a frozen version, but:

- the kernel, system software, libraries, etc. are updated each time a security breach is detected,

- browsers and email clients are updated each time their developers update them.

The first rules to apply are therefore:

- stop to use distributions that are no longer maintained,

- apply all the updates proposed by the update manager,

- update the user installed programs,

- always use the latest versions of security programs, downloaded from their developers' websites (such as Flatpak framework, Firejail, Firetools, rootkits and viruses scanners, Veracrypt, GnuPG, Lynis…) since they are not, in most cases, maintained by Ubuntu.

Update your computer firmware. This is specially true if you use an EFI system: Secure Boot security has been compromised by a test key, see https://arstechnica.com/security/2024/07/secure-boot-is-completely-compromised-on-200-models-from-5-big-device-makers/, including a list of 215 compromised devices; and another weakness drove to the LogoFAIL attack, where the manufacturer logo can be replaced by a malformed BMP file, see https://www.binarly.io/blog/blind-trust-and-broken-fixes-the-ongoing-battle-with-logofail-vulnerabilities (several manufacturers, such as AMI, Insyde, Intel, Lenovo and Phoenix have issued advisories).

[And do not disable Secure Boot: the first Linux targeting bootkit malware, called Bootkitty, has been discovered in November 2024; it can use the LogoFAIL weakness, or can install itself on computers with Secure boot disabled, see https://www.binarly.io/blog/logofail-exploited-to-deploy-bootkitty-the-first-uefi-bootkit-for-linux]

4.3) Increase your system intrinsic security with Ubuntu Pro

This is generic prevention against LSA1, LSA2, DSA1, DSA2, DSA3, DSA4.

Ubuntu, and Linux Mint as a consequence, maintains for security 2300 packages from the main repository for 5 years. When Ubuntu Pro is enabled, those 2300 packages are maintained for 10 years, and 23000 extra packages from Universe are maintained for 10 years. (Coverage for both critical, high and selected medium CVEs).

Moreover, critical fixes and many zero-day vulnerabilities are applied in under 24 h on average.

Ubuntu Pro is available for free to personal users, with a license for up to 5 computers. It can be used on Linux Mint distributions with hacking of some system text files.

Professional customers should pay for it and are bound to use it on Ubuntu LTS unmodified distributions only.

Annex 5: How to Enable Ubuntu Pro on Linux Mint .

4.4) Use trusted sources

This is generic prevention against LSA1, LSA2, DSA1, DSA2, DSA3, DSA4.

As said, Linux Mint or Ubuntu distributions give access to trusted, secure packages:

- From Linux Mint repositories, distributed under Linux Mint team responsibility.

- From Ubuntu repositories, distributed under Canonical/Ubuntu team responsibility.

However, they are stable, long term support distributions. It means that most applications are not updated, and users may want to use more recent ones or install programs not supplied by Linux Mint / Ubuntu.

→ as soon as a user does that, he needs to be careful and install programs from trusted sources only.

How are extra programs distributed, and what are trusted sources?

- Programs can be downloaded as deb packages, dedicated to one distribution only (dynamically linked with the distribution libraries), or working with all distributions (distributed with their own libraries).

- Programs can be downloaded with installers, inside a compressed "tar.gz" file.

- Programs can be downloaded as flatpaks, snaps, AppImages.

- Programs can be downloaded from Personal Packages Archives (PPAs).

- Programs can be downloaded using other package managers: GUIX (for fully Open Source programs), pip (for python scripts).

- Sources can be downloaded and compiled…

→ There is a large variety of programs that can be found on a user's personalized system.

Whatever the way programs are downloaded and install, user should only use sources he can trust.

- Programs directly downloaded from LibreOffice, Mozilla, Tor project, FreeFileSync, Gimp, Darktable, Dia, Inkscape websites etc. and from other major developers websites can be trusted.

- Flatpak programs are generally published by independent developers on Flathub, with some exception such as Mozilla or Gimp; when users installs such programs, he should trust both the original developer and the flatpak creator. Note that, since flatpaks run in a sandbox, security risk is low.

- Snap programs are generally made by independent developers and published on snapcraft.io under Canonical control. Some snaps are official ones (Mozilla…). Since nothing is known about the checks done by Canonical, the same precautions should be taken as with flatpaks. Note that, since snaps run in a sandbox, security risk is low.

- AppImages is another way to deliver applications with their own dependencies. AppImages are generally generated by independent developers publishing them to websites such as appimagehub.com and others. Users should be very careful when downloading those AppImages. However, since AppImages are launched without superuser rights, security risk is low.

Some AppImages are official ones (Audacity, GnuPG Desktop, LibreOffice…) and can be downloaded without risk.

Users can generate their own AppImages, see:

AppImageKit: https://github.com/AppImage/AppImageKit

pkg2appimage: https://github.com/AppImageCommunity/pkg2appimage

and, to make their own version of LibreOffice AppImage:

https://github.com/antoniofaccioli/libreoffice-appimage

or https://git.libreitalia.org/libreitalia/loaih

Official AppImages or user generated ones present no more security risk than the original application.

- Compiling sources downloaded from trusted developers websites presents no risk.

- Risks are high with PPAs, called "Untrusted PPAs" by Ubuntu. Those PPAs can be created on launchpad by any developer. They bring users some programs not provided by Linux Mint/Ubuntu, or provide new versions of programs, and are not checked and not endorsed by Linux Mint/Ubuntu.

Some are official or semi-official ones:

* They can be linked from the original developer website.

* They are maintained by the team in charge of the project (examples: "flatpak stable PPA" or "Graphics Drivers team PPA").

Most are maintained by individual developers or users; even if they are not malevolent, they may not be fully competent, or they may work on an insecure computer, and their use implies high risks.

Some of those PPAs are known to not only provide programs, but also change many system libraries; their use may break some programs, and they are very difficult or impossible to remove (Rob Savoury PPAs, Oibaf graphical drivers PPA…).

→ User should pay a lot of attention to PPA choice.

- Finally, risk is maximal when user founds on social networks, forums, or when browsing internet, links to download programs which normally need to be paid, are presented as free or with high cost-reduction, and links don't point to official developers websites.

→ That is one of the main ways to spread malware.

4.5) Use a firewall

This is prevention against DSA1, DSA2.

The Linux kernel includes the Netfilter subsystem, which is used to manipulate or decide the fate of network traffic headed into or through your server; running on a kernel subsystem, Netfilter based firewalls are robust ones.

The Netfilter firewall installed on Linux Mint/Ubuntu is Iptables. It can be configured in command line mode (see its help or its man), or with Uncomplicated Firewall, UFW, in command line mode and with its graphic interface GUFW.

The first step is to launch GUFW and enable it. You can launch it from Linux Mint control center.

There are three profiles: Home, Public and Office, you can set different rules for each profile.

Home users, without any server on their computer will simply deny incoming connections and allow outgoing ones for each of the three profiles:

[Note: some people recommend allowing outgoing connections for legit applications (browsers, mail clients…) and then blocking outgoing connections for all applications that have not been formally allowed, in the hope it will block malware programs and prevent them to connect to their servers. It is an illusion: malware can attach in memory at the end of an allowed program and can get outgoing internet access hidden from the firewall].

Some users may host a server on their computer. They will add rules to allow incoming connections to their server.

- The rules will allow the minimum access as possible (port or service, TCP or UDP protocol, limited range of incoming IP addresses).

- The hosted services will use encryption (ssh, sftp, https) and strong authentication (strong password, or two-factor authentication, see Annex 3: Password Selection). The corresponding server applications will be chosen and/or set to limit the access to computer directories to the minimum needed, they should be preferably sandboxed, or run in virtual machines.

- The incoming connections should be logged, and user should examine them, at least at the beginning in order to check that the incoming connections do reflect the way user wanted they work.

- The worst thing to do would be to use peer-to-peer software such as BitTorrent or eMule: you would need to open a port, to the entire world range of IP addresses, without any password… Ransomware, cryptocurrencies miners and other malware do love that!

GUFW allows implementing incoming rules:

- it includes a list of preconfigured rules,

- it allows setting specific rules, with simple or advanced interfaces.

Here are some external documentation links about GUFW/UFW:

https://help.ubuntu.com/community/Gufw

https://itsfoss.com/set-up-firewall-gufw/

https://ubuntu.com/server/docs/security-firewall

There is a large offer or firewalls for Linux, with command line or GUI interface; here are some (untested):

- firewalld, https://firewalld.org/

- IPFire, https://www.ipfire.org/

- Portmaster, https://safing.io/

Some may have larger capabilities than firewall ones (filters, intrusion detection etc.).

4.6) Sandbox your applications

This is prevention against DSA2, DSA3, DSA4.

There are two kinds of sandboxing.

In the first kind, a program runs totally isolated from the operating system; such sandboxing is used to test suspicious programs in a secure environment.

In the second kind, programs can run normally (they can access network, print, open and save files), and the sandboxing limits files access and system resources access to what is needed. We are here interested in that second kind of sandboxing.

The applications that need to be sandboxed are internet connecting ones (browsers, mail clients, ftp clients, servers, virtual machines, multimedia players, video capture applications, virtual machines etc.) and applications used to open internet downloaded files (LibreOffice, documents viewers, file compression/uncompression utilities, images viewers/editors etc.).

Of course, other kinds of applications can also be sandboxed, without any damage.

In order to understand why sandboxing is necessary, I will take an example, using a browser.

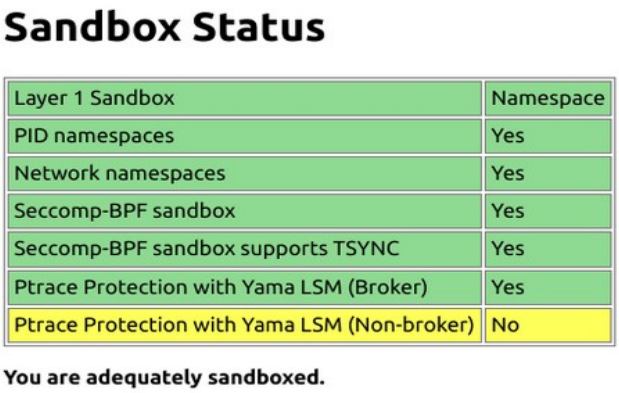
In a normal use, the browser saves its disk cache and configuration on a hidden directory of home user (as an example, "~/.mozilla"); when a file is downloaded with user action, a window appears offering the user the choice where to download the file; user has set to block webcam use, and the browser respects the webcam setting.

The browser, being launched by the user without superuser rights, has the same file access rights as the user; its write rights are limited to user home, "/tmp" (*) and any connected device (USB disk, key…) where user can write.

Malicious script action is limited by the browser (with its own sandboxing) and by the operating system: an application launched without superuser rights cannot write on the system "/".

Here are details of browsers sandboxing.

Chromium own sandboxing, from "chrome://sandbox/" (Chromium 130):



Sandbox Status

Layer 1 Sandbox	Namespace
PID namespaces	Yes
Network namespaces	Yes
Seccomp-BPF sandbox	Yes
Seccomp-BPF sandbox supports TSYNC	Yes
Ptrace Protection with Yama LSM (Broker)	Yes
Ptrace Protection with Yama LSM (Non-broker)	No

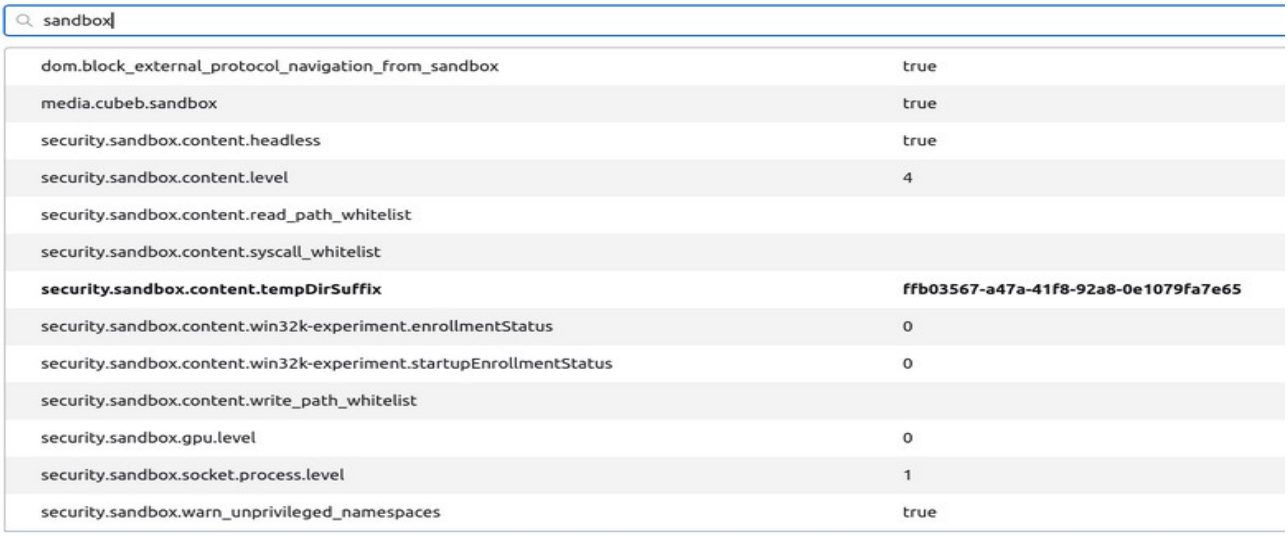
You are adequately sandboxed.

Firefox own sandboxing, from: "about:support" (Firefox 131):

Sandbox

Seccomp-BPF (System Call Filtering)	true
Seccomp Thread Synchronization	true
User Namespaces	true
Content Process Sandboxing	true
Media Plugin Sandboxing	true
Content Process Sandbox Level	4
Effective Content Process Sandbox Level	4

and from "about:config":



[Firefox has sandboxed the gpu on Windows systems; however, it is not enabled on Linux systems, even in 131.0.3 version; in future releases of Firefox check "security.sandbox.gpu.level": if its value is 1, it is enabled, if its value is 0 it is not enabled].

Any supplemental sandboxing (flatpak, snap or Firejail) adds an external sandboxing layer to existing built-in one.

But browsers are complex software, and they have bugs. Those bugs are the results of human errors at the specification, code writing, code compiling and testing stages. They are found in the software itself, in the libraries/dependencies it uses, or in the tools used by developers. Some of those bugs induce security weaknesses that could be used by a malicious script: at each revision of a browser, there are several (in the range 1 to 50) security fixes, and a few times a year, a highly critical weakness is exploited before a fix is available (they are called "zero-day exploits").

Exploiting such a critical weakness, a malicious script could find a way to circumvent browser settings (and to use webcam without your consent), or to put the system in an unstable state, gain superuser privileges, and find a way to write files on the system "/" and corrupt it.

[(*): on Linux there is only one directory for temporary files, for system and user; as a consequence, any user has a write access on that directory; this is different on Windows where system temp and user temp are different directories, with different access rights.]

Moreover, on Linux Mint 22.x / Ubuntu 24.04, "unprivileged user namespaces" is no longer enabled by default, while it is still enabled in former versions of Linux Mint.

An explanation is given by Mozilla here: https://support.mozilla.org/en-US/kb/install-firefox-linux#w_security-features-warning.

> The sandbox in Firefox makes use of unprivileged user namespaces when creating new processes for enforcing more security. This can be considered a security risk, therefore some Linux distributions have started to restrict its usage and only allow it to work where there is an AppArmor profile.

Since the use of unprivileged user namespaces may be at risk, distributions such as Debian / Ubuntu / Linux Mint disable it. But Firefox, and other browsers, make a large use of unprivileged user namespaces: when it is disabled their sandbox security is notably reduced.

From Ubuntu 23.10, and so from Linux Mint 22, the use of an AppArmor profile, even a pseudo one, allowing all and blocking nothing, can entitle Firefox to use unprivileged user namespaces. See https://ubuntu.com/blog/ubuntu-23-10-restricted-unprivileged-user-namespaces.

Linux Mint 22 installation includes AppArmor and AppArmor profiles for Firefox and for Chromium. Those profile include "userns" rule and allow Firefox or Chromium (as installed from Linux Mint repositories) to use "unprivileged user namespaces".

But other browsers, derived from Firefox (such as LibreWolf, Tor Browser, Mullvad Browser…) or derived from Chromium (such as Google Chrome, Ungoogled-Chromium, Brave Browser, Vivaldi Browser…) do need such an AppArmor profile to have full sandboxing.

If not provided with the browser installation, user should write one, inspired from the following example, in "/etc/apparmor.d/opt.google.chrome.chrome" in this case:

[code]

```
 abi <abi/4.0>,
include <tunables/global>
/opt/google/chrome/chrome flags=(default_allow) {
  userns,
  # Site-specific additions and overrides. See local/README for details.
  include if exists <local/opt.google.chrome.chrome>
}
```

Extra sandboxing is strongly recommended for browsers.

What are the sandboxing tools available on LinuxMint/Ubuntu? Mainly five: flatpak, snap, Firejail, AppArmor and systemd sandboxing.

How can we compare their security? A good sounding approach could be to test sandboxed applications, with the five different tools, on different LinuxMint / Ubuntu flavors, with a set of attacks and to compare the results got with sandboxed applications to the results got with

unsandboxed ones. That would be very expensive and would take much time, and the results would be valid only within the set of tested operating systems / applications / sandboxing tools / attacks. AFAIK, that has not been done.

Without that test approach, there are only opinions. Here is mine, based on the way those five tools work.

1) Flatpak

Flatpak uses Bubblewrap https://github.com/containers/bubblewrap. (Bubblewrap is installed with flatpak framework).

From Bubblewrap github page:

> The goal of bubblewrap is to run an application in a sandbox, where it has restricted access to parts of the operating system or user data such as the home directory.
>
> Bubblewrap works by creating a new, completely empty, mount namespace where the root is on a tmpfs that is invisible from the host, and will be automatically cleaned up when the last process exits.
>
> The user can specify exactly what parts of the filesystem should be visible in the sandbox. Any such directories you specify mounted nodev by default, and can be made readonly.
>
> The maintainers of this tool believe that it does not, even when used in combination with typical software installed on that distribution, allow privilege escalation.
>
> In particular, bubblewrap uses PR_SET_NO_NEW_PRIVS to turn off setuid binaries, which is the traditional way to get out of things like chroots.

Permissions settings (established by flatpak publishers or user changed with Flatseal) allow restricting file accesses: as an example, restricting file access of your browser or e-mail client to "xdg-download" will allow it to only read /write your Downloads directory, with no access to any other directory (except with user control, through file chooser portal).

What changes when a browser is used with flatpak?

Flatpak applications are not intrinsically more secure than non-flatpak ones: application software is the same, and libraries/dependencies are replaced by the ones in runtimes. But they run in a sandbox.

In the normal use, the browser saves its disk cache and configuration on a hidden directory of home user (as an example, "~/.var/app/org.mozilla.firefox"); when a file is downloaded on user request, a window appears offering the user the choice where to download the file: this is done through the "file chooser portal", an interface between the flatpak sandbox and the system, having the same write privileges as normal user (file chooser portal CANNOT have superuser rights); if the use of devices has been prevented in the browser flatpak permissions, webcam cannot be used.

Suppose now a malicious script exploits a browser security weakness:

- Webcam block cannot be circumvented, since it is controlled by flatpak sandbox permissions, and not by the browser only.

- No file can be written on the system "/":

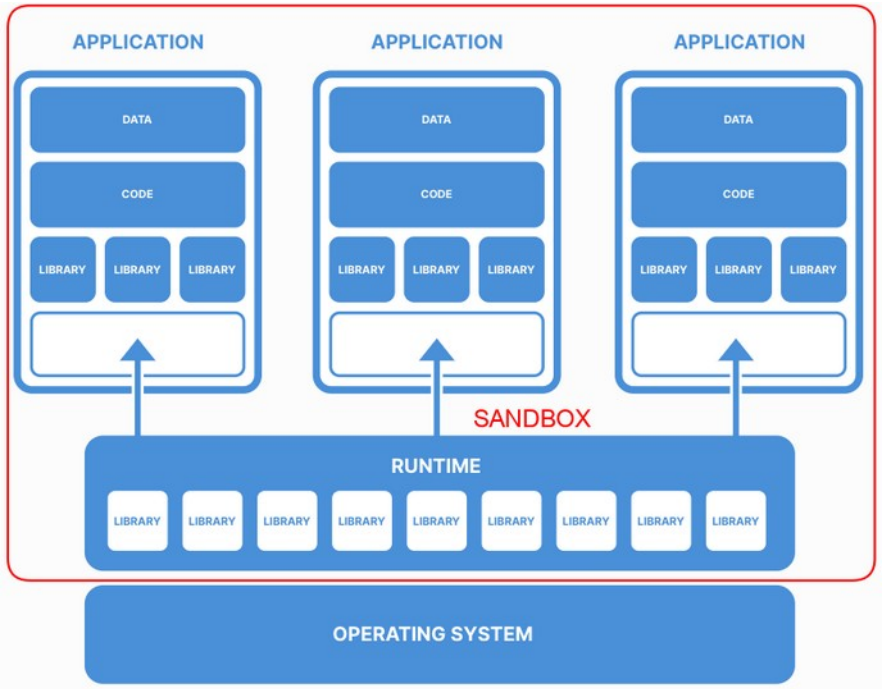
* It is not possible through file chooser portal, with user interaction, since that portal cannot write on the system "/".

* It is not possible in an unattended way, without user interaction, since that way is controlled by flatpak filesystem permissions, and since the most permissive possible one is the same as file chooser portal: no file can be written on the system "/".

→ **System corruption by a malicious script is not possible.**

* Moreover, since the operating system is isolated from the browser by flatpak sandbox, it cannot be put in an unstable state.

→ **Privileges escalation is blocked by both flatpak sandbox and the operating system and is considered as not possible. Browsers zero-days attacks can be prevented.**



[Flatpak framework, of course, does use operating system libraries; but it is not in direct contact with applications and is not connected to internet].

Flatpaks are available for all Linux distributions. Fedora Silverblue distribution is entirely based on flatpaks.

See Annex 6: Flatpak Tutorial.

2) Snap

Snaps work in similar ways as flatpaks. Snapd needs to be installed first on the system, and it installs a core, including libraries. When a snap application is installed, it is downloaded with its own core, where are the extra libraries that it will use.

Snaps go further than flatpaks, since services can be sandboxed.

Snap sandboxing makes an extensive use of AppArmor.

→ Since snapped applications don't use system libraries, but the ones in cores, operating system is isolated from those applications, and snaps security is comparable to flatpaks one.

Snaps are disabled by default on Linux Mint 20.x and 21.x. Snaps can be enabled by:

- removing "/etc/apt/preferences.d/nosnap.pref" or commenting its content,

- installing snapd,

    [code]

    sudo apt install snapd

Snaps applications can be installed from the snap store, https://snapcraft.io/snap-store.

Snaps are available for several Linux distributions. However, their security is heavily degraded on distributions where SELinux is enabled (RedHat, Fedora…), since only one "major" security module such as AppArmor or SELinux can be activated at the same time. Snaps mainly target Ubuntu and its derivatives, and Debian based distributions. Ubuntu core, an internet of things embedded version of Ubuntu, is entirely based on snaps.

General information on snaps can be found here: https://en.wikipedia.org/wiki/Snap_(software).

3) Firejail

From its website, https://firejail.wordpress.com/:

> Firejail is a SUID program that reduces the risk of security breaches by restricting the running environment of untrusted applications using Linux namespaces and seccomp-bpf. It allows a process and all its descendants to have their own private view of the globally shared kernel resources, such as the network stack, process table, mount table.

Reading its profiles contents show that it also uses AppArmor, Netfilter…

Unlike with flatpaks or snaps, firejailed applications use the operating system libraries. The operating system is not isolated from firejailed applications, and a zero-day exploit is still possible.

→ Firejail offers a lower security than flatpaks or snaps.

However, it has some interesting functionalities:

- it allows sandboxing AppImages,

- it allows sandboxing applications not available as flatpaks or snaps.

Firejail should be installed using the downloads from its website:

https://firejail.wordpress.com/download-2/,

since LinuxMint/Ubuntu have obsolete versions only.

It can be completed by Firetools, a GUI application used to launch firejailed applications and to create applications profiles.

Linux Mint 21.x users can download Firetools deb from Firejail download page and install it with gdebi, while Linux Mint 20.x users need to download the source (tar.gz) and compile it (compiling procedure is detailed in the README file inside the tar.gz).

Firejail stores its application profiles in "/etc/firejail". With "0.9.72" release, there are some 1247 profiles. They should not be edited: at next Firejail update installation they will be replaced.

User can create its own profiles in "~/.config/firejail". If two profiles with the same name exist in "/etc/firejail" and "~/.config/firejail", the one that will be launched is the one in "~/.config/firejail".

To launch an installed application with Firejail:

[code]

firejail application_name

Example:

[code]

firejail firefox

To launch an AppImage application with Firejail:

[code]

firejail --appimage full_application_path

Example:

[code]

firejail --appimage ~/opt/LibreOffice.AppImage

[Note that, without an existing profile with the same name as the AppImaged application, a default AppImage profile, a very restrictive one, would be used. Firetools, with its configuration wizard, can be used to write a profile; the profile should then be copied and saved in a "application_name.profile" file, in ~/.config/firejail" directory].

Once tested, a shortcut can be added or modified in launch menu or in Firetools, to launch a firejailed application. Firejail documentation mentions an automatic installation of Firejail; the command:

[code]

sudo firecfg

creates links for all the applications that can be firejailed, and allows launching them from launch menu without any change; But the use of Firejail should be tested for each application: application profile might not work correctly, and I don't advise to use firecfg.

Once firecfg has been set, it can be unset with:

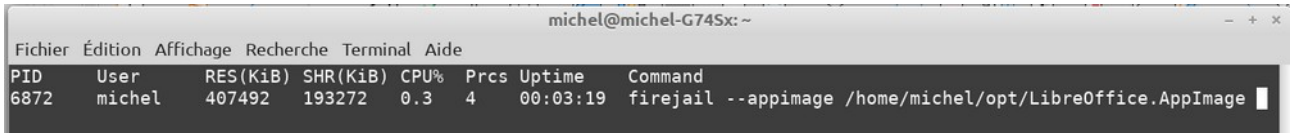
[code]

sudo firecfg --clean

Please read Firejail complete documentation, https://firejail.wordpress.com/documentation-2/.

Since Firejail uses AppArmor, its security is heavily degraded on distributions where SELinux is enabled (RedHat, Fedora…). Firejail mainly targets Ubuntu and its derivatives, and Debian.

Launching in a terminal "firejail --top" before to launch any firejailed application allows seeing the list of those applications (here, LibreOffice AppImage):



Firejail has profiles for some virtual machines: Vmware Workstation Pro, VirtualBox, qemu.

Is it an efficient way to sandbox a virtual machine?

I use VMware Workstation Player (a successor to both Vmware Player and Vmware Worsktation, since 2015), installed with Vmware Workstation Pro, and I have Firejail sandboxing.

- 1st test, Firejail has no profile for the executable "vmplayer", but one for the older "vmware-player"; I copy "/etc/firejail/vmware-player.profile" to "~/.config/firejail/vmplayer.profile" and in a terminal I execute the command "firejail vmplayer"; but "vmplayer" fails to launch correctly.

- 2nd test, Firejail includes a "noprofile.profile" used to test an application compatibility with Firejail; in a terminal I execute the command "firejail --profile=noprofile vmplayer"; this time, "vmplayer" is launched correctly, I can run a Windows 10 guest. → VMware Workstation Player is compatible with Firejail.

Is it worth the time to write and test a "vmplayer.profile"?

- When VMware Workstation Player is installed, 2 modules are compiled in the kernel: "vmmon" and "vmnet".

- At system startup:

* 8 processes are launched: "vmnet-bridge", "vmnet-dhcpd" (two instances), "vmnet-natd", "vmnet-netifup" (two instances), "vmware-authdlauncher", "vmware-usbarbitrator"; "systemd-analyze security" finds two services: "vmware-USBArbitrator.service" and "vmware.service".

- When VMware Workstation Player is launched, the process "vmplayer" runs.

- When a guest runs, there are three more processes running: "vmware-vmx", and two invisible ones "mks", which writes logs in the virtual machines directories, and "vmmon" .

In fact, the following three processes, "Virtual Machine Executable (VMX) process", "Virtual Machine Monitor (VMM) process" and "Mouse Keyboard Screen (MKS) process" make a group acting as a liaison between virtual machines and the physical hardware that supports them.

→ But the only firejailed executable is "vmplayer"; none of the twelve other processes or services can be firejailed.

An analysis of the security advisories published by VMware from 2020 to 2023 shows that eight critical or important advisories concerned VMware Workstation Player for Linux. In seven cases, "vmware-vmx" was involved, and in one case it was "vmnet-dhcpd".

→ The process "vmplayer" can be firejailed but was not concerned by security advisories; the processes "vmware-vmx" and "vmnet-dhcpd" cannot be firejailed, but were concerned by security advisories.

Firejail is NOT suitable to sandbox such a complex software as a virtual machine program.

Firejail has profiles for browsers, including Firefox.

- I download Firefox from Mozilla website, uncompress it and copy it to my "~/opt" directory.

- In a terminal, I execute the command "firejail ~opt/firefox/firefox". Firefox launch fails with "Error: no suitable ~/opt/firefox/firefox executable found" (though, of course, "~/opt/firefox/firefox" exists and is the Firefox executable).

→ The existing "/etc/firejail/firefox.profile" does not work with Firefox installed in any other place than the one expected when using a deb.

Firejail is NOT suitable to sandbox Firefox as downloaded from its website, since it is not installed in the same directories as the deb version.

Finally, what is Firejail use?

Firejail should be used only with simple applications, with a single running process, when no more secure solution (as flatpak or snap) is available.

Here is an example, a specific Firejail profile written for z-library application. Z-library access constantly changes; you can install "zlibrary-setup-latest.deb" to have the z-library application (a complete application, including its own browser, self-looking for available z-library websites and using Tor network for anonymity). Once installed, you can use Firejail to sandbox it with the following custom profile:

```
[code]
# Custom profile for /opt/Z-Library/z-library
# file system
include /etc/firejail/disable-common.inc
whitelist ${DOWNLOADS}
include /etc/firejail/whitelist-common.inc
private-tmp
private-dev
blacklist /mnt
blacklist /media
# network
```

multimedia

nosound

no3d

nodvd

novideo

notv

kernel

seccomp

nonewprivs

caps.drop all

noroot

apparmor

Copy this code, and save it in "~/.config/firejail/z-library.profile" and launch z-library with "firejail /opt/Z-Library/z-library %U" command: z-library is now sandboxed by Firejail.

Some comments on the code.

File system:

The profile uses all the standards Firejail exclusions and authorizations; the read/write access to Downloads is allowed (and any other directory of your home cannot be accessed); access to "/tmp" and "/dev" is prohibited, temporary private directories are created for that; "/mnt" and "/media" access is prohibited.

Network:

No restriction below "# network", full access to network is allowed.

Multimedia:

Multimedia access is prohibited.

Kernel:

"seccomp" is disabled (commented), since it would block all system calls and prevent z-library to function; "seccomp" should be used with exceptions allowing the system calls used by the application, it is a long process, explained in Firejail Seccomp Guide:

https://firejail.wordpress.com/documentation-2/seccomp-guide/comment-page-1/.

"nonewprivs" prevents any child process of the application to get higher privileges than the ones given to the application by Firejail.

"caps.drop all" drops all capabilities for the processes running in the sandbox. This option is recommended for running GUI programs or any other program that doesn't require root

privileges. It is a must-have option for sandboxing programs installed from unofficial sources - such as games, Java programs, etc.

"noroot" installs a user namespace with a single user - the current user; root user does not exist in the new namespace.

"apparmor" allows using firejail own AppArmor profile to run the application (see below).

Such a specific profile can be generated using Firejail Configuration Wizard ("firejail-ui" command).

4) AppArmor

AppArmor is installed by default in Linux Mint/Ubuntu operating systems. The default installation comes with few AppArmor profiles. Extra profiles can be added by installing two packages:

[code]

sudo apt install apparmor-profiles apparmor-profiles-extra

The AppArmor profiles installed that way are community maintained and are in complain mode: they work, with some exceptions allowed. Once user has tested the profiles do work, he can enforce them.

The installation of "apparmor-utils" package is necessary to be able to enforce an apparmor profile with "aa-enforce" command:

[code]

sudo apt install apparmor-utils

The following command:

[code]

sudo aa-status

shows the list of available AppArmor profiles, the active ones, and their mode (complain or enforced).

Ubuntu includes an AppArmor profile for some of its deb packages, see:

https://wiki.ubuntu.com/SecurityTeam/KnowledgeBase/AppArmorProfiles.

More documentation on AppArmor:

https://ubuntu.com/server/docs/security-apparmor

Firejail supports direct integration with AppArmor through a generic AppArmor profile. During installation, the profile, "firejail-default", is placed in "/etc/apparmor.d" directory.

This profile is automatically installed and loaded when you install a Firejail apparmor package from Firejail website (example: "firejail_0.9.72-apparmor_1_amd64.deb").

Local customizations of the apparmor profile are supported by editing the file:

"/etc/apparmor.d/local/firejail-local".

AppArmor is enabled for many Firejail profiles. There are several ways to enable AppArmor confinement on top of a Firejail security profile:

* AppArmor is automatically enabled for Firejail profiles with "apparmor" written in the profile. Example: I use LibreOffice AppImage. When I launch it with Firejail:

[code]

firejail --appimage ~/opt/LibreOffice.AppImage

the "sudo aa-status" output shows LibreOffice is apparmored (oosplash and soffice.bin):

```
6 processes are in enforce mode.
   /usr/sbin/cups-browsed (1710)
   /usr/sbin/cupsd (1004)
   /usr/sbin/ntpd (1002)
   /run/firejail/appimage/opt/libreoffice7.4/program/oosplash (7048) firejail-default//&unconfined
   /run/firejail/appimage/opt/libreoffice7.4/program/soffice.bin (7082) firejail-default//&unconfined
   /usr/bin/tor (1746) system_tor
2 processes are in complain mode.
   /usr/sbin/avahi-daemon (825) avahi-daemon
   /usr/sbin/avahi-daemon (887) avahi-daemon
0 processes are unconfined but have a profile defined.
```

* Add the "--apparmor" option in a custom profile.

* Enable Apparmor globally in "/etc/firejail/globals.local" and disable as needed through the use of "ignore apparmor" option in "/etc/firejail/ProgramName.local" or in "~/.config/firejail" specific profile.

Note that you should avoid conflicts for applications having an AppArmor profile, and a Firejail profile with "apparmor" option. If you rather want to use a specific, non Firejail, AppArmor profile for an application, you have to use the above-mentioned "ignore apparmor" option. However, that is not recommended, as using both Firejail and AppArmor for the same applications often creates problems.

Since apparmored applications still use system libraries, operating system and applications are not isolated, and the security level reached with AppArmor is lower than what you get with flatpaks or snaps.

5) Systemd sandboxing

Services (or daemons, synonymous), are launched with "systemd". By default, they are launched in root mode, with the maximum permissions, and that can be potentially dangerous.

The following command:

[code]

systemd-analyze security

lists all the services with their security level:

```
UNIT                                          EXPOSURE PREDICATE HAPPY
NetworkManager.service                             7.8 EXPOSED   😕
acpid.service                                      9.6 UNSAFE    😨
alsa-state.service                                 9.6 UNSAFE    😨
anacron.service                                    9.6 UNSAFE    😨
avahi-daemon.service                               9.6 UNSAFE    😨
cockpit-wsinstance-http-redirect.service           9.2 UNSAFE    😨
cockpit-wsinstance-http.service                    9.2 UNSAFE    😨
cockpit.service                                    9.2 UNSAFE    😨
colord.service                                     8.8 EXPOSED   😕
cron.service                                       9.6 UNSAFE    😨
cups-browsed.service                               9.6 UNSAFE    😨
cups.service                                       9.6 UNSAFE    😨
dbus.service                                       9.6 UNSAFE    😨
dm-event.service                                   9.5 UNSAFE    😨
dmesg.service                                      9.6 UNSAFE    😨
emergency.service                                  9.5 UNSAFE    😨
getty@tty1.service                                 9.6 UNSAFE    😨
getty@tty7.service                                 9.6 UNSAFE    😨
hddtemp.service                                    9.6 UNSAFE    😨
irqbalance.service                                 6.1 MEDIUM    😐
lightdm.service                                    9.6 UNSAFE    😨
lvm2-lvmpolld.service                              9.5 UNSAFE    😨
lines 1-23
```

You can get more information about a service:

[code]

systemd-analyze security NetworkManager.service

```
  NAME                                                    DESCRIPTION                                                >
✗ PrivateNetwork=                                         Service has access to the host's network                   >
✗ User=/DynamicUser=                                      Service runs as root user                                  >
✗ CapabilityBoundingSet=~CAP_SET(UID|GID|PCAP)            Service may change UID/GID identities/capabilities         >
✓ CapabilityBoundingSet=~CAP_SYS_ADMIN                    Service has no administrator privileges                    >
✓ CapabilityBoundingSet=~CAP_SYS_PTRACE                   Service has no ptrace() debugging abilities                >
✗ RestrictAddressFamilies=~AF_(INET|INET6)                Service may allocate Internet sockets                      >
✗ RestrictNamespaces=~CLONE_NEWUSER                       Service may create user namespaces                         >
✗ RestrictAddressFamilies=~…                              Service may allocate exotic sockets                        >
✓ CapabilityBoundingSet=~CAP_(CHOWN|FSETID|SETFCAP)       Service cannot change file ownership/access mode/c         >
✗ CapabilityBoundingSet=~CAP_(DAC_*|FOWNER|IPC_OWNER)     Service may override UNIX file/IPC permission chec         >
✗ CapabilityBoundingSet=~CAP_NET_ADMIN                    Service has network configuration privileges               >
✓ CapabilityBoundingSet=~CAP_RAWIO                        Service has no raw I/O access                              >
✗ CapabilityBoundingSet=~CAP_SYS_MODULE                   Service may load kernel modules                            >
✓ CapabilityBoundingSet=~CAP_SYS_TIME                     Service processes cannot change the system clock           >
✗ DeviceAllow=                                            Service has no device ACL                                  >
✗ IPAddressDeny=                                          Service does not define an IP address whitelist            >
✓ KeyringMode=                                            Service doesn't share key material with other serv         >
✗ NoNewPrivileges=                                        Service processes may acquire new privileges               >
✓ NotifyAccess=                                           Service child processes cannot alter service state         >
✗ PrivateDevices=                                         Service potentially has access to hardware devices         >
✓ PrivateMounts=                                          Service cannot install system mounts                       >
✗ PrivateTmp=                                             Service has access to other software's temporary f         >
lines 1-23
```

Note that "systemd-analyze security" displays the security level as known by "systemd", and does not take into account AppArmor protection.

Services that do need to be launched as root can have their permissions restricted to the minimum with "systemd" sandboxing.

Services that do not need to be launched as root can be launched as a user, and will have the user privileges. If they need some root capability, they can borrow it using "AmbientCapabilities" option.

An example of "systemd" sandboxing, "How To Sandbox Processes With Systemd On Ubuntu 20.04":

https://www.digitalocean.com/community/tutorials/how-to-sandbox-processes-with-systemd-on-ubuntu-20-04.

Documentation: "man systemd.exec", "man capabilities".

Since "systemd" sandboxed services still use system libraries, operating system and services are not isolated, and the security level reached with "systemd" sandboxing is lower than what you get with snaps (services can be sandboxed as snaps).

4.7) Safe browsing

This is a protection against DSA1, DSA3, DSA4 and PT2.

Safe browsing is, at first, a user care. However, it can be helped by automatic dispositions.

1) Google Safe Browsing API

Google maintains a list of websites that should be avoided, and sells it through an API. All major browsers (Chromium and derivatives, Firefox and derivatives) use that API and provide some safe browsing.

[Note that LibreWolf, being a community project, can't afford a full API license and so offers a limited safe browsing; Ungoogled-Chromium, by choice of its developers, has no communication at all with Google servers and so doesn't offer Google Safe Browsing; Tor Browser and Mullvad Browser, for privacy respect reasons, do not use Google Safe Browsing API. In those cases, the use of a filter extension such as uBlock Origin with malware filter list is mandatory].

2) Filters

It can be completed by filters. Your hosts file can be used as an IP addresses filter, for all your system connections; however, since it cannot be disabled when browsing, I recommend using it with malware oriented filters, and to complete it in your browser with a filter extension such as uBlock Origin.

- Use your "/etc/hosts" as an IP addresses filter:

* Copy your "/etc/hosts" to your home "~/" and rename it "hosts_base.txt".

* Create an "update_hosts.sh" bash file. Launch Xed, and copy/paste the following content to the empty Xed window:

[code]

#!/bin/bash

anti malware and anti spam hosts file writer/updater

filters any connection

sudo mv /etc/hosts /etc/hosts.bak

malware lists download

```
wget "https://raw.githubusercontent.com/davidonzo/Threat-Intel/master/lists/latestdomains.piHole.txt" -O hosts1.txt

wget "https://urlhaus.abuse.ch/downloads/hostfile/" -O hosts2.txt

wget "https://malware-filter.gitlab.io/malware-filter/urlhaus-filter-hosts.txt" -O hosts3.txt

# spam list download

wget "https://raw.githubusercontent.com/FadeMind/hosts.extras/master/add.Spam/hosts" -O hosts4.txt

# no coin lists download

wget "https://raw.githubusercontent.com/hoshsadiq/adblock-nocoin-list/master/hosts.txt" -O hosts5.txt

wget "https://raw.githubusercontent.com/greatis/Anti-WebMiner/master/hosts" -O hosts6.txt

# merging lists

cat hosts_base.txt hosts1.txt hosts2.txt hosts3.txt hosts4.txt hosts5.txt hosts6.txt > hosts

# changing hosts ownership

sudo chown root:root hosts

# moving created hosts

sudo mv hosts /etc/hosts

# cleaning

rm hosts1.txt

rm hosts2.txt

rm hosts3.txt

rm hosts4.txt

rm hosts5.txt

rm hosts6.txt

# DNS cache purge - Uncomment the command line corresponding to your system

# Linux Mint 20.x and former

#sudo systemd-resolve --flush-caches

# Linux Mint 21.x and later

#sudo resolvectl flush-caches

read -s -n1 -p "Hit any key to continue..."; echo
```

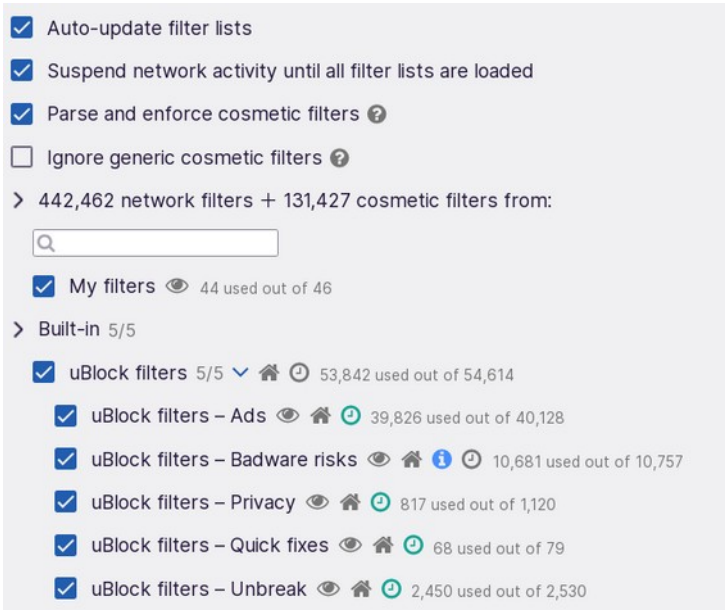
Save this file to your home "~/update_hosts.sh". Execute it to create your "/etc/hosts" filter, and to update it (once a week, or once every two weeks).

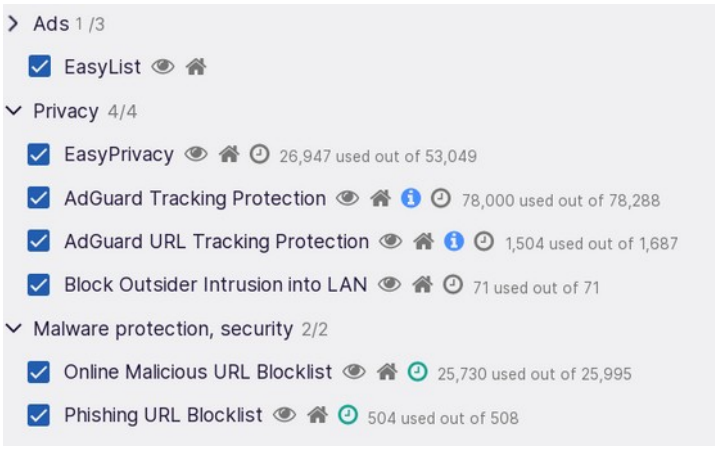
You can also use Savage Blocker, https://github.com/100savage/Savage-Blocker, to manage your hosts file.

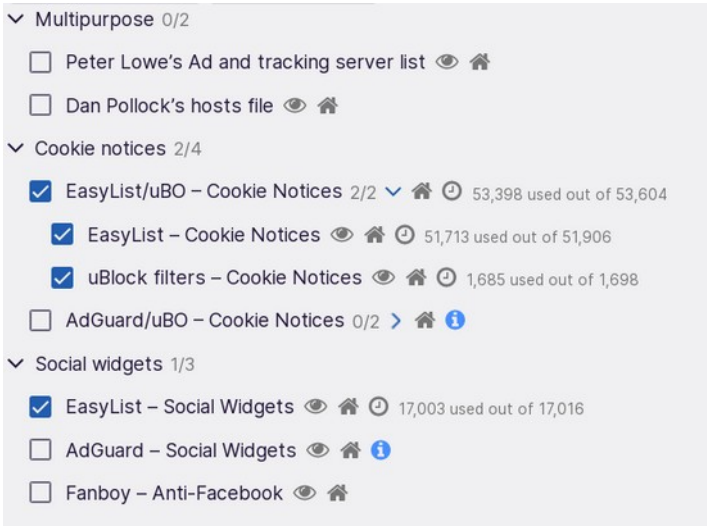
See also filtering DNS servers in §2 "Do not use your ISP DNS servers" of Reduce what your ISP can know.

- Install "uBlock Origin" extension in your browser and set its parameters.

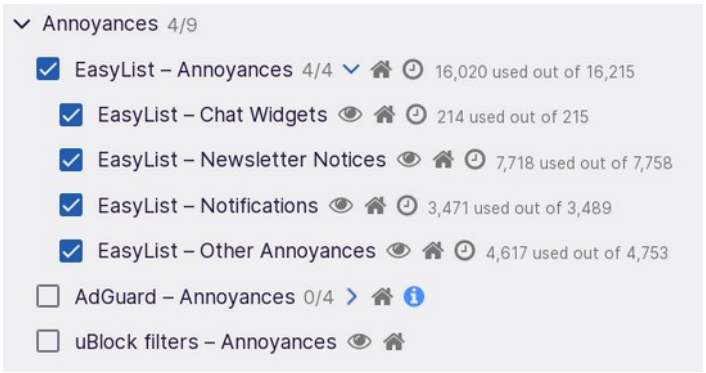
On the Settings tab, you can keep default settings. On the Filters tab:





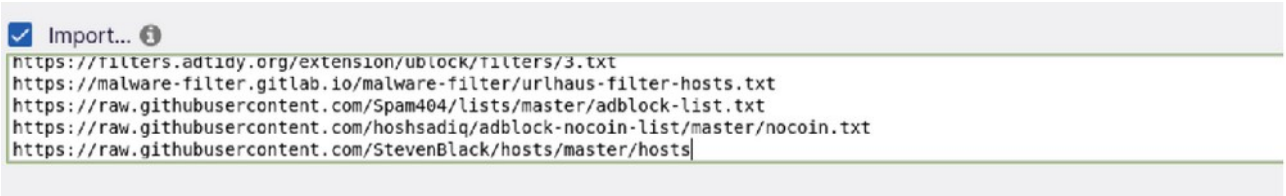


Don't select anything in the "Multipurpose" category, this will be treated in the "Custom" one.



In the "Regions, languages" category, select one for your language if it is not English.

On the "Custom" category, tick the case; this opens a window where you will copy/paste the following lists:



[code]

https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts

https://raw.githubusercontent.com/stamparm/blackbook/master/blackbook.txt

https://raw.githubusercontent.com/Spam404/lists/master/adblock-list.txt

https://raw.githubusercontent.com/hagezi/dns-blocklists/refs/heads/main/adblock/tif.mini.txt

If you intend to use your browser with a VPN or with Tor Network, your hosts file will not be used; so, add the following lines:

[code]

https://raw.githubusercontent.com/davidonzo/Threat-Intel/master/lists/latestdomains.piHole.txt

https://urlhaus.abuse.ch/downloads/hostfile/

https://malware-filter.gitlab.io/malware-filter/urlhaus-filter-hosts.txt

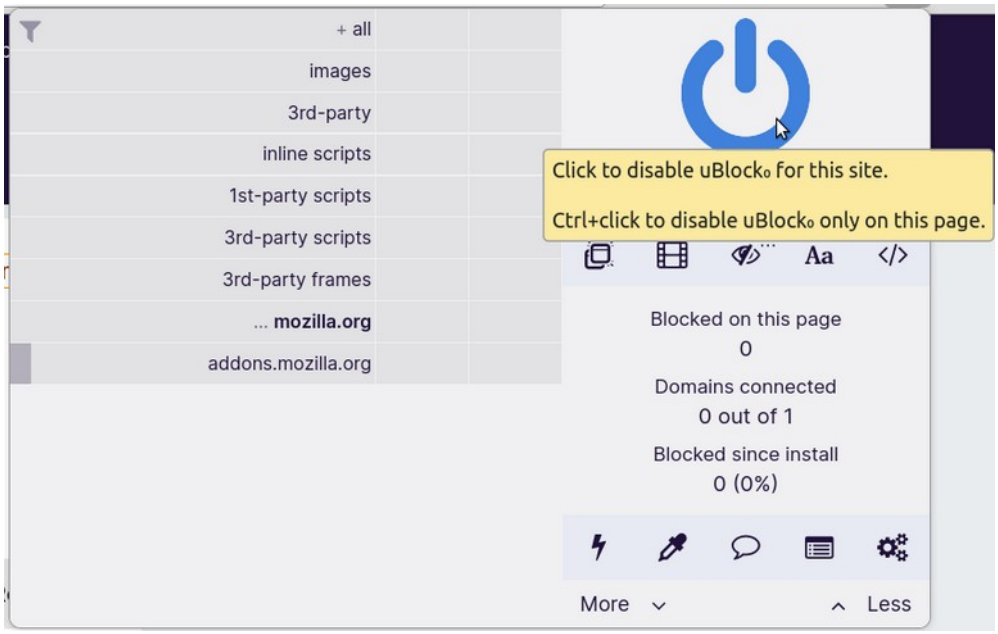
https://raw.githubusercontent.com/FadeMind/hosts.extras/master/add.Spam/hosts

https://raw.githubusercontent.com/hoshsadiq/adblock-nocoin-list/master/hosts.txt

https://raw.githubusercontent.com/greatis/Anti-WebMiner/master/hosts

Then click on "Apply changes" button and that's done, your filters are configured.

When browsing internet, don't forget to add in uBlock Origin the websites where uBlock Origin should not apply the filters (trusted websites, or websites not functioning with the filters).



It may arrive that uBlock Origin displays a message saying that a web page cannot be loaded because of a filter; you can then temporarily disable the filter.

Once you have set your hosts file, and set uBlock Origin in your browser, you are now protected by filters against spam, adware, malware, cryptocurrencies miners and tracking.

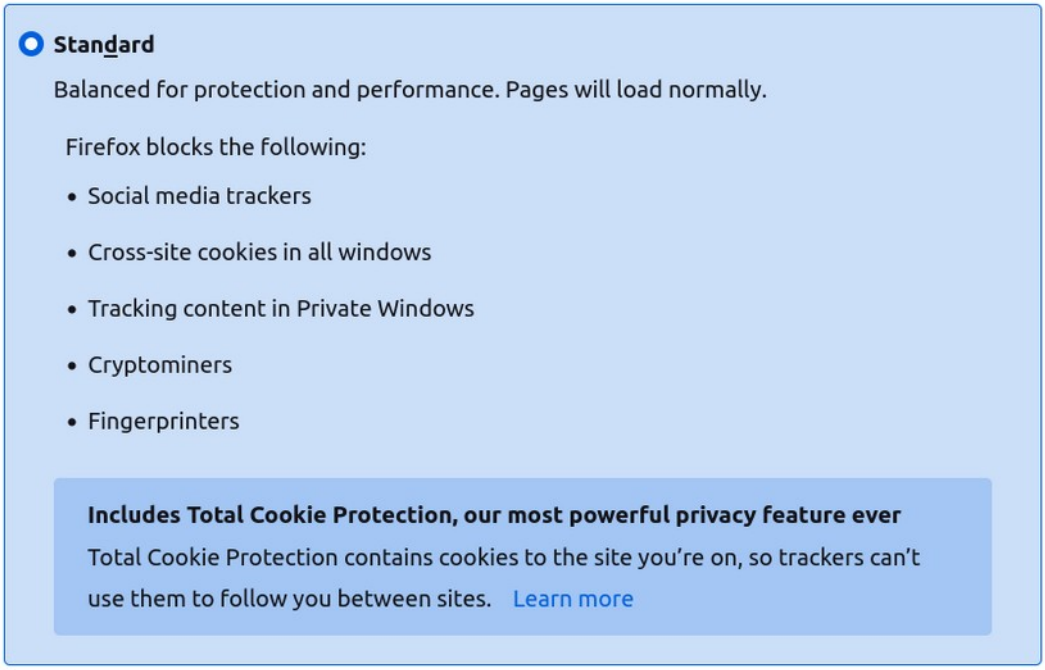
You can also use uBlock Origin as an extension in Thunderbird, with the same settings, if you use feeds.

NB1: Google has decided to drop V2 extensions in Chromium family browsers; uBlock Origin does not work any longer in Chrome, Chromium, Edge etc. browsers; it still works in Firefox. In Chromium browsers family, use uBlock Origin Lite extension, still efficient but with less tailoring.

NB2: When you add filters in your hosts file, or when you choose a filtering DNS server, you filter all your system connections, but you cannot remove the filters for a given website; at the opposite, when you use filters in your browser with uBlock Origin or uBlock Origin Lite, you filter only your

browser connections, but you can remove the filters for a given website. So, you should balance the filters you add at the system level or in your browser: my choice is to use malware filters at the system levels, and to complete them by ads and tracking filters in the browser.

Finally, Firefox offers default privacy settings, that you can adjust to your needs and that participate to safe browsing:



3) Browsers extensions

You can complete anti-tracking by adding these browsers extensions:

- "Privacy Badger", available for Firefox-based and Chromium-based browsers, blocking usual cookies tracking;

- "Decentraleyes", available for Firefox-based and Chromium-based browsers, protecting against tracking linked to centralized "free" content distributors;

- "ClearURLs", available for Firefox-based and Chromium-based browsers, this extension will automatically remove tracking elements from URLs to help protect your privacy when browse through the Internet.

- "Facebook Container", available for Firefox-based browsers, preventing Facebook to follow you everywhere on internet.

4) Fingerprinting protection

With filters in "/etc/hosts", uBlock Origin, Privacy Badger, Decentraleyes, ClearURLs and Facebook container, you have a strong protection against traditional cookies tracking.

However, the browser characteristics can be used to track, in a complement of cookies tracking: this is called fingerprinting tracking.

Here is how to be protected against this tracking:

- A set of two extensions, "CanvasBlocker" and "Font Fingerprint Defender", available for Firefox and Chromium families browsers, report fake, random, changing values and prevent corresponding fingerprinting.

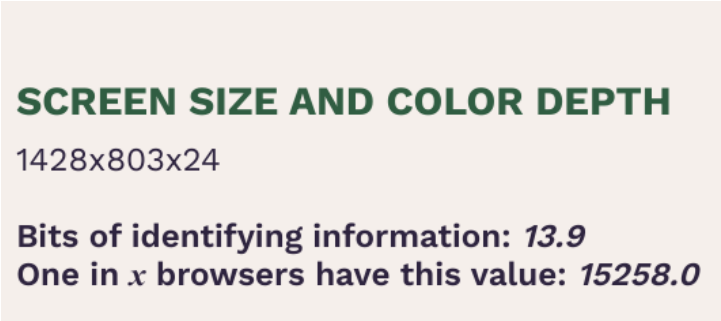
- Screen size, as reported by the browser, is also used for fingerprinting; when the monitor dpi is lower than 100, browsers will display full monitor size; when the monitor dpi is higher than 100, browsers need to adapt screen size to have a readable display.

In Firefox, "about:config", look for "layout.css.devPixelsPerPx"; change default value for a positive value; my laptop has a 17.4" screen, with 1920x1080 resolution and 129 dots per inch (dpi); with a value of 0.9 for "layout.css.devPixelsPerPx", the screen size (as checked with https://browsersize.com/) is 1600x900.

Mullvad Browser, based on Firefox, can use letter boxing; my computer has a screen size of 1400x600 once "privacy.resistfingerprinting.letterboxing.didforcesize" is set to "true" in "about:config". You can combine it with "layout.css.devPixelsPerPx".

A rounded value is rather good for screen size fingerprinting, since it increases the number of computers with the same value.

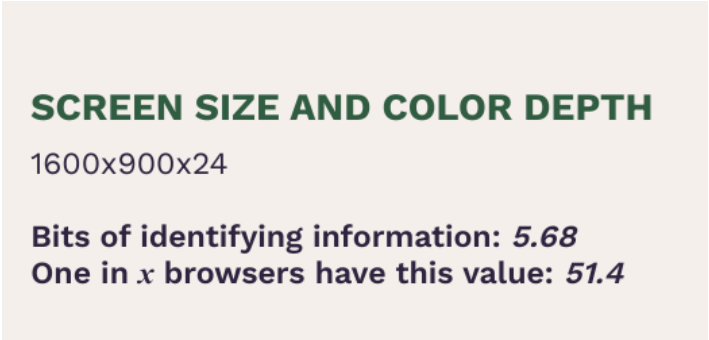
On the same monitor, Chromium displays a screen size of 1428x803; as per EFF Cover Your Tracks, https://coveryourtracks.eff.org/, this is a rather unique value:



If I launch Chromium with "--high-dpi-support=1" option, Chromium will use the full screen size, here 1920x1080. It is a good value for fingerprinting protection, but tabs and bookmarks police is very small, almost unreadable.

With "--high-dpi-support=1 --force-device-scale-factor=1.2", Chromium will use a screen size of 1600x900 (since 1920÷1.2=1600 and 1080÷1.2=900).

This time, tabs and bookmarks police is small but readable; I can adjust zoom page with "Zoom Page WE" extension (but this has no effect on tabs and bookmarks police size). The 1600x900 value is more common than 1428x803:



So, you can reduce fingerprinting tracking, but not completely remove it; browser headers, plugins, language, use of an ad-blocker, subscribed lists etc. can be used for fingerprinting tracking.

5) Web RTC leak

The WebRTC Leak is critical for anyone using a VPN or a browser on Tor Network, as it leverages the WebRTC API to communicate with a STUN server and potentially reveal the user's real local and public IP addresses, even when using a VPN, proxy server, or behind a NAT.

To disable WebRTC in Firefox-based browsers:

* Type "about:config" in the address bar and press Enter.

* In the search bar, type "media.peerconnection.enabled" and double-click the preference to set its value to false.

WebRTC in Chromium-based browsers:

The recommended solution to limit the risk of IP leakage via WebRTC is to use the official Google extension called "WebRTC Network Limiter", which provides a range of options with varying levels of protection.

6) Browser choice

Finally, the choice of the browser itself impacts your privacy and tracking:

- Firefox browser sends some data to Mozilla, but it can be disabled in Firefox settings. Some users prefer LibreWolf because it does not send any data to Mozilla.

- Google Chrome browser has links with Google servers; it is still the case with Open Source Chromium browser, since Chromium project is piloted by Google; in Ungoogled-Chromium, all the internal links pointing to Google servers are removed, and it does not send any data to Google. Finally, Microsoft Edge, based on Chromium, has links with Bing and Microsoft servers.

Firefox users can read Annex 11: Enhancing Firefox Security and Privacy for detailed instructions.

7) Protection against JavaScript and CSS attacks

JavaScript and CSS can be used as vectors attacks:

- NoScript Security Suite, available for Firefox-based browsers, or NoScript, available for Chromium-based browsers, allow to selectively enable JavaScript on websites you trust. It allows JavaScript, and other executable content (such as script, object, media, frame, font, webgl, fetch, ping, noscript, unrestricted CSS, rendering of plain HTML frames...) to run only from trusted

domains of your choice (e.g. your banking site), thus mitigating remotely exploitable vulnerabilities, such as Spectre and Meltdown.

It protects your "trust boundaries" against cross-site scripting attacks (XSS), cross-zone DNS rebinding / CSRF attacks (router hacking), and Clickjacking attempts, thanks to its unique ClearClick technology. Moreover, without JavaScript, websites cannot gather much information about your browser and computer, and NoScript improves your privacy.

- CSS Exfil Protection extension, available for Firefox family and for Chromium family browsers protect against the "CSS data exfiltration attack", an attack that might gather, as an example, your username and password when connecting to a website. Though this attack has been described in 2020, browsers have not yet fixed against this attack in 2024! A CSS Exfil vulnerability tester is available here: https://www.mike-gualtieri.com/css-exfil-vulnerability-tester.

- Just in time compilation, JIT, increases JavaScript attack surface. Disabling it will reduce the attack surface and increase security, JavaScript will be interpreted and not compiled, and web pages making a large use of JavaScript may appear slower.

JIT can be easily disabled in Firefox: in "about:config", set "javascript.options.baselinejit" to false.

In the Chromium family browsers, Microsoft Edge is the only one allowing to disable JIT; Microsoft Edge allows having a "normal" and "strict" security settings. With those two settings, Edge disables JIT, for the websites you don't visit often ("normal" security settings) or for all websites ("strict" security settings). Moreover, when JIT is not used, Microsoft Edge can use some extra protection features that don't work with JIT.

- Firefox and Thunderbird PDF readers enable scripting by default; scripting is used to enhance PDF reader capability, but can also be used to perform an attack with a malformed PDF document. You can disable scripting in PDF readers, with "about:config" in Firefox and Advanced Configuration in Thunderbird. In both cases, change the value of "pdfjs.enableScripting" from true to false.

8) Testing

You can test your browser at EFF Cover Your Tracks, https://coveryourtracks.eff.org/; do not consider the absolute values (they may change in a day, since Cover Your Tracks keeps tests results for one week, and since it is not very widely used). Use it for before/after comparisons.

You can also test your browser at Browser Leaks, https://browserleaks.com/. When you test AudioContext Fingerprint, Canvas Fingerprint, Font Fingerprint and WebGL Fingerprint, refreshing the page will show you that random fingerprint values have changed, if you use the extensions mentioned in Fingerprinting protection.

4.8) Be careful with downloaded files or attachments

This is prevention against DSA4.

You have followed preceding advises: you update your system, you have enabled Ubuntu Pro, you use trusted sources, you use sandboxing for internet connecting applications and for applications used to open downloaded files, and you practice safe browsing.

However, when you download a file or a mail attachment and save it on your computer, you are at risk.

Of course, you use your brain; but you may have been convinced by social engineering, or by receiving a mail from a friend (with a forged sender mail address…), that the file you saved is secure and legit.

1) What are the complementary precautions to take?

- You should be very careful with uncompressed files: they may have executable permission; check it in their properties.

- The first time you open a file or an attachment, you should not double-click on it, but launch the corresponding application and open it.

- In LibreOffice options / security, you should set the macros security to its maximum level. It would block the macros execution in a LibreOffice malicious document.

- You should prevent some files in your home "~/" to be modified, because they are automatically executed at your system launch or logout:

> [code]
>
> sudo chattr +i ~/.profile ~/.bashrc ~/.bash_aliases ~/.bash_logout ~/.bash_profile ~/.bash_login

That makes those files immutable; if you need to modify one, use "sudo chattr -i", then modify it and again "sudo chattr +i"

- You should enable all the thumbnailers in your file manager. Xreader is installed by default in Linux Mint, and it includes xreader-thumbnailer for documents such as PDF ones.

- You should use the "file" command to determine the file type.

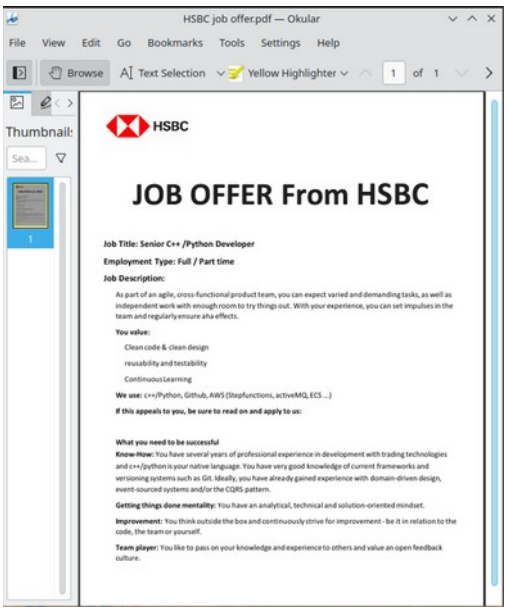
- You should test for viruses those files. Virus testing is normally a detection activity, but if it is done to avoid to execute a malicious program, it is prevention. See Malware and viruses detection.

2) Are those dispositions efficient?

A malware targets job-seekers (see "Operation DreamJob with a Linux payload" at:

https://www.welivesecurity.com/2023/04/20/linux-malware-strengthens-links-lazarus-3cx-supply-chain-attack/).

People receive a zip file, "HSBC_job_offer.pdf.zip" supposed to contain an HSBC job offer, "HSBC_job_offer.pdf". The "HSBC_job_offer.pdf" seems, at a first glance, legit. However, it is a fake PDF file: it is an executable malicious program, with a forged ".pdf" extension: the character used to display the dot is not the usual one, but the U+2024 Unicode character, resembling to the dot. When user double-clicks on the fake PDF, it launches the malicious program execution. The malicious program uses the default system PDF viewer and displays a decoy:



* The first suspicious thing is a PDF in a zip, that's not useful (PDF are compressed files) and not usual.

* Of course, the file has executable permission, and a PDF does not need to be executed.

* If you try to open the fake PDF with a document viewer (such as Okular flatpak), it will fail, while double-clicking on the fake PDF will launch the program and displays the decoy PDF document in your default PDF reader.

* The malicious program will attempt to change some files within: "~/.profile", "~/.bashrc", "~/.bash_aliases", "~/.bash_logout", "~/.bash_profile", "~/.bash_login", to launch a payload that could be executed automatically at system launch or logout. With immutable attribute, those files cannot be modified without superuser rights.

* In a computer with PDF thumbnailer enabled, as soon as a PDF file is saved, a thumbnail is displayed in the file manager, and allows making the difference between true or fake PDF.

Here is the thumbnail for the document you are reading:

If I copy it and rename it in "Linux_Mint_Security.txt", then delete thumbnails cache, the thumbnail does not change and shows that "Linux_Mint_Security.txt" is a document and not a text file:

Now, here is the normal thumbnail of a binary executable:

If I copy it and rename it in "SmillaEnlarger.pdf", the PDF thumbnailer cannot read the content of the fake PDF file and just displays a PDF icon :

* The "file" command can give the file type; example:

[code]

file Linux_Mint_Security.pdf

[output]

Linux_Mint_Security.pdf: PDF document, version 1.6

→ "PDF document" shows it is a true PDF.

With an executable file, the output is different:

[code]

file SmillaEnlarger

[output]

SmillaEnlarger: ELF 64-bit LSB shared object, x86-64, version 1 (GNU/Linux), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=ac8a58434be842b2dea74693c06cf162d102768c, for GNU/Linux 3.2.0, not stripped

→ "ELF 64-bit LSB shared object" shows it is a Linux executable.

And, with a fake PDF:

[code]

file SmillaEnlarger.pdf

[output]

SmillaEnlarger.pdf: ELF 64-bit LSB shared object, x86-64, version 1 (GNU/Linux), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=ac8a58434be842b2dea74693c06cf162d102768c, for GNU/Linux 3.2.0, not stripped

→ "ELF 64-bit LSB shared object" shows it is a Linux executable and NOT a PDF document.

The use of thumbnailers or of "file" command allows making the difference between true PDF documents and fake ones.

* Finally, VirusTotal, combining over 70 antivirus scanners and being continuously updated, has the maximum probability to detect the fake PDF as a malicious program.

3) The specific case of SVG files

SVG (Scalable Vector Graphics) is a vectorial image format; but SVG files are written in xml code, and can contain the equivalent of an html page with scripts; at launch, they can look for resources, locally or on internet. They have recently become attack vectors. As an example, an application able to open and display/modify SGV files is Inkscape, and Inkscape can even run javascript!

A supplemental precaution is needed: use a sandboxed application to open SVG files; as an example, Inkscape in its flatpak version.

[code]

flatpak install org.inkscape.Inkscape

Using Flatseal, itself a flatpak application (com.github.tchx84.Flatseal), disable Inkscape network access, and strongly reduce its unattended write permissions (disable "filesystem=host" default permission and add "xdg-download" one: in unattended mode, Inkscape will be authorized to write files in your downloads directory only).

[NB: this is just a reminder of a more general recommendation, "The applications that need to be sandboxed are internet connecting ones (browsers, mail clients, ftp clients, servers, virtual machines, multimedia players, video capture applications, virtual machines etc.) and applications used to open internet downloaded files (LibreOffice, documents viewers, file

compression/uncompression utilities, images viewers/editors etc.)". See Sandbox your applications.]

4.9) Don't use Wine or Mono to run Windows programs

This is prevention against VST1.

Some Windows programs can be run in Linux Mint using Wine. Some Windows programs purely written in C# (or .Net) language, can be run in Linux Mint with Mono or .Net runtimes.

It is an insecure way to run Windows programs in Linux Mint: Windows and Linux processes are not separated, and Windows weaknesses could impact Linux Mint security.

There are three secure methods allowing to run Windows programs securely on a Linux Mint computer.

1) Don't use Wine, use flatpak Bottles

Bottles is an easy way to use Windows programs with Wine, a much easier way than to use Wine without Bottles:

- you can download preconfigured, ready to use versions of Wine within Bottles (software, games),

- you can have keep different Wine configurations or install all your Windows programs in the most complete one (games),

- you can have automatic Wine updates.

Flatpak version of Bottles, an official one, whose use is recommended by Bottles developers, offers a secure isolation between Windows and Linux processes, without any risk that a Windows weakness can be used to compromise your Linux Mint system.

Some useful links:

- Bottles website: https://usebottles.com/

- Bottles documentation: https://docs.usebottles.com/

- How To Run Windows Software On Linux With Bottles on OSTECHNIX:

https://ostechnix.com/run-windows-software-on-linux-with-bottles/

- Bottles on Flathub: https://flathub.org/apps/com.usebottles.bottles

See Flatpak and Annex 6: Flatpak Tutorial.

2) Run Windows in a virtual machine

You can run Windows in a virtual machine such as libvrt/qemu, VirtualBox or VMware Workstation Pro.

You will need to install the virtual machine software you choose on Linux Mint, then install Windows in the virtual machine (with an ISO and a license number), install VirtualBox Guest additions or VMware tools on Windows, and finally install your programs on Windows.

The main advantage of this solution is that Windows programs will work correctly, in a complete Windows environment.

There are caveats: you need to have enough resources on your computer (CPU cores, RAM, GPU RAM) to allocate them to the virtual machine; you need to pay for a Windows license (some OEM licenses can be bought online from 15 to 20 $).

Virtual machines offer a secure isolation between Windows processes, run on Windows guest inside the virtual machine, and Linux processes, run on Linux Mint host outside the virtual machine.

But virtual machine programs are not free of vulnerabilities. Malware programs running in virtual machines can detect them and adapt their behavior.

The most secure virtual machine is Gnome Boxes, a GUI frontal for libvrt/qemu/kvm, used as a Flatpak or a Snap.

Users preferring VMware Workstation Pro or Virtual Box should keep them up-to-date, since releases often contain security fixes.

Unneeded devices (such as CD/DVD player physical devices) should be removed from the virtual guest settings.

Finally, Windows guest should be secured in the same way as if running on a physical host (updates applied; firewall with blocked incoming connections; trusted sources; sandboxing; antivirus etc., see your Windows security settings). To avoid malware in Windows guest is the best way to prevent that malware targets the host through the virtual machine software.

3) Use multiboot

Of course, multiboot does not allow running Linux Mint and Windows programs at the same time, and can so be considered as an old, obsolete technology.

However, it allows having both Linux Mint and Windows operating systems installed on the same computer, and provides the best possible processes isolation (!).

It may also be the only possible solution when a program does not work with Wine and cannot work in a virtual machine because of some specific needs (Cuda, Vulkan…).

Some precautions should be taken when installing a LinuxMint/Windows multiboot, in order to avoid that Windows does not boot any longer after its first update.

See Annex 7: Multiboot.

4.10) Set your system security

This is prevention against VST2.

Linux Mint/Ubuntu are very permissive:

- user can choose a password with 5 characters only,

- user can keep his password for years without mandatory requirements to change it,

- user can start his computer and open a Linux Mint session without having to enter any password,

- […]

Of course, they allow having a simple computer user experience, but they also allow defective security parameters.

The way to stronger security is to perform a security audit, then change settings. It can be done using Lynis or Ubuntu Pro User Security Guide (USG).

1) Using Lynis

- Download the tar.gz from this page: https://cisofy.com/downloads/lynis/ by clicking on "Download" button. (Don't use the version available in Linux Mint/Ubuntu repositories, since it is obsolete).

- Uncompress the downloaded file, uncompress it and copy "lynis" directory where you want in your home (for me it is "~/opt/lynis"); open a terminal inside "lynis" directory, then launch the audit:

[code]

./lynis audit system

- In the terminal window, Lynis will display its audit results, with warnings in red.

- At the end, there will be a list of suggestions (I got 40…):

Here are some of the suggestions I got:

```
 * Consider hardening system services [BOOT-5264]
   - Details  : Run '/usr/bin/systemd-analyze security SERVICE' for each service
     https://cisofy.com/lynis/controls/BOOT-5264/
 * Check the output of ps for dead or zombie processes [PROC-3612]
     https://cisofy.com/lynis/controls/PROC-3612/
 * Run pwck manually and correct any errors in the password file [AUTH-9228]
     https://cisofy.com/lynis/controls/AUTH-9228/
 * Configure password hashing rounds in /etc/login.defs [AUTH-9230]
     https://cisofy.com/lynis/controls/AUTH-9230/
 * Install a PAM module for password strength testing like pam_cracklib or pam_passwdqc [AUTH-9262]
     https://cisofy.com/lynis/controls/AUTH-9262/
* Configure minimum password age in /etc/login.defs [AUTH-9286]
     https://cisofy.com/lynis/controls/AUTH-9286/
```

After each suggestion, there is a link; example, https://cisofy.com/lynis/controls/AUTH-9286/ points to a web page explaining how to set a password duration.

Once you have applied some or all of the suggestions, your computer security is better set, but your user experience may be more complicated: each applied suggestion adds restraints on user.

2) Using Ubuntu Pro User Security Guide

Once you have installed Ubuntu Pro, you need to enable Ubuntu Pro USG service:

> [code]
>
> sudo pro enable usg
>
> [output]
>
> One moment, checking your subscription first
>
> Updating package lists
>
> Ubuntu Security Guide enabled
>
> Visit https://ubuntu.com/security/certifications/docs/usg for the next steps

As said, visit https://ubuntu.com/security/certifications/docs/usg.

USG allows auditing your security settings, comparing them to security models proposed by CIS (Center for Internet Security), and automatically fixing them to be compliant with those models. It may be necessary for companies to be compliant with their customers requirements.

The CIS models taken into account by USG are "cis_level1_workstation", "cis_level1_server", "cis_level2_workstation" and "cis_level2_server".

Remember it will add lots of restraints and will change your computer user experience. Before to apply the changes, think to back up your system or to make a snapshot.

See Annex 5: How to Enable Ubuntu Pro on Linux Mint .

[To have USG work on Linux Mint you need to have your operating system fully recognized as Ubuntu LTS, and to perform the changes on "/etc/issue", "/etc/issue.net" and "/etc/lsb-release" mentioned in "8. Kernels management / Use kernel livepatch" of Annex 5].

4.11) Reduce what your ISP can know

This is prevention against PT1.

When you normally use internet, your internet service provider (ISP) knows all what you do on internet (DNS requests, accessed and read web-pages) and logs all. To have all that information centralized in one place is a strong threat on your privacy.

Here is how to reduce what you ISP can know about what you do on internet:

1) Enable HTTPS protocol on all the websites you visit

Your ISP will know where you go, since all your internet access is done through its gateway, but will not know what you have seen, since the web pages are sent encrypted to your browser.

This setting can be enabled on all modern browsers:

See "HTTPS-Only Mode in Firefox", https://support.mozilla.org/en-US/kb/https-only-prefs, to enable it on Firefox and its derived browsers.

See "How to Enable HTTPS Only Mode in Chrome", https://www.nirmaltv.com/2021/11/22/how-to-enable-https-only-mode-in-chrome/, to enable it on Chromium and its derived browsers.

Once set, browsers will ask for an HTTPS version of any web page. If the HTTPS version is not available, the browser will not load the web page, will warn you and propose you to load it in HTTP.

2) Do not use your ISP DNS servers

If your connection uses your default ISP settings (with automatic DNS choice), you use your ISP DNS server, with UDP unencrypted protocol for DNS queries and answers.

→ Your ISP knows all your DNS queries and answers.

It is preferable to use one of the following public servers; they are all compatible with DNS over HTTPS, DNS over TLS, and they conform to DNSSEC; you can use them either in your connection settings, or in one of the encrypted protocols, DNS over HTTPS and DNS over TLS.

[Note that, if you do not use an encrypted protocol, since all your internet connections are done through your ISP gateway, it will still know your DNS queries and answers, though they will not be logged in its DNS servers, but in the gateway logs].

Cloudflare:

Primary server IPV4 address: 1.1.1.1

Secondary server IPV4 address: 1.0.0.1

Primary server IPV6 address: 2606:4700:4700::1111

Secondary server IPV6 address: 2606:4700:4700::1001

DNS over HTTPS: https://cloudflare-dns.com/dns-query

DNS over TLS: cloudflare-dns.com

Cloudflare with malware block:

Primary server IPV4 address: 1.1.1.2

Secondary server IPV4 address: 1.0.0.2

Primary server IPV6 address: 2606:4700:4700::1112

Secondary server IPV6 address: 2606:4700:4700::1002

DNS over HTTPS: https://security.cloudflare-dns.com/dns-query

DNS over TLS: security.cloudflare-dns.com

Cloudflare with malware and adult content block:

Primary server IPV4 address: 1.1.1.3

Secondary server IPV4 address: 1.0.0.3

Primary server IPV6 address: 2606:4700:4700::1113

Secondary server IPV6 address: 2606:4700:4700::1003

DNS over HTTPS: https://family.cloudflare-dns.com/dns-query

DNS over TLS: family.cloudflare-dns.com

OpenDNS (Cisco owned):

Primary server IPV4 address: 208.67.222.222

Secondary server IPV4 address: 208.67.220.220

Primary server IPV6 address: 2620:119:35::35

Secondary server IPV6 address: 2620:119:53::53

DNS over HTTPS: https://doh.opendns.com/dns-query

DNS over TLS: opendns.com

OpenDNS Family Shield:

Primary server IPV4 address: 208.67.222.123

Secondary server IPV4 address: 208.67.220.123

Primary server IPV6 address: 2620:119:35::123

Secondary server IPV6 address: 2620:119:53::123

DNS over HTTPS: https://doh.familyshield.opendns.com/dns-query

DNS over TLS: familyshield.opendns.com

NB: Cisco has stopped OpenDNS activities in France, effective on June 28, 2024. See:

https://support.opendns.com/hc/en-us/articles/27951404269204-OpenDNS-Service-Not-Available-To-Users-In-France-and-Portugal (service has been reactivated in Portugal).

Google public DNS servers:

Primary server IPV4 address: 8.8.8.8

Secondary server IPV4 address: 8.8.4.4

Primary server IPV6 address: 2001:4860:4860::8888

Secondary server IPV6 address: 2001:4860:4860::8844

DNS over HTTPS: https://dns.google/dns-query

Quad 9:

Primary server IPV4 address: 9.9.9.9

Secondary server IPV4 address: 149.112.112.112

Primary server IPV6 address: 2620:fe::fe

Secondary server IPV6 address: 2620:fe::9

DNS over HTTPS: https://dns.quad9.net/dns-query

DNS over TLS: dns.quad9.net

Rethink configurable DNS resolver:

Rethink offers, for free at the moment, DNS resolver with filters that the user can configure.

There are two ways to do it:

* "advanced" configuration, https://rethinkdns.com/configure, you select the filters you want among more than 190 filters;

* "simple" configuration, from the former web page you click on "simple" button, and you select full categories of filters (Adult, Piracy, Gambling, Dating, Social Media, Security Full, Security Extra, Privacy Lite, Privacy Aggressive, Privacy Extreme);

In both cases, once your selection done, you get a DNS resolver name as DNS over TLS (DoT) or DNS over HTTPS (DoH).

Example: having selected from the simple configuration page "Security Full" and "Privacy Agressive", you get the following DNS resolver DoH name: "https://sky.rethinkdns.com/1:-B9_AP_9-P_Q3pNQEBkCQwEAACAAEA==".

NB: in Linux Mint you can use Rethink DNS resolvers only in browsers or mail clients in their DoH settings, since the resolvers are not available as IP addresses, see §3 "Use DNS over HTTPS protocol" of this chapter.

Mullvad DNS resolvers:

Mullvad has opened its DNS servers, used with Mullvad VPN, to everybody as a free service.

Hostnames and content blockers

The table below shows the different hostnames options and their content blockers. Refer to this when configuring the DNS with the instructions below.

Hostname	Ads	Trackers	Malware	Adult	Gambling	Social media
dns.mullvad.net						
adblock.dns.mullvad.net	✅	✅				
base.dns.mullvad.net	✅	✅	✅			
extended.dns.mullvad.net	✅	✅	✅			✅
family.dns.mullvad.net	✅	✅	✅	✅	✅	
all.dns.mullvad.net	✅	✅	✅	✅	✅	✅

IP-addresses and ports

The table below shows the corresponding IPV4 and IPV6 addresses.

Hostname	IPV4 address	IPV6 address	DoH port	DoT port
dns.mullvad.net	194.242.2.2	2a07:e340::2	443	853
adblock.dns.mullvad.net	194.242.2.3	2a07:e340::3	443	853
base.dns.mullvad.net	194.242.2.4	2a07:e340::4	443	853
extended.dns.mullvad.net	194.242.2.5	2a07:e340::5	443	853
family.dns.mullvad.net	194.242.2.6	2a07:e340::6	443	853
all.dns.mullvad.net	194.242.2.9	2a07:e340::9	443	853

DNS over HTTPS:

https://dns.mullvad.net/dns-query

https://adblock.dns.mullvad.net/dns-query

https://base.dns.mullvad.net/dns-query

https://extended.dns.mullvad.net/dns-query

https://family.dns.mullvad.net/dns-query

https://all.dns.mullvad.net/dns-query

Mullvad DNS resolvers only use encrypted DoH or DoT, and not unencrypted DNS over UDP/53. The filters lists used by Mullvad DNS resolvers are available here: https://github.com/mullvad/dns-blocklists.

More on Mullvad DNS resolvers here: https://mullvad.net/en/help/dns-over-https-and-dns-over-tls.

Recursive DNS Resolvers, sovereign and GDPR-compliant, for EU citizens:

There are two initiatives of sovereign, GDPR-compliant, recursive DNS Resolver:

- "sovereign": servers are inside EU, and there is at least one server in each of the 27 EU countries,

- "GDPR-compliant": offering the best privacy protection to its users, in accordance with EU General Data Protection Regulation, see https://gdpr-info.eu/,

- "DNSSEC compatible": DNS answers are signed by the servers.

* EU founded DNS4EU, see https://www.joindns4.eu/. This DNS exists in five flavors:

Hostname	Ads	Trackers	Malware	Adult
unfiltered.joindns4.eu				
protective.joindns4.eu			✅	
noads.joindns4.eu	✅	✅	✅	
child.joindns4.eu			✅	✅
child-noads.joindns4.eu	✅	✅	✅	✅

Non filtering DNS:

Primary IPV4 DNS server: 86.54.11.100

Secondary IPV4 DNS server: 86.54.11.200

Primary IPV6 DNS server: 2a13:1001::86:54:11:100

Secondary IPV6 DNS server: 2a13:1001::86:54:11:200

DNS over HTTPS: https://unfiltered.joindns4.eu/dns-query

DNS over TLS: unfiltered.joindns4.eu

Protective resolution DNS:

This DNS is a hardened one, security oriented.

Primary IPV4 DNS server: 86.54.11.1

Secondary IPV4 DNS server: 86.54.11.201

Primary IPV6 DNS server: 2a13:1001::86:54:11:1

Secondary IPV6 DNS server: 2a13:1001::86:54:11:201

DNS over HTTPS: https://protective.joindns4.eu/dns-query

DNS over TLS: protective.joindns4.eu

Protective resolution with ad-blocking:

Primary IPV4 DNS server: 86.54.11.13

Secondary IPV4 DNS server: 86.54.11.213

Primary IPV6 DNS server: 2a13:1001::86:54:11:13

Secondary IPV6 DNS server: 2a13:1001::86:54:11:213

DNS over HTTPS: https://noads.joindns4.eu/dns-query

DNS over TLS: noads.joindns4.eu

Protective resolution with child protection:

Primary IPV4 DNS server: 86.54.11.12

Secondary IPV4 DNS server: 86.54.11.212

Primary IPV6 DNS server: 2a13:1001::86:54:11:12

Secondary IPV6 DNS server: 2a13:1001::86:54:11:212

DNS over HTTPS: https://child.joindns4.eu/dns-query

DNS over TLS: child.joindns4.eu

Protective resolution with child protection & ad-blocking:

Primary IPV4 DNS server: 86.54.11.11

Secondary IPV4 DNS server: 86.54.11.211

Primary IPV6 DNS server: 2a13:1001::86:54:11:11

Secondary IPV6 DNS server: 2a13:1001::86:54:11:211

DNS over HTTPS: https:/child-noads.joindns4.eu/dns-query/

DNS over TLS: child-noads.joindns4.eu

* Privately founded, DNS0.EU. This DNS exists in three flavors:

Non filtering DNS:

See https://www.dns0.eu

Primary IPV4 DNS server: 193.110.81.0

Secondary IPV4 DNS server: 185.253.5.0

Primary IPV6 DNS server: 2a0f:fc80::

Secondary IPV6 DNS server: 2a0f:fc81::

DNS over HTTPS: https://dns0.eu

DNS over TLS: dns0.eu

Zero filtering DNS:

This DNS is a hardened one, security oriented, using both human and heuristics filtering, see details at https://www.dns0.eu/zero

Primary IPV4 DNS server: 193.110.81.9

Secondary IPV4 DNS server: 185.253.5.9

Primary IPV6 DNS server: 2a0f:fc80::9

Secondary IPV6 DNS server: 2a0f:fc81::9

DNS over HTTPS: https://zero.dns0.eu

DNS over TLS: zero.dns0.eu

Kids filtering DNS:

This DNS protects kids against adult content, see details at https://www.dns0.eu/kids

Primary IPV4 DNS server: 193.110.81.1

Secondary IPV4 DNS server: 185.253.5.1

Primary IPV6 DNS server: 2a0f:fc80::1

Secondary IPV6 DNS server: 2a0f:fc81::1

DNS over HTTPS: https://kids.dns0.eu

DNS over TLS: kids.dns0.eu

3) Use DNS over HTTPS protocol

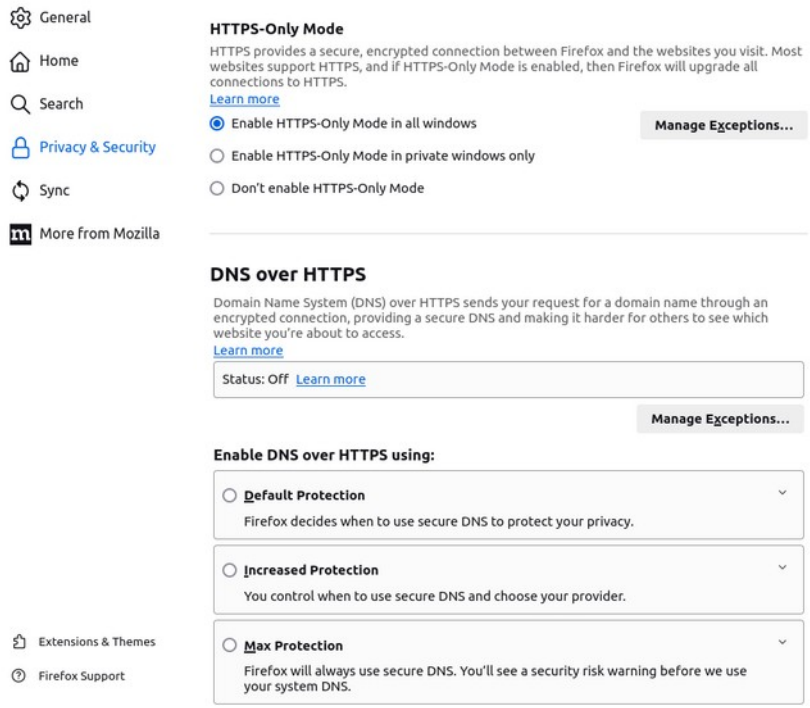
With DNS over HTTPS protocol, the DNS queries and the DNS answers are encrypted, and your ISP does not know the content of your DNS queries. But this is limited to browsers, and mail clients (available in Thunderbird).

The other DNS queries are still unencrypted and can be spied by your ISP. Of course, since browsers DNS requests represent most DNS traffic and are the most sensitive, this is an acceptable solution.

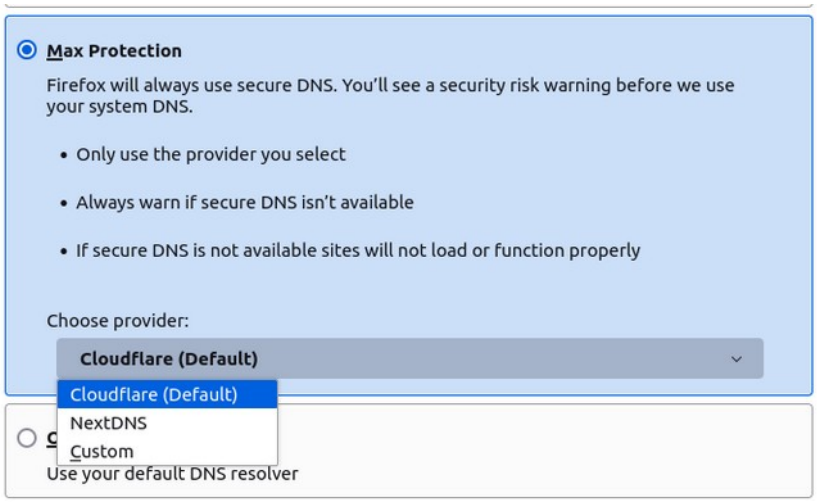
[In this mode, the browsers (Firefox and similar, Chromium and similar) and Thunderbird, once set, will directly send DNS requests to the chosen DNS server, using HTTPS mode. But other applications will still use UPD unencrypted requests sent to local server (set by Network Manager in "/etc/resolv.conf"), and "systemd-resolved" service will send unencrypted DNS requests to the DNS servers set in your connection settings].

DNS over HTTPS can be set in the browsers:

In Firefox, settings menu, privacy and security:



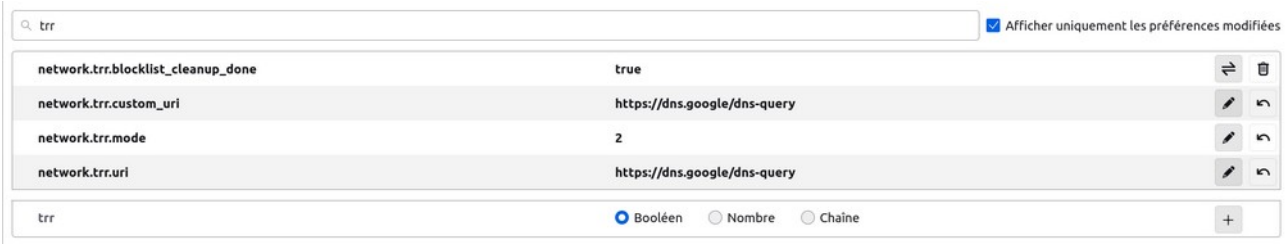
Choose your protection level (Default, Increased or Max), my advice is to use Max; then choose your DNS server (Cloudfare, NextDNS or Custom):



See "How to Enable DNS over HTTPS in Chrome",

https://www.bollyinside.com/articles/how-to-enable-dns-over-https-in-chrome/, to enable it on Chromium and its derived browsers.

For Thunderbird, open advanced configuration and change the following values:

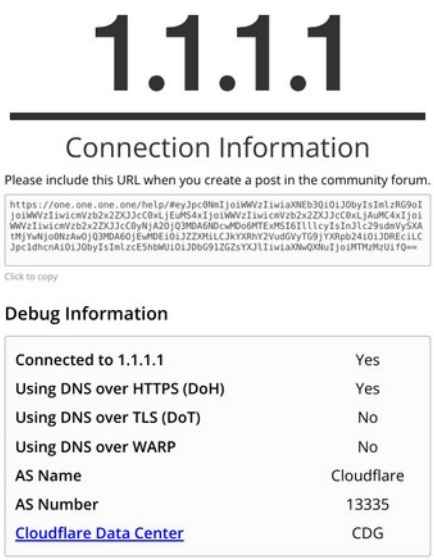


[You could also use FDNS, a Firejail DOH proxy server, to enable it at your operating system level, for all your DNS requests. See https://firejaildns.wordpress.com/ *for download and documentation].*

If you have set Cloudflare as DNS server for HTTPS mode, you can test it at:

https://one.one.one.one/help/

Here is the result:



If you use another public DNS server, you can still test that DNSSEC works:

(test with https://dnscheck.tools/):

```
IE-GOOGLE-2a00-1450-4000-1
    2a00:1450:4025:1801::101          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1801::103          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1801::105          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1803::101          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1803::102          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1803::104          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1803::105          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1805::101          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1805::102          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1805::103          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1805::104          ns: ns1.google.com          San Francisco, California, US
    2a00:1450:4025:1805::105          ns: ns1.google.com          San Francisco, California, US

Great! Your DNS responses are authenticated with DNSSEC:

    correct P-256 signature... connected
    invalid P-256 signature... not connected
    expired P-256 signature... not connected
    missing P-256 signature... not connected
    correct P-384 signature... connected
    invalid P-384 signature... not connected
    expired P-384 signature... not connected
    missing P-384 signature... not connected
    correct Ed25519 signature... connected
    invalid Ed25519 signature... not connected
    expired Ed25519 signature... not connected
    missing Ed25519 signature... not connected
```

4) Use DNS over TLS protocol

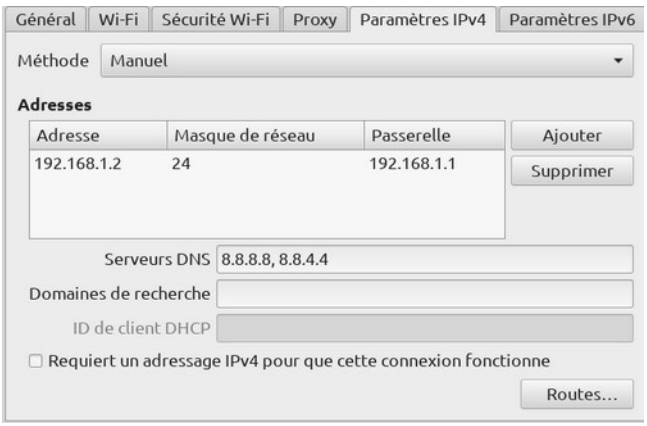
With this mode, all your connections will use a kind of encrypted tunnel between your computer and the selected DNS server(s). You will not enable DNS over HTTPS for your browsers and Thunderbird. All applications will send unencrypted internal UDP DNS queries to your computer port 53, and it will use TLS to encrypt DNS queries and send them to the public DNS you choose.

Some people suggest to use "stubby", an application that acts as a local DNS Privacy stub resolver (using DNS over TLS). Stubby encrypts DNS queries sent from a client machine (desktop or laptop) to a DNS Privacy resolver, increasing end user privacy. Since "stubby" has no cache, it is necessary to use "dnsmaq" as a non-permanent cache for "stubby". It is a complex installation and setting process to have "stubby" and "dnsmaq" working together.

Here is one method using "systemd-resolved" service, without additional package installation.

First, my initial configuration:

* In my connection settings I have set the use of Google Public DNS servers, for IPV4 and IPV6:



* "/etc/systemd/resolved.conf" has its default value:

```
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it under the
#  terms of the GNU Lesser General Public License as published by the Free
#  Software Foundation; either version 2.1 of the License, or (at your option)
#  any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the resolved.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
#
# Use 'systemd-analyze cat-config systemd/resolved.conf' to display the full config.
#
# See resolved.conf(5) for details.
[Resolve]
# Some examples of DNS servers which may be used for DNS= and FallbackDNS=:
# Cloudflare: 1.1.1.1#cloudflare-dns.com 1.0.0.1#cloudflare-dns.com
2606:4700:4700::1111#cloudflare-dns.com 2606:4700:4700::1001#cloudflare-dns.com
# Google:     8.8.8.8#dns.google 8.8.4.4#dns.google 2001:4860:4860::8888#dns.google
2001:4860:4860::8844#dns.google
# Quad9:      9.9.9.9#dns.quad9.net 149.112.112.112#dns.quad9.net
2620:fe::fe#dns.quad9.net 2620:fe::9#dns.quad9.net
#DNS=
#FallbackDNS=
#Domains=
#DNSSEC=no
#DNSOverTLS=no
#MulticastDNS=no
#LLMNR=no
```

```
#Cache=no-negative
#CacheFromLocalhost=no
#DNSStubListener=yes
#DNSStubListenerExtra=
#ReadEtcHosts=yes
#ResolveUnicastSingleLabel=no
```

* My browser does not use DNS over HTTPS mode.

In the browser I open BrowserLeaks DNS leak test page, https://browserleaks.com/dns; this page generates a hundred of DNS requests.

During the test, packets are captured with Wireshark (a well known packets sniffer).

DNS leak results:

172.253.12.195	Google LLC	Italy, Milan
172.253.12.196	Google LLC	Italy, Milan
172.253.12.197	Google LLC	Italy, Milan
172.253.13.129	Google LLC	Italy, Milan
172.253.13.130	Google LLC	Italy, Milan
172.253.13.131	Google LLC	Italy, Milan
172.253.13.132	Google LLC	Italy, Milan
172.253.13.133	Google LLC	Italy, Milan
2a00:1450:4025:1801::103	Google LLC	Italy, Milan
2a00:1450:4025:1801::102	Google LLC	Italy, Milan
2a00:1450:4025:1801::101	Google LLC	Italy, Milan

Wireshark captured packets:

No.	Time	Source	Destination	Protocol	Length	Info
340	3.922511756	8.8.8.8	192.168.1.2	DNS	122	Standard query response 0x34f6 A xpete7rv5fmqeqzs.4.browserle…
324	3.908935573	8.8.8.8	192.168.1.2	DNS	122	Standard query response 0x6189 A 8aa62na3x00uuy8r.4.browserle…
323	3.908916458	8.8.8.8	192.168.1.2	DNS	134	Standard query response 0x612d AAAA 8aa62na3x00uuy8r.6.browse…
305	3.897931397	8.8.8.8	192.168.1.2	DNS	122	Standard query response 0x90bd A 02j5deipsg10wmxk.4.browserle…
302	3.895513558	8.8.8.8	192.168.1.2	DNS	122	Standard query response 0x38a2 A 2mrv5ayip0vzeo07.4.browserle…
301	3.895489562	8.8.8.8	192.168.1.2	DNS	122	Standard query response 0x3df4 A 2mrv5ayip0vzeo07.6.browserle…
297	3.891486719	8.8.8.8	192.168.1.2	DNS	122	Standard query response 0x292d A xh0svdbvtjad2euk.4.browserle…
277	3.870083284	8.8.8.8	192.168.1.2	DNS	146	Standard query response 0x0769 AAAA 2mrv5ayip0vzeo07.4.browse…
274	3.867899677	8.8.8.8	192.168.1.2	DNS	134	Standard query response 0xc00d AAAA 02j5deipsg10wmxk.6.browse…
270	3.866161808	8.8.8.8	192.168.1.2	DNS	134	Standard query response 0xa5a9 AAAA 2mrv5ayip0vzeo07.6.browse…
257	3.851247402	8.8.8.8	192.168.1.2	DNS	134	Standard query response 0x09b5 AAAA rf7kinndxa34u10s.6.browse…
252	3.845132117	8.8.8.8	192.168.1.2	DNS	122	Standard query response 0x5ae3 A a76ai3twxfqqcop6.6.browserle…
251	3.844627119	8.8.8.8	192.168.1.2	DNS	122	Standard query response 0x965c A 02j5deipsg10wmxk.6.browserle…

As expected, DNS queries has been sent to 8.8.8.8 and answers returned (screen capture) using "Standard query", which means UDP protocol on port 53, without encryption.

Now, to use DNS over TLS protocol, you need to change settings in "/etc/systemd/resolved.conf".

First step is to save the original file.

[code]

sudo cp /etc/systemd/resolved.conf /etc/systemd/resolvedorig.conf

Then, edit "resolved.conf".

[code]

sudo nano /etc/systemd/resolved.conf

Select all the content, delete it, and replace it by:

[Resolve]

DNS=193.110.81.9#zero.dns0.eu

DNS=2a0f:fc80::9#zero.dns0.eu

DNS=185.253.5.9#zero.dns0.eu

DNS=2a0f:fc81::9#zero.dns0.eu

Domains=~

DNSOverTLS=yes

Cache=no-negative

ReadEtcHosts=yes

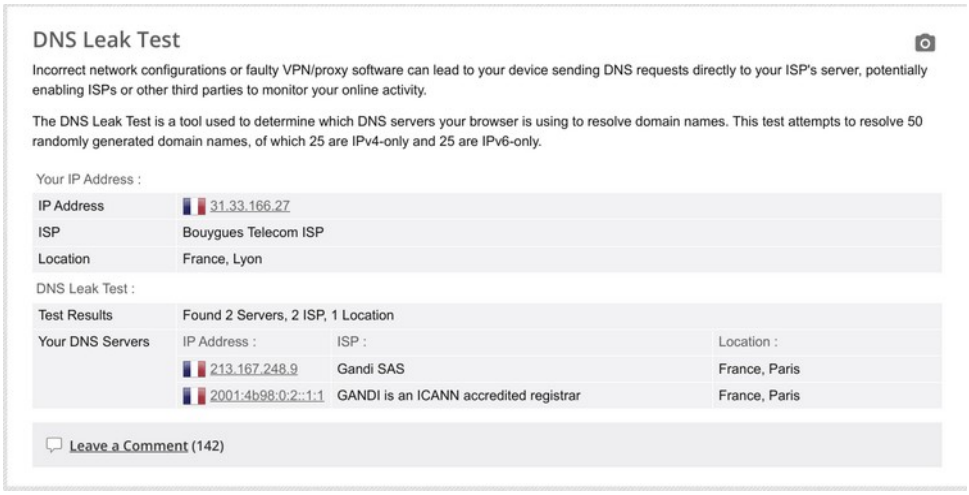
Here I have usedzero.dns0.eu DNS resolvers. Of course, you can use the DNS resolvers of your choice. Save "/etc/systemd/resolved.conf". Then, open your network settings, and delete the DNS servers references in Ipv4 and Ipv6 parameters:

Restart "systemd-resolved" service:

[code]

sudo systemctl restart systemd-resolved

I perform the same tests as before. DNS leak results:



Gandi SAS DNS server is a France based DNS server, part of "Zero" filtering DNS network.

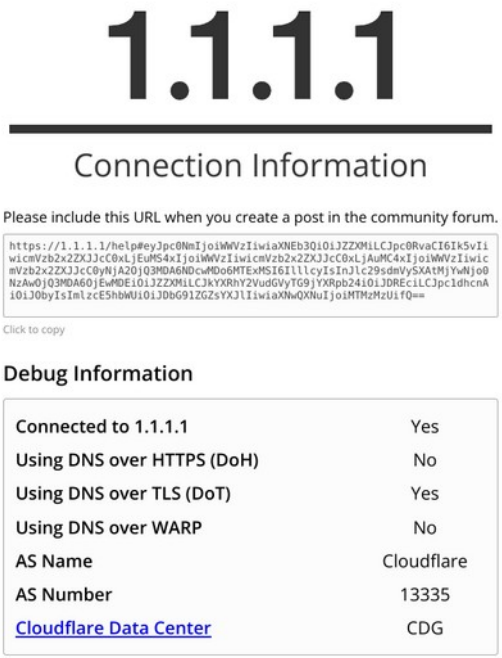
Wireshark captured packets:

266	4.689263648	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
265	4.689209613	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
263	4.680747750	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
262	4.680704332	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
260	4.670810704	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
259	4.670761116	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
257	4.660179459	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
256	4.660118219	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
254	4.650194321	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
253	4.650096619	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
250	4.639893764	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
249	4.639839890	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
247	4.629576815	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
246	4.629528501	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
243	4.623271904	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
242	4.623222815	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
240	4.616931477	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
239	4.616882677	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
238	4.609491537	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
237	4.609442962	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data
236	4.602410703	192.168.1.2	193.110.81.9	TLSv1.3	155	Application Data
235	4.602357569	192.168.1.2	193.110.81.9	TLSv1.3	90	Application Data

This time, queries have been sent to "Zero" filtering DNS server (193.110.81.9) using TLSv1.3 encryption.

If you had set Cloudflare as DNS server for TLS mode, you could test it at:

https://one.one.one.one/help/. Here is the result:



Like with DNS over HTTPS, if you don't use Cloudfare DNS servers, you still can use https://dnscheck.tools/ to check that DNSSEC works.

Now, all your DNS queries are addressed with UDP to your computer internal server on port 53, where "systemd-resolved" listens and addresses them encrypted to the chosen DNS servers on TCP port 853, using TLS v1.3 encryption: DNS over TLS protocol is used for all your operating system queries.

Your "/etc/hosts" is read, and the DNS answers are cached, except negative (failed) answers. Your ISP can detect you send DNS queries, and can detect what are the DNS servers you use, but it cannot know the content either of your DNS queries or of the answers.

NB: in "/etc/systemd/resolved.conf", "DNSOverTLS=opportunistic" attempts to make an encrypted DNS query and, is this query fails, it makes an unencrypted one; once you have checked with Wireshark that your DNS server does accept TLSV3 queries, you may change "opportunistic" by "yes".

5) What about a VPN?

A VPN, Virtual Private Network, will allow you having more functionalities;

* your IP address is not known by the websites you visit, it is masked by your VPN provider *[NB: it is almost mandatory to use a VPN when you have a fixed IP address from your ISP, since anti-tracking and fingerprint protection techniques are useless when your IP address never changes]*,

* you will appear to use your VPN provider DNS server,

* you can appear from another country than yours, and benefit of services not offered in your country.

However, your VPN provider knows all what you do on internet:

* your DNS requests,

* the websites you have visited (and their content, if you don't have set HTTPS only mode; but even with that setting, your VPN provider, using the "man-in-the-middle" attack principles, could know the content of the websites you visit).

With a VPN, you have just changed the location where all what you do on internet is centralized, from your Internet Service Provider location to your Virtual Private Network Provider one…

→ **You need to trust your VPN Provider, and to carefully choose it.**

You can use a free VPN service, free Proton VPN, see Annex 10: Install and Set Up Free Proton VPN.

[If you use an Anonymous mail, based on Proton mail, do not use the same e-mail address for Proton VPN than the anonymous one, but create a new Proton e-mail address for Proton VPN].

Or you can use a paid VPN service, that you will pay in bitcoin or other cryptocurrencies, in order to not reveal your name and means of payment:

* Proton VPN, https://protonvpn.com/, (paid service offers more protection and tools than free one),

* Mullvad VPN, https://mullvad.net/.

4.12) Protect your mails

This is prevention against PT3, PT5 and PT6.

Mails are basically pieces of text. Their content can be read by any server between sender and recipient; they can be altered; sender mail address can be spoofed. They are threats on user privacy.

Here are ways to reduce those threats.

1) Don't use free mail services

Free mail services such as Gmail, Hotmail, Yahoo Mail are tempting and used by lots of users.

However, you should remember that, when a service is free, it is paid for by user data. Those free mail services will display ads and sell your personal data.

→ For privacy, it is better to use pay services that can be trusted.

2) Don't let your mails on the servers

Most people use internet webmails, and access them with their browsers. This has the inconvenience to let their mails, sent and received ones, on a server.

Others will use IMAP protocols. This is useful when sharing a mailbox between several persons, but mails are let on the server.

Having all your mails available on one server is a strong threat on your privacy (and what about a server crash if you have no local copy?).

So, don't use webmails, and don't use IMAP protocol (except for sharing your mails on a dedicated mailbox). Use POP protocol, that you can configure with mail clients such as Thunderbird.

If you use several appliances (computer, smartphone…) to read your mails, configure auxiliary one (the smartphone) to let your mails on the server, and the main one (the computer) to remove the copy on the server.

3) Use encryption

* To connect to your servers, use SSL/TLS connection with port 995 to connect to POP server, and SSL/TLS connection with port 465 to connect to SMTP server.

* Encrypt your mails, with OpenPGP or S/MIME protocols, in order they can be read only by your recipient; sign them with OpenPGP or S/MIME, in order your recipient can be sure that you are the sender and that the mail has not been altered. Ask your senders to send you encrypted, signed mails.

See Annex 4: Encryption.

4) Use strong passwords to protect your mail accounts

See Annex 3: Password Selection.

5) Avoid mail domain spoofing

Mails are normally signed by mail servers, confirming that the mail domain "@domain.extension" has not been spoofed. This is called a dkim signature (dkim stands for "DomainKeys Identified Mail"). See https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail.

Using Thunderbird, the dkim signature validity can be easily checked with "DKIM Verifier" module, https://addons.thunderbird.net/en-GB/thunderbird/addon/dkim-verifier/.

Once installed, in its parameters go to "Display" and select the two boxes.

When you receive a mail, DKIM Verifier will automatically check the dkim signature validity and display the results of the verification.

Without Thunderbird and DKIM Verifier, in order to check the validity of a DKIM signature, you need to verify that:

* The DKIM-Signature header exists and is properly formed.

* The public key is available in DNS (via a TXT record).

*The hashes of the signed headers and body match what is in the signature.

* The cryptographic signature can be validated with the public key.

You can use Google Admin Toolbox to automate this process; in your mail client, display the source of the message and copy the full header; then go to:

https://toolbox.googleapps.com/apps/messageheader/,

paste the full headers of your message, and click on "Analyze Header".

6) Protect your own mail domain from spoofing

If you have your own mail domain, you should add a SPF record to your mail domain. SPF stands for "Sender Policy Framework"; it is a text added to your Domain Name System record, specifying what are the mail servers allowed to send a mail for your mail domain. If a mail presumably coming from your mail domain is sent by another mail server than the ones in the SPF record, this will prove the mail is a spoofed one, and recipients servers will reject it.

See https://en.wikipedia.org/wiki/Sender_Policy_Framework.

If your mail domain is managed by an ISP, check with it how to write this SPF record.

Once done, your mails will also have an automatic dkim signature.

If you manage your own mail server, you know how to help yourself and how to add an SPF record to your mail domain!

4.13) Protect yourself from spam

This is prevention against PT4.

Without precautions, your mailbox will be very soon full of spam. It makes more time-consuming to read your mails (you have first to sort and delete the spam) and enables tracking you.

There is no miracle recipe to avoid spam, just some precautions.

1) Use adaptive, self-learning spam filters

That kind of filter is incorporated in Thunderbird. When you receive a spam, you declare it as unwanted; when you receive a mail wrongly classified as unwanted, you declare it acceptable.

Progressively, the spam filter will learn to make the difference between acceptable and unwanted mails.

2) Use your mail hosting service anti-spam

If you use a pay hosting service, it might propose you an anti-spam service; use it.

3) Thunderbird specific protections

In Edition / Parameters / Privacy and security, let unselected "Allow distant content in e-mails" (it prevents tracking, and you can still display distant content on a case by case basis).

If you use feeds, you can install uBlock Origin in Thunderbird, and configure it in the same way as for a browser. It adds some anti-spam capability and prevents tracking. See Safe browsing.

4) Various protections

- When you buy something online, be careful to not select to receive a mail letter (it is often selected by default).

- If you receive unwanted mail letters, try to unsubscribe to those mail letters. It may work (with *bona fide* senders) or not (with others…).

- Finally, you may need two mail addresses. A "rotten" one (Gmail, Hotmail, Yahoo Mail…) that you will use when buying something online and that will receive spam, and a "good" one that you will use for your other mails.

- To avoid spams, you can use disposable e-mail addresses, from https://yopmail.com/, https://www.disposablemail.com/ or https://maildrop.cc/; when you need an e-mail address to register to an internet service or website, you use the disposable address once, to retrieve a register code, and avoid having your main e-mail address spammed. You can also use alias e-mail addresses from DuckDuckGo https://duckduckgo.com/email/, Firefox Relay https://relay.firefox.com/, Proton Pass https://proton.me/pass/aliases.

4.14) Install LanguageTool local server

This is prevention against PT7.

LanguageTool, https://languagetool.org/, is a nice, free, multilingual grammar and spell checker. It is available for office suites (LibreOffice, OpenOffice on Linux), browsers (Firefox, Chromium, and their derivatives on Linux), mail clients (mainly Thunderbird on Linux).

The use of LanguageTool may be a threat to your privacy by sending texts you want to be corrected to LanguageTool servers (located in Germany).

Initially, LanguageTool was only an extension for LibreOffice and OpenOffice. That extension is still available, and works locally (without sending texts to LanguageTool servers), for free, without any text length limitation. It can be still downloaded from https://languagetool.org/download/, download the latest ".oxt" file. From LibreOffice 7.4.x, the extension is no longer necessary (while it can be still used); LibreOffice code has been modified and can send text for correction to LanguageTool servers. It is a privacy threat. Moreover, the free Basic service allows correcting texts with a maximum of 20,000 characters and the pay Premium one allows correcting text with up to 150,000 ones. Browsers and Thunderbird extensions do not work locally by default and send the texts for correction to LanguageTool servers, with the same 20,000/150,000 characters limitation.

When using LanguageTool on LibreOffice, browsers and Thunderbird, it is interesting for both privacy protection and no limitation reasons to set up a local LanguageTool server that will be used by all applications.

This tutorial, based on https://dev.languagetool.org/http-server, shows how to set a local LanguageTool server, that can be used by LO >= 7.4 versions integrated code and by mail clients and browsers extensions. That way, the text to correct has no size limit, and the correction is done locally on your computer, without privacy leakage.

Note that the local server, or the full LibreOffice extension, cannot offer the AI improvements available with a pay Premium account and distant corrections. You have to choose privacy protection or improved service.

1) Installation detailed steps

- If not already done, install Java; note that LanguageTool version 6.6 requires Java minimum version 17; "openjdk-17-jre" is available in Linux Mint repositories.

- Download LanguageTool Desktop for offline use:

https://languagetool.org/download/LanguageTool-stable.zip.

- Once downloaded, decompress it; it will create a "LanguageTool-6.6" directory in your "Downloads" one. *[NB: 6.6 is the latest version on May 2025, the version number will change with time]*.

- Move "LanguageTool-6.6" directory into your home "~/opt" one. Rename it simply "LanguageTool".

- Create, where you want, a void "config.txt" file, example in "~/.config/LanguageTool/config.txt".

- Create in your "~/opt" directory a file named "LTserver.sh":

[code]

```
#!/bin/bash

# launch of LanguageTool server

nohup java -cp ~/opt/LanguageTool/languagetool-server.jar
org.languagetool.server.HTTPServer --port 8081 --allow-origin --config
~/.config/LanguageTool/config.txt > /dev/null 2>&1 &
```

Some code explanations:

* "nohup" at the beginning and "&" at the end allow the server to be permanent even after the closure of the terminal window in which the code will execute,

* "java -cp ~/opt/LanguageTool/languagetool-server.jar org.languagetool.server.HTTPServer --port 8081 --allow-origin --config ~/.config/LanguageTool/config.txt" is the code used to launch the server,

* "> /dev/null 2>&1" redirects the output and the errors to "null", the terminal window will not be seen.

- Make "LTserver.sh" executable, changing its permissions in your file manager or using chmod.

- Add "LTserver.sh" in the list of applications launched at startup.

Example, in Mate, launch "control center" then "startup applications", click on the "+" button to add an application, name it "LanguageTool server launch", and in the "command" line copy "~/opt/LTserver.sh", put a 5 to 10 seconds delay after startup and save. LanguageTool server will be automatically launched at each system startup.

[NB: if you prefer a server to launch and close manually, replace the command line in "LTserver.sh" by:

"java -cp ~/opt/LanguageTool/languagetool-server.jar org.languagetool.server.HTTPServer --port 8081 --allow-origin --config ~/.config/LanguageTool/config.txt".

Launch the server by executing "LTserver.sh"; close the server by closing its terminal window].

- Shutdown and restart your computer. You are now going to test that the server functions. Click on the following URL: http://localhost:8081/v2/check?language=en-US&text=my+text; after a few seconds the browser will display something similar to:

```
software
name  "LanguageTool"
version     "6.6"
buildDate   "2025-03-27 20:50:25 +0100"
apiVersion  1
premium     false
premiumHint "You might be missing errors only the Premium version can
find. Contact us at support<at>languagetoolplus.com."
status      ""
warnings
incompleteResults false
language
name  "English (US)"
code  "en-US"
detectedLanguage
name  "English (US)"
code  "en-US"
confidence  0.65618557
sourc
[...]
```

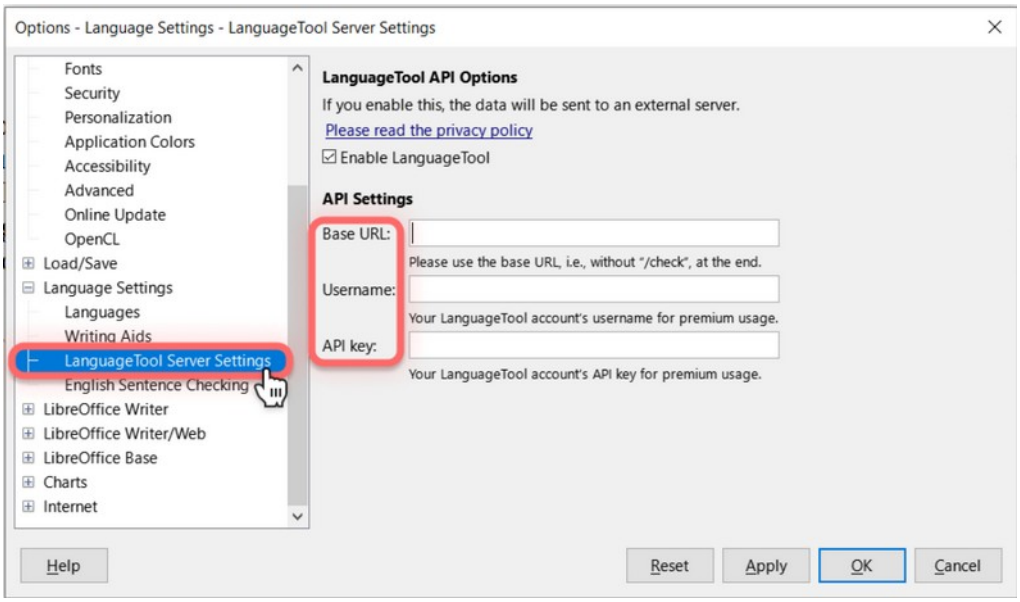
[If the server does not work, a message with "connection refused" will be displayed. For information: on my computer LanguageTool server uses some 816 MiB of RAM when idle, a bit more when correction is active.]

- It is now time to set up your mail clients and browsers extensions: on the LanguageTool extensions option, go to "advanced parameters" and select "local server (localhost)" instead of "cloud server (languagetool.org)".

- If you use LibreOffice 7.4.x or later, follow this tutorial:

https://languagetool.org/insights/post/product-libreoffice/#how-to-enable-languagetool-on-libreoffice.

In the following window, write "http://localhost:8081/v2" in "Base URL":



- You can now use LanguageTool locally, for free, without text size limitation and without privacy leakage in your mail clients, browsers, and LO 7.4.x and further !

- Optional: you can increase LanguageTool capabilities, finding errors using n-gram data, for English, German, French, Spanish and Dutch. LanguageTool will make use of large n-gram data sets to detect errors with words that are often confused, like "their" and "there". Follow instructions here: https://dev.languagetool.org/finding-errors-using-n-gram-data.

2) Security

* If you are a single user, set your firewall to block all incoming connections.

* If your computer is used as a LanguageTool server on an internal network, add "--public" in the command line in "Ltserver.sh"; the command line becomes:

"nohup java -cp ~/opt/LanguageTool/languagetool-server.jar org.languagetool.server.HTTPServer --port 8081 --allow-origin --config ~/.config/LanguageTool/config.txt --public > /dev/null 2>&1 &";

Allow incoming connections from the computers on your network to your 8081 port, then block all other incoming connections.

[If your LanguageTool server is public and can be accessed from internet, it is recommended to use the LanguageTool HTTP server and to run it behind an Apache or nginx reverse proxy with SSL/TLS support].

3) Updates

Check for LanguageTool updates at https://languagetool.org/download/, download again "LanguageTool-stable.zip" when an update is available, and always copy the contents of its decompressed directory to your "~/opt/LanguageTool" one (after having deleted the former directory content).

4.15) Use local translation programs

This is prevention against PT8.

Online translation (Google Translate, Bing Microsoft Translator…) has considerably improved, and will probably take profit of AI for still better translation.

Whatever the way you use it (copy/paste text on web page, use a browser extension, use Crow Translate or use a LibreOffice extension), the text you want to translate is sent to Google or Bing servers, and it is not suitable if the text is confidential.

The only way to avoid that privacy leakage is to use a translation program that can work locally, without sending anything to servers. Don't expect from that kind of tool the same performance as with online ones, use them for confidential texts.

1) Use Translate Locally

See https://translatelocally.com/.

From its download page, https://translatelocally.com/downloads/, you can download debs for Ubuntu 20.04 LTS, 18.04 LTS and compatible systems. Linux Mint 21.x/Ubuntu 22.04 LTS users can try to install the 20.04 version with gdebi, or will need to compile it from source.

Translate Locally is also available as a browser extension for Firefox.

2) Use CATs

You can use Computer Aided Translation Programs, such as OmegaT, Anaphraseus, Lokalize, Swordfish etc.

4.16) Avoid to have your personal data stolen

This is prevention against PT9 and PT10.

There are several ways your personal information (passwords, payment data, personal information) stolen.

1) In Browsers

- Browsers propose to use autofill for passwords, payment data (credit card reference…) and personal information (name, postal address…). This is a dangerous facility:

* There are known attacks against the use of autofill: cross-site scripting, see:

https://en.wikipedia.org/wiki/Cross-site_scripting; it can be used to steal passwords saved in your browser.

* A browser weakness can be used to get access to the saved information (passwords, payment data, personal information) in your browser; recently, Google has informed about a critical vulnerability, CVE-2023-3214: this new security issue is rated as critical and impacts the autofill payments function of the Google Chrome browser. It is not the first, it will not be the last one, and this kind of vulnerability can affect all browsers.

Since the data are saved in browser settings, sandboxing does not protect against this kind of vulnerability.

→ The only prevention mean is: do not use autofill in browsers.

2) Phishing

Your personal data can be stolen with phishing; here is an example, very common now in France:

* You receive a mail or an SMS asking you to pay something, generally a small amount: you need to pay to receive a parcel, or you are late to pay a fine; if you click on the received link, you will be directed to a website, resembling to the legit one (of course, it uses the same html code, captured from the legit one).

→ Never click on a link asking you to pay something; on the browser, enter the url link manually.

* If you enter your name and payment data, you will then be contacted by people pretending being members of your bank staff; they will know all about you (of course, you have given the

information at the preceding step). Once in confidence, they will tell you that someone has pirated your bank account, and they will ask you to give your card pin, or to use your bank application, on a smartphone or through your bank website, to block the pirate use of your bank account. In both cases, they will incite you to validate payments to the pirate.

→ Never give your card pin by phone; when you use your bank application, check what you do before to validate something.

3) e-commerce

e-commerce websites (Amazon, eBay...), payment websites (PayPal…) and bank online applications do keep a lot of information about you. Sometimes their servers are not well protected, and the data they keep are stolen (personal information, passwords…), and sold on Darknet or used.

→ Enable two-factor authentication on e-commerce, payment and banks websites, in order to avoid that stolen data can be used. See Annex 3: Password Selection.

4) Applications phoning home

Some, or most, applications initiate outgoing connections by themselves; they are said to "phone home", and can expose private data concerning you as a user or your computer / operating system / programs.

* Some of the prevention means already exposed in this guide, such as "/etc/hosts" IP addresses filtering do reduce this threat, see 4.7). You can complete the lists used in "/etc/hosts" by more specific ones; as an example, if you use Microsoft products you may be interested in lists preventing Microsoft applications to phone home, such as the ones found at https://github.com/crazy-max/WindowsSpyBlocker/tree/master/data/hosts.

* Choose your application.

As an example, Google Chrome has strong relations with Google servers; its Open Source version, Chromium, has less but still has hundreds of internal links to Google Chrome, coded in the source and that cannot be affected by settings; finally, Ungoogled Chromium has no link at all with Google servers, while keeping the same GUI as Chromium. Ungoogled Chromium will not expose your private information to Google (except, of course, if you use some Google services, but it will be your choice).

Microsoft Edge browser is probably very linked to Microsoft servers.

Electron applications, see https://www.electronjs.org/, embed Chromium and Node.js to create desktop applications. They are known to contact Google servers. Avoid them if you want to avoid your data being sent to Google.

* Learn why your applications initiate outgoing connections and adjust their settings.

Some are normal: Mint Update Manager periodically checks the availability of new packages and connects to Linux Mint repositories, to Ubuntu ones and to all extra repositories you have added (other main repositories, PPAs, flathub etc.); if you use Ubuntu Pro, it periodically connects to Ubuntu servers to check the validity of your subscription, free or paid one.

Some applications just connect to internet to check for a new version availability.

Firefox has a support page, "How to stop Firefox from making automatic connections" explaining why automatic connections are made, see https://support.mozilla.org/en-US/kb/how-stop-firefox-making-automatic-connections. Once you have read and understood this page you can use Firefox settings to remove the automatic connections you don't want.

* Block internet access.

Some applications do not need to access internet. You can use sandboxing to prevent them to access internet.

You can use flatpak version of an application and block its internet access in GUI mode with Flatseal (you deselect "share=network" with a slider).

If a Firejail profile is available for your application, you can use Firejail with a specific launch command "firejail --net=none <your_app_full_path_name>".

* Limit outgoing connections.

For this, you need firewalls with more functionalities than UFW/GUFW.

OpenSnitch, https://github.com/evilsocket/opensnitch, is a GNU/Linux interactive application firewall inspired by Little Snitch. It logs your connections and makes provisional rules that you can further transform in permanent ones. Using it, you can allow an application some outgoing connections and block other ones.

NB: do not think to block illegitimate applications with this method, but only to control the way legitimate ones connect to internet; illegitimate applications have known from long how to load themselves in RAM just below a legitimate one, and profit of the authorized outgoing connections of this application.

4.17) Stay anonymous

This is prevention against AT1.

We have seen how to avoid tracking (see Safe browsing), and how to avoid that all what you do on internet can be known and logged by your ISP (see Reduce what your ISP can know).

However, that may not be enough for some users who want to hide further what they do on internet and stay anonymous.

Anonymity can be reached by using dedicated internet encrypted subnetworks that don't use the public IP addressing but their own one, not allowing to identify a user with its internal address.

They are mainly three of those subnetworks:

"The Onion Router", or Tor, https://www.torproject.org/,

"The Invisible Internet Project", or I2P, https://geti2p.net/en/,

"Freenet", https://freenetproject.org/.

All are available for Linux Mint.

How do they work?

1) From I2P website

> The Invisible Internet Project (I2P) is a fully encrypted private network layer. It protects your activity and location. Every day people use the network to connect with people without worry of being tracked or their data being collected. In some cases people rely on the network when they need to be discrete or are doing sensitive work.
>
> I2P hides the server from the user and the user from the server. All I2P traffic is internal to the I2P network. Traffic inside I2P does not interact with the Internet directly. It is a layer on top of the Internet. It uses encrypted unidirectional tunnels between you and your peers. No one can see where traffic is coming from, where it is going, or what the contents are. Additionally, I2P offers resistance to pattern recognition and blocking by censors. Because the network relies on peers to route traffic, location blocking is also reduced.
>
> The network is people powered. Peers make a portion of their resources, particularly bandwidth, available to other network participants. This allows the network to function without relying on centralized servers.
>
> I2P has created transport protocols that resist DPI censorship, and continuously improves its end-to-end encryption.
>
> Outproxies to the Internet are run by volunteers, and are centralized services. The privacy benefits from participating in the I2P network come from remaining in the network and not accessing the internet. Tor Browser or a trusted VPN are better options for browsing the Internet privately.

→ I2P main use is to stay within I2P network, and use or host services on the network.

You can install a flatpak i2p client, i2p daemon:

> [code]
>
> flatpak install website.i2pd.i2pd

Here is its documentation: https://i2pd.readthedocs.io/en/latest/.

Using a VPN, you can have a double anonymity layer, see Annex 10: Install and Set Up Free Proton VPN.

2) From Freenet website

> Freenet is a peer-to-peer platform for censorship-resistant and privacy-respecting publishing and communication.
>
> Freenet makes it easy to publish and follow what others publish with strong privacy protections.
>
> Plugins built on its decentralized data store make it very easy to host your own website and provide microblogging and forums, media sharing from files to video-on-demand and decentralized version tracking, blogging and spam resistance without central authority.
>
> For an easy start you can join the global Opennet. For maximum privacy, connect to your friends and build a friend-to-friend network independent of and invisible to any centralized

server. To access the global network, you either need some friends who also connect to opennet, or use the Shoeshop plugin to build a sneakernet that can even bridge separate friend-to-friend networks when your regional internet itself gets severed from the global information network.

Lots of additional information about Freenet and its history is available on Wikipedia, https://en.wikipedia.org/wiki/Freenet.

Freenet is under move from Fred, analogous to a decentralized hard drive, to Locutus, analogous to a full decentralized computer. Locutus development is not yet fully completed.

Download and install Freenet from hyphanet.org, https://www.hyphanet.org/pages/download.html.

Documentation, https://www.hyphanet.org/pages/documentation.html.

Using a VPN, you can have a double anonymity layer, see Annex 10: Install and Set Up Free Proton VPN.

3) Tor

- With Tor, there are three kinds of users: basic users, internal relays and internet relays. When a basic user wants to access internet through Tor network, his request is sent encrypted to an internal relay; it can be sent through several internal relays, and finally reach an internet relay.

The first used internal relay knows who has emitted the request, but does not know its content (it cannot decipher it); further internal relays don't know the original emitter and the request content; internet relay knows the request content but does not know the original emitter. It ensures anonymity.

Internal relays can anonymously host services, known through their ".onion" internal address.

Tor has some limitations:

* most users are basic ones, they don't dedicate one part of their resources to be internal or internet relays, and the traffic is slow,

* the internet relays, some 2,000, are well identified and can be blocked easily by any website admin.

→ Tor first use should be to access Tor services (onion websites), and occasionally to browse internet.

* The usual way to access Tor network and to browse internet is to download Tor Browser, from its download page, https://www.torproject.org/download/.

The download is a compressed "tar.xz" file; once downloaded, you uncompress it and copy its directory anywhere in your home, "~/opt" as an example. Further installation and launch is described in Tor Browser installation manual, see https://tb-manual.torproject.org/installation/.

If Tor Browser is good for anonymity, however it is poor, on a security point of view. Its security can be reinforced by Firejail sandboxing, but it could not prevent zero-day attacks, since Tor Browser uses some of the operating system libraries and is not isolated from the system as flatpaks or snaps are. And Tor Browser is not available as a flatpak or a snap.

There is a more secure alternative. Mullvad Browser is Tor Browser without Tor network; it offers the same anti-tracking features as Tor Browser, and is designed to be used with VPNs. It is available as a flatpak, offering the maximum security, and can be used with Tor network.

See Annex 8: Mullvad Browser Flatpak on Tor Network, a Secure Alternative to Tor Browser.

Note that some people recommend to access Darknet with a double anonymity protection, using Tor Network through a VPN, see Annex 10: Install and Set Up Free Proton VPN.

* Other applications than browsers can use Tor network. For that, Tor should be installed as a service on your operating system, see https://support.torproject.org/apt/#apt_tor-deb-repo.

Applications able to use socks proxy can be used directly on Tor network, by setting the proxy use. Other applications can be "Torified" using Torsocks.

How to install Tor service, and an example of proxy setting on Mullvad Browser, are detailed in Annex 8: Mullvad Browser Flatpak on Tor Network, a Secure Alternative to Tor Browser.

* Tails is a portable operating system that expands Tor use to all applications. It protects against surveillance and censorship. See https://tails.boum.org/index.en.html.

You can use Tails in several ways:

- install it on a computer,

- download a USB key image,

- download an ISO and burn it on a DVD,

- or use it in a virtual machine (you create a virtual machine with VirtualBox or VMware Workstation Pro; the machine boots on the ISO image, and an allocated disk can be used by Tails to make an encrypted container where to save files).

[Tails changes often, with each version of Tor Browser; always use the latest Tails version].

4) Anonymous mail

You can create and use an anonymous mail using Tor Browser or Annex 8: Mullvad Browser Flatpak on Tor Network, a Secure Alternative to Tor Browser.

Launch Tor Browser or Mullvad Browser on Tor Network:

- Create a disposable e-mail address, go to https://yopmail.com/ and create your disposable address, of the kind "anything_you_want@yopmail.com"; it requires no password; once your address created, the inbox will open, keep it in a tab of your browser.

- Create a secure encrypted Proton mail address: go to https://proton.me/mail and create a free account; select a pseudo (your mail address will be "pseudo@proton.me"), enter twice a password (see Annex 3: Password Selection); during the creation process, you will be asked to enter an e-mail address in order to receive a verification code, enter the yopmail address you have created, retrieve the verification code, enter it in your proton mail creation page.

[Proton mail is a secure, encrypted, mail service with servers based in Switzerland].

- Once your mail account created, you will be proposed to use your yopmail address as recover; deselect it, and confirm account creation without recover method. That's done! You have a permanent, anonymous e-email address and a secure, encrypted webmail service.

Use your Proton webmail service with Tor Browser or Mullvad Browser on Tor Network only!

You can further connect at the following address: https://account.proton.me/login.

5) Private Messaging

We are looking here for messaging software with the following characteristics:

- Full Open Source Software (FOSS);

- Full End To End Encryption (E2EE);

- Not requiring the use of a smartphone to register;

- Actively maintained;

- Secure, running in a sandbox.

That excludes Skype, Microsoft Teams, Google Meet, Trillian (undisclosed proprietary software); Pidgin (encryption no longer maintained); WhatsApp (proprietary, requires a smartphone); Signal, Telegram, Olvido (require a smartphone)… and many others.

Here are my proposals:

- I2P clients (see §1 of this chapter) include a fully encrypted IRC chat.

- Several encrypted chat rooms run on Tor network (see §3 of this chapter), you need to be "invited" to join such chat rooms, or you can create your own one (as a Tor service).

You can easily create your own chat room on Tor Network with OnionShare, an application allowing to chat anonymously, share files, receive files, and host a website; it is available as a flatpak at Flathub (https://flathub.org/apps/org.onionshare.OnionShare).

Once OnionShare installed, you can create your chat room; you then need to send the credentials securely to the people you invite to join it (using encrypted mail, see Annex 4: Encryption).

People can join your chat room using Tor Browser (available on Linux, macOS, Windows, Android); Onion Browser or Orbot on iOS (with reduced confidentiality protection); Mullvad Browser flatpak on Linux, with increased security (see Annex 8: Mullvad Browser Flatpak on Tor Network, a Secure Alternative to Tor Browser).

- Applications running Extensible Messaging and Presence Protocol (XMPP), see https://xmpp.org/.

Two applications running XMPP are available as verified flatpaks, on Flathub:

* Dino (https://flathub.org/apps/im.dino.Dino);

* Gajim (https://flathub.org/apps/org.gajim.Gajim).

In all cases, you can still reinforce your privacy / anonymity by using a VPN (see Annex 10: Install and Set Up Free Proton VPN).

4.18) Protect your LAN against wireless intrusions

This is prevention against DSA5.

An attacker in your vicinity could penetrate your Local Area Network (LAN) using unprotected or insufficiently protected wireless connections (Wi-Fi, Bluetooth). Once done, it could access your LAN computers (using shared directories or open ports) and equipment (LAN disks…).

Here are the precautions to take, in order to avoid those intrusions.

1) If you do not use Wi-Fi or Bluetooth, disable them

- If your modem-router offers Bluetooth, always disable it.

- Disable Wi-Fi on your modem-router by any available mean: hardware (remove antenna), software (using the modem-router settings).

- Disable Bluetooth on your computer when not used (enable it only when you use it, then disable it).

- Disable Wi-Fi on your computer by any available mean: hardware (remove Wi-Fi board, remove antenna) or software (uninstall Wi-Fi chip driver, do not set a Wi-Fi connection).

2) Choose a secure Wi-Fi protocol

- Your modem-router, your computer and your equipment may offer several Wi-Fi protocols: WEP, WPS, WPA, WPA2, WPA3. You can use only the protocols available for all.

- An attacker will typically use a computer with two Wi-Fi boards; one will be used to perturb the Wi-Fi traffic and to force the modem-router and connected equipment to exchange again keys, the second one will be used to record this; a software program (some are available as OpenSource) will analyze the recorded traffic and attempt to gain Wi-Fi access to your LAN.

- WEP should NOT be used, since it cannot resist to this kind of attack.

- WPS security is doubtful, and, if possible, WPS use should be avoided.

- WPA has been designed as a temporary solution to correct WEP security breaches, and partially complies with 802.11i standard. If possible, WPA use should be avoided.

- WPA2 is fully compliant with 802.11i standard. However, this protocol is old, 2004, and has shown a severe security breach in 2018. This breach has been corrected, and WPA2 can now be used.

- WPA3 is the most modern protocol, defined in 2018, but its implementation in modem-routers, computers and equipment is slow. WPA3 replaces PSK encryption algorithm by AES one.

→ the use of WPA2 with AES, or the use of WPA3 are considered secure, and should be the preferred choices.

3) Use strong passwords

- See Annex 3: Password Selection, for the choice of a password and its strength.

- Modem-router and computer: you should set two different passwords, one to administrate your modem-router, and another one for the Wi-Fi connection.

- Computer as an access point: the password of the Wi-Fi connection should be different from all the other ones used on your computer.

- Bluetooth connection to your computer: the password of the Bluetooth connection should be different from all the other ones used on your computer.

4) Other precautions

Applying these precautions depends on their availability.

- Keep your computer, modem-router and equipment up-to-date: for your computer, see Update your system; for your modem-router and equipment, update their firmware and software, as per their manufacturers recommendations.

- Make the name of your Wi-Fi connection not visible: it will be more difficult to be detected, and an attacker would have to find its name.

- Filter by their MAC address the authorized computers and equipment allowed to connect to your modem-router.

- In your modem-router do not enable DHCP for the connected computers and equipment; attribute them a fixed IP address and disable DHCP.

- Change the IP addresses used in your LAN: a modem-router will often use by default the IP address block 192.168.1.0 to 192.168.1.255, with the modem router itself being 192.168.1.1. But several ranges of private IP addresses are available: 10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.31.255.255, 192.168.0.0 to 192.168.255.255. Changing the IP address block used by your LAN will make an intrusion more difficult, particularly when DHCP is disabled (see former precaution).

- Don't give your guests access to your main LAN; set for guests, in your modem-router, a dedicated connection giving access to internet only, with a different password. After use, disable this connection, and next time change the password.

- In your modem-router, enable IPV4 and IPV6 firewalls.

- If your modem-router can accept it, use Open Source firmware such as OpenWrt, https://openwrt.org/.

5. Detection

5.1) Malware and viruses detection

1) Are they Linux specific malware and viruses?

The answer is yes.

- Linux Malware Detect, https://www.rfxn.com/projects/linux-malware-detect/, has several thousands signatures of malware targeting Linux.

- This Wikipedia article, https://en.wikipedia.org/wiki/Linux_malware, details what are the malware and viruses targeting Linux.

The true question, how many of those threats are now existing in the wild, is much more difficult to answer!

2) What are the Linux available tools?

- Linux Malware Detect, https://www.rfxn.com/projects/linux-malware-detect/ is still maintained. It can be installed on Linux Mint, https://www.rfxn.com/downloads/maldetect-current.tar.gz. It is an on-demand scanner, specialized on some malware.

- Chkrootkit and Rkhunter, rootkit scanners, are no longer maintained and are now obsolete.

→ The corresponding section has been removed.

- Though there are plenty of offers for Linux antivirus endpoints for professional use, the only offer for a FOSS Linux virus scanner program for personal use is ClamAV, https://www.clamav.net/.

3) What are the tools to be used?

System scan:

- When a system is suspected to be infected, the good way is to scan the system from a running live CD or live USB key: in that way it is launched when your operating system is not running, it cannot compromise it, and cannot be lured.

* You can scan your computer using the Kaspersky Rescue Disk, an application that allows scanning your computer without booting an operating system, https://support.kaspersky.com/krd18. To use it, you must create a bootable USB drive or CD/DVD disk and boot your computer from the external media.

* You can use "Antivirus Live CD" for this, see https://4mlinux.com/index.php?page=fork1 for download and tutorial, it is an official 4MLinux fork including the ClamAV scanners.

On-access scan:

- On-access system scan faces the same problem as system scan: if a malware, virus or rootkit is running, it could lure any virus scanner. But on access user files scan could have some interest.

- In a professional environment, with a secured network (firewalls, filtering proxies, mail servers with their own virus scanner, intrusion detection probes) and strong computer security policy, the use of a pay endpoint protection software (from Cisco, Eset, Kaspersky, Microsoft, Sophos etc.) would bring benefits in the case of mixed operating systems (Linux, Windows, macOS…).

- But using ClamAV on-access scan is not adequate:

* As most virus scanners, it can be lured by viruses/malware self-launching before ClamAV during the computer start process.

* Deleting its log files directory prevents ClamAV use (this security breach could be used by malware launched at boot, having acquired superuser privileges). ClamAV should be installed without logging.

* Its use is controversial, since it parses malicious data from unknown sources; moreover, with on-access scan several services are running → ClamAV has a large attack surface.

* Ubuntu offers an AppArmor profile for ClamAV, but this profile prevents on-access scan; it should be edited to allow on-access scan. ClamAV security should be set-up relying on Firejail, AppArmor or Systemd sandboxing; but these sandboxing tools do not offer such a security as Flatpak or Snap (and ClamAV on-access scan is not available as a flatpak or snap). Installed on your system, with ClamAV services constantly running in the background, security is not optimal.

* Its installation can be done from Ubuntu repositories: the installation is facilitated by preinstalled configurations, but the version offered if very old or obsolete. Its installation can also be done by using the latest stable or LTS version available from ClamAV website, but all the configuration and settings need to be done manually. In all cases, setting on-access scan to work and setting sandboxing is a complicated and manual task.

* As per https://en.wikipedia.org/wiki/Clam_AntiVirus:

> ClamAV was tested against other antivirus products on Shadowserver. In 2011, Shadowserver tested over 25 million samples against ClamAV and numerous other antivirus products. Out of the 25 million samples tested, ClamAV scored 76.60% ranking 12 out of 19, a higher rating than some much more established competitors.
>
> In the 2008 AV-TEST of antivirus tools, ClamAV scored poorly in on-demand detection, avoiding false positives, and rootkit detection.
>
> In a Shadowserver six-month test between June and December 2011, ClamAV detected over 75.45% of all viruses tested, putting it in fifth place behind AhnLab, Avira, BitDefender and Avast. AhnLab, the top antivirus, detected 80.28%.
>
> In 2022 Splunk conducted an efficacy study involving ~400,000 malware samples sourced from MalwareBazaar. The study concluded ClamAV is 59.94% effective overall at detecting commodity malware.

This shows that ClamAV detection rate, using official signatures only, has been between ~60% to ~75% on the period 2011-2020. The detection rate can be improved by using extra unofficial signatures, though without reaching the detection rate of commercial software.

* Finally, the on-access scan interest is low: ClamAV scanning is slow and uses a lot of RAM and CPU, and on-access scan cannot be used system-wide; the numbers of directories to protect with on-access scan should be very limited in order to avoid system freeze.

→ For all of those reasons, I don't recommend installing ClamAV on-access scan on your operating system.

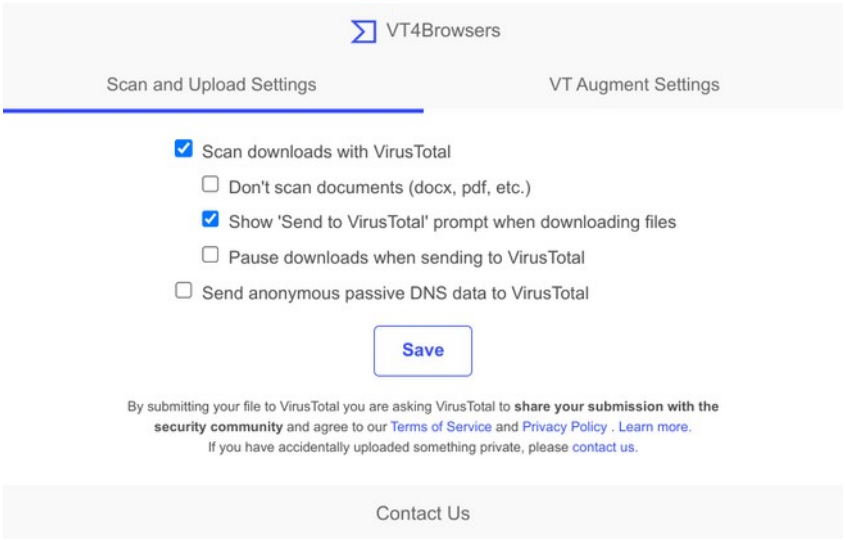
On-demand scan:

- As we have seen, Linux Malware Detect can be used (see its website for installation and use). Since its signatures are limited to a specific category of malware, don't expect a high usefulness; moreover, as any malware scanner launched from your operating system, it can be lured. LMD signatures can be used by ClamAV.

- Use an online tool, VirusTotal:

* Upload an attachment or scan a web page from https://www.virustotal.com/gui/home/upload.

* Use "VT4Browsers" extension to upload a file to be scanned to VirusTotal, with specific settings:



* Use "Just Verify It" extension for Thunderbird. It allows sending mail attachments to VirusTotal; it requires an API key, and user should subscribe one for free (it allows sending up to four attachments per minute to VirusTotal).

VirusTotal is a very nice tool, and since it uses more than 70 antivirus programs and is continuously updated, it has the best possible detection rate, including for very new malware. Not running on your operating system, it cannot be lured. It is used by large organizations (as an example, archive.org sends all the submitted files to VirusTotal).

However, it has two limitations:

* The maximum uploadable file size is 650 MB (is it really a limitation?).

* VirusTotal should NOT be used for confidential files. Here is an extract of VirusTotal Privacy Policy, https://support.virustotal.com/hc/en-us/articles/115002168385-Privacy-Policy:

> We also use the information we collect to provide, maintain, protect, and improve the Services, to develop new features of the Services, and to protect the Community and our malware-fighting mission. This includes using Samples and other collected information for any of the following activities:
>
> . Sharing Samples with antivirus, scanning, sandbox, and other security partners in order to generate requested malware verdicts for the user who uploaded the Samples.
>
> . Making Samples available to verified security professionals, companies, and security researchers, many of whom are VirusTotal customers or partners, for threat detection and research.
>
> . Further analyzing and scanning Samples submitted by the Community to generate useful information and corresponding security reports and further publishing and updating the reports to the Community and making such material available through the Services - including Comments, mentions, and trusted ratings.
>
> . Adding Samples to our database of known or potential malware (the VirusTotal "Corpus"), in order to continue to advance the security industry's understanding of online threats.

- The flatpak Raspirus is a simple virus scanner, using Yara rules (it uses several thousands rules, don’t expect a high detection rate). You can install it the following way:

> [code]
>
> flatpak install io.github.raspirus.raspirus

- The flatpak version of ClamTk, that includes ClamAV executable binaries, is no longer maintained:

* the runtime org.freedesktop.Platform/x86_64/22.08 is no longer maintained,

* ClamTk is no longer maintained,

* ClamAV executable binaries are obsolete (1.1.1 version instead of 1.4.2)

→ The corresponding section has been removed from this document.

- Scanning locally confidential files can be done by following the tutorial in Annex 12: On-Demand Scan of Confidential Files.

Messages and attachment scans:

- Finally, your mail provider might offer a free or paid antivirus service; with that service, your mails and attachments would be scanned for viruses on your mail provider servers.

5.2) Intrusion detection

The recommendations in Prevention chapter, when applied, do prevent system intrusion and reduce its probability.

However, an intrusion, though very improbable, is always possible; would it occur, it should be detected.

There are several tools with some intrusion detection capabilities; the most often mentioned are:

- OSSEC and OSSEC+, a scalable, multi-platform, open source Host-based Intrusion Detection System (HIDS) https://www.ossec.net/;

- Fail2ban scans log files (e.g. /var/log/apache/error_log) and bans IPs that show the malicious signs -- too many password failures, seeking for exploits, etc.

https://www.fail2ban.org/wiki/index.php/Main_Page;

- Snort, a network intrusion prevention and detection software, https://www.snort.org/;

- Suricata, a high performance, open source network analysis and threat detection software used by most private and public organizations, https://suricata.io/;

- Zeek, a free and open-source software network analysis framework, https://zeek.org/;

- Security-Onion, a free and open Linux distribution for threat hunting, enterprise security monitoring, and log management, https://securityonionsolutions.com/software;

- Samhain, a host-based intrusion detection system (HIDS) provides file integrity checking and log file monitoring/analysis, as well as rootkit detection, port monitoring, detection of rogue SUID executable programs, and hidden processes, https://www.la-samhna.de/samhain/.

→ Most of those tools are enterprise ones, and require some admin knowledge.

Three firewalls have network intrusion detection (and prevention) capabilities:

- IPFire, a hardened, versatile, state-of-the-art Open Source firewall based on Linux. Its ease of use, high performance in any scenario and extensibility make it usable for everyone, https://www.ipfire.org/; it is available as a distribution, used to change an old computer in a firewall (the firewall computer is connected to internet and acts as a server for the other computers of the LAN).

- Portmaster, a free and open-source application firewall that does the heavy lifting for you. Restore privacy and take back control over all your computer's network activity, https://safing.io/.

- OpenSnitch, https://github.com/evilsocket/opensnitch, is a GNU/Linux interactive application firewall inspired by Little Snitch. It logs your connections and makes provisional rules that you can transform in permanent ones.

Note that any firewall logs can be used for network intrusion detection

Finally, system monitors have some network activity analysis capabilities, coupled to active processes list.

The network intrusion detection tool to choose depends on your system (single computer or small network).

Ideally, a tool with network intrusion detection capability should be used in conjunction with Tripwire, a system files change detection program.

At initialization, Tripwire creates an encrypted database with the cryptographic hashes of the system files. Then, at check, Tripwire compares the newly calculated hashes to the stored ones.

User knows what files have been changed or added; he has then to decide if the change is legit (he can use Synaptic history for that) or not. If the change is legit, user updates the database. If not, he reinstalls the changed files, and delete the added ones.

Tripwire can be useful to detect system files alterations by malware.

See Annex 9: Tripwire Tutorial.

6. Pre-Established Arrangements

"Noah built his Ark before the Flood"

6.1) Elaborate a recover strategy

You should be ready to recover a functional computer after anything wrong:

- hardware failure,

- operating system crash,

- successful attack on your computer.

For this, you should elaborate, as soon as possible, and even before installing your system, a recover strategy.

This strategy depends on:

- The time you are allowed, by yourself or others, to recover a fully operational computer:

* After a hardware failure, if you can wait days, you just note the references of parts that are more prone to fail (power supply, battery, battery charger, disks, fans, cables…) and you reorder them; if you can wait some hours only, you should buy some spares; if you can't wait more than minutes, you should have a second computer ready to work.

* After a software failure, it may be faster to restore a system backup than to make a complete reinstatement of the system.

- The complexity of the tasks needed to recover a fully operational computer, and people skill:

* Can you repair your system, in that case just download and print the necessary documentations.

* Or do you need assistance, in that case note the name and telephone number of a repairman, or even sign a maintenance contract.

This strategy should precise what to do in what occasion, how to dot it, and it should include all the hardware parts and software tools to use.

6.2) Backup and restore strategy

Backup and restore strategy should allow recovering a fully operational computer after any software problem (system crash, successful attack having compromised your system).

It should be thought before installation: by default, Linux Mint installs the operating system files "/", swap file and user home files "~/" or "/home/username" in the same partition; however, having them in separate partitions strongly facilitates system backup (it avoids to back up all when backing up the system).

→ A "minimal" preferable partitioning is "/", "/home/" and swap partitions.

What are the preferred hardware supports for backups?

You should avoid making backups on the same disk as your system: in case of disk crash, you would lose original files and their backups.

You should avoid making backups on another internal disk: in case of power supply over-voltage, disk controller failure, PCI bus failures (and more generally any common cause failure), you would lose original files and their backups.

You should avoid having your backup disk permanently mounted: in case of common cause failure, or user error, you would lose original files and their backups; in case of system compromising, you would increase chances that your backup is compromised; and you would increase disk wear, and reduce its life duration.

→ Best backup supports are removable one, external HDDs or SSDs, that are mounted for backup and restore only.

How many backup hardware supports?

Of course, the minimal answer is one. In that case, it should be used with caution, and stored apart from the computer (in case of computer steal or loss, the backup should still be available).

There are other possible strategies:

Three backup disks: one near the computer, that you can use in case of problem; in case of problem (user error) when using it, a second backup is used; if problem still persists, an expert will use the third backup, stored in a safe.

The "3,2,1,1,0" rule: the optimal backup strategy for your business continuity:

- 3 copies of your data.
- 2 different supports to avoid loss, corruption or piracy.
- 1 copy stored on a different location.
- 1 network isolated copy or air gapped.
- 0 data unusable after your tests (this implies you test your backups).

The more important and valuable are your personal files and data, the more expensive is the strategy you should use. Even a single home user could lose several years of his computer life (mails, documents, photos, music, videos) without adequate personal files backup.

6.3) Proposed minimum backup and restore strategy

Here is a proposal for a minimum backup and restore strategy.

1) System partitioning

3 partitions, "/", "/home" and swap.

2) Tools

* An external disk, fast (USB3 or USB-C, SSD or high rpm HDD), whose size is at least equal to the added sizes of "/" and "/home" partitions (larger is preferable).

* On a Ventoy USB key, see https://www.ventoy.net/en/index.html, copy the ISOs of your system installation live DVD (Linux Mint 21.3 Mate for me), Foxclone, see https://www.foxclone.org/ and System Rescue, see https://www.system-rescue.org/. (It may be a good idea to have two Ventoy keys, one being used and one as backup).

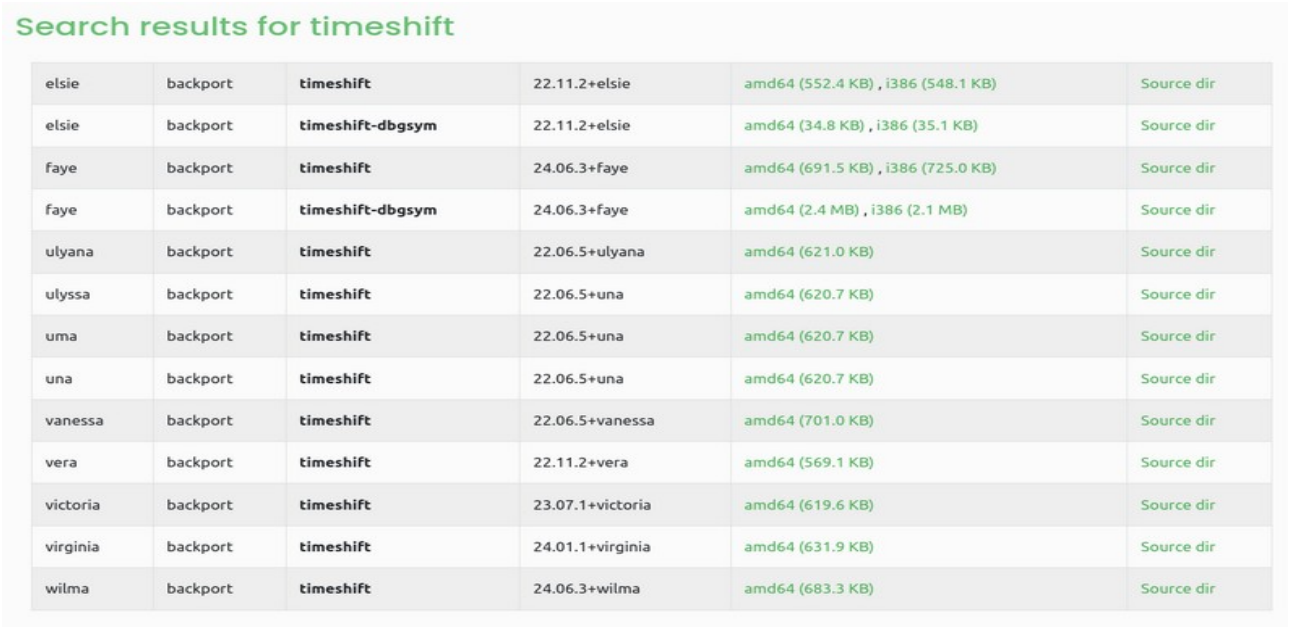
* On your operating system, install FreeFileSync, preferably its flatpak version *[since FreeFileSync reads / writes hundreds of thousands files, it offers a large surface attack; flatpak version reduces this security risk; caveat: FreeFileSync cannot be used for system files backup and restore]*:

[code]

flatpak install org.freefilesync.FreeFileSync

* On your operating system, install Timeshift, version 22.11.2 or later.

At the moment of writing this document, here are available Timeshift versions, from http://packages.linuxmint.com/search.php?release=any&section=any&keyword=timeshift:



Search results for timeshift

elsie	backport	**timeshift**	22.11.2+elsie	amd64 (552.4 KB) , i386 (548.1 KB)	Source dir
elsie	backport	**timeshift-dbgsym**	22.11.2+elsie	amd64 (34.8 KB) , i386 (35.1 KB)	Source dir
faye	backport	**timeshift**	24.06.3+faye	amd64 (691.5 KB) , i386 (725.0 KB)	Source dir
faye	backport	**timeshift-dbgsym**	24.06.3+faye	amd64 (2.4 MB) , i386 (2.1 MB)	Source dir
ulyana	backport	**timeshift**	22.06.5+ulyana	amd64 (621.0 KB)	Source dir
ulyssa	backport	**timeshift**	22.06.5+una	amd64 (620.7 KB)	Source dir
uma	backport	**timeshift**	22.06.5+una	amd64 (620.7 KB)	Source dir
una	backport	**timeshift**	22.06.5+una	amd64 (620.7 KB)	Source dir
vanessa	backport	**timeshift**	22.06.5+vanessa	amd64 (701.0 KB)	Source dir
vera	backport	**timeshift**	22.11.2+vera	amd64 (569.1 KB)	Source dir
victoria	backport	**timeshift**	23.07.1+victoria	amd64 (619.6 KB)	Source dir
virginia	backport	**timeshift**	24.01.1+virginia	amd64 (631.9 KB)	Source dir
wilma	backport	**timeshift**	24.06.3+wilma	amd64 (683.3 KB)	Source dir

Linux Mint 22.x users can have Timeshift version 24.06.3 or later with minimal version 22 of Linux Mint.

Linux Mint 21.x users can have Timeshift version 22.11.2 or later with minimal version 21.1 of Linux Mint.

Ubuntu 22.04 users can directly download this deb from Ubuntu Universe and install it:

http://archive.ubuntu.com/ubuntu/pool/universe/t/timeshift/timeshift_22.11.2-1_amd64.deb.

Linux Mint 20.x/Ubuntu 20.04 users can download the source from Ubuntu Universe and compile it: http://archive.ubuntu.com/ubuntu/pool/universe/t/timeshift/timeshift_22.11.2.orig.tar.gz

Once downloaded, uncompress it; with file manager, open the uncompressed directory "/timeshift-22.11.2", then launch a terminal in this directory and execute the following code:

[code]

Dependencies installation

sudo apt install make gettext valac libvte-2.91-dev libgee-0.8-dev libjson-glib-dev

Make

make all

Old Timeshift version uninstall

sudo apt remove timeshift

Final installation

sudo make install

As usual, there are lots of warnings during "make" and "make install" steps, ignore them.

To uninstall this version, once compilation and installation is done, save the "/timeshift-22.11.2" directory; in "/timeshift-22.11.2/src" you will find a "timeshift-uninstall" file, you can uninstall Timeshit by executing this file with superuser rights:

[code]

sudo timeshift-uninstall

Why to use at least Timeshift "22.11.2" and not former versions?

TimeShift is not reliable, and has several annoying bugs:

file "info.json" may be corrupted or missing, this seems to be corrected in "22.11.2" version, see https://github.com/linuxmint/timeshift/issues/108 *[former versions users should use the workaround mentioned in issue 108],*

on Linux Mint 21.x / Ubuntu 22.04, a rsync update breaks TimeShift and prevents it to work, see https://github.com/linuxmint/timeshift/issues/152,

Timeshift issues, ~203 open ones when writing this document, can be followed at

https://github.com/linuxmint/timeshift/issues.

Timeshift should not be considered as a system backup tool, since it is not reliable, and since it is not always able to recover a functional system after a crash, see:

https://forums.linuxmint.com/viewtopic.php?t=331605

Moerover, Timeshift snapshots do not contain all the directories found in the computer "/". The following directories are excluded from the snapshots, in Timeshift sources:

https://raw.githubusercontent.com/linuxmint/timeshift/master/src/Core/Main.vala

```
exclude_list_default.add("/dev/*");
exclude_list_default.add("/proc/*");
exclude_list_default.add("/sys/*");
```

```
exclude_list_default.add("/media/*");
exclude_list_default.add("/mnt/*");
exclude_list_default.add("/tmp/*");
exclude_list_default.add("/run/*");
exclude_list_default.add("/var/run/*");
exclude_list_default.add("/var/lock/*");
//exclude_list_default.add("/var/spool/*");
exclude_list_default.add("/var/lib/dhcpcd/*");
exclude_list_default.add("/var/lib/docker/*");
exclude_list_default.add("/var/lib/schroot/*");
exclude_list_default.add("/lost+found");
exclude_list_default.add("/timeshift/*");
exclude_list_default.add("/timeshift-btrfs/*");
exclude_list_default.add("/data/*");
exclude_list_default.add("/DATA/*");
exclude_list_default.add("/cdrom/*");
exclude_list_default.add("/sdcard/*");
exclude_list_default.add("/system/*");
exclude_list_default.add("/etc/timeshift.json");
exclude_list_default.add("/var/log/timeshift/*");
exclude_list_default.add("/var/log/timeshift-btrfs/*");
exclude_list_default.add("/swapfile");
exclude_list_default.add("/snap/*");
```

If you can add excluded directories using Timeshift GUI, you cannot remove from exclusion list the ones excluded in the source. The full list of excluded directories is found in the "exclude.list" file of each snapshot.

The excluded directories are normally not backed up in a snapshot, and not restored. If you use snaps, a bug has a catastrophic consequence: your "/snap/" directory is not backed up, and a restore of any snapshot will distroy your snaps installation.

See: https://github.com/linuxmint/timeshift/issues/179

Finally, Timeshift use is a security risk: Timeshift reads / writes hundreds of thousands files, in the operating system. It so offers a large attack surface. Can this risk be mitigated?

- Timeshift needs to write system files, including kernel and GRUB launcher. This prevents any kind of sandboxing.

- Timeshift would be used more securely when executed from a live DVD / USB or ISO image.

Risk mitigation when using it from the operating system to back up would imply a stable, reliable Timeshift, with quick resolution of bugs. We are very far from that.

However, Timeshift is a nice tool, when it works, and when user is not affected by its bugs. It can so be used for system snapshots, but not alone: if Timeshift fails to restore a system snapshot, Foxclone will allow restoring the system from a former image.

[Of course, the security risk does not exist with Foxclone: it is launched from an ISO (immutable), without the need to access internet, running on its own operating system (a small Ubuntu 18.04 or 20.04), using sectors and not files].

3) Settings

Using Gnome Disks, write caching should be disabled on all disks, internal and external; write caching is a vestige of a time when disks were slow, it DOES NOT speed writing on disks, and is a very good way to lose data (on a power failure, on a too early disk removal).

4) User files strategy

Use FreeFileSync to backup files from your home to the external disk. FreeFileSync is very fast, and it works differentially: on the first time all files are backed up, then only what has changed is backed up (use "Mirror" setting).

Periodicity: once a day, or once every two days.

Use FreeFileSync to restore them, or use your file manager if FreeFileSync is not yet installed on the computer you restore.

FreeFileSync Manual: https://freefilesync.org/manual.php.

5) System files backup strategy

* Boot on Ventoy key, choose Foxclone ISO and make an image of your system that you save on external disk.

Periodicity: once a week, or once every two weeks.

Keep one image, the latest.

Foxclone User Guide: https://www.foxclone.org/uguide.html, also found on ISO.

* When your operating system is running, make a Timeshift snapshot.

Periodicity: once a day, or once every two days (manual snapshot, no scheduling, since I recommend using an external disk).

Keep the latest two snapshots.

Timeshift documentation: https://www.fossmint.com/backup-restore-linux-with-timeshift/.

6) System files restore strategy

* If your operating system can run, restore the latest known good system snapshot with Timeshift.

* If your operating system cannot run, boot on Ventoy key, choose Linux Mint ISO and restore the latest known good system snapshot with Timeshift.

* If this does not work, boot on Ventoy key, choose Foxclone ISO and restore the system from its backed up image; once done, boot on the restored Linux Mint and restore the latest known good system snapshot with Timeshift.

7) Other recovering problems

System Rescue can be used to recover from lots of problems. It includes the following main tools, and much more (from its website):

> GNU Parted: creates, resizes, moves, copies partitions, and file systems (and more).
>
> GParted: GUI implementation using the GNU Parted library.
>
> FSArchiver: flexible archiver that can be used as both system and data recovery software.
>
> ddrescue : Attempts to make a copy of a block device that has hardware errors, optionally filling corresponding bad spots in input with user defined pattern in the copy.
>
> File systems tools (for Linux and Windows file systems): format, resize, and debug an existing partition of a hard disk.
>
> Test-disk : tool to check and undelete partition, supports reiserfs, ntfs, fat32, ext3/ext4 and many others.
>
> Memtest: to test the memory of your computer (first thing to test when you have a crash or unexpected problems).
>
> Rsync: very-efficient and reliable program that can be used for remote backups.
>
> Network tools (Samba, NFS, ping, nslookup, …): to back up your data across the network.

System Rescue Manual: https://www.system-rescue.org/manual/.

System Rescue Book: https://www.system-rescue.org/Books/.

Annex 1: Launching Commands and GUI Applications with Superuser Rights

1) Sudo

Command "sudo" can be used to launch commands, within a terminal window, with superuser rights. It applies only to commands, not to GUI applications.

An example: you want to install "firefox" package, with "apt install" command. You launch a terminal, then, in the terminal window you enter the following line of code:

[code]

sudo apt install firefox

When you hit the "Enter" key after having written the command line, you will be prompted to enter a password; once this password entered, the command will be executed. It will not be executed without entering the password. And, if the entered password is wrong, you will be asked to enter it again. The password is the one that has been defined during system installation. It can be changed with:

[code]

sudo passwd username

where "username" is YOUR user name; you will then be prompted to enter actual, old, password and then to enter twice the new password.

2) Su

Command "su" is used to enter superuser mode, within a terminal window. Once done, it replaces the usual prompt "username@computername:~$" by "root@computername:/home/username#" and all the commands executed in the terminal window are executed with superuser rights without the need to add "sudo" before the command.

You can use "su" command with "sudo" one:

[code]

sudo su

and enter your superuser password.

Or you can enable root account:

[code]

sudo passwd root

and enter once your superuser password and twice the root password you choose.

Once done, you can enter root mode in a terminal with:

[code]

su

and enter your root password.

Note that this is a controversial topic: some people recommend to not enable root account, with the main argument that "if root account has no password, it cannot be guessed by an attacker".

This argument is wrong: if the attacker guesses the superuser password, he can become root with "sudo su" or enable the root account with "sudo passwd root"; one password guess = superuser + root access. With root password set, and different as superuser one, attacker needs two good guesses to have superuser and root access.

3) Graphical applications

Formerly, "gksudo" and "gksu" commands played the same role for graphical applications than "sudo" and "su" for commands. However, they have been deprecated and are no longer available.

Here are several solutions to launch a graphical application with super-user privileges:

- Use of "pkexec" command:

Some GUI applications whose use necessary implies superuser rights are automatically launched with a window asking to enter superuser password (an example is Synaptic).

Some applications, for which a Polkit policy exist, can be launched with "pkexec" command.

Example:

[code]

pkexec caja

opens a window asking to enter superuser password and, once password has been correctly entered, launches the Mate file manager "caja" with superuser privileges.

The available Polkit policies are found in "/usr/share/polkit-1/actions"; example, the existence of "org.freedesktop.caja-admin.policy" file shows that a policy exists to launch "caja" with superuser privileges.

When such a policy exists for an application, "pkexec" is the preferable way to launch it with superuser privileges.

A graphical application without Polkit policy can be launched with "pkexec", once a Polkit policy has been written for it.

[This is out of the scope of this tutorial, see next paragraph for a simple workaround].

- You can create the missing "gksudo" command as an alias of "pkexec" with a generic pseudo Polkit policy.

Create a file ".bash_aliases" that you will save in your home "~/" with the following content:

[code]

alias gksudo='pkexec env DISPLAY=$DISPLAY XAUTHORITY=$XAUTHORITY'

At each system launch, ".bashrc", also found in your home "~/" will execute your ".bash_aliases" and create the aliases it contains, here "gksudo".

You can now launch any GUI application with "gksudo" alias, equivalent to "pkexec env DISPLAY=$DISPLAY XAUTHORITY=$XAUTHORITY" command; an example:

[code]

gksudo xed /etc/hosts

will open "xed" GUI text editor with superuser privileges and edit "/etc/hosts" file.

- You can launch GUI text editors with "admin://" prefix, before the file full path.

Example:

[code]

xed admin:///etc/hosts

will open "xed" GUI text editor with superuser privileges and edit "/etc/hosts" file.

[Because of an uncorrected bug, you will be asked to enter your superuser password twice].

- You can use "gksudo2" command, a replacement of deprecated "gksudo".

See https://github.com/furryfixer/gksudo2.

Annex 2: Password Protect your GRUB Menu

GRUB menu can be easily hacked to gain root privileges. Attacks are explained here:

Booting from live DVD:

https://www.cyberciti.biz/tips/howto-recovering-grub-boot-loader-password.html

Booting from computer disk:

http://3wymlmcsvxiaqzmbepsdawqpk6o2qsk65jhms72qqjulk5u4bgmvs3qd.onion/grub/boot_from_command_prompt (onion link, in Tor Network)

https://www.tecmint.com/how-to-hack-your-own-linux-system/

The solution is derived from this source: https://linuxconfig.org/set-boot-password-with-grub.

The solution against those attacks is to:

- password protect your GRUB menu,

- change the boot order in your computer UEFI/BIOS settings to have HDD/SSD first,

- and password protect access to your UEFI/BIOS settings,

- and password protect your computer boot with another password (to protect against USB key boot with BIOS computers).

[Note that this will induce a restraint when you will want to boot on CD-ROM, DVD-ROM or USB key to use external tools such as system backup; you will be obliged to access your UEFI/BIOS, change boot order, boot on your removable media, use the tools, then think to change back boot order once you have finished].

1) First, a reminder on GRUB menu

GRUB menu may be visible or not.

If GRUB menu is not visible, you can see it once, at boot time:

if your system uses BIOS, press and hold the shift key while the system is booting,

if your system uses UEFI, press and hold the ESC key while the system is booting.

[Note that you may have conflicts with your computer, Esc might be used to show the BIOS / UEFI menu].

If you want to see it permanently (my preferred solution):

* once your system is running, edit your GRUB file, type in a terminal:

[code]

sudo nano /etc/default/grub file

* in front of the line "GRUB_TIMEOUT_STYLE=hidden", add a "#" character to comment it, add a non zero value at the end of the line "GRUB_TIMEOUT=" (as an example, if the value is 5,

GRUB menu will display during 5 seconds), save your modified GRUB file by "CTRL+O" and leave nano by "CTRL+X".

Here is an example of edited GRUB file, edited lines are in bold:

[code]

If you change this file, run 'update-grub' afterwards to update

/boot/grub/grub.cfg.

For full documentation of the options in this file, see:

info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0

#GRUB_TIMEOUT_STYLE=hidden

GRUB_TIMEOUT=5

GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`

GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"

GRUB_CMDLINE_LINUX=""

Uncomment to enable BadRAM filtering, modify to suit your needs

This works with Linux (no patch required) and with any kernel that obtains

the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)

#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

Uncomment to disable graphical terminal (grub-pc only)

#GRUB_TERMINAL=console

* Update GRUB:

[code]

sudo update-grub

* Restart your system, GRUB menu is there.

2) Password protect your GRUB menu

* Get started by opening a command line terminal and typing the following command to generate a password hash.

[code]

grub-mkpasswd-pbkdf2

You will be prompted to enter twice the GRUB password you want, and then a hash will be output to your terminal.

Here is a typical output:

[output]

Enter password:

Reenter password:

PBKDF2 hash of your password is

grub.pbkdf2.sha512.10000.EB3855C01C410FAD7D0F45FECCC9A62A1329D77FE03F155D1BA69405D998856CC7F9EAD29A718EBE039E3863945CB752B972E48B17C3B517C4646DB5A0EFED55.025840A2EF34B8D1F6C607EBF33FB3344872D6C26A3327F126BCB0CD15845AE6731459C11CA571ECBCCD5E9C34EBBEF27E3C6E295470F3B19E6661EC080E6A42

In bold is the complete password hash. Copy it to your clipboard (this includes the part that starts with "grub.").

* Next, we will make some edits to the "/etc/grub.d/00_header" GRUB configuration file. Use nano or your preferred text editor to open this file with root privileges.

[code]

sudo nano /etc/grub.d/00_header

At the bottom of this file, you will need to paste the following code, while replacing username with the name of your user account, and replacing INSERT-HASH with the password hash that you generated earlier.

[code]

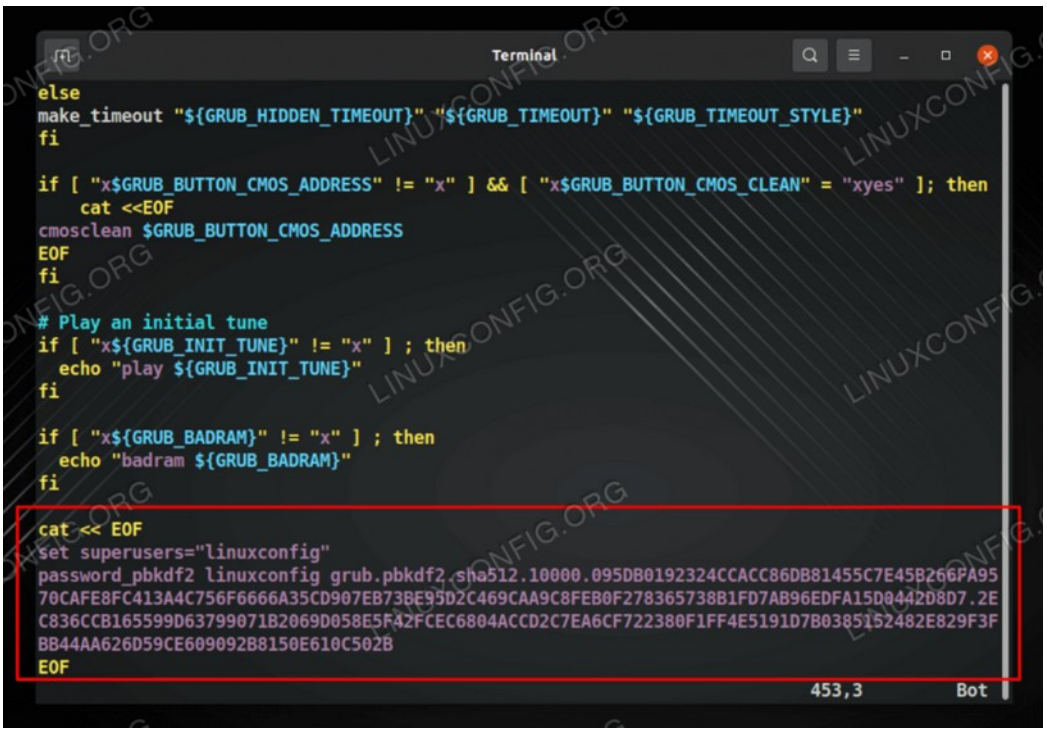
cat << EOF

set superusers="username"

password_pbkdf2 username INSERT-HASH

EOF

The following screen capture is an example of modified "/etc/grub.d/00_header".



* After you have made the change from the previous step, you can exit and save your changes to the GRUB configuration file. After that, execute the update-grub command with root privileges for the GRUB password settings to take effect.

[code]

sudo update-grub

* Restart your system and make sure that you are presented with a prompt for your GRUB password, and enter your password (its hash will be computed and compared with the stored one; if they match you will access GRUB menu).

3) Boot order

Finally, change the boot order in your computer UEFI/BIOS settings to have HDD/SSD first, and password protect access to your UEFI/BIOS settings.

This would prevent an attacker to boot from a live DVD or live USB and access to your "/etc/grub.d/00_header" and remove the password protection.

Annex 3: Password Selection

Home Security Heroes have published a web page dedicated to password cracking using an AI, https://www.homesecurityheroes.com/ai-password-cracking/.

The result is this table, showing time need to crack a password of given length and characteristics:

# OF CHARACTER	Numbers Only	Lowercase Letters	Lowercase Upper & Letters	Numbers, Upper & Lowercase Letters	Numbers, Upper & Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	4 Seconds
7	Instantly	Instantly	22 Seconds	42 Seconds	6 Minutes
8	Instantly	3 Seconds	19 Minutes	48 Minutes	7 Hours
9	Instantly	1 Minutes	11 Hours	2 Days	2 Weeks
10	Instantly	1 Hours	4 Weeks	6 Months	5 Years
11	Instantly	23 Hours	4 Years	38 Years	356 Years
12	25 Seconds	3 Weeks	289 Years	2K Years	30K Years
13	3 Minutes	11 Months	16K Years	91K Years	2M Years
14	36 Minutes	49 Years	827K Years	9M Years	187M Years
15	5 Hours	890 Years	47M Years	613M Years	14Bn Years
16	2 Days	23K Years	2Bn Years	26Bn Years	1Tn Years
17	3 Weeks	812K Years	539.72M Years	2Tn Years	95Tn Years
18	10 Months	22M Years	7.23Bn Years	96Tn Years	6Qn Years

The method used is "brute-force attack", passwords are generated by AI and tested until one matches.

The result is the following: in 2023, a secure password is 11 characters long, with characters randomly taken from numbers, upper and lower case letters, symbols.

Here is such a secure password: "X14an/)UV12".

This kind of password is efficient, but humans have difficulties to remember them, when the sequence of characters is really random and shows no pattern.

Moreover, as soon as we speak of encryption, it becomes still more complicated. One of the most frequently used encryption algorithm is "AES256". It requires a random key of 256 bits. Here is an example of use.

Suppose AES256 is used to encrypt a disk, a partition, or a container. The user chooses a password. The encryption program generates a random 256 bits key and encrypt the disk. To decipher the disk, the user enters the password, used to recover the encryption key, then the disk is deciphered with the key.

Password should have the same strength as encryption key: 256 bits, or 37 characters from ASCII 128 table:

0	*NUL*	16	*DLE*	32	SPC	48	0	64	@	80	P	96	`	112	p
1	*SOH*	17	*DC1*	33	!	49	1	65	A	81	Q	97	a	113	q
2	*STX*	18	*DC2*	34	"	50	2	66	B	82	R	98	b	114	r
3	*ETX*	19	*DC3*	35	#	51	3	67	C	83	S	99	c	115	s
4	*EOT*	20	*DC4*	36	$	52	4	68	D	84	T	100	d	116	t
5	*ENQ*	21	*NAK*	37	%	53	5	69	E	85	U	101	e	117	u
6	*ACK*	22	*SYN*	38	&	54	6	70	F	86	V	102	f	118	v
7	*BEL*	23	*ETB*	39	'	55	7	71	G	87	W	103	g	119	w
8	*BS*	24	*CAN*	40	(	56	8	72	H	88	X	104	h	120	x
9	*HT*	25	*EM*	41	)	57	9	73	I	89	Y	105	i	121	y
10	*LF*	26	*SUB*	42	*	58	:	74	J	90	Z	106	j	122	z
11	*VT*	27	*ESC*	43	+	59	;	75	K	91	[	107	k	123	{
12	*FF*	28	*FS*	44	,	60	<	76	L	92	\	108	l	124	|
13	*CR*	29	*GS*	45	-	61	=	77	M	93	]	109	m	125	}
14	*SO*	30	*RS*	46	.	62	>	78	N	94	^	110	n	126	~
15	*SI*	31	*US*	47	/	63	?	79	O	95	_	111	o	127	DEL

And still more characters, if some ones from this table are not used, because not easy to type.

If an 11 random characters long password was difficult to remember, a 37 characters long one is almost impossible to remember. Are they other methods?

1) Use a passphrase

One of the methods is to use a passphrase. A passphrase, done with several words, is easier to remember than a password.

Suppose you use a 5 words long passphrase, with words randomly selected from 5 merged dictionaries (English/French/German/Italian/Spanish); such a passphrase could be:

"hablamosbaguettetodayachtungvenerdi".

Once you have realized it is done with the words "hablamos", "baguette", "today", "achtung", "venerdi", it seems easy to remember.

What is its strength?

A first approach could be to count the letters (35 in this example), and say that there are 26^{35} possibilities, so a strength of log(26^{35})/log(2) ~ 164.5 bits.

However, this approach is wrong: you have not selected 35 randoms characters, but 5 random words. Suppose you have merged 5 dictionaries with ~ 60,000 words each, you have a dictionary of 300,000 words, and the number of possibilities is $300{,}000^5$.

So, its strength is log($300{,}000^5$)/log(2) ~ 91 bits. This is equivalent to the strength of a 13 characters long password, randomly selected in ASCII 128 table. This is acceptable for the requirements of a password in 2023.

But it is far to be enough to have the same strength as a 256 bits encryption key. You would need a passphrase long of 14 words, randomly selected from a 300,000 words dictionary to password protect a 256 bits long encryption key.

14 words… it begins to be difficult to remember!

Why is the first approach to calculate strength wrong? Modern cryptography is not based on obscurantism, and you should always consider that an attacker knows the method you use. *[Kerckhoffs' principle: the security of a cryptosystem should only be based on the secrecy of the key, not on the secrecy of the method or algorithm]*.

The sad conclusion resulting from those considerations is that the use of a password only is an obsolete technology, particularly when used in conjunction with encryption software.

2) What to do else?

- You can defeat the brute-force attack by increasing progressively time between two failed trials: this is what is implemented in my computer UEFI/BIOS when I have enabled a boot password protection; after 3 or 4 failed trials, I have to wait a few minutes before a new attempt; and this can grow up to an hour.

- Another way is to limit the maximum numbers of attempts: this is implemented in iPhones, user can set a maximum number of trials of 10. When the maximum number is reached, iPhone is unusable (even for owner knowing the password…). I don't think this can be implemented today in a Linux computer.

- Two or multiple factors authentication is the best possibility.

* With a computer equipped with a smartcard reader, you can use a strong key to password protect the computer access or to encrypt it; then the key is itself encrypted on a smartcard, protected by a password that human can remember; and the smartcard is NOT stored with the computer.

An attacker having access to the computer without the smartcard, trying to brute-force attack the key, will have to test all the possible 256 bits keys, it would take in 2023 some 95 trillion years. While user with his smartcard will just enter his password, then the key will be deciphered and used to access the computer or to decipher disk.

* Another possibility is to use both password and fingerprint, on a computer with fingerprint reader.

* More generally you can combine something the user knows (password), something he owns (smartcard, USB key), and/or something that characterizes him (fingerprint, eye iris).

- Two-factor authentication is also used on e-commerce websites and online payments.

* Connection to a website: user identifies himself by his username and password, then a one time code is sent to its mailbox or to its smartphone using SMS; user then has to enter the received OTC to finalize the connection. This kind of two factors authentication can be enabled on Amazon, eBay and others.

* Online payments: in EU two factors authentication is mandatory for online payments using credit cards; once he has entered his credit card owner name, number, date of expiration and cryptographic code, the user is directed to his bank payment server and a supplemental verification is done (with my bank, I have to complete authentication with my bank app on my smartphone, and a secret pin or fingerprint).

Annex 4: Encryption

The main encryption uses are:

- encrypt a disk, a partition, or a container,

- encrypt a file, a message,

- sign a file, a message.

1) Symmetrical encryption

The first encryption method is symmetrical encryption: an encryption algorithm is chosen, say AES256 (Advanced Encryption Standard with 256 bits key). Then, an encryption key (256 bits, or 16 bytes) is generated or chosen, and this key is used to encrypt a file.

The algorithm is symmetrical, since the key used to encrypt the file is also used to decipher it, once encrypted.

Symmetrical encryption is included in any encryption program.

But, when you want to send an encrypted message or file to a recipient, there is a problem: how to send him securely the decryption key?

Asymmetrical encryption solves the problem.

2) Asymmetrical encryption

The sender, John Doe, and the recipient, Jane Doe, will generate each one a pair of keys; each pair contains a public key and a private one. John exports his public key and sends it to Jane; Jane exports her public key and sends it to John.

When John wants to send an encrypted message to Jane, he first uses symmetrical encryption, then he encrypts the key with Jane's public asymmetrical key, and sends encrypted message and encrypted key to Jane.

When Jane receives the message and the key, she deciphers the key with her own private key, and uses the deciphered key to decipher the message.

When Jane answers to John, she will use John's public key to encrypt the symmetrical key, and John will be able to decipher the key with his own public key, then decipher the message.

Asymmetrical encryption is also used as a signature algorithm, in conjunction with a hash algorithm.

When John wants to send a message to Jane he wants that Jane can be sure that he, John, was the sender, and that the content of the message has not been altered during internet transfer.

He writes his message, then he takes a cryptographic hash of this message (a cryptographic hash changes totally even if one character of the message is changed, and it cannot be reversed without brute-force attack). John encrypts the hash with his own private key, this is the signature, and sends the message and the signature to Jane.

Jane receives the message and the signature, she computes the hash of the message, then uses John's public key to decipher the signature and get the original hash; if both hashes are identical, the message has not been altered, and John is the sender.

Jane, of course, can also sign the messages she sends to John with her private key, and John will use Jane's public key to confirm the message is an original, unaltered one.

The same thing can be done to a file: a file can be signed, in the same way as a message.

Of course, encryption and signature can be combined.

OpenPGP and S/MIME are two different standards that use asymmetrical encryption. The main difference concerns key generation: with OpenPGP, users generate their own public/private key pair; with S/MIME a certificate is issued by a certification authority (CA). This certificate contains a key pair, and this key pair is signed by the CA, providing an identity authentication, or at least a mail address authentication.

3) What are the programs to use?

- To encrypt a container, the program to use is Veracrypt. It can be downloaded from https://veracrypt.eu/en/. Linux Mint/Ubuntu users will choose the deb package for their distribution (Linux Mint 20.x users will choose the Ubuntu 20.04 deb; Linux Mint 21.x users will choose the Ubuntu 22.04 deb) and install it with "gdebi" (in the file manager, select the downloaded deb, then right click and "open with gdebi package installer").

A full documentation is available online, at https://veracrypt.eu/en/Documentation.html.

- To encrypt files and messages for a recipient public key, or to sign a message, there are two possibilities:

* The first one is to use Thunderbird; it includes an OpenPGP and S/MIME compliant encryption and signature module.

Here is a documentation about the OpenPGP module:

https://support.mozilla.org/en-US/kb/openpgp-thunderbird-howto-and-faq

* The second possibility is to use Gnu Privacy Guard (or GnuPG). The "2.x" versions are compliant with both OpenPGP and S/MIME.

[The "1.4" version is an insecure one, using some obsolete insecure algorithms, just kept for compatibility reasons; if you still have old encrypted mails or attachment, you will need this version and your old keys to decipher them].

Linux Mint 21.x / Ubuntu 22.04 LTS have an obsolete version of GnuPG in their repositories. Their users can install Richard Hansen backport of gnupg2 PPA and get a recent gnupg version, see https://launchpad.net/~rhansen/+archive/ubuntu/gnupg2.

Latest GnuPG version can be downloaded from https://gnupg.org/, more precisely from the downloads page:

https://gnupg.org/download/index.html.

Formerly, gnupg.org provided a ready to use GnuPG Desktop AppImage. It is no longer the case; it is now necessary to download the latest source version and to compile it.

You need then to install a GUI, I advise using Kleopatra.

If you use Linux Mint 20.x or 21.x, it may be preferable to install kubuntu ppa backports, https://launchpad.net/~kubuntu-ppa/+archive/ubuntu/backports, in order to have a more recent version of Kleopatra and KDE dependencies:

[code]

sudo add-apt-repository ppa:kubuntu-ppa/backports

sudo apt update

Kleopatra installation:

[code]

sudo apt install kleopatra

Kleopatra English manual: https://docs.kde.org/stable5/en/kleopatra/kleopatra/index.html.

Note that GnuPG keys, once generated with Kleopatra, are stored in "~/.gnupg/".

4) Sandboxing Kleopatra and GnuPG binaries

Kleopatra and GnuPG security can be enhanced by sandboxing. This can be done using Firejail.

Download the latest version of Firejail from https://sourceforge.net/projects/firejail/files/firejail/. At the time of writing this document, the latest version is "firejail_0.9.74_1_amd64.deb".

* Install Firejail with:

[code]

replace firejail deb name by the one you downloaded in the following command

gdebi firejail_0.9.74_1_amd64.deb

* Firejail does not include a profile for Kleopatra. However, it includes one for GPA, another GnuPG GUI. This allows to derive the following "kleopatra.profile":

[code]

Firejail profile for Kleopatra

Description: GnuPG GUI

Persistent global definitions

include globals.local

noblacklist ${HOME}/.gnupg

noblacklist ${HOME}/.config

noblacklist ${HOME}/.local/share/kleopatra

include disable-common.inc

include disable-devel.inc

include disable-interpreters.inc

include disable-programs.inc

caps.drop all

netfilter

nodvd

nogroups

noinput

nonewprivs

noroot

nosound

notv

nou2f

novideo

protocol unix,inet,inet6

seccomp

tracelog

private-dev

restrict-namespaces

Copy this code, paste it in a text editor and save it in "~/.config/firejail/kleopatra.profile", where "~" is a system shortcut for "/home/**username**/".

Once done, we have to link "kleopatra", "gpg" and "gpg-agent" to Firejail:

[code]

sudo ln -s /usr/bin/firejail /usr/local/bin/kleopatra

sudo ln -s /usr/bin/firejail /usr/local/bin/gpg

sudo ln -s /usr/bin/firejail /usr/local/bin/gpg-agent

From now, "kleopatra", "gpg" and "gpg-agent" will always be executed in Firejail sandboxes, and security is greatly improved.

Annex 5: How to Enable Ubuntu Pro on Linux Mint

1) Introduction

Linux Mint versions have a limited maintenance duration of 5 years.

Ubuntu Pro is available for Ubuntu 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 24.04 LTS and extends in time and in depth the maintenance support, with better reactivity.

Any individual Ubuntu user can subscribe for free to Ubuntu Pro, and receive a token valid for up to 5 computers, for personal use.

What are Ubuntu Pro advantages? They are explained here https://ubuntu.com/pro.

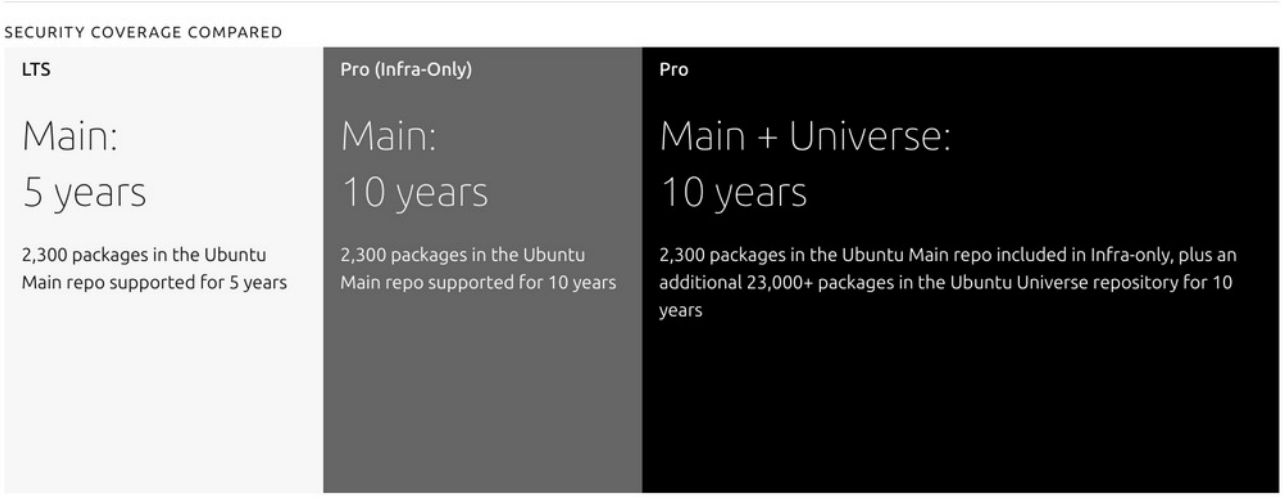
In summary, they are:

* maintenance duration 10 years with Ubuntu Pro instead of 5 years with Ubuntu (up to April 2026 for 16.04 LTS / Linux Mint 18.x; up to April 2028 for 18.04 LTS / Linux Mint 19.x; up to April 2030 for 20.04 LTS / Linux Mint 20.x; up to April 2032 for 22.04 LTS / Linux Mint 21.x; up to April 2034 for Ubuntu 24,04 LTS / Linux Mint 22.x),

* 2300 packages from main Ubuntu repository + 23000 packages from Universe are maintained with Ubuntu Pro instead of 2300 packages from main Ubuntu repository with Ubuntu,

* Live kernel patch, allowing to update kernel without interrupting a session (mainly interesting for servers, or after the 5 years of initial maintenance),

* CVE fix after one day with Ubuntu Pro, instead of up to 98 days with Ubuntu.



This is interesting for:

- Linux Mint 18.x and 19.x users, whose distributions are no longer maintained,

- Linux Mint 20.x, 21.x and 22.x users, whose distributions are still maintained, but who will take profit of 23000 more maintained packages, live kernel patch and quick CVE fix.

At the moment of writing (2023/05), Linux Mint users can attach a computer to Ubuntu Pro, but they cannot enable Ubuntu Pro services: Linux Mint distributions are not recognized as valid Ubuntu ones, though Linux Mint distributions are based on Ubuntu LTS ones.

That's why this tutorial shows how to enable Ubuntu Pro on Linux Mint 18.x / 19.x / 20.x / 21.x / 22.x.

2) Warning

For Linux Mint 18.x and 19.x users, in decreasing order of precedence, it is preferable to:

* Upgrade to a newer version of Linux Mint, 20.x or 21.x.

* Install Ubuntu 16.04 LTS (instead of Linux Mint 18.x) or Ubuntu 18.04 LTS (instead of Linux Mint 19.x) and enable Ubuntu Pro the official way, for in that case the whole distribution will be maintained.

* Follow this tutorial, for in that case only the Ubuntu part of the distribution will be maintained, not the Linux Mint part of the distribution. With time, some uncorrected weaknesses in Linux Mint applications might become critical security breaches, and some incompatibilities might appear with Ubuntu still maintained applications.

(This does not apply now to Linux Mint 20.x, 21.x and 22.x, still maintained)

This tutorial shows an unofficial and unsupported way to enable Ubuntu Pro on Linux Mint.

What follows is a hack and, though working, is not a satisfactory way to have Ubuntu Pro works: it is unsupported, it might stop to work after an update, and might break things.

So, I don't give any advice about using or not using the tutorial: each user has to make his own opinion, after having carefully read all this post.

Before to follow the tutorial, I strongly recommend updating your system and making a full backup of your system and of your home.

3) External links

Ubuntu Pro presentation: https://ubuntu.com/pro

Ubuntu Pro installation official tutorial: https://ubuntu.com/pro/tutorial

Ubuntu Pro discourse: https://discourse.ubuntu.com/c/ubuntu-pro/116

Thanks to Pjotr: https://easylinuxtipsproject.blogspot.com/p/faq-3.html#ID16

and https://easylinuxtipsproject.blogspot.com/p/ubuntuplus.html

4) Subscribe to Ubuntu Pro

Follow the official Ubuntu Pro installation tutorial https://ubuntu.com/pro/tutorial, up to the moment you have got an Ubuntu Pro token. Save this token in a convenient place, you will need to use it later.

5) Have your system recognized by Ubuntu Pro client tool

Open your file manager, go to "/etc", have a look at "os-release" file. It may be a shortcut, or a file; right-click on it and read its properties:

* if it is a link, you will have a "link target: ../usr/lib/os-release",

* if it is a real file, you will have a "location: /etc".

For what follows, we assume, as it is in most cases, that the real file is "/usr/lib/os-release"; if not, replace by "/etc/os-release").

Make a copy of this file:

[code]

sudo cp /usr/lib/os-release /usr/lib/os-release_orig

Edit os-release:

[code]

xed admin:///usr/lib/os-release

If you use Linux Mint 18.x, replace the file content by (and save it):

[code]

NAME="Ubuntu"

VERSION="16.04 LTS (Xenial Xerus)"

ID=ubuntu

ID_LIKE=debian

PRETTY_NAME="Ubuntu 16.04 LTS"

VERSION_ID="16.04"

HOME_URL="https://www.ubuntu.com/"

SUPPORT_URL="https://help.ubuntu.com/"

BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"

PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"

VERSION_CODENAME=xenial

UBUNTU_CODENAME=xenial

If you use Linux Mint 19.x, replace the file content by (and save it):

[code]

NAME="Ubuntu"

VERSION="18.04 LTS (Bionic Beaver)"

ID=ubuntu

ID_LIKE=debian

PRETTY_NAME="Ubuntu 18.04 LTS"

VERSION_ID="18.04"

HOME_URL="https://www.ubuntu.com/"

SUPPORT_URL="https://help.ubuntu.com/"

BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"

PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"

VERSION_CODENAME=bionic

UBUNTU_CODENAME=bionic

If you use Linux Mint 20.x, replace the file content by (and save it):

[code]

NAME="Ubuntu"

VERSION="20.04 LTS (Focal Fossa)"

ID=ubuntu

ID_LIKE=debian

PRETTY_NAME="Ubuntu 20.04 LTS"

VERSION_ID="20.04"

HOME_URL="https://www.ubuntu.com/"

SUPPORT_URL="https://help.ubuntu.com/"

BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"

PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"

VERSION_CODENAME=focal

UBUNTU_CODENAME=focal

If you use Linux Mint 21.x, replace the file content by (and save it):

[code]

NAME="Ubuntu"

VERSION="22.04 LTS (Jammy Jellyfish)"

ID=ubuntu

ID_LIKE=debian

PRETTY_NAME="Ubuntu 22.04 LTS"

VERSION_ID="22.04"

HOME_URL="https://www.ubuntu.com/"

SUPPORT_URL="https://help.ubuntu.com/"

BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"

PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"

VERSION_CODENAME=jammy

UBUNTU_CODENAME=jammy

If you use Linux Mint 22.x, replace the file content by (and save it):

[code]

NAME="Ubuntu"

VERSION="24.04 LTS (Noble Numbat)"

ID=ubuntu

ID_LIKE=debian

PRETTY_NAME="Ubuntu 24.04 LTS"

VERSION_ID="24.04"

HOME_URL="https://www.ubuntu.com/"

SUPPORT_URL="https://help.ubuntu.com/"

BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"

PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"

VERSION_CODENAME=noble

UBUNTU_CODENAME=noble

6) Have mintsources work

Unfortunately, the change of os-release content breaks mintsources. When you launch it from a command line, once entered your admin password, you get the following message:

[output]

mintsources

LSB codename: 'focal'.

Version of base-files: '20.3.0'.

Your LSB codename isn't a valid Linux Mint codename.

Please check your LSB information with "lsb_release -a".

Or a similar one, depending on the version of Ubuntu LTS (xenial, bionic, focal, jammy, noble).

Here is a workaround, allowing to have mintsources work.

In "/usr/share/mintsources" you have a list of directories with the names of Linux Mint versions; here is, as an example, this list for my computer (running Linux Mint 21.3 Virginia):

[code]

ls

[output]

betsy elsie qiana rosa sonya tessa ulyana una victoria

cindy faye rafaela sarah sylvia tina ulyssa vanessa virginia

debbie rebecca serena tara tricia uma vera

Now, copy the directory corresponding to your distribution (tina, una etc.) and name it as Ubuntu codename (bionic, focal etc.); for me, using Linux Mint 21.3, I have to copy "virginia" to "jammy".

[code]

sudo cp -r /usr/share/mintsources/virginia /usr/share/mintsources/jammy

"/usr/share/mintsources" content is now (change in bold):

[code]

ls

[output]

betsy elsie qiana rosa sonya tessa ulyana una victoria

cindy faye rafaela sarah sylvia tina ulyssa vanessa virginia

debbie **jammy** rebecca serena tara tricia uma vera

Check that mintsources does work with command line:

[code]

mintsources

NB: the change made in "os-release" may also break the "Linux kernels" tool in Update Manager view menu, and mintinstall, the software library. We will see later how to cope with it.

7) Install Ubuntu Pro and enable Pro services

Restart your computer.

The main Pro services are the following ones, depending on your Ubuntu LTS base:

SERVICE	DESCRIPTION
esm-apps	Expanded Security Maintenance for Applications
esm-infra	Expanded Security Maintenance for Infrastructure
fips	NIST-certified core packages
fips-updates	NIST-certified core packages with priority security updates
livepatch	Canonical Livepatch service
usg	Security compliance and audit tools

NB:

* "fips" and "fips-updates" services are not compatible with Canonical Livepatch service (you have to choose between fips + fips-update and Livepatch).

* "livepatch" allows to patch the kernel without session interruption; this service is a snap application; if you want to use it, you need to enable snaps in Linux Mint in 20.x and 21.x versions (for this you can remove "/etc/apt/preferences.d/nosnap.pref" or comment its content).

* "usg" stands for "Ubuntu Security Guide". If you enable it, install then the "usg" package with the usual "sudo apt install usg"; usg command will be used to audit the system to a given security model, and to automatically fix it.

Attach your computer to Ubuntu Pro:

[code]

sudo pro attach C1...Be

where "C1...Be" is the token (30 characters long for mine) you got when subscribing to Ubuntu Pro (if lost, connect again to Ubuntu Pro and open your "Pro dashboard", the token is there).

Some services will be automatically enabled:

* esm-apps,

* esm-infra,

* there will be an attempt to install livepatch, but it will fail (see 8. Kernels management).

Once done, you can get Ubuntu Pro status:

[code]

pro status

[output]

SERVICE ENTITLED STATUS DESCRIPTION

esm-apps yes enabled Expanded Security Maintenance for Applications

esm-infra yes enabled Expanded Security Maintenance for Infrastructure

fips yes disabled NIST-certified core packages

fips-updates yes disabled NIST-certified core packages with priority security updates

livepatch yes disabled Canonical Livepatch service

usg yes disabled Security compliance and audit tools

Enable services with: pro enable <service>

Account: user@domain

Subscription: Ubuntu Pro - free personal subscription

You can now download the list of new/upgradable packages and update your system:

[code]

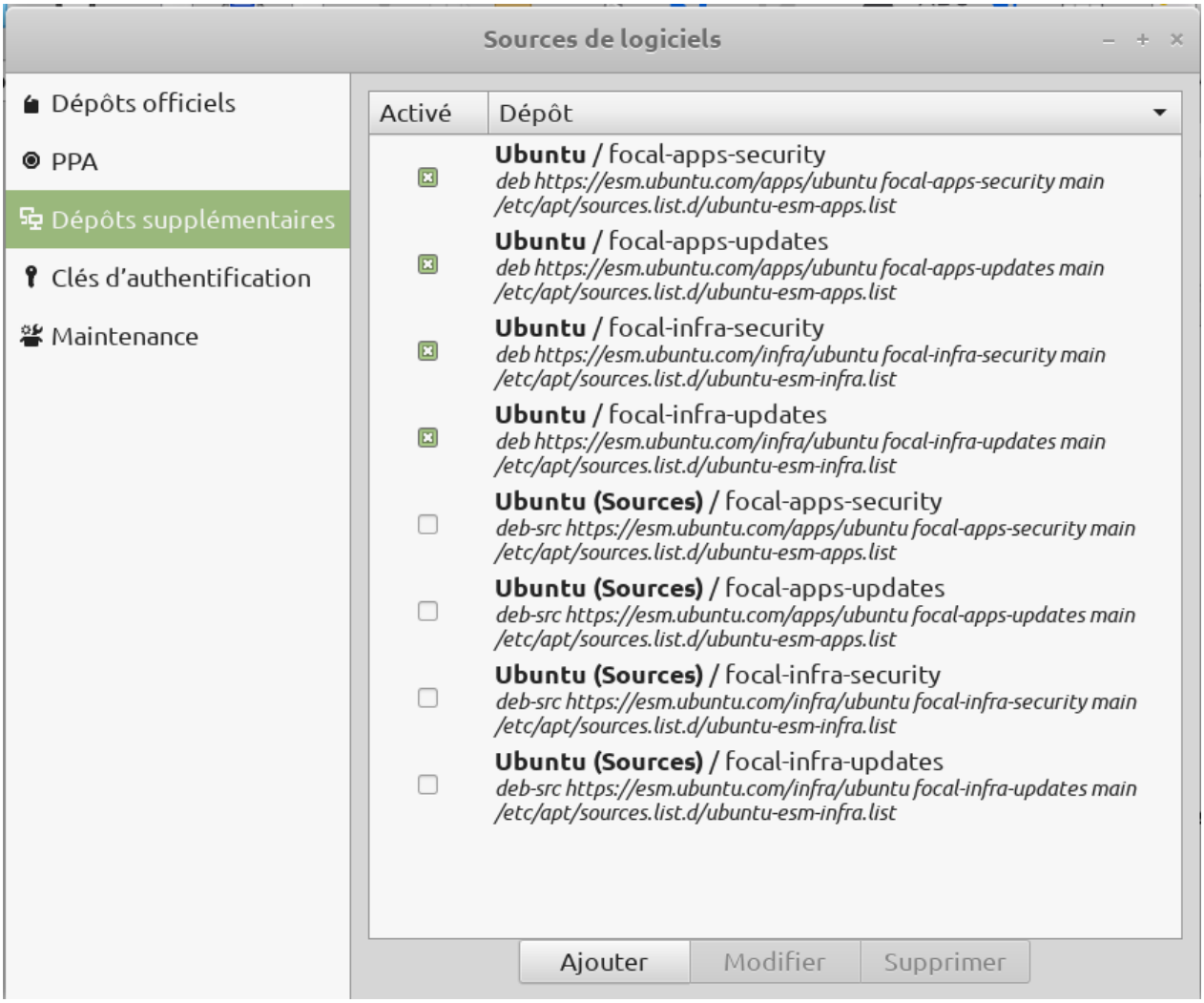
sudo apt-get update

sudo apt-get upgrade

(or use the update manager).

I received 23 packages updates after installation on Linux Mint 21,3; Linux Mint 18.x, 19.x or 20.x users may receive a larger list of packages to update.

"mintsources" now shows extra-repositories:



You can further enable a Pro service by:

[code]

sudo pro enable servicename

You can further disable a Pro service by:

[code]

sudo pro disable servicename

You can detach your computer from Ubuntu Pro by:

[code]

sudo pro detach

The packages that were updated by Ubuntu Pro (with "esm" at the end of package name) will stay and appear in Synaptic in the category "Installed (local or obsolete)".

Linux Mint 18.x and 19.x users can remove Linux Mint repositories from their sources lists, in order to avoid a potential Linux Mint update breaks this installation:

[code]

xed admin:///etc/apt/sources.list.d/official-package-repositories.list

Then comment the line with "packages.linuxmint.com": "# deb http://packages.linuxmint.com" and save the edited file.

NB: for all Linux Mint distributions users, if you didn't disable Linux Mint repositories, and if Update Manager proposes to update "base-files" package, do NOT install it, it would override the changes made to "/usr/lib/os-release".

8) Kernels management

As said, the kernel management tool in Update Manager may be no longer available, depending on your Linux Mint version.

Here are the different possibilities to manage kernels:

- Use Synaptic:

Installed kernels can be view in Synaptic / Sections / kernels; you can remove old kernels from this place.

- Use command line:

* To show your complete current running kernel version string:

[code]

uname -a

[output]

Linux computername 5.4.0-139-generic #156-Ubuntu SMP Fri Jan 20 17:27:18 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux

* To show only the current running kernel version number:

[code]

uname -r

[output]

5.4.0-139-generic

* To list GRUB available bootable kernels installed:

[code]

find /boot/config-*

[output]

/boot/config-5.4.0-139-generic

* Get a list of installed kernels via dpkg:

[code]

dpkg -l |grep -e 'linux\-\(image\|header\).*'

[output]

ii linux-headers-5.4.0-139 5.4.0-139.156 all Header files related to Linux kernel version 5.4.0

ii linux-headers-5.4.0-139-generic 5.4.0-139.156 amd64 Linux kernel headers for version 5.4.0 on 64 bit x86 SMP

ii linux-headers-generic 5.4.0.139.137 amd64 Generic Linux kernel headers

ii linux-image-5.4.0-139-generic 5.4.0-139.156 amd64 Signed kernel image generic

ii linux-image-generic 5.4.0.139.137 amd64 Generic Linux kernel image

* Remove old kernels (it will keep the latest two):

[code]

sudo apt-get autoremove --purge

* more here: https://hackmd.io/@YzaMEc3fRrGC0i-_CCCIXg/BkO1yAigs

- Use kernel Livepatch:

NB: Livepatch service works only on generic kernels (initial GA or HWE ones); it provides kernels update for 10 years. Livepatch will patch the kernel in silent mode without session interruption; when it is not possible, it will ask the user to stop and restart.

Snaps are enabled with Linux Mint 18.x and 19.x; they are disabled with Linux Mint 20.x, 21.x and 22.x, you can enable them by removing "/etc/apt/preferences.d/nosnap.pref" or commenting its content for Linux Mint 20.x and 21.x).

Livepatch service can be installed, but it requires further adjusting of "/etc/issue", "/etc/issue.net", "/etc/lsb-release".

First, make copies of those files.

Then, edit them. Here is how, for Linux Mint 20.x. Please adapt for other Linux Mint versions.

Launch text editor xed in admin mode to edit the files:

[code]

xed admin:///etc/issue

Replace the file content by:

[code]

Ubuntu 20.04 LTS \n \l

Then:

[code]

xed admin:///etc/issue.net

Replace the file content by:

[code]

Ubuntu 20.04 LTS

Then:

[code]

xed admin:///etc/lsb-release

Replace the file content by:

[code]

DISTRIB_ID=Ubuntu

DISTRIB_RELEASE=20.04

DISTRIB_CODENAME=focal

DISTRIB_DESCRIPTION="Ubuntu 20.04 LTS"

Once done, enable livepatch service:

[code]

sudo pro enable livepatch

It may occur an error. To force install: install snapd, and force install livepatch:

[code]

sudo apt install snapd

snap install canonical-livepatch

sudo pro enable livepatch

- Mainline kernels users:

Mainline kernels users can manage their kernels with Ubuntu Mainline Kernel Manager, see:

https://ubuntuhandbook.org/index.php/2020/08/mainline-install-latest-kernel-ubuntu-linux-mint/

or with Ubuntu Kernel Update Utility (Ukuu), see:

https://github.com/teejee2008/ukuu.

Note that the paid version of Ukuu, https://teejeetech.com/tag/ukuu/ can be also used to manage generic kernels.

- Linux Mint 20.x users:

Linux Mint 20.x users can replace mintupdate, the update manager, by a tailored version for Bodhi Linux. This version works on Ubuntu 20.04 LTS and its derivatives, kernel tool is available and still functions with Linux Mint 20.x. See:

https://easylinuxtipsproject.blogspot.com/p/ubuntuplus.html.

Note that this version no longer automatically updates flatpaks; users can add a simple bash script to the list of their applications launched at startup, to have flatpak automatic update at each session start:

[code]

Flatpak_update.sh

#!/bin/bash

flatpak update -y

This can be completed by a crontab:

[code]

Name this file with your username and copy it to '/var/spool/cron/crontabs' directory

If you already have a user crontab, just add this line after having changed the command path

0 */2 * * * ~/Flatpak_update.sh

It will update flatpaks every two hours.

- Linux Mint 21.x users:

It seems that kernel management tool is not affected by the changes in "/usr/lib/os-release". (At least on Linux Mint 21.3 Virginia). Nothing to do...

9) Mintinstall

Mintinstall may not work any longer, once the changes have been made in "/usr/lib/os-release".

Workaround: uninstall mintinstall package, install gnome-software package and, if you use flatpaks and/or snaps, gnome-software-plugin-flatpak and/or gnome-software-plugin-cnap.

10) Applications long term support

Internet connecting applications are the most exposed ones. One way to increase their security and to have a recent version is to use flatpaks or snaps applications, since they are run in sandboxes. Moreover, since flatpaks or snaps run in their own environment, they may not interfere with system.

Flatpaks have three components: the framework, the runtimes, the applications. Flatpak framework can be kept updated using flatpak stable PPA. However, the flatpak support from the PPA stops at the end of the 5 years usual Ubuntu support. Flatpak use should so be abandoned at the end of the usual 5 years maintenance period, or newer version compiled and installed by user.

Snaps: snapd and core will be maintained by Ubuntu for 10 years, once Ubuntu Pro enabled, since livepatch service is based on a snap application (canonical-livepatch). Migrating exposed

applications to snap ones is a way to have both long term support and security increase. Note that a lot of applications are available as snaps.

Snaps are enabled with Linux Mint 18.x and 19.x; they are disabled with Linux Mint 20.x and 21.x, you can enable them by removing "/etc/apt/preferences.d/nosnap.pref" or commenting its content for Linux Mint 20.x and 21.x).

Unexposed applications, not connecting to internet, can be downloaded from their websites and manually updated. Several are working on any version of Ubuntu / Linux Mint (LibreOffice and most of Office suites, XnView MP, FreeFileSync…). Some are delivered as AppImages (Audacity, Avidemux…), allowing them also to work on any Linux Mint / Ubuntu versions.

11) Legal stuff

Ubuntu Pro service is described here (valid since 05 October 2022):

https://ubuntu.com/legal/ubuntu-pro-description.

Some extracts (underlines are mine):

> As an Ubuntu Pro customer, you are entitled to the following coverage, depending on the appropriate support level on a per-machine basis. Each subscription can cover either (i) Infrastructure-only: Ubuntu Pro (Infra-only) with or without support (previously known as Ubuntu Advantage for Infrastructure), or (ii)Infrastructure and Applications: (Ubuntu Pro), with or without support:
>
> 1. Physical server: A subscription attached to a physical host running Ubuntu or a Covered Hypervisor. If all physical hosts in the Environment are attached, then Ubuntu Pro subscription also covers all Ubuntu guests on those hosts
>
> 2. Desktop: A subscription limited to Desktop use-cases. It covers packages in the base Ubuntu desktop image as well as packages necessary for basic network authentication and connectivity using sssd, winbind, network-manager, and network-manager plugin. It can also cover support (weekday or 24/7) for Ubuntu distribution for Windows Subsystem for Linux (WSL) and developer tools such as MicroK8s and Multipass

→ As a user running Linux Mint, a desktop operating system, you are concerned by paragraph 2.

Ubuntu Pro service terms are found on:

https://ubuntu.com/legal/ubuntu-pro-service-terms.

At a first glance, they may be restrictive, since they mention:

> Ubuntu: a version of the operating system known as "Ubuntu", which is supported by Canonical pursuant to Canonical's public announcements and support schedule and is either Canonical's standard version with no modifications or a version modified by Canonical.

But the 1st paragraph precises (underlines are mine):

> Service terms

> These service terms (the "Agreement") take effect as of the effective date of an Order incorporating its terms or otherwise accepts its terms as part of a Registration Process (the "Effective Date") between, Canonical Group Limited, a company registered in England (company number 110334C) whose registered office is at 5 New Street Square, London EC4A 3TW, United Kingdom ("Canonical"), and the customer identified in the Order ("Customer").

From this paragraph, you are bound to service terms when you have made an order, when this order has been accepted by Ubuntu, and when this order identifies you as a customer.

What is the definition of an order? On the same page (underlines are mine):

> Order: If Customer is buying Services directly from Canonical: Customer's order for Services which incorporates this Agreement and identifies Customer's name, contact information, Services, Fees, and Term. An Order may be in the form of a statement of work. If Customer is buying Services through Canonical's reseller: Customer's agreement to purchase the Services and pay applicable Fees.

So, you are bound by Service terms when you make an order, i.e. when you buy a service, and when you are identified to Ubuntu by your name and contact information.

→ Personal users, not buying anything, and not being identified, are not bound by those service terms.

[Note that personal users getting a free Ubuntu Pro licensing are known only to Ubuntu by their name, username, mail address, password. Since name is not verified and can be a pseudo, and since mail address is not a physical location address, this cannot constitute a legal identity and contact information]

So, nothing legal prevents the use of Ubuntu Pro on Linux Mint (note that this might change with further versions of those Ubuntu Pro pages).

12) Conclusion

Once you have enabled Ubuntu Pro, if you are happy with it, make a new backup / a new snapshot of the system.

If not, you can go back by:

* restoring your previous backup,

* restoring your previous snapshot,

* detaching your computer from Ubuntu Pro with "sudo pro detach" and restoring your "/usr/lib/os-release" and maybe your "/etc/issue", "/etc/issue.net", "/etc/lsb-release" files to their original contents from the copies you made.

If an official supported way to enable Ubuntu Pro on Linux Mint were one day available, this tutorial would be obsolete.

Annex 6: Flatpak Tutorial

1) What is flatpak?

- Like Firejail and snap, flatpak is a sandboxing solution, allowing to run applications with controlled permissions. This increases security, particularly for internet connecting applications. Flatpak is based on sandboxing software bubblewrap and on ostree file system.

- Like snaps and AppImages, flatpaks are a way to distribute applications packages that run unchanged on several distributions.

- Flatpak launch is fast: flatpaks don't need to be uncompressed before launch (as snaps need); flatpaks don't need to have a SquashFS file system mounted before launch (as AppImages need).

- Unlike snap, flatpak is not controlled by an organization (Canonical). Any user can make a flatpak for his own use or publish it in a repository (Flathub is the main one).

- Flatpak has three categories of components:

* flatpak framework, packages installed in the operating system,

* flatpak runtimes, including libraries, necessary to run applications,

* flatpak applications.

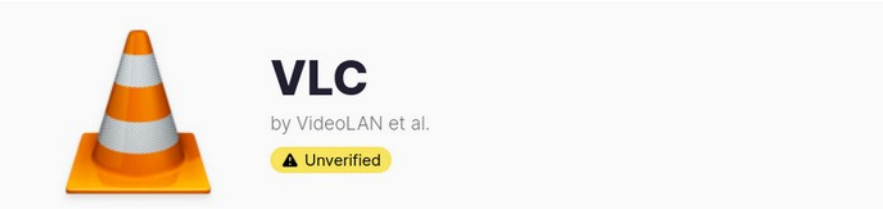
- Flatpak also allows to have fresh applications, with automatic updates. With stable distributions, such as Linux Mint or Ubuntu LTS, applications (except some such as browsers and mail clients) are not updated. With rolling distributions, such as Arch Linux or Manjaro, applications are updated, but the operating system is not stable. Flatpak is a way to have updated applications in a stable operating system; flatpak runtimes and applications are a kind of rolling distribution running in a stable operating system.

- Caveats:

* Flatpaks take a lot of disk space, since runtimes are needed to run applications. The installation of the first flatpak might increase system disk use by several hundreds of MB; following installations will increase disk use less if the installed applications use the same runtime.

* Most flatpaks are published by developers on Github and are not endorsed by the application author (a notable exception is Firefox flatpak, published by Mozilla). Flathub is in the process of verifying flatpak applications; when an application has been verified, its authors have been identified and are the same as the non-flatpak application (if it exists):

At the opposite, flatpak applications with an orange unverified label are published by independent developers unrelated to the original non-flatpak application (or are not yet verified):



I don't recommend the use of an unverified flatpak security application such as password manager or encryption, and I recommend prudence when using another kind of unverified flatpak application (note that VLC flatpak publishers are well known ones, and flatpak VLC can be used without risk. The Software Manager of Linux Mint 22.x does not display, by default, unverified flatpaks.

* Flatpak runtimes change very quickly; daily Timeshift snapshots show that several thousands files change or are created every day. Timeshift snapshots should be so done on a daily basis, completed by operating system backups on a weekly basis.

2) Reference links

- Flatpak website: https://www.flatpak.org/

- Flatpak documentation: https://docs.flatpak.org/en/latest/

- Flatpak on Github: https://github.com/flatpak/

- Flatpak ppa on Github: https://github.com/flatpak/ppa-flatpak

- Flatpak stable versions on Launchpad: https://launchpad.net/~flatpak/+archive/ubuntu/stable

- Flathub repository: https://flathub.org/home

3) Installing flatpak framework and Flathub repository

Flatpak framework is normally installed by default in Linux Mint. However, as with any security application, user should always use the latest available flatpak framework, with the latest bugs and security failures corrections. Ubuntu does not update flatpak framework. Linux Mint has updated it, one shot, in Mint 20.3 (version 1.12.1). But the latest one is 1.12.5 (on February the 20th, 2022).

The solution to always use the latest available stable framework is to install the flatpak stable versions ppa, maintained by flatpak developers team, and available for stable supported versions of Ubuntu, and so for Linux Mint 20.x and 21.x. To add this ppa to your system:

[code]

sudo add-apt-repository ppa:flatpak/stable

Once this ppa is installed, launch the Update Manager and refresh the list of packages:

* If flatpak is installed on your system, Update Manager will propose you to update it; accept, update and close Update Manager.

* If flatpak is not installed, Update Manager will not propose you to update it, close Update Manager; launch Synaptic, look for flatpak, select it for installation and apply. Synaptic will install flatpak framework with all its dependencies (when writing this document, the following packages are installed, at a minimum: flatpak, gir1.2-flatpak-1.0, libflatpak0, libostree-1-1, xdg-dbus-proxy,

xdg-desktop-portal). Optionally, user can install flatpak-builder and its dependencies, to build his own flatpaks.

Once flatpak framework is installed, it is time to install Flathub repository (to allow command line search in it). This is done by the following command:

[code]

flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo

4) Installing a flatpak application

There are three ways to install a flatpak application:

* Use the Software Manager to look for and install a flatpak application (flatpak applications are identified as such in the Software Manager; and search can be restricted to flatpaks).

* Use Flathub: from its home main page, look for an application (search function at the top of the page) or browse applications per categories; once found the application you want to install, copy the installation code from the application page, paste it in the terminal and launch it.

* Use flatpak commands; as an example, you want to install Gimp, you first search for it:

[code]

flatpak search gimp

[output]

Name Description
Application ID Version Branch Remotes

Éditeur d'imag… Créer des images et modifier des photographies
org.gimp.GIMP 2.10.30 stable flathub

GIMP User Manu… GIMP User Manual
org.gimp.GIMP.Manual 2.10 2.10 flathub

Resynthesizer Set of GIMP plug-ins that heal (in-paint), synthesize texture, theme an image, and more
…mp.GIMP.Plugin.Resynthesizer 2.0.3 2-40 flathub

Resynthesizer Set of GIMP plug-ins that heal (in-paint), synthesize texture, theme an image, and more
…mp.GIMP.Plugin.Resynthesizer 2.0.3 2-3.36 flathub

GimpLensfun GimpLensfun is a Gimp plugin to correct lens distortion using the lensfun library and database.
org.gimp.GIMP.Plugin.Lensfun 0.2.4 2-40 flathub

GimpLensfun GimpLensfun is a Gimp plugin to correct lens distortion using the lensfun library and database.
org.gimp.GIMP.Plugin.Lensfun 0.2.4 2-3.36 flathub

Fourier A simple GIMP plug-in to do fourier transform on your image.
org.gimp.GIMP.Plugin.Fourier 0.4.3 2-40 flathub

Fourier A simple GIMP plug-in to do fourier transform on your image.
org.gimp.GIMP.Plugin.Fourier 0.4.3 2-3.36 flathub

BIMP Batch Image Manipulation Program, a GIMP plugin to apply a set of manipulations to an entire group of i… org.gimp.GIMP.Plugin.BIMP 2.6 2-40 flathub

BIMP Batch Image Manipulation Program, a GIMP plugin to apply a set of manipulations to an entire group of i… org.gimp.GIMP.Plugin.BIMP 2.5 2-3.36 flathub

LiquidRescale LiquidRescale plugin to resize pictures non uniformly while preserving their features, i.e. avoiding di… …mp.GIMP.Plugin.LiquidRescale 0.7.2 2-40 flathub

LiquidRescale LiquidRescale plugin to resize pictures non uniformly while preserving their features, i.e. avoiding di… …mp.GIMP.Plugin.LiquidRescale 0.7.2 2-3.36 flathub

G'MIC GREYC's Magic for Image Computing org.gimp.GIMP.Plugin.GMic 3.0.2 2-40 flathub

G'MIC GREYC's Magic for Image Computing org.gimp.GIMP.Plugin.GMic 2.9.6 2-3.36 flathub

FocusBlur Focus Blur plug-in crete a blurring effect similar to Depth of Field. …g.gimp.GIMP.Plugin.FocusBlur 3.2.6 2-40 flathub

FocusBlur Focus Blur plug-in crete a blurring effect similar to Depth of Field. …g.gimp.GIMP.Plugin.FocusBlur 3.2.6 2-3.36 flathub

Glimpse Créer des images et modifier des photographies org.glimpse_editor.Glimpse 0.2.0 stable flathub

Scans to PDF Create small, searchable PDFs from scanned documents com.github.unrud.djpdf 0.1.6 stable flathub

The results are much richer than what you get with Software Manager of Flathub:

* Gimp program is "org.gimp.GIMP",

* Gimp user manual is "org.gimp.GIMP.Manual",

* Several plugins are found:

"org.gimp.GIMP.Plugin.Resynthesizer",

"org.gimp.GIMP.Plugin.Lensfun",

"org.gimp.GIMP.Plugin.Fourier",

"org.gimp.GIMP.Plugin.BIMP",

"org.gimp.GIMP.Plugin.Plugin.LiquidRescale",

"org.gimp.GIMP.Plugin.Gmic",

"org.gimp.GIMP.Plugin.FocusBlur"

that you would not find with Software Manager or Flathub online search.

Now you can install what you need, as an example the following commands will install Gimp, its manual, and G'MIC plugin:

[code]

flatpak install org.gimp.GIMP

flatpak install org.gimp.GIMP.Manual

flatpak install org.gimp.GIMP.Plugin.GMic

NB:

- applications will be installed automatically in your system language,

- you will have to choose the place for applications, system or user home, during the installation; default is system.

5) Useful tricks and commands

Automatic updates:

Linux Mint 20.x users:

You can set automatic silent flatpaks update: launch the Update Manager / Edition / Preferences / Automation and move to the right the slider "Update flatpaks automatically", then close Update Manager windows.

Linux Mint 21.x and 22.x users:

Flatpaks updates are proposed in Update Manager in the same way as deb packages updates.

All users:

You can add a simple bash script "Flatpak_update.sh" to the list of the applications launched at startup, to have flatpaks automatic update at each session start:

[code]

Flatpak_update.sh

#!/bin/bash

flatpak update -y

This can be completed by a crontab:

[code]

Name this file with your username and copy it to '/var/spool/cron/crontabs' directory

If you already have a user crontab, just add this line after having changed the command path

0 */2 * * * ~/Flatpak_update.sh

It will update flatpaks every two hours.

Some useful commands:

Flatpak framework version:

[code]

flatpak --version

Flatpak manual update:

[code]

flatpak update

example: after a change or an update of your GPU driver, this command will update the corresponding runtime.

Flatpak runtimes and applications installation repair:

[code]

flatpak repair

Flatpak unused runtimes purge (will not uninstall anything if nothing unused):

[code]

flatpak uninstall --unused

Flatpak application uninstall:

[code]

flatpak uninstall appname

where "appname" is the name of the app you want to uninstall, same name as used during installation; example, the following command will uninstall Gimp:

[code]

flatpak uninstall org.gimp.GIMP

To list all the runtimes and applications found on your system:

[code]

flatpak list

To list only the applications:

[code]

flatpak list --app

Output example:

[output]

Name	Application ID	Version	Branch	Installation
calibre	com.calibre_ebook.calibre	5.37.0	stable	system

Pinta	com.github.PintaProject.Pinta	2.0.2	stable	system
Flatseal	com.github.tchx84.Flatseal	1.7.5	stable	system
OBS Studio	com.obsproject.Studio	27.2.0	stable	system
Transmission	com.transmissionbt.Transmission	3.00	stable	system
XnView MP	com.xnview.XnViewMP	0.99.7	stable	system
HandBrake	fr.handbrake.ghb	1.5.1	stable	system
Audacity	org.audacityteam.Audacity	3.1.3	stable	system
Avidemux	org.avidemux.Avidemux	2.8.0	stable	system
Chromium Web Browser	org.chromium.Chromium	98.0.4758.102	stable	system
FileZilla	org.filezillaproject.Filezilla	3.58.0	stable	system
Geeqie	org.geeqie.Geeqie	v1.7.2	stable	system
Inkscape	org.inkscape.Inkscape	1.1.2	stable	system
Okular	org.kde.okular	21.12.2	stable	system
Thunderbird	org.mozilla.Thunderbird	91.6.1	stable	system
Firefox	org.mozilla.firefox	97.0.1	stable	system
VLC	org.videolan.VLC	3.0.16	stable	system

To launch a flatpak application:

[code]

flatpak run appname

where appname is the name of the application (as listed with "flatpak list --app").

NB: the preferred way to launch a flatpak is by the shortcut added to your menu during the flatpak application installation; the command line may be more complex and include some extra arguments. Example, for chromium browser:

"/usr/bin/flatpak run --branch=stable --arch=x86_64 --command=/app/bin/chromium --file-forwarding org.chromium.Chromium @@u %U @@"

The whole list of commands and their syntax are described in Flatpak's documentation.

Files on your disk:

- With the default (system) place, runtimes and applications are installed in "/var/lib/flatpak" and do not disturb in any way your operating system installation (no system file is changed or created in other system directory during runtimes or applications installation).

- Configuration and cache files, including browsers or mail clients profiles, are in "~/.var/app"

Flatpaks permissions editor:

Flatpak permissions are set by flatpak publishers. They are generally well adapted to the application. If permissions of an application need to be edited, Flatseal is a flatpak application allowing to do it. It is done easily, with a graphical UI. Flatseal includes an integrated documentation explaining the meaning of each permission setting. To install Flatseal:

[code]

flatpak install com.github.tchx84.Flatseal

Once installed, Flatseal will appear in your Accessories menu.

Check with command line if a flatpak application is verified:

When you look for a flatpak application with command line you normally get no information about its verified or unverified status. An example with Recordbox, a verified flatpak:

[code]

flatpak search recordbox

Returned answer:

Name Description Application ID Version Branch Remotes

Recordbox Browse and play your local music ca.edestcroix.Recordbox 0.8.2 stable flathub

Install now flathub-verified remote:

[code]

flatpak remote-add --if-not-exists --subset=verified flathub-verified https://flathub.org/repo/flathub.flatpakrepo

Search again Recordbox:

[code]
flatpak search recordbox

Returned answer:

Name Description Application ID Version Branch Remotes

Recordbox Browse and play your local music ca.edestcroix.Recordbox 0.8.2 stable flathub,flathub-verified

You know now that Recordbox is verified, since it is found in flathub-verified remote.

Search Avidemux (unverified flatpak):

> [code]
> flatpak search avidemux

Returned answer:

> Name Description Application ID Version Branch Remotes
>
> Avidemux Multi-purpose video editing and processing software org.avidemux.Avidemux 2.8.1 stable flathub

You know now that Avidemux is unverified, since it is not found in flathub-verified remote.

6) About Flatpak security

Flatpak uses Bubblewrap, https://github.com/containers/bubblewrap.

From Bubblewrap github page:

> The goal of bubblewrap is to run an application in a sandbox, where it has restricted access to parts of the operating system or user data such as the home directory.
>
> Bubblewrap works by creating a new, completely empty, mount namespace where the root is on a tmpfs that is invisible from the host, and will be automatically cleaned up when the last process exits.
>
> The user can specify exactly what parts of the filesystem should be visible in the sandbox. Any such directories you specify mounted nodev by default, and can be made readonly.
>
> The maintainers of this tool believe that it does not, even when used in combination with typical software installed on that distribution, allow privilege escalation.
>
> In particular, bubblewrap uses PR_SET_NO_NEW_PRIVS to turn off setuid binaries, which is the traditional way to get out of things like chroots.

Permissions setting (with Flatseal) allows restricting file accesses: as an example, restricting file access of your browser or e-mail client to "xdg-download" will allow it to only read /write your Downloads directory, with no access to any other directory (except with user control, through file chooser portal).

More explanations on how flatpak security works.

I will take an example: you use a browser, and you have set this browser to forbid the use of webcam.

Using browser only:

In the normal use, the browser saves its disk cache and configuration on a hidden directory of home user (as an example, "~/.mozilla"); when a file is downloaded with user action, a window appears offering the user the choice where to download the file; and the browser respects the webcam setting.

The browser, being launched by the user without superuser rights, has the same file access rights as the user; its write rights are limited to user home, "/tmp" (*) and any connected device (USB disk, key...) where user can write.

Malicious script action is limited by the browser (with its own sandboxing) and by the operating system: an application launched without superuser rights cannot write on the system "/".

But browsers are complex software, and they have bugs. Those bugs are the results of human errors at the specification, code writing, code compiling and testing stages. They are found in the software itself, in the libraries/dependencies it uses, or in the tools used by developers. Some of those bugs induce security weaknesses that could be used by a malicious script: at each revision of a browser, there are several (in the range 1 to 50) security fixes, and a few times a year, a highly critical weakness is exploited before a fix is available (they are called "zero-day exploits").

Exploiting such a critical weakness, a malicious script could find a way to circumvent browser settings (and to use webcam without your consent), or to put the system in an unstable state, gain superuser privileges, and find a way to write files on the system "/" and corrupt it.

[(*): on Linux there is only one directory for temporary files, for system and user; as a consequence, any user has a write access on this directory; this is different on Windows where system temp and user temp are different directories, with different access rights.]

Using flatpak browser:

Flatpak applications are not intrinsically more secure than non-flatpak ones: application software is the same, and libraries/dependencies are replaced by the ones in runtimes. But they run in a sandbox.

In the normal use, the browser saves its disk cache and configuration on a hidden directory of home user (as an example, "~/.var/app/org.mozilla.firefox"); when a file is downloaded on user request, a window appears offering the user the choice where to download the file: this is done through the "file chooser portal", an interface between the flatpak sandbox and the system, having the same write privileges as normal user (file chooser portal CANNOT have superuser rights); if the use of devices has been prevented in the browser flatpak permissions, webcam cannot be used.

Suppose now a malicious script exploits a browser security weakness: webcam block cannot be circumvented, since it is controlled by flatpak sandbox permissions, and not by the browser only.

No file can be written on the system "/":

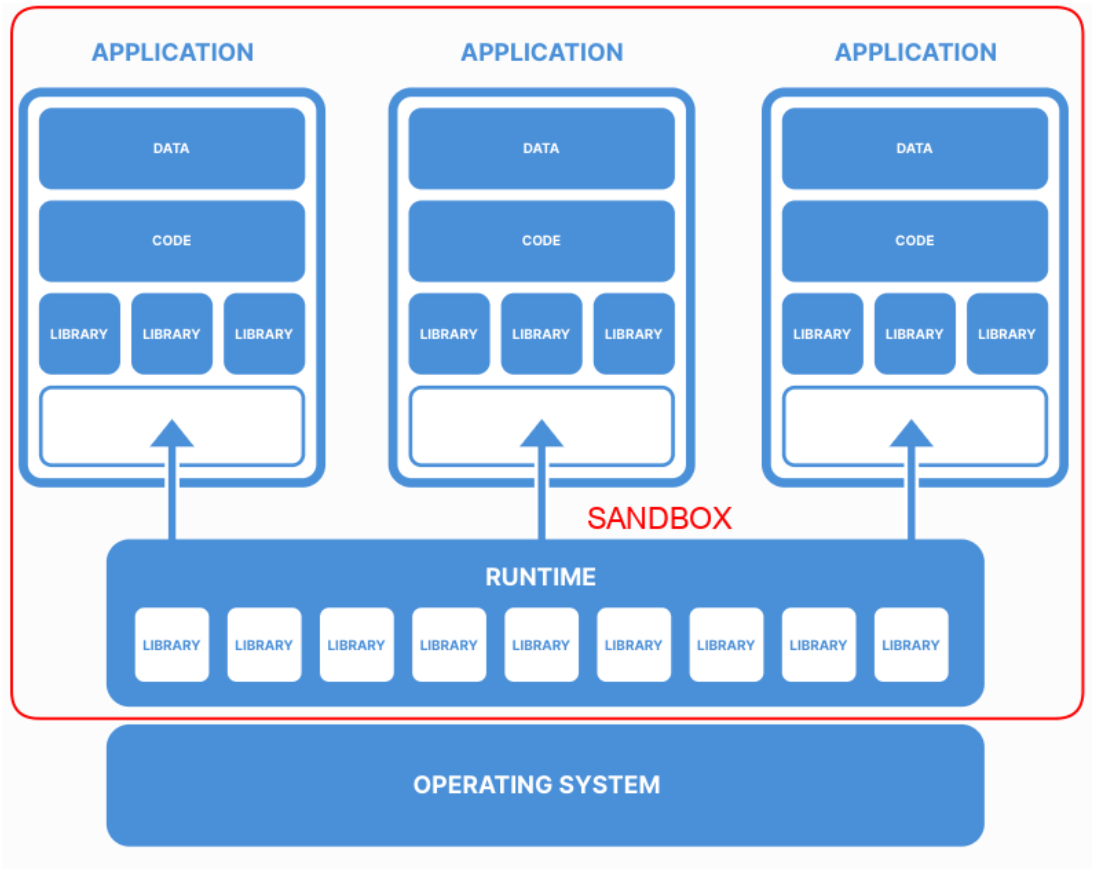
- It is not possible through file chooser portal, with user interaction, since this portal cannot write on the system "/".

- It is not possible in an unattended way, without user interaction, since this way is controlled by flatpak filesystem permissions, and since the most permissive possible one is the same as file chooser portal: no file can be written on the system "/".

→ System corruption by a malicious script is not possible

Moreover, since the operating system is isolated from the browser by flatpak sandbox, it cannot be put in an unstable state.

→ Privileges escalation is blocked by both flatpak sandbox and the operating system and is considered as not possible.

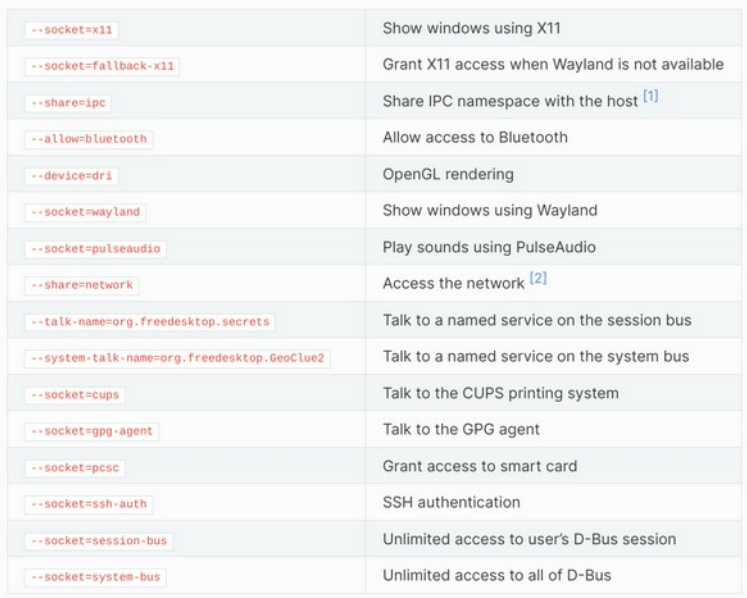


Another example of use of filesystem permission:

I have an AppImage application with its help in the AppImage; when I want to read the help, html help files are copied to "/tmp", then browser is launched; since I have associated html files to my flatpak browser, it is automatically launched; however, without the corresponding filesystem permission, the browser cannot open help files: I need to add a "/tmp:ro" filesystem permission (allowing to read only files in "/tmp" directory) to have the browser opening the help files.

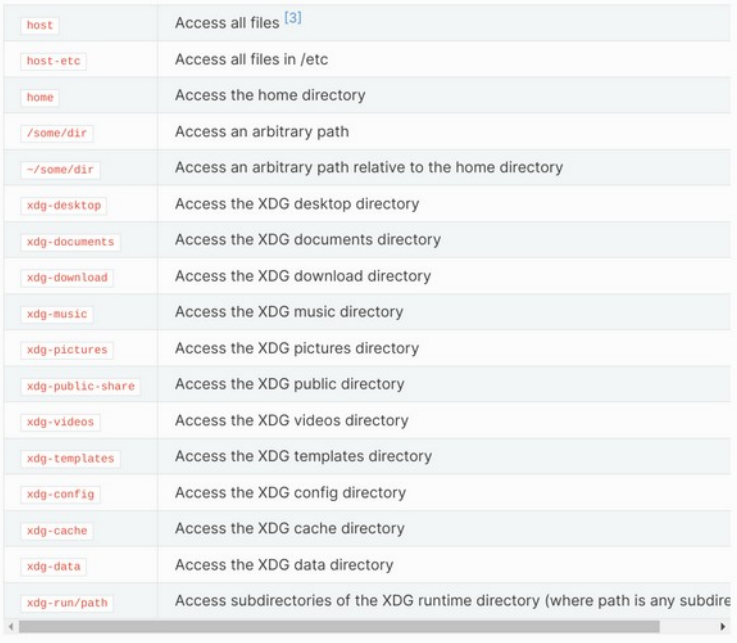
More precision on sandbox, filesystem and portals permissions:

- Sandbox permissions can be adjusted:



`--socket=x11`	Show windows using X11
`--socket=fallback-x11`	Grant X11 access when Wayland is not available
`--share=ipc`	Share IPC namespace with the host [1]
`--allow=bluetooth`	Allow access to Bluetooth
`--device=dri`	OpenGL rendering
`--socket=wayland`	Show windows using Wayland
`--socket=pulseaudio`	Play sounds using PulseAudio
`--share=network`	Access the network [2]
`--talk-name=org.freedesktop.secrets`	Talk to a named service on the session bus
`--system-talk-name=org.freedesktop.GeoClue2`	Talk to a named service on the system bus
`--socket=cups`	Talk to the CUPS printing system
`--socket=gpg-agent`	Talk to the GPG agent
`--socket=pcsc`	Grant access to smart card
`--socket=ssh-auth`	SSH authentication
`--socket=session-bus`	Unlimited access to user's D-Bus session
`--socket=system-bus`	Unlimited access to all of D-Bus

- Filesystem permissions can be adjusted (they apply to unattended files read/write):



`host`	Access all files [3]
`host-etc`	Access all files in /etc
`home`	Access the home directory
`/some/dir`	Access an arbitrary path
`~/some/dir`	Access an arbitrary path relative to the home directory
`xdg-desktop`	Access the XDG desktop directory
`xdg-documents`	Access the XDG documents directory
`xdg-download`	Access the XDG download directory
`xdg-music`	Access the XDG music directory
`xdg-pictures`	Access the XDG pictures directory
`xdg-public-share`	Access the XDG public directory
`xdg-videos`	Access the XDG videos directory
`xdg-templates`	Access the XDG templates directory
`xdg-config`	Access the XDG config directory
`xdg-cache`	Access the XDG cache directory
`xdg-data`	Access the XDG data directory
`xdg-run/path`	Access subdirectories of the XDG runtime directory (where path is any subdire

Sandbox permissions and filesystem permissions are normally set by flatpak applications publishers in the application manifest. They can be easily adjusted by the user with Flatseal.

[User should take care to read Flatseal documentation or Flatpak reference and understand what each setting does before to change those settings].

- Portals permissions cannot be adjusted: file chooser portal has the same read/write permissions as the user, and printer portal can use the same printers as the user.

[This point is still discussed on flatpak GitHub, some users would prefer to have the capability to adjust (reduce) portals permissions; it is not a security concern, since portals permissions cannot exceed user ones and since portals use is under user control, it is simply a facility request.]

Of course, flatpak framework itself, providing the sandbox, can have bugs and security weaknesses. That's why it should be kept update, as any security software. The simplest way to do it is by using flatpak stable PPA.

Annex 7: Multiboot

Linux users forums are full of users reporting problems with multiboot:

- Windows is installed, then Linux is installed and Windows does not boot any longer,

- Linux boot is not detected, no GRUB menu, but boot from BIOS works,

- [...]

This tutorial is here to explain how those problems arrive, and what is the good way to have a working multiboot.

Warning: don't attempt to use multiboot if your system does not allow having at least two disks, and/or if disks cannot be removed and installed to another slot, and/or if disks boot order cannot be changed.

1) First, a reminder on GRUB menu

GRUB menu may be visible or not.

If GRUB menu is not visible, you can see it once, at boot time:

if your system uses BIOS, press and hold the shift key while the system is booting,

if your system uses UEFI, press and hold the ESC key while the system is booting.

[Note that you may have conflicts with your computer, Esc might be used to show the BIOS / UEFI menu].

If you want to see it permanently (my preferred solution):

* once your system is running, edit your GRUB file, type in a terminal:

[code]

sudo nano /etc/default/grub

* in front of the line "GRUB_TIMEOUT_STYLE=hidden", add a "#" character to comment it, add a non zero value at the end of the line "GRUB_TIMEOUT=" (as an example, if the value is 5, GRUB menu will display during 5 seconds), save your modified GRUB file by "CTRL+O" and leave nano by "CTRL+X".

Here is an example of edited GRUB file, edited lines are in bold:

[code]

If you change this file, run 'update-grub' afterwards to update

/boot/grub/grub.cfg.

For full documentation of the options in this file, see:

info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0

#GRUB_TIMEOUT_STYLE=hidden

GRUB_TIMEOUT=5

GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`

GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"

GRUB_CMDLINE_LINUX=""

Uncomment to enable BadRAM filtering, modify to suit your needs

This works with Linux (no patch required) and with any kernel that obtains

the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)

#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

Uncomment to disable graphical terminal (grub-pc only)

#GRUB_TERMINAL=console

* Update GRUB:

[code]

sudo update-grub

* Restart your system, GRUB menu is there.

2) Multiboot Windows/Linux Mint on a system using BIOS

In order to have a GRUB menu allowing to choose between Windows and Linux Mint, Linux Mint should be "in the first position":

- on a single disk: Linux Mint on the 1st partition, Windows on the second one

- with two disks: Linux Mint on disk "0", Windows on disk "1", or Linux Mint disk set in the BIOS as the 1st disk to boot.

With a BIOS system, GRUB menu will be written on Linux Mint disk / partition, and nothing will be written on Windows one.

[If Linux Mint is in the 1st position, GRUB menu will show at boot (maybe after having commented the line with "hidden", see § 1). Of course, if Windows is in the first position, the GRUB menu will not be shown! In that case, however, entering your system BIOS at boot (generally by pressing and holding "Esc") will allow you to choose as boot the disk where Linux Mint is installed and to launch it].

If necessary, update GRUB:

[code]

sudo update-grub

3) Failing multiboot Windows/Linux Mint on a system using UEFI

Ubuntu (and, as a consequence, Linux Mint) version of GRUB has a BIG BUG: it writes itself on the first UEFI partition it sees, whatever the partition, Windows or Linux.

As a consequence, here is the most frequent multiboot failure: Windows is installed on your system, you add Linux Mint (on a partition or on a separate disk), GRUB menu is written over the 1st UEFI partition, Windows one; it seems to work, at boot you will have a GRUB menu offering the choice between Windows and Linux Mint; but, at next Windows version update (every 6 months or year), Windows will write a new UEFI, and GRUB menu will disappear, and Linux Mint boot will become impossible.

And this is in the best case, writing GRUB on Windows UEFI might also prevent Windows to work.

With UEFI there is only one way to install multiboot: just read the following paragraph.

4) Multiboot Windows / Linux Mint that always and definitely works

This is a multiboot that works with both BIOS or UEFI.

Windows is installed on a system, and your computer allows having at least two disks.

* shut down your system, remove Windows disk from your system,

* add a new disk on your system, in slot "0", where your Windows disk was; install Linux Mint, enable GRUB menu (see § 1), shut down your system,

* once done, install your Windows disk on your system, in the second position, slot "1",

* boot your system, you should see a GRUB menu offering the choice between Windows and Linux Mint; if not, boot on Linux Mint, and update GRUB:

    [code]

    sudo update-grub

* at next boot, your GRUB menu will offer the choice between Windows and Linux Mint.

If your system allows two disks with different and incompatible hardware that cannot be exchanged, remove Windows 10 disk while you install Linux Mint on the second, then insert Windows 10 disk again and set your Linux Mint disk as boot disk in the UEFI / BIOS.

If necessary update GRUB:

    [code]

    sudo update-grub

This will work for both BIOS and UEFI:

- BIOS: since Linux Mint disk is in first position, GRUB menu will be automatically launched.

- UEFI: since Linux Mint disk is in first position, GRUB menu will be automatically launched; since GRUB has not modified Windows UEFI, Windows will be able to update its UEFI without any problem.

5) My advice: don't use multiboot

Even if multiboot can work, it is an old and obsolete technology; it may break your system; and it does not allow using Windows and Linux Mint apps at the same time…

Several technologies should be preferred.

With Linux host:

- use a virtual machine, you will be able to run any Windows app; use libvrt/qemu/kvm (Gnome Boxes is an easier way, and its flatpak or snap version give the best isolation between Linux and Windows), Virtual Box or VMware Workstation Pro, in which you will install Windows (of course you need a Windows license, but you can buy a legal OEM or second-hand one for some $15 to $20).

- if you need some Windows apps only, use Bottles: the best, simplest and most secure way to use Wine is Bottles, with its flatpak version giving the best isolation between Windows apps and Linux.

With Windows host:

- use VirtualBox or VMware Workstation Pro, in which you will install Linux Mint.

[Windows 10 Pro or Windows 11 Pro users should not use WSL or WSL2: Windows and Linux Processes are not well isolated, and there are attacks targeting Windows by using WSL/WSL2].

Annex 8: Mullvad Browser Flatpak on Tor Network, a Secure Alternative to Tor Browser

About Mullvad Browser:

> The Mullvad Browser is a privacy-focused web browser developed in a collaboration between Mullvad VPN and the Tor Project. It's designed to minimize tracking and fingerprinting. You could say it's a Tor Browser to use without the Tor Network. Instead, you can use it with a trustworthy VPN. The idea is to provide one more alternative --beside the Tor Network --to browse the internet with more privacy. To get as many people as possible to fight the big data gathering of today. To free the internet from mass surveillance.

Since Mullvad Browser is available as flatpak (while Tor Browser is not), its security is much higher than Tor Browser one.

Mullvad Browser flatpak, and Tor service (AppArmor protected) allow having a secure alternative to Tor Browser, with the same privacy and anonymity features.

1) Install Tor service

This is explained on Tor project support page, https://support.torproject.org/apt/#apt_tor-deb-repo, and detailed here.

- Adding Tor repository:

For Linux Mint 20.x:

> [code]
>
> sudo apt install apt-transport-https curl
>
> sudo -i
>
> echo "deb https://deb.torproject.org/torproject.org/ focal main" > /etc/apt/sources.list.d/tor.list
>
> curl https://deb.torproject.org/torproject.org/A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89.asc | gpg --import
>
> gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-key add -
>
> apt update
>
> exit

For Linux Mint 21.x:

> [code]
>
> sudo apt install apt-transport-https curl
>
> sudo -i

echo "deb https://deb.torproject.org/torproject.org/ jammy main" > /etc/apt/sources.list.d/tor.list

curl https://deb.torproject.org/torproject.org/A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89.asc | gpg --import

gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-key add -

apt update

exit

For Linux Mint 22.x:

[code]

sudo apt install apt-transport-https curl

sudo -i

echo "deb https://deb.torproject.org/torproject.org/ noble main" > /etc/apt/sources.list.d/tor.list

curl https://deb.torproject.org/torproject.org/A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89.asc | gpg --import

gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-key add -

apt update

exit

- Installing "tor", "tor-geoipdb" (to be able to use country specific exit nodes), "torsocks" (a library to easily torify applications) and "deb.torproject.org-keyring" (a package that makes sure you have the latest repository signing key):

[code]

sudo apt install tor tor-geoipdb torsocks deb.torproject.org-keyring

- Launching Tor service:

[code]

sudo systemctl restart tor

2) Mullvad Browser flatpak installation and configuration

- Mullvad Browser flatpak installation:

[code]

flatpak install net.mullvad.MullvadBrowser

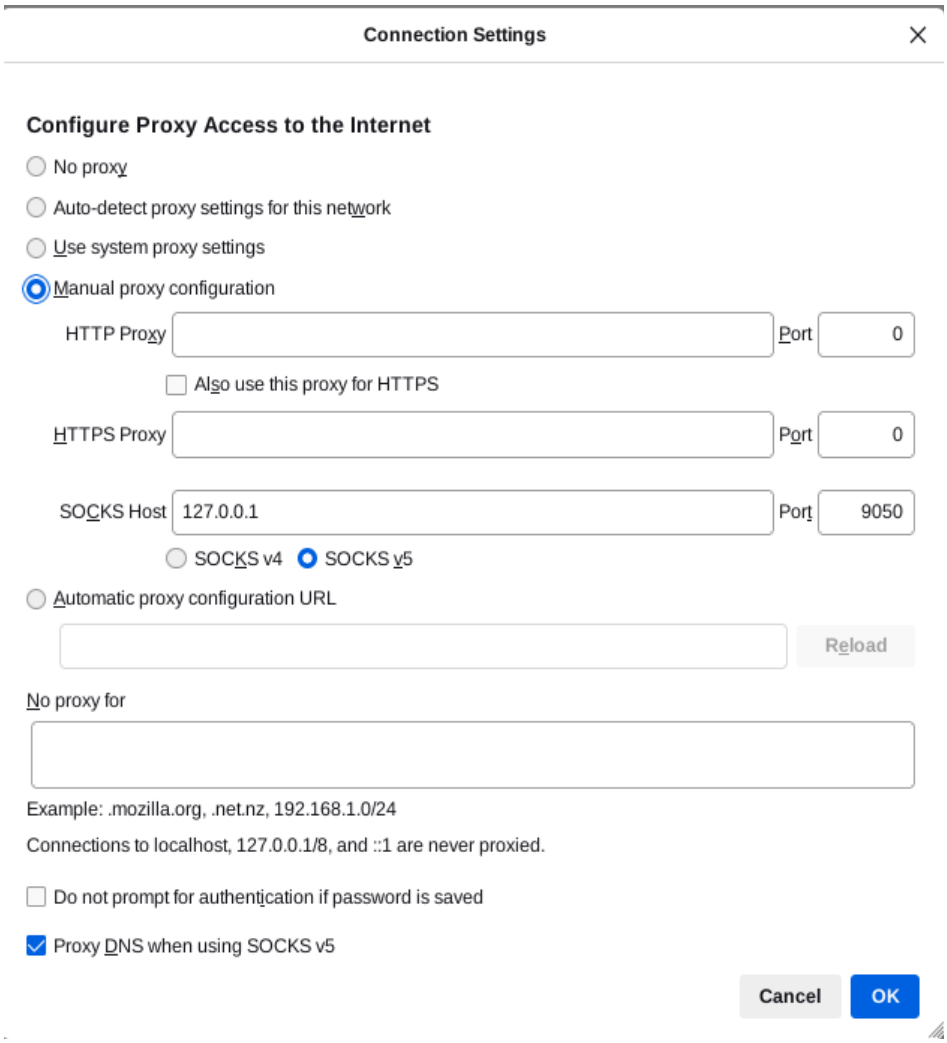
- Mullvad Browser configuration to use Tor:

Once installed, launch Mullvad from your menu. Click on the three lines at the right high corner of Mullvad Browser Windows and edit the settings.

On the General settings tab, go to Network Settings at the end of the page, and click on "Settings…" button.

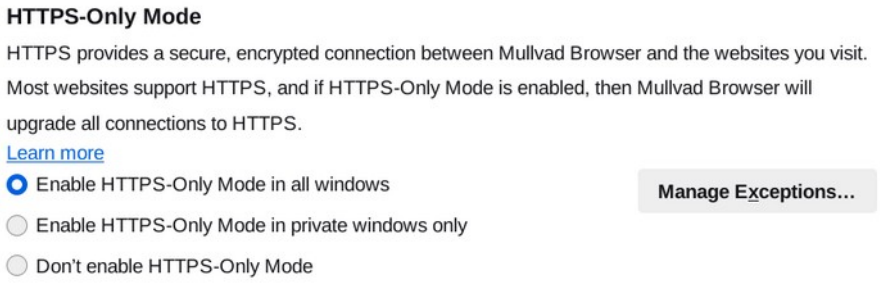
Go to "Configure Proxy Access to the Internet", select "Manual proxy configuration"; in "SOCKS" write "127.0.0.1" after "Host", and "9050" after "Port".

In the bottom of the windows, select "Proxy DNS when using SOCKS v5" and unselect "Enable DNS over HTTPS".

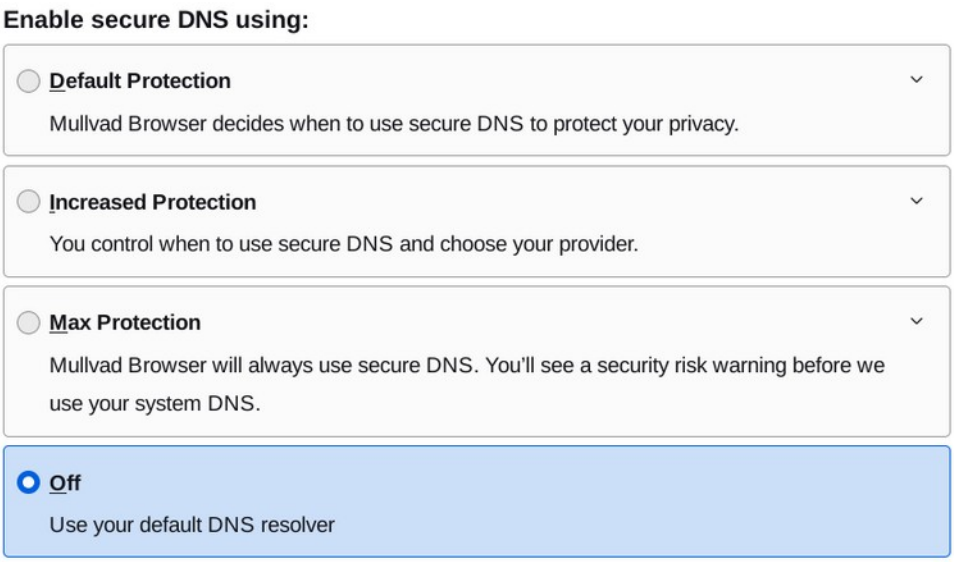


Click OK, your network configuration is saved.

In "Privacy & Security" enable HTTP-Only mode:



Then, disable secure DNS (Mullvad uses Tor Network resolving DNS capabilities):

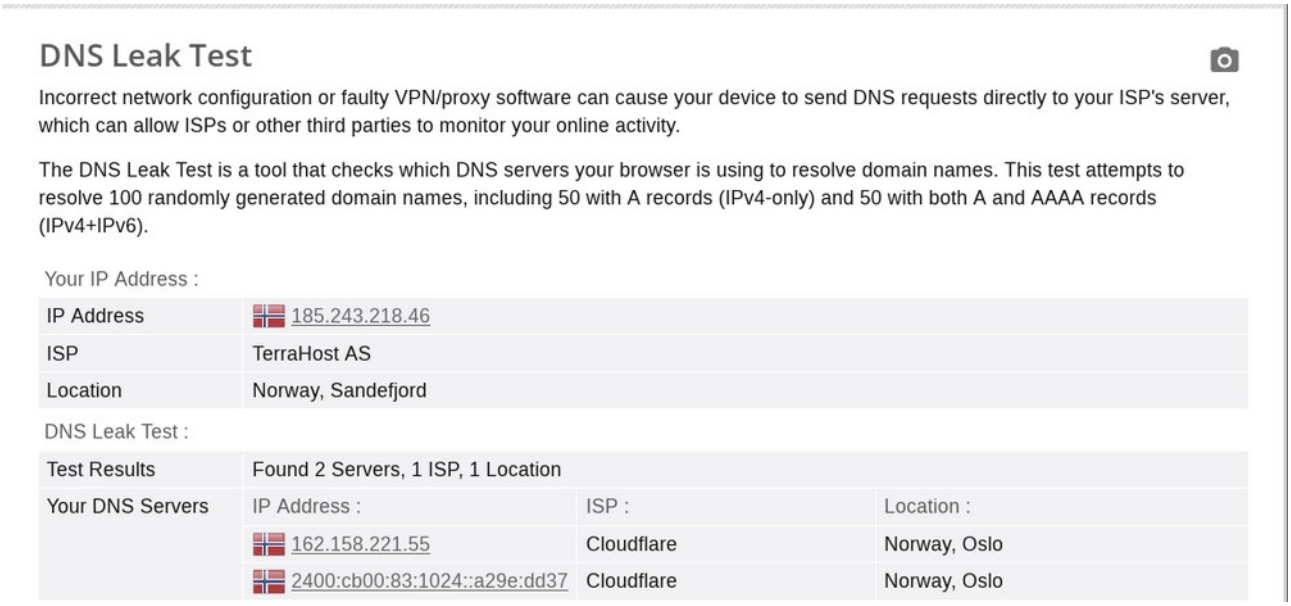


Check that Mullvad now works with Tor: https://check.torproject.org/.

You should get this:

You can check that there is no DNS leak test (the used DNS is not yours, Tor Network has its own DNS resolving service, your DNS should not appear):

(test with https://browserleaks.com/dns):



DNS Leak Test

Incorrect network configuration or faulty VPN/proxy software can cause your device to send DNS requests directly to your ISP's server, which can allow ISPs or other third parties to monitor your online activity.

The DNS Leak Test is a tool that checks which DNS servers your browser is using to resolve domain names. This test attempts to resolve 100 randomly generated domain names, including 50 with A records (IPv4-only) and 50 with both A and AAAA records (IPv4+IPv6).

Your IP Address :

IP Address	185.243.218.46		
ISP	TerraHost AS		
Location	Norway, Sandefjord		

DNS Leak Test :

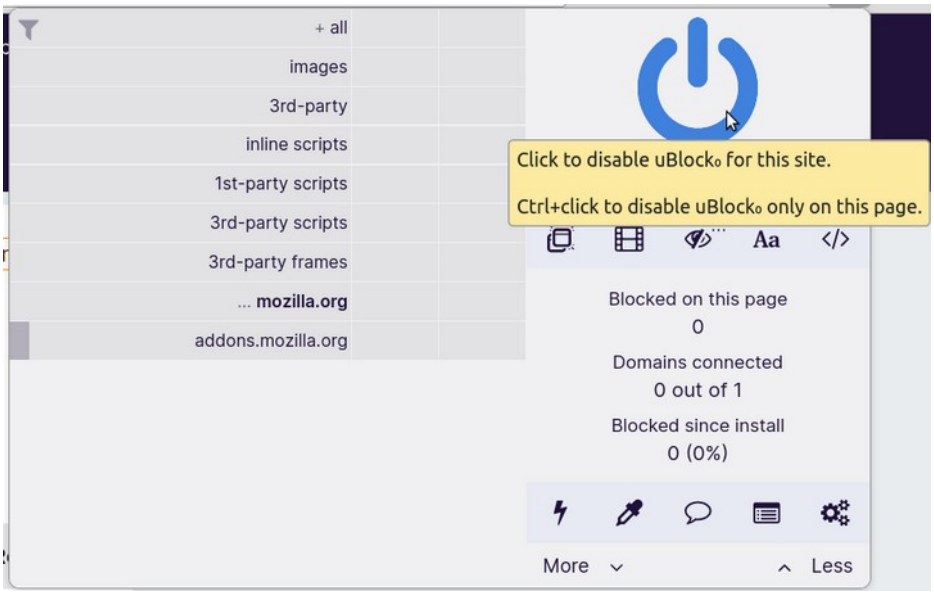
Test Results	Found 2 Servers, 1 ISP, 1 Location		
Your DNS Servers	IP Address :	ISP :	Location :
	162.158.221.55	Cloudflare	Norway, Oslo
	2400:cb00:83:1024::a29e:dd37	Cloudflare	Norway, Oslo

- uBlock Origin configuration:

Mullvad Browser has uBlock Origin extension pre-installed. We are going to configure it to have protection against spam, adware, malware, cryptocurrencies miners and tracking. See Install "uBlock Origin in §4.7.

NB: Since Tor network has its own DNS system, the filters that are in "/etc/hosts" files are not used, and should be added to uBlock Origin.

When browsing internet, don't forget to add in uBlock Origin the websites where uBlock Origin should not apply the filters (trusted websites, or websites not functioning with the filters).



- Add the following Firefox extensions:

* CanvasBlocker,

* CSS Exfil Protection,

* Font Fingerprint Defender,

* Privacy Badger or DuckDuckGo Search and Tracker Protection (it installs DuckDuckGo search as home page, DuckDuckGo search engine by default and DuckDuckGo Privacy Essentials extension).

It is time now to perform some changes in "about:config":

* set "javascript.options.baselinejit" to "false",

* set "webgl.enable-debug-renderer-info" to "false",

* set "pdfjs.enableScripting" to "false",

* set "media.peerconnection.enabled" to false,

* check that cache disk is disabled (you do NOT want to let traces on your disk) and that memory cache is enabled.

- You can now choose to customize Mullvad security and privacy parameters, to display menu toolbar, to customize toolbars and to personalize home page.

* "NoScript Security Suite" is installed in Mullvad Browser; it allows enabling javascript on selected trusted websites (when javascript is disabled, very little information from your browser and computer leaks to the websites). Make it visible by customizing toolbar. You can keep the default Mullvad NoScript profiles (same as Tor Browser ones) or use your own ones.

* Remove "Mullvad Browser extension", and remove from toolbar menu "Security level" and "Change identity" (they work only when Mullvad Browser is used on Mullvad VPN).

- Optional: configure Tor exit country

You can select the country where you want to use a Tor exit node.

Edit your torrc file: in a terminal open it

[code]

sudo nano /etc/tor/torrc

then add the following two lines:

[code]

ExitNodes {COUNTRY_CODE}

StrictNodes 1

where COUNTRY_CODE is the country code of the country where you want to exit Tor network (us for USA, ge for Germany, fr for France, ca for Canada, uk for United Kingdom etc.). Example, if you have written "ExitNodes {us}", you will use one of the tor network exit nodes located in USA and appear to have an USA IP address.

Shutdown Linux Mint and reboot, Mullvad Browser will now use the Tor exit country you chose.

- When browsing you can change the Tor exit node, and so change your IP address; in a terminal:

[code]

sudo systemctl reload tor

- You can check the IP address given by your ISP:

[code]

curl ipv4.icanhazip.com

and compare it to the one you have in Tor network:

[code]

torsocks curl ipv4.icanhazip.com

(with torsocks you can torify any application, when launched with torsocks it will use tor network).

3) Tor security

When installed from Tor Project repository, "tor" package comes with an AppArmor profile, automatically installed and enforced.

When tor service runs, it runs apparmored, as shown by the "sudo aa-status" command output:

```
6 processes have profiles defined.
6 processes are in enforce mode.
   /usr/sbin/cups-browsed (1707)
   /usr/sbin/cupsd (971)
   /usr/sbin/ntpd (996)
   /usr/sbin/avahi-daemon (849) avahi-daemon
   /usr/sbin/avahi-daemon (917) avahi-daemon
   /usr/bin/tor (1742) system_tor
0 processes are in complain mode.
0 processes are unconfined but have a profile defined
```

Moreover, tor service is partially sandboxed by systemd; the output of "systemd-analyze security" shows that tor service security is "medium":

```
systemd-udevd.service                        8.4 EXPOSED 😕
thermald.service                             9.6 UNSAFE  😨
tor@default.service                          6.5 MEDIUM  😐
ubuntu-advantage.service                     9.6 UNSAFE  😨
udisks2.service                              9.6 UNSAFE  😨
upower.service                               2.3 OK      🙂
user@1000.service                            9.4 UNSAFE  😨
```

You can get more details with the command "systemd-analyze security tor@default.service":

```
  NAME                                                  DESCRIPTION
✗ PrivateNetwork=                                       Service has access to the host's network
✗ User=/DynamicUser=                                    Service runs as root user
✗ CapabilityBoundingSet=~CAP_SET(UID|GID|PCAP)          Service may change UID/GID identities/cap
✓ CapabilityBoundingSet=~CAP_SYS_ADMIN                  Service has no administrator privileges
✓ CapabilityBoundingSet=~CAP_SYS_PTRACE                 Service has no ptrace() debugging abiliti
✗ RestrictAddressFamilies=~AF_(INET|INET6)              Service may allocate Internet sockets
✗ RestrictNamespaces=~CLONE_NEWUSER                     Service may create user namespaces
✗ RestrictAddressFamilies=~…                            Service may allocate exotic sockets
✓ CapabilityBoundingSet=~CAP_(CHOWN|FSETID|SETFCAP)     Service cannot change file ownership/acce
✗ CapabilityBoundingSet=~CAP_(DAC_*|FOWNER|IPC_OWNER)   Service may override UNIX file/IPC permis
✓ CapabilityBoundingSet=~CAP_NET_ADMIN                  Service has no network configuration priv
✓ CapabilityBoundingSet=~CAP_RAWIO                      Service has no raw I/O access
✓ CapabilityBoundingSet=~CAP_SYS_MODULE                 Service cannot load kernel modules
✓ CapabilityBoundingSet=~CAP_SYS_TIME                   Service processes cannot change the syste
✓ DeviceAllow=                                          Service has a minimal device ACL
✗ IPAddressDeny=                                        Service does not define an IP address whi
✓ KeyringMode=                                          Service doesn't share key material with o
✓ NoNewPrivileges=                                      Service processes cannot acquire new priv
✗ NotifyAccess=                                         Service child processes may alter service
✓ PrivateDevices=                                       Service has no access to hardware devices
✓ PrivateMounts=                                        Service cannot install system mounts
✓ PrivateTmp=                                           Service has no access to other software's
✗ PrivateUsers=                                         Service has access to other users
✗ ProtectClock=                                         Service may write to the hardware clock o
✗ ProtectControlGroups=                                 Service may modify the control group file
✓ ProtectHome=                                          Service has no access to home directories
```

Each time there is a green tick, some exposure is prevented; as an example, ProtectHome has a green tick, this means "Service has no access to home directories".

So, tor service has a double sandboxing, with AppArmor and with systemd. Since these settings have been chosen by Tor Project, we will not touch them.

The latest thing to do is to edit "torrc", the tor configuration file.

[code]

sudo nano /etc/tor/torrc

Look at this paragraph:

Entry policies to allow/deny SOCKS requests based on IP address.

First entry that matches wins. If no SocksPolicy is set, we accept

all (and only) requests that reach a SocksPort. Untrusted users who

can access your SocksPort may be able to learn about the connections

you make.

#SocksPolicy accept 192.168.0.0/16

#SocksPolicy reject *

If you use tor on a standalone computer, replace "#SocksPolicy accept 192.168.0.0/16" by "SocksPolicy accept 127.0.0.1". This will restrict the use of tor to your computer only.

If you use tor on a server for your local network, replace "#SocksPolicy accept 192.168.0.0/16" by "SocksPolicy accept 192.168.0.0/16" (if necessary, replace "192.168.0.0/16" by your local network block of addresses); in your firewall, open 9050 port for incoming connections, allowing them for your local network block of addresses.

Save the edited "torrc", and relaunch tor to take into account the changes in "torrc" :

[code]

sudo systemctl restart tor

4) What to do with Mullvad Browser on Tor?

- Internet browsing can be done anonymously, but there are several caveats.

* Among the major search engines, DuckDuckGo seems to be the only to work. Bing gives no answer to any request, and Google redirects you to a captcha because it sees too many requests from Tor internet relays IP addresses (and, even if you succeed the captcha, it refuses to answer the request).

* All the traffic to internet is routed from Tor network to internet by some ~2000 internet relays (at the time of writing, June 2022). Consequences: browsing speed may be slow; Tor internet relays are known, and they can be easily filtered by web servers; some of those relays are used by irrespective users and their IP addresses are blocked by security appliances, preventing the connection from this internet relay to a protected website.

- The main use of Mullvad Browser on Tor is to access "DarkNet", and particularly ".onion" hidden websites (that can be accessed only from within Tor network). Though on the dark web you cannot search for webpages (you are "invited" to those pages, from some discussion forums), here are some entry points:

* A webpage dedicated to guide you (on "ClearNet"):

https://www.webhostingsecretrevealed.net/blog/security/dark-web-websites-onion-links/.

* Z-library connection page (inside Tor network):

http://fr.loginzlib2vrak5zzpcocc3ouizykn6k5qecgj2tzlnab5wcbqhembyd.onion/.

* The Hidden Wiki (inside Tor network):

http://zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page.

5) Conclusion

Mullvad Browser flatpak on Tor Network offers the same privacy protection as Tor Browser.

With flatpak sandboxing of Mullvad Browser, and with AppArmor and "systemd" sandboxing of Tor service, security is much stronger than the one offered by Tor Browser.

Annex 9: Tripwire Tutorial

This tutorial will explain how to detect system files changes and additions using Tripwire.

At initialization, Tripwire creates an encrypted database with the cryptographic hashes of the system files.

Then, at check, Tripwire compares the newly calculated hashes to the stored ones.

User knows what files have been changed or added; he has then to decide if the change is legit (he can use Synaptic history for this) or not. If the change is legit, user updates the database. If not, he reinstalls the changed files, and delete the added ones.

This method can be useful to detect system files alterations by malware.

This tutorial is based on two online ones:

https://www.howtoforge.com/tutorial/how-to-monitor-and-detect-modified-files-using-tripwire-on-ubuntu-1604/,

https://computingforgeeks.com/install-and-configure-tripwire-on-ubuntu/.

1) Install tripwire

Tripwire is available in Linux Mint repositories.

[code]

sudo apt update

sudo apt install -y tripwire

During installation, you will be asked some questions about Tripwire configuration:

- Create new 'site-key' for Tripwire - choose 'Yes' and press Next button to continue.

- For new 'local-key', choose 'Yes' and press Next again.

- For the Rebuild Tripwire Configuration option, choose 'Yes' and press Next.

- Same for Rebuild Tripwire Policy option - choose 'Yes' and press Next.

- Now you will be prompted for the 'site-key' passphrase. Type a passphrase. Repeat the 'site-key' passphrase and Next.

- Now you will be prompted for the 'local-key' passphrase. Type a passphrase. Repeat the 'local-key' passphrase and Next.

And now, tripwire installation completes.

2) Configure Tripwire policy for Linux Mint

After Tripwire installation, we need to initialize the database system. Run the following command for it.

[code]

sudo tripwire --init

You will be asked your super-user password, type it an press Enter; next you will be asked about your local-key passphrase - type your local-key passphrase and press Enter.

And you will likely get several 'No such directory' errors on the terminal output. To solve this error, we need to edit Tripwire configuration file and regenerate the configuration.

Before editing the Tripwire configuration, we need to check which directory doesn't exist, something that you can do using the following command:

[code]

sudo sh -c "tripwire --check | grep Filename > no-directory.txt"

Now you can see all directories and files that do not exist in the following way:

[code]

cat no-directory.txt

Output:

Filename: /etc/rc.boot
Filename: /root/mail
Filename: /root/Mail
Filename: /root/.xsession-errors
Filename: /root/.xauth
Filename: /root/.tcshrc
Filename: /root/.sawfish
Filename: /root/.pinerc
Filename: /root/.mc
Filename: /root/.gnome_private
Filename: /root/.gnome-desktop
Filename: /root/.gnome
Filename: /root/.esd_auth
Filename: /root/.elm
Filename: /root/.cshrc
Filename: /root/.bash_profile
Filename: /root/.bash_logout

Filename: /root/.amandahosts

Filename: /root/.addressbook.lu

Filename: /root/.addressbook

Filename: /root/.Xresources

Filename: /root/.Xauthority

Filename: /root/.ICEauthority

Filename: /proc/9187/fd/3

Filename: /proc/9187/fdinfo/3

Filename: /proc/9187/task/9187/fd/3

Filename: /proc/9187/task/9187/fdinfo/3

Next, go to the Tripwire configuration directory and edit the configuration file twpol.txt:

[code]

cd /etc/tripwire

sudo nano twpol.txt

On the 'Boot Scripts' rule, comment the line as below:

```
(
  rulename = "Boot Scripts",
  severity = $(SIG_HI)
 )
 {
      /etc/init.d          -> $(SEC_BIN) ;
      #/etc/rc.boot        -> $(SEC_BIN) ;
      /etc/rcS.d           -> $(SEC_BIN) ;
```

On the 'System Boot Changes' rule, since these files change at each boot, comment as below (unless you want to keep an eye on what happens during boot):

```
(
  rulename = "System boot changes",
  severity = $(SIG_HI)
 )
 {
      #/var/lock           -> $(SEC_CONFIG) ;
```

#/var/run -> $(SEC_CONFIG) ; # daemon PIDs

#/var/log -> $(SEC_CONFIG) ;

On the 'Root config files' rule, make the following comments, in bold:

```
(
  rulename = "Root config files",
  severity = 100
)
{
      /root                    -> $(SEC_CRIT) ; # Catch all additions to /root
      #/root/mail              -> $(SEC_CONFIG) ;
      #/root/Mail              -> $(SEC_CONFIG) ;
      #/root/.xsession-errors  -> $(SEC_CONFIG) ;
      #/root/.xauth            -> $(SEC_CONFIG) ;
      #/root/.tcshrc           -> $(SEC_CONFIG) ;
      #/root/.sawfish          -> $(SEC_CONFIG) ;
      #/root/.pinerc           -> $(SEC_CONFIG) ;
      #/root/.mc               -> $(SEC_CONFIG) ;
      #/root/.gnome_private    -> $(SEC_CONFIG) ;
      #/root/.gnome-desktop    -> $(SEC_CONFIG) ;
      #/root/.gnome            -> $(SEC_CONFIG) ;
      #/root/.esd_auth         -> $(SEC_CONFIG) ;
      #/root/.elm              -> $(SEC_CONFIG) ;
      #/root/.cshrc            -> $(SEC_CONFIG) ;
      /root/.bashrc            -> $(SEC_CONFIG) ;
      #/root/.bash_profile     -> $(SEC_CONFIG) ;
      #/root/.bash_logout      -> $(SEC_CONFIG) ;
      /root/.bash_history      -> $(SEC_CONFIG) ;
      #/root/.amandahosts      -> $(SEC_CONFIG) ;
      #/root/.addressbook.lu   -> $(SEC_CONFIG) ;
      #/root/.addressbook      -> $(SEC_CONFIG) ;
      #/root/.Xresources       -> $(SEC_CONFIG) ;
```

#/root/.Xauthority -> $(SEC_CONFIG) -i ; # Changes Inode number on login

/root/.ICEauthority -> $(SEC_CONFIG) ;

Finally, on the 'Device and Kernel information' rule, comment as below:

```
(
  rulename = "Devices & Kernel information",
  severity = $(SIG_HI),
)
{
    /dev        -> $(Device) ;
    #/proc      -> $(Device) ;
```

Normally, that's it. Check a last time with the content of "no-directory.txt" file that all the errors have been treated. Save the changes and exit the editor.

After editing the config file, implement all changes by recreating the encrypted policy file using the "twadmin" command as shown below:

[code]

sudo twadmin -m P /etc/tripwire/twpol.txt

When asked, type your super-user password and press Enter, then type the 'site-key' passphrase and press Enter. With this, new Tripwire policy is created.

Now, reinitialize the Tripwire database:

[code]

sudo tripwire --init

You will be asked your super-user password, type it and press Enter; next you will be asked about your local-key passphrase - type your local-key passphrase and press Enter.

Check that you get no error this time:

Parsing policy file: /etc/tripwire/tw.pol

*** Processing Unix File System ***

Performing integrity check...

The object: "/root/.cache/doc" is on a different file system...ignoring.

The object: "/root/.cache/gvfs" is on a different file system...ignoring.

The object: "/dev/hugepages" is on a different file system...ignoring.

The object: "/dev/mqueue" is on a different file system...ignoring.

The object: "/dev/pts" is on a different file system...ignoring.

The object: "/dev/shm" is on a different file system...ignoring.

Wrote report file: /var/lib/tripwire/report/michel-G74Sx-20220518-130122.twr

[…]

==========

Error Report:

==========

No Errors

*** End of report ***

The database can be printed with the following command:

[code]

twprint -m d -d /var/lib/tripwire/[b]computername[/b].twd

Where "computername" is found in your terminal prompt:

[code]

username@computername:~$

Be careful, this database is huge, it contains the paths and hashes of 74295 files among 581905 (on my computer). You can display the result on terminal, or redirect the display to a text file.

3) Check integrity of system files

Tripwire has been installed, and the Tripwire policy has been updated and reinitialized. In this step, we will be manually checking the system using Tripwire.

Verify all system files using the following command:

[code]

sudo tripwire --check

When asked, type your super-user password and press Enter, then type the 'site-key' passphrase and press Enter.

If no system file has been changed or added, you will get a report with no violation and no error:

Parsing policy file: /etc/tripwire/tw.pol

*** Processing Unix File System ***

Performing integrity check...

The object: "/dev/hugepages" is on a different file system...ignoring.

The object: "/dev/mqueue" is on a different file system...ignoring.

The object: "/dev/pts" is on a different file system...ignoring.

The object: "/dev/shm" is on a different file system...ignoring.

Wrote report file: /var/lib/tripwire/report/computername-20230531-140307.twr

Open Source Tripwire(R) 2.4.3.7 Integrity Check Report

Report generated by: root

Report created on: mer. 31 mai 2023 14:03:07

Database last updated on: Never

==========

Report Summary:

==========

Host name: computername

Host IP address: 127.0.1.1

Host ID: None

Policy file used: /etc/tripwire/tw.pol

Configuration file used: /etc/tripwire/tw.cfg

Database file used: /var/lib/tripwire/michel-G74Sx.twd

Command line used: tripwire --check

==========

Rule Summary:

==========

Section: Unix File System

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0

Root file-system libraries 100 0 0 0

(/lib)

Critical system boot files 100 0 0 0

Other configuration files 66 0 0 0

(/etc)

Boot Scripts 100 0 0 0

Security Control 66 0 0 0

Root config files 100 0 0 0

Devices & Kernel information 100 0 0 0

(/dev)

Invariant Directories 66 0 0 0

Total objects scanned: 74295

Total violations found: 0

==========

Object Summary:

==========

Section: Unix File System

No violations.

==========

Error Report:

==========

No Errors

*** End of report ***

All rights reserved.

Integrity check complete.

If a system file has been changed or added, the report will include its full path. The user will then need to analyze and decide if the change / addition is legit or not.

Legit changes examples: user has made an update with update manager, or user has installed a new package. Synaptic history of changes will help to decide if change is legit. In that case, user needs to accept changes and update database:

[code]

sudo tripwire --init

If the change is not legit, user needs to reinstall the changed system files (using Synaptic "re installation" option for an installed package), or to delete the added suspicious file (with "sudo rm" command) or to make a complete fresh installation of the system or to restore it from a backup when suspicious changes are too large.

User can plan a periodic check using cron.

Annex 10: Install and Set Up Free Proton VPN

This Annex shows you how to:

- Create a free Proton e-mail address.

- Install and set-up Proton VPN and use its free version with your e-mail credentials.

Warning: as usual, backup your files before to install Proton VPN.

Why a VPN?

- With a VPN you have a partial anonymity: while your true IP address is hidden to the websites you visit, the VPN provider knows all what you do: it just replaces your ISP provider.

- A VPN can be used to change the apparent connection country and can be used to circumvent country related access restrictions.

- You can use Tor Network within your VPN: visiting the Darknet, your anonymity is reinforced (in the past, flaws in Tor could be used to identify users; and your browser javascript can reveal things about you; with Tor in a VPN you have a double anonymity layer: if Tor anonymity were broken, the revealed IP address would be the VPN attributed one).

[NB: paid versions of Proton VPN include access to Tor network without the need of Tor service or Tor Browser installation; with free Proton VPN you need to use Tor Browser, or Mullvad Browser over Tor, see Annex 8: Mullvad Browser Flatpak on Tor Network, a Secure Alternative to Tor Browser; in any case, the use of Proton VPN Tor access should be avoided, since it offers only half anonymity, Proton VPN knowing your ISP provided IP address].

- You can use I2P or Freenet within your VPN, your anonymity is reinforced with a double layer.

Why Proton VPN?

- You need to trust your VPN provider: Proton is based in Switzerland, its VPN code is Open Source, and it has been independently audited; Proton VPN can be trusted. See https://en.wikipedia.org/wiki/Proton_VPN.

- Its free VPN has the same speed as its paid versions: pay VPN users pay for free VPN ones. There are however some limitations:

    * Free VPN is limited to one connection.

    * Free VPN has five exit countries (Japan, Netherlands, Poland, Romania, USA), while pay versions have more than 120 exit countries.

    * Secure Core and Netshield are not available, with free version the DNS resolver set in your system is used.

    * Pay versions speed is higher.

    * [...]

See comparison at https://protonvpn.com/pricing.

- Proton VPN is simple to install and to use.

Installation steps:

1) Get a free Proton e-mail address

- Create your Proton account here: https://account.proton.me/mail/signup

  * You choose your username and password, and get an e-mail address of the kind "username@proton.me".

  * If you use your usual ISP parameters, you will probably be just asked to solve a captcha; if you use Tor or a VPN, you will be probably asked to use an e-mail address that will be verified, and if you don't want to use your usual e-mail address, you will use a disposable one (go to https://yopmail.com/ and create your disposable address, of the kind "username@yopmail.com"; it requires no password; once your address is created, the inbox will open, keep it in a tab of your browser until you receive a validation e-mail from Proton).

- When proton e-mail creation account is completed, save your Proton credentials (e-mail address, password).

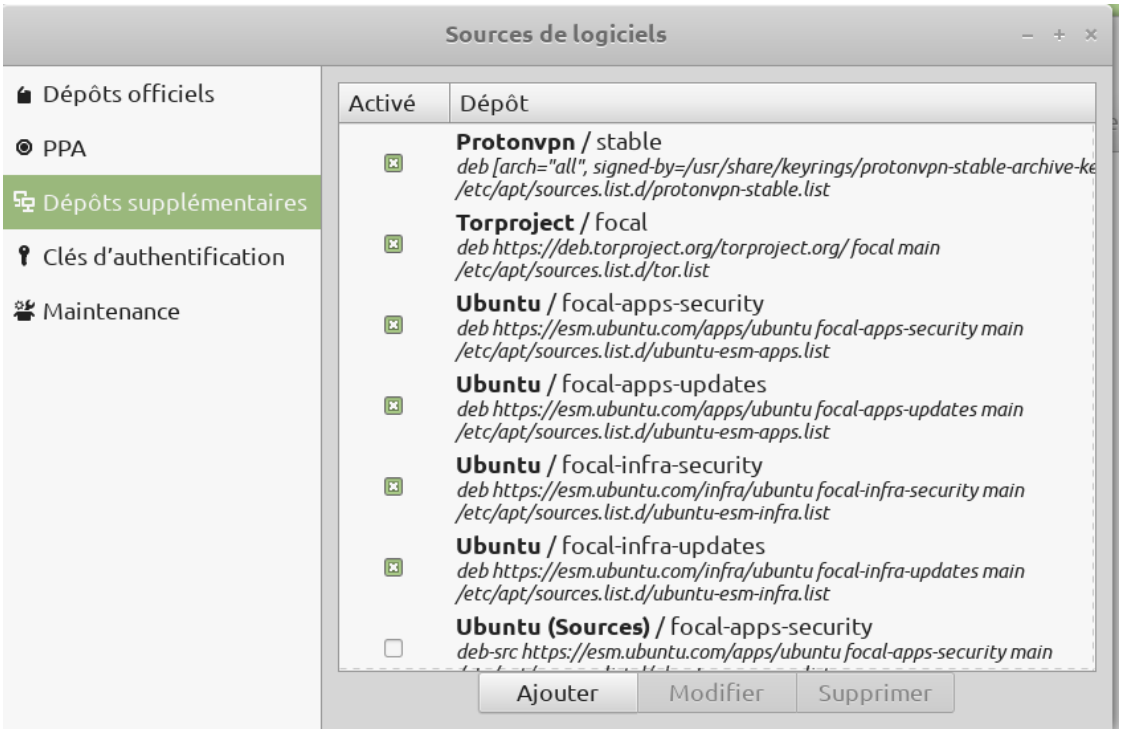
2) Get and install free Proton VPN

1st method, using Proton VPN stable repository:

- Download Proton VPN repository installer from this page https://protonvpn.com/support/official-linux-vpn-ubuntu/.

- Once downloaded, open and install the deb with gdebi (open your file manager, select the deb, right-click, and "Open with package installer Gdebi") or with Synaptic, dpkg command…

When done, mintsources will show a new "Protonvpn / stable" repository:



- Install Proton VPN

[code]

sudo apt-get update

sudo apt-get install proton-vpn-gnome-desktop

installation of tray icon

sudo apt install gir1.2-appindicator3-0.1

When installing tray icon, you may be requested to select your default displays manager. You can get its full path with:

[code]

grep '/usr/s\?bin' /etc/systemd/system/display-manager.service

(for me the answer is "ExecStart=**/usr/sbin/lightdm**").

- Restart your computer.

2nd method, using flatpak version:

Proton VPN is also available as a flatpak. It can be installed with the command:

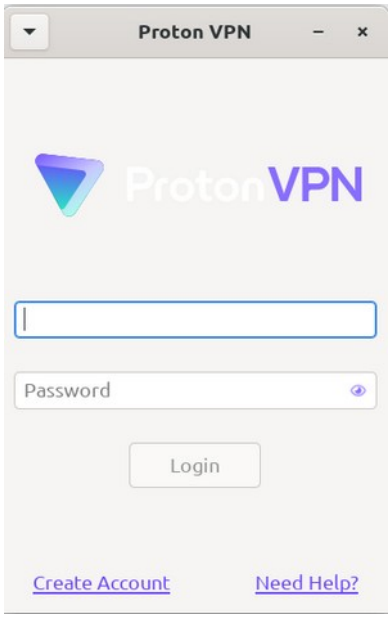
[code]

flatpak install com.protonvpn.www

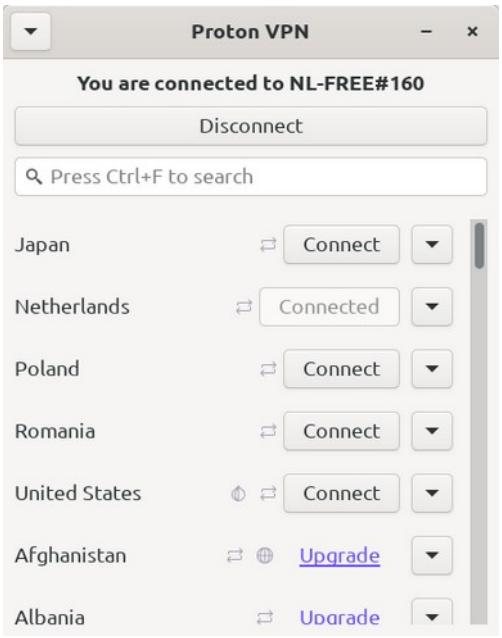
See Annex 6: Flatpak Tutorial.

3) Launch and connect Proton VPN

- From your menu, in internet category, launch Proton VPN GUI client. At launch, enter your Proton e-mail address (username@proton.me) and password:

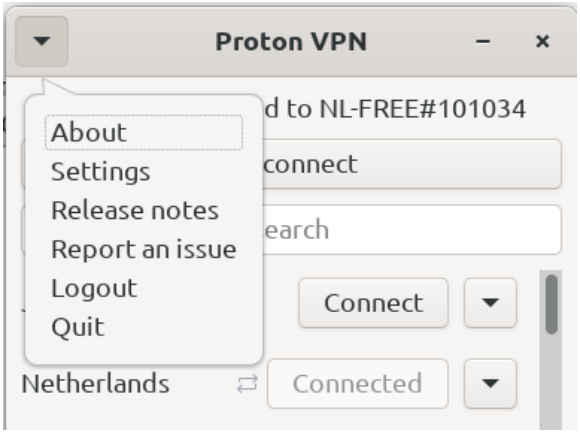


At next screen, connect by clicking on "Quick Connect" button. Here is the result in Proton VPN window:



The tray icon shows Proton VPN is connected:

You can access the settings:



4) Disconnect Proton VPN

You can disconnect Proton VPN using its main window (click on "Disconnect" button). You can also use the tray icon.

Once done, you will get your former connection again, without VPN

5) Some tests using Proton VPN

All the tests are done with a Netherlands connection.

DNS checking (https://dnscheck.tools/), showing DNSSEC conformity:

dnscheck.tools - check your dns resolvers

Results More

Hello! Your public IP addresses are:

WORLDSTREAM
89.39.107.193 ptr: 89-39-107-193.hosted-by-worldstream.net Naaldwijk, South Holland, NL

Your DNS resolvers are:

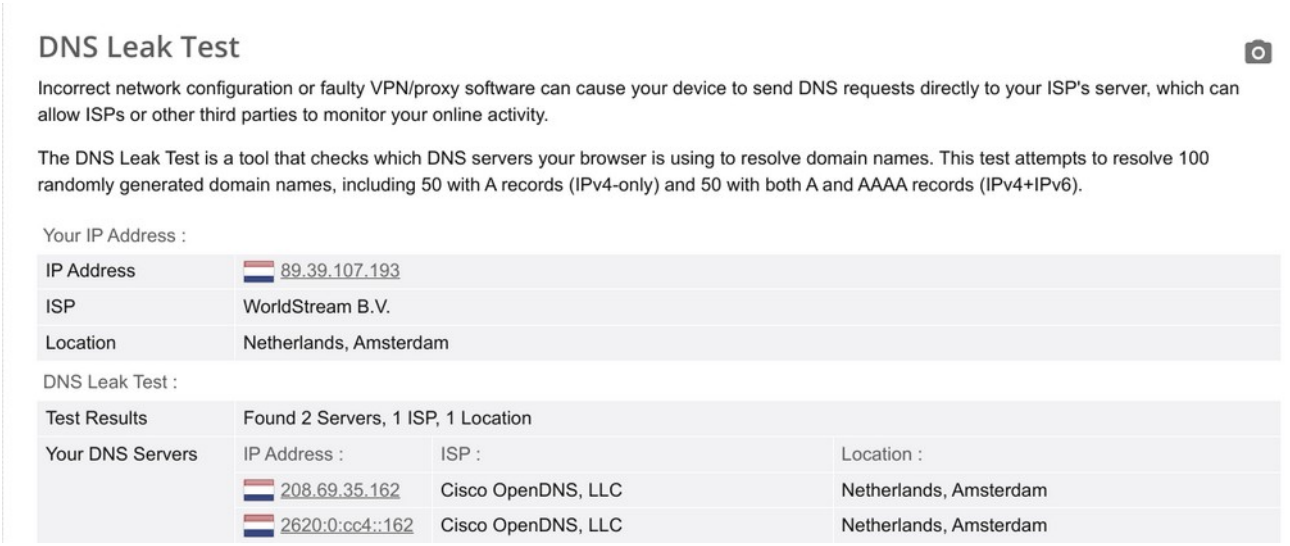
Cisco OpenDNS
208.69.35.163 ptr: r2003.compute.ams1.edc.strln.net Amsterdam, North Holland, NL
2620:0:cc4::163 ptr: r2003.compute.ams1.edc.strln.net Amsterdam, North Holland, NL

WORLDSTREAM
89.39.107.190 ptr: 89-39-107-190.hosted-by-worldstream.net Naaldwijk, South Holland, NL
89.39.107.191 ptr: 89-39-107-191.hosted-by-worldstream.net Naaldwijk, South Holland, NL
89.39.107.192 ptr: 89-39-107-192.hosted-by-worldstream.net Naaldwijk, South Holland, NL
89.39.107.193 ptr: 89-39-107-193.hosted-by-worldstream.net Naaldwijk, South Holland, NL

Great! Your DNS responses are authenticated with DNSSEC:

	ECDSA P-256	ECDSA P-384	Ed25519
Good signature	✓	✓	✓
Bad signature	✓	✓	✓
Expired signature	✓	✓	✓
Missing signature	✓	✓	✓

DNS Leak test (https://browserleaks.com/), showing no DNS leak:



DNS Leak Test

Incorrect network configuration or faulty VPN/proxy software can cause your device to send DNS requests directly to your ISP's server, which can allow ISPs or other third parties to monitor your online activity.

The DNS Leak Test is a tool that checks which DNS servers your browser is using to resolve domain names. This test attempts to resolve 100 randomly generated domain names, including 50 with A records (IPv4-only) and 50 with both A and AAAA records (IPv4+IPv6).

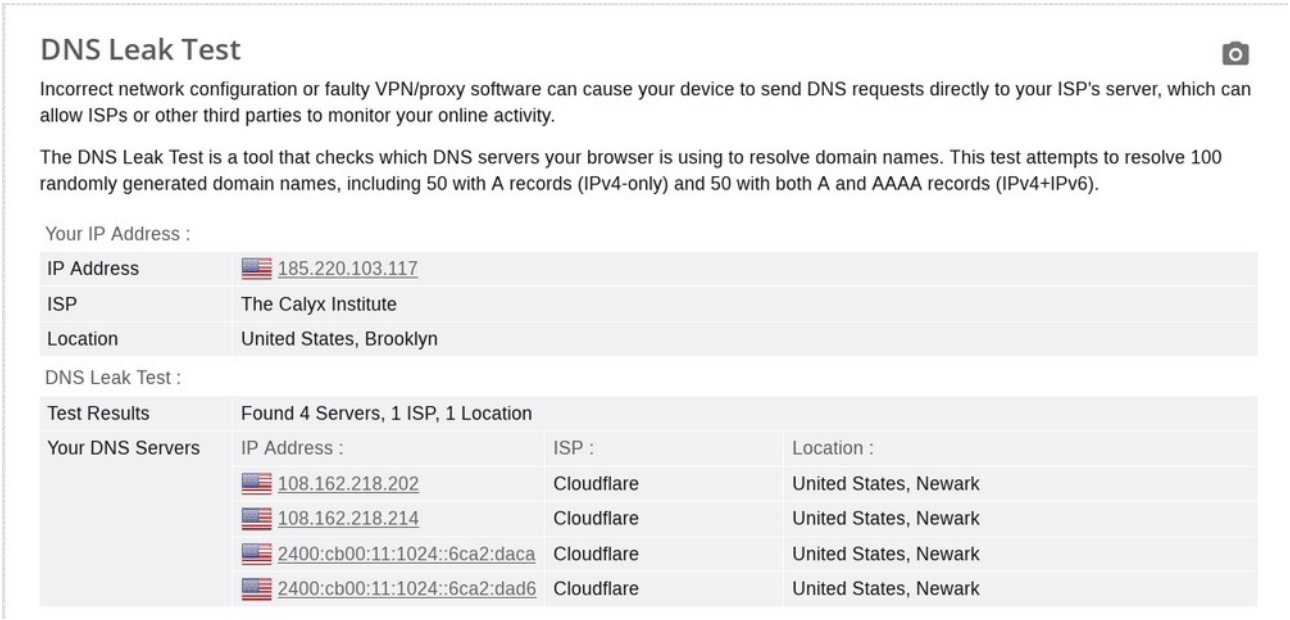
Your IP Address :

IP Address	89.39.107.193		
ISP	WorldStream B.V.		
Location	Netherlands, Amsterdam		

DNS Leak Test :

Test Results	Found 2 Servers, 1 ISP, 1 Location		
Your DNS Servers	IP Address :	ISP :	Location :
	208.69.35.162	Cisco OpenDNS, LLC	Netherlands, Amsterdam
	2620:0:cc4::162	Cisco OpenDNS, LLC	Netherlands, Amsterdam

Using Mullvad Browser on Tor, through Proton VPN, showing no DNS leak:



DNS Leak Test

Incorrect network configuration or faulty VPN/proxy software can cause your device to send DNS requests directly to your ISP's server, which can allow ISPs or other third parties to monitor your online activity.

The DNS Leak Test is a tool that checks which DNS servers your browser is using to resolve domain names. This test attempts to resolve 100 randomly generated domain names, including 50 with A records (IPv4-only) and 50 with both A and AAAA records (IPv4+IPv6).

Your IP Address :

IP Address	185.220.103.117		
ISP	The Calyx Institute		
Location	United States, Brooklyn		

DNS Leak Test :

Test Results	Found 4 Servers, 1 ISP, 1 Location		
Your DNS Servers	IP Address :	ISP :	Location :
	108.162.218.202	Cloudflare	United States, Newark
	108.162.218.214	Cloudflare	United States, Newark
	2400:cb00:11:1024::6ca2:daca	Cloudflare	United States, Newark
	2400:cb00:11:1024::6ca2:dad6	Cloudflare	United States, Newark

WebRTC Leak Test (https://browserleaks.com/), showing no WebRTC Leak:

[NB: during the test, Ungoogled-Chromium browser is unprotected against WebRTC leak; protection is given by Proton VPN only; in normal use, without VPN, Ungoogled-Chromium browser is protected against WebRTC leak by settings in chrome://flags/].

Using Mullvad Browser on Tor, through Proton VPN, showing no WebRTC leak:

[NB: during the test, Mullvad browser is unprotected against WebRTC leak; protection is given by Proton VPN only; in normal use, without VPN, Mullvad browser is protected against WebRTC leak by "media.peerconnection.enabled" set to false in about:config].

Tests conclusions:

- DNS provided by Proton VPN, at least with Netherlands connection, conform to DNSSEC.

- Proton VPN does not leak any information regarding the DNS used by your computer without VPN and the ISP provided IP address.

- Mullvad Browser on Tor, being used through Proton VPN, does not leak any information regarding the DNS used by Proton VPN and the IP address its provides.

6) Proton VPN removal

If you want to remove Proton VPN from your operating system and have installed it from Proton VPN repository:

- In Proton VPN GUI, disable Kill Siwtch (set it to OFF), then exit Proton VPN.

- Uninstall it:

    [code]

    sudo apt autoremove proton-vpn-gnome-desktop && sudo apt purge protonvpn-stable-release

    rm -rf ~/.cache/proton

    rm -rf ~/.config/proton

You can keep "gir1.2-appindicator3-0.1" since it can be used by other applications.

Finally, check in mintsources that "Protonvpn / stable" repository has been removed.

If you want to remove Proton VPN from your operating system and have installed it as a flatpak:

- In Proton VPN GUI, disable Kill Siwtch (set it to OFF), then exit Proton VPN.

- Uninstall it:

    [code]

    flatpak uninstall com.protonvpn.www

    flatpak uninstall --unused

    rm -rf ~/.var/app/com.protonvpn.www

Uninstall troubleshooting:

- When the app is still installed, you can disable Kill Switch from within the app. If you have uninstalled the app with the kill switch still enabled, it will be no longer possible to disable Kill Switch, and you may not be able to access the internet.

Here is how to fix the problem, once Proton VPN has been uninstalled:

[from: https://protonvpn.com/support/official-linux-vpn-mint/, "2. How to disable Kill Switch if you have uninstalled the app"]

a) Identify the name of the Proton VPN connection. To do this, run:

[code]

nmcli connection show --active

You will now see a list of active connections.

b) Look for any connections names that start with **pvpn-.** This usually includes **pvpn-killswitch** and **pvpn-ipv6leak-protection**, and may include **pvpn-routed-killswitch**. Delete all these connections using the following command:

[code]

nmcli connection delete [connection name]

As an example:

[code]

nmcli connection delete pvpn-killswitch

c) Once you've done this, run the following command again to make sure that all Proton VPN connections have been deleted:

[code]

nmcli connection show --active

If any remain, delete them as described above.

7) Troubleshooting

- You need to disable Kill Switch to see your local network devices.

- You need to disable Kill Switch to print on a Wi-Fi printer connected to your local network. This may not be enough:

I have a Wi-Fi laser printer, Brother HLL2375DW, on my LAN, connected to the same router as my computer and within the same subnetwork:

* router IP address on LAN: 192.168.1.1

* computer IP address on LAN: 192.168.1.2, static

* printer IP address on LAN: 192.168.1.10, static

On my first tests, I could not print when connected to Proton VPN, (even with Kill Switch disabled). With the help of Proton VPN customer support, I found the solution; it consists of using a connection URI of the kind "ipp://Printer_IP_address:Port_number/ipp" in CUPS:

* Launch CUPS interface, http://localhost:631/admin,

* then click on "Manage printers",

* click on the printer (HLL2375DW for me),

* in "Administration" menu, select "Modify printer",

* enter your username and password in the windows that requests them, and change the connection URI and driver as in the following example.

My connection URI and driver were:

Old connection URI:

dnssd://Brother%20HL-L2375DW%20series._ipp._tcp.local/?uuid=e3248000-80ce-11db-8000-3c2af4d27073

Old driver:

Brother HLL2375DW for CUPS (grayscale, 2-sided printing)

And I replaced them by:

New connection URI:

ipp://192.168.1.10:631/ipp

New driver:

HL-L2375DW series - IPP Everywhere (grayscale, 2-sided printing)

[Changing the driver for an IPP Everywhere one was necessary to use ipp connection].

You can get your printer IP address and port, and find if it is compatible with IPP Everywhere using the command:

[code]

avahi-browse -rt _ipp._tcp

- If you shut down your computer without having disconnected and exited Proton VPN, at next system start your connection will not work, and Proton VPN will not launch correctly and will not be able to solve the problem.

Solution is the same as when having uninstalled Proton VPN with Kill Switch enabled, see Here is how to fix the problem.

- If you use Chromium browser, or any browser based on Chromium (Google Chrome, Ungoogled-Chromium, Microsoft Edge etc.) you may have a keyring conflict between Proton VPN and your browser, with the browser or Proton VPN GUI asking for a password to access the keyring content, or requesting to create a new default keyring.

The only way to solve this conflict is to have a default keyring unlocked at login.

Here is how to proceed; this supposes that "gnome-keyring" and "libpam-gnome-keyring" are installed on your system (installed by default on Linux Mint).

You need a keyring manager. You can use seahorse, available from Linux Mint / Ubuntu repositories or as a flatpak.

Install seahorse:

[code]

sudo apt install seahorse

or

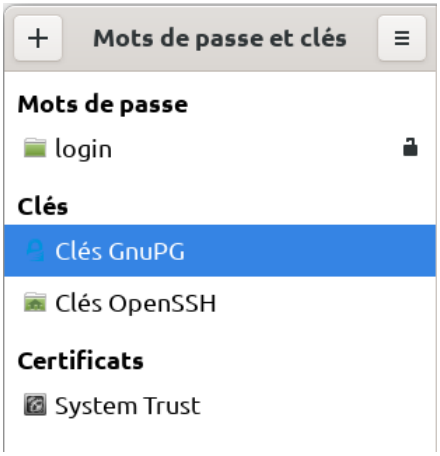
[code]

flatpak install org.gnome.seahorse.Application

(if you install flatpak version, see Annex 6: Flatpak Tutorial)

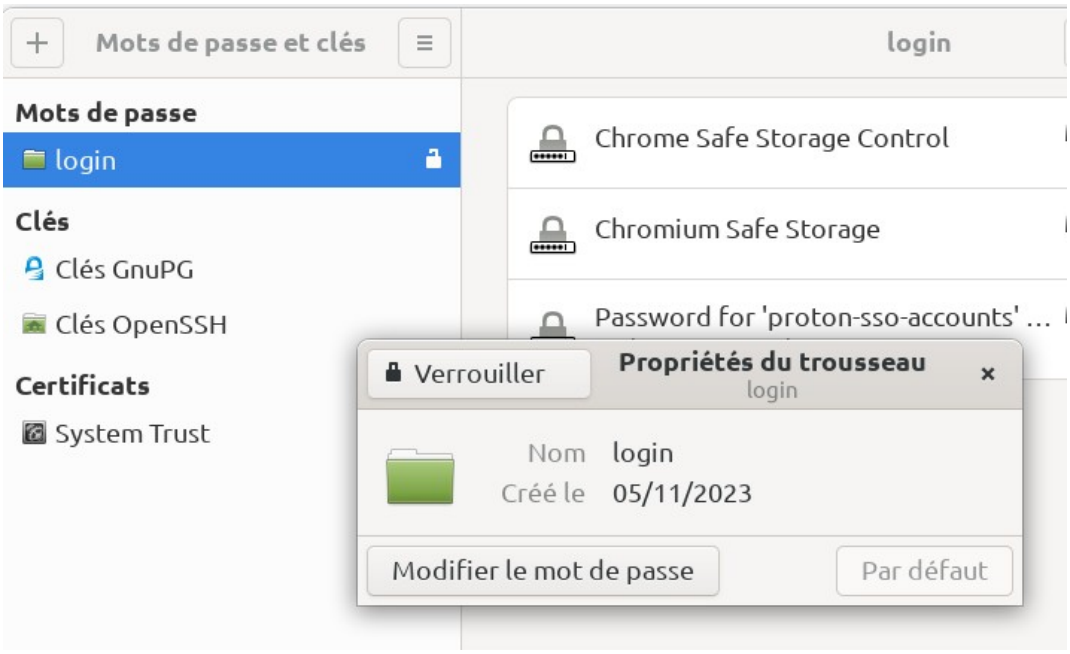
Once seahorse installed, launch it.

Under the "Passwords" category, you may have one or several keyrings. Create a new keyring, named "login".



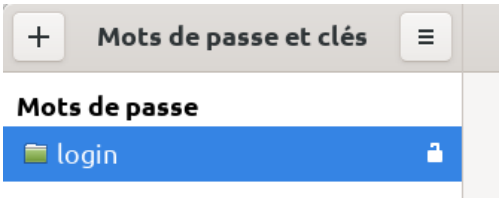
The password of this keyring needs to be the same as your login password (and, if you don't use any password to start your computer, there should be no password on the "login" keyring).

Set this keyring as default, with the keyring properties (dialog box accessed with a right click):

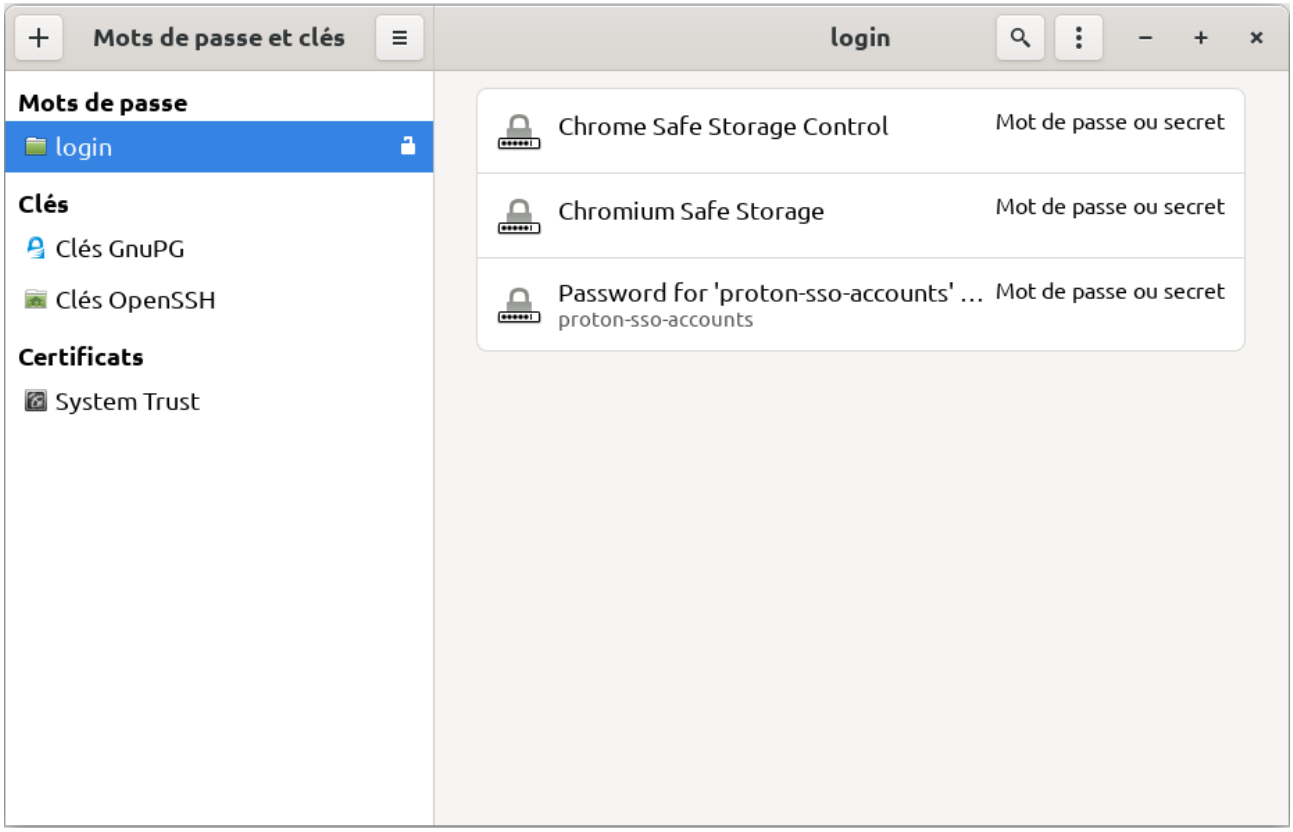


As you can see, once done, "Default", or here its French translation "Par défaut", are grayed.

With these settings, when you start your computer, the "login" keyring will be automatically unlocked by "libpam-gnome-keyring", as you can see:



Your Chromium browser and Proton VPN will now be able to write their keys in the default "login" keyring, without asking for any password, or without requesting to create a new default keyring:



To avoid later problems, limit writing rights:

[code]

chmod 700 ~/.local/share/keyrings

chmod 700 ~/.local/share/keyrings/*.*

8) Use Proton VPN browsers extensions

Instead of installing Proton VPN "system-wide", you can just use browsers extensions, now even with free version of Proton VPN.

Those extensions are available for Firefox, LibreWolf, Waterfox, Mullvad Browser and generally for Firefox-based browsers; and for Chrome, Chromium, Ungoogled-Chromium, Brave, Edge, and generally for Chromium-based browsers.

Download and installation instructions for Firefox-based browsers:

https://protonvpn.com/download-firefox-extension.

Download and installation instructions for Chromium-based browsers:

https://protonvpn.com/download-chrome-extension.

Note that the extensions, unlike the system installed version, do not allow the user to choose its connection among the five countries available for free subscription (Japan, Netherlands, Poland, Romania, USA); an arbitrary choice is done by the extension when connecting.

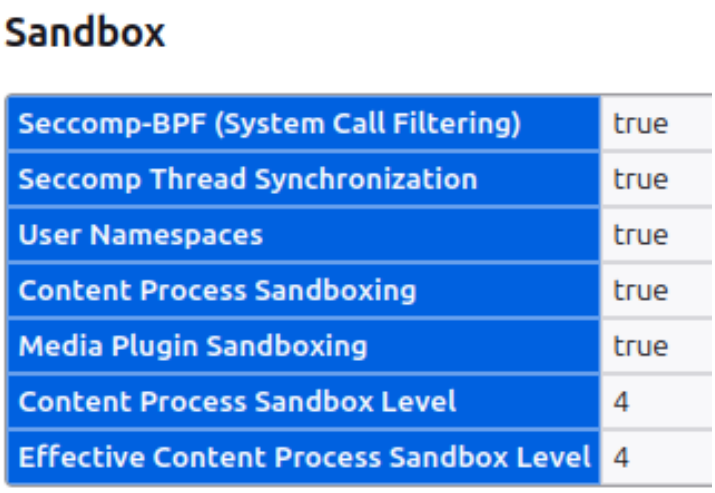
Annex 11: Enhancing Firefox Security and Privacy

This annex shows detailed techniques to enhance Firefox security and privacy. It is valid for Firefox 131 versions, and might not apply in future versions of Firefox.

1) Sandboxing:

- Firefox includes its own sandboxing.

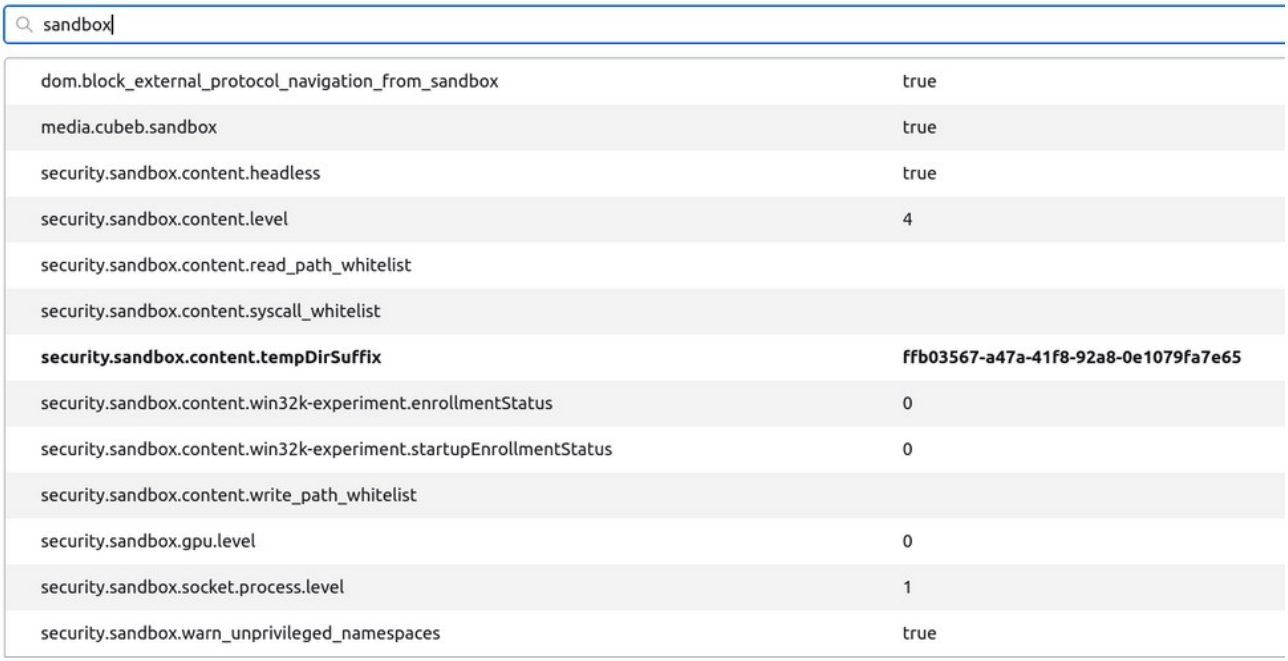
Screen capture from "about:support":



Sandbox

Seccomp-BPF (System Call Filtering)	true
Seccomp Thread Synchronization	true
User Namespaces	true
Content Process Sandboxing	true
Media Plugin Sandboxing	true
Content Process Sandbox Level	4
Effective Content Process Sandbox Level	4

This other screen capture, from "about:config", shows that GPU is not sandboxed, since "security.sandbox.gpu.level" value is "0" (Mozilla did sandbox the GPU on Firefox for Windows, but did not care to do the same with Firefox for Linux):



sandbox

dom.block_external_protocol_navigation_from_sandbox	true
media.cubeb.sandbox	true
security.sandbox.content.headless	true
security.sandbox.content.level	4
security.sandbox.content.read_path_whitelist	
security.sandbox.content.syscall_whitelist	
security.sandbox.content.tempDirSuffix	**ffb03567-a47a-41f8-92a8-0e1079fa7e65**
security.sandbox.content.win32k-experiment.enrollmentStatus	0
security.sandbox.content.win32k-experiment.startupEnrollmentStatus	0
security.sandbox.content.write_path_whitelist	
security.sandbox.gpu.level	0
security.sandbox.socket.process.level	1
security.sandbox.warn_unprivileged_namespaces	true

Moreover, without sandboxing, Firefox has full read / write access on your home files, including in unattended mode.

Finally, Firefox does not run fully isolated from the operating system and, each year, several successful "Zero Day" exploits are reported.

For those reasons, using Firefox without an extra sandboxing is at risk.

- With Firefox on Linux Mint 22.x / Ubuntu 24.04, "unprivileged user namespaces" is no longer enabled by default, while it is still enabled in former versions of Linux Mint.

An explanation is given here: https://support.mozilla.org/en-US/kb/install-firefox-linux#w_security-features-warning.

> The sandbox in Firefox makes use of unprivileged user namespaces when creating new processes for enforcing more security. This can be considered a security risk, therefore some Linux distributions have started to restrict its usage and only allow it to work where there is an AppArmor profile.

Since the use of unprivileged user namespaces may be at risk, distributions such as Debian / Ubuntu / Linux Mint disable it. But Firefox makes a large use of unprivileged user namespaces: when it is disabled, Firefox sandbox security is notably reduced.

From Ubuntu 23.10, and so from Linux Mint 22, the use of an AppArmor profile, even a pseudo one, allowing all and blocking nothing, can entitle Firefox to use unprivileged user namespaces. See https://ubuntu.com/blog/ubuntu-23-10-restricted-unprivileged-user-namespaces.

Linux Mint 22 installation includes AppArmor and an "/etc/apparmor.d/firefox" AppArmor profile for Firefox, from Ubuntu repositories; this profile includes "userns" rule and allows Firefox (installed as a deb) to use "unprivileged user namespaces".

Users who prefer using "Mozilla builds" self-updating versions of Firefox should create a specific AppArmor profile; to create it:

In /etc/apparmor.d/, create a file with the name firefox-local; in the file, add the following (this assumes the Firefox install is at $HOME/bin/):

> [code]
>
> # This profile allows everything and only exists to give the
>
> # application a name instead of having the label "unconfined"
>
> abi <abi/4.0>,
>
> include <tunables/global>
>
> profile firefox-local
>
> /home/<USER>/bin/firefox/{firefox,firefox-bin,updater}
>
> flags=(unconfined) {
>
> userns,

```
    # Site-specific additions and overrides. See local/README for details.
    include if exists <local/firefox>
}
```

Replace <USER> with your Linux username. Once you have saved the file, run "sudo systemctl restart apparmor.service" in the Linux terminal.

Here are the possibilities to add extra sandboxing to Firefox:

- Use Firejail: download and install Firejail on your computer, then use "firejail firefox %u" to launch Firefox with Firejail sandboxing.

Pros:

* Firefox own sandboxing is reinforced by Firejail one, successful exploits are more difficult, files permissions are restricted.

Cons:

* Firejail sandboxing has very restrictive file permissions; adding extra permissions requires copying "/etc/firejail/firefox.profile" to "~/.config/firejail/firefox.profile" and modify "~/.config/firejail/firefox.profile" with whitelist and/or noblacklist instructions. But Firejail does not offer the possibility to access a directory in read-only mode: if you whitelist your Documents directory in "~/.config/firejail/firefox.profile", Firefox will access it in read and write modes.

* Firejail sandboxed applications still use operating system libraries and dependencies, separation between the application and the operating system is not complete, and attacks exploiting "Zero Day" vulnerability might put the system in an unstable state and be successful (though Firejail reduces this risk).

* GPU is still not sandboxed.

* On linux Mint 22 / Ubuntu 24.04 LTS, unprivileged users namespaces is not enabled in Firefox sandbox, since Firejail sandboxes Firefox with AppArmor but the "/etc/apparmor.d/firejail-default" profile it uses does not contain the "userns" rule nor "CAP_SYS_ADMIN" and Firefox own AppArmor profile is replaced by Firejail one.

See Firejail and read its online documentation at https://firejail.wordpress.com/.

- Use Flatpak or Snap versions of Firefox: since they do not use system libraries and dependencies but their own runtimes or cores, separation between application and operating system is total, including GPU sandboxing.

Flatpak Firefox and snap Firefox are verified applications, published by Mozilla.

I recommend the use of flatpak Firefox.

Pros:

* GPU is sandboxed

* Firefox own sandboxing is reinforced by flatpak sandboxing, based on bubblewrap.

* Firefox runs fully isolated from the operating system.

* It is very easy to change the files permissions in unattended mode, using Flatseal, itself a flatpak, in GUI mode: in the screen capture that follows, "~/Documents", "/tmp" and "~/opt/vdhcoapp" have been user added, in read-only mode ":ro".

Cons:

* Firefox own sandbox cannot use unprivileged users namespaces, it seems that Flatpak disables unprivileged user namespaces due to the increased attack surface they bring with them;

See https://discuss.privacyguides.net/t/does-flatpak-weaken-chromium-firefoxs-sandbox/13373/2, https://github.com/flatpak/flatpak/issues/5921 and https://github.com/flatpak/flatpak/issues/5879#issuecomment-2255568180 ("threat model B").

Screen capture from "about:support", flatpak version:

Sandbox

Seccomp-BPF (System Call Filtering)	true
Seccomp Thread Synchronization	true
User Namespaces for privileged processes	true
User Namespaces	false
Content Process Sandboxing	true
Media Plugin Sandboxing	true
Content Process Sandbox Level	4
Effective Content Process Sandbox Level	4

See Flatpak and Annex 6: Flatpak Tutorial.

[Snap version of Firefox might also be a good choice; snap includes the userns interface, allowing an application to create new user namespaces, see https://snapcraft.io/docs/userns-interface; this

could allow snap Firefox to use unprivileged user namespaces in its own sandbox, to be checked. However, snaps file permissions cannot be easily changed, with command line only, see https://ubuntu.com/blog/a-guide-to-snap-permissions-and-interfaces *and that may be a blocker for unskilled users].*

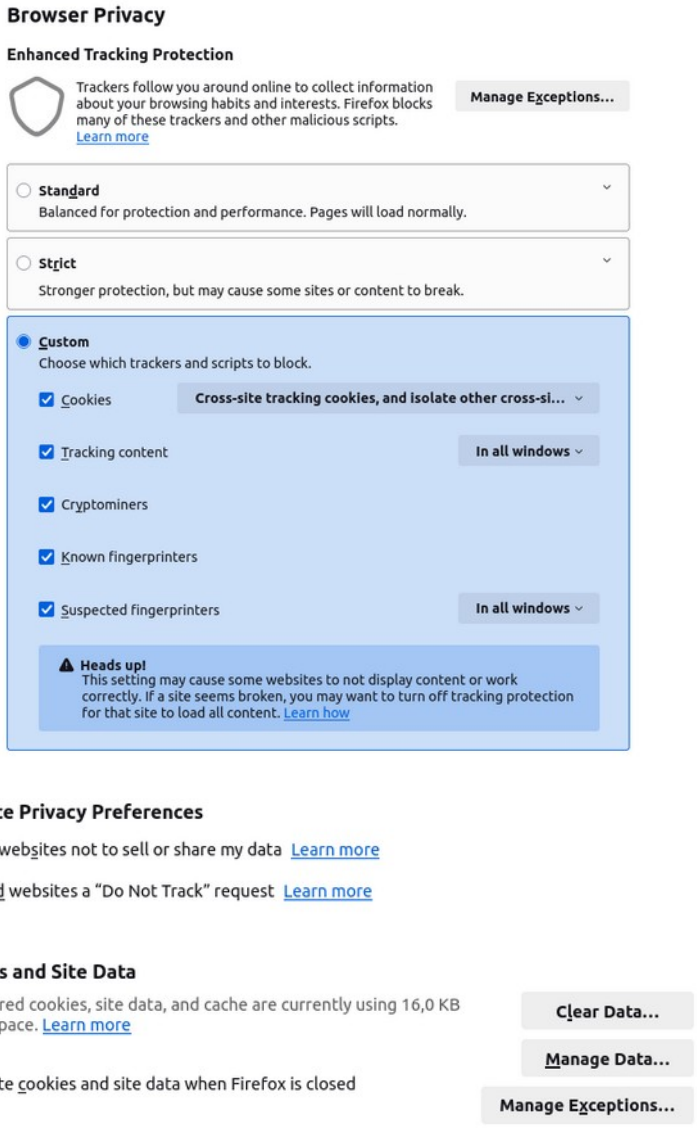
2) Firefox privacy and security settings:

Firefox does make automatic connections; these connections are detailed and explained in the following support page:

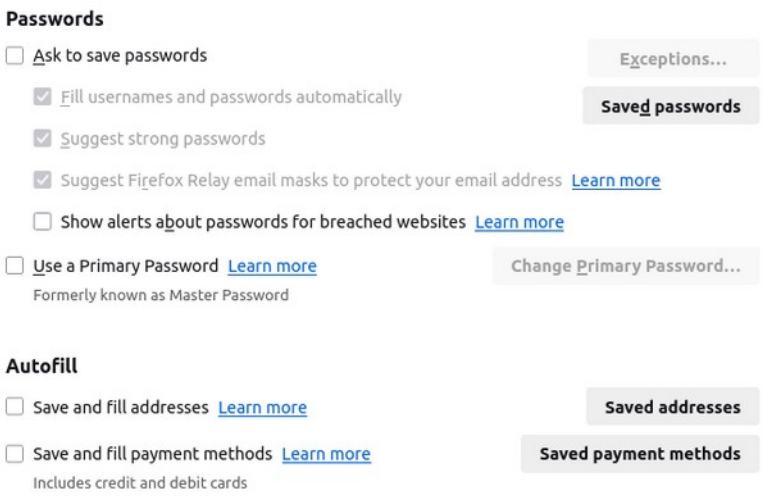
https://support.mozilla.org/en-US/kb/how-stop-firefox-making-automatic-connections

Once having read and understood why Firefox makes automatic connections and how to stop them, user can set Firefox; here are my proposed settings:

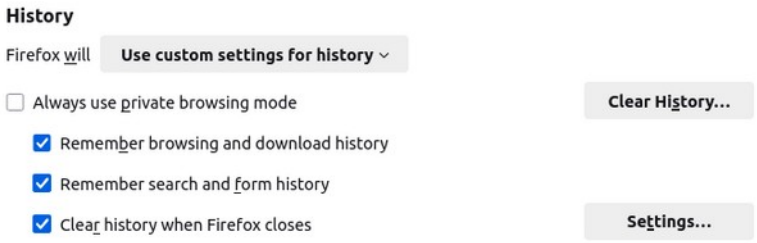
Browser privacy:



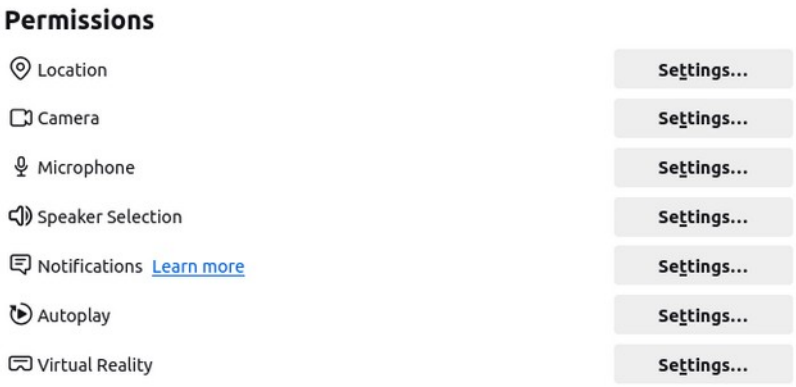
Saving passwords and using Autofill is at risk:



History settings:



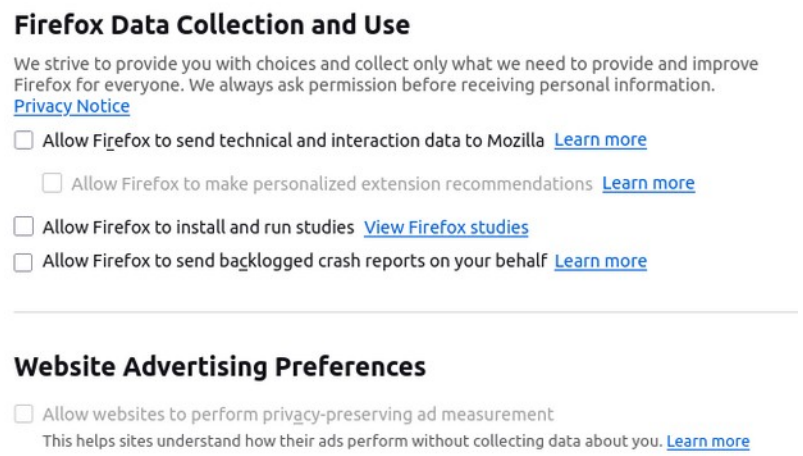
Permissions settings:



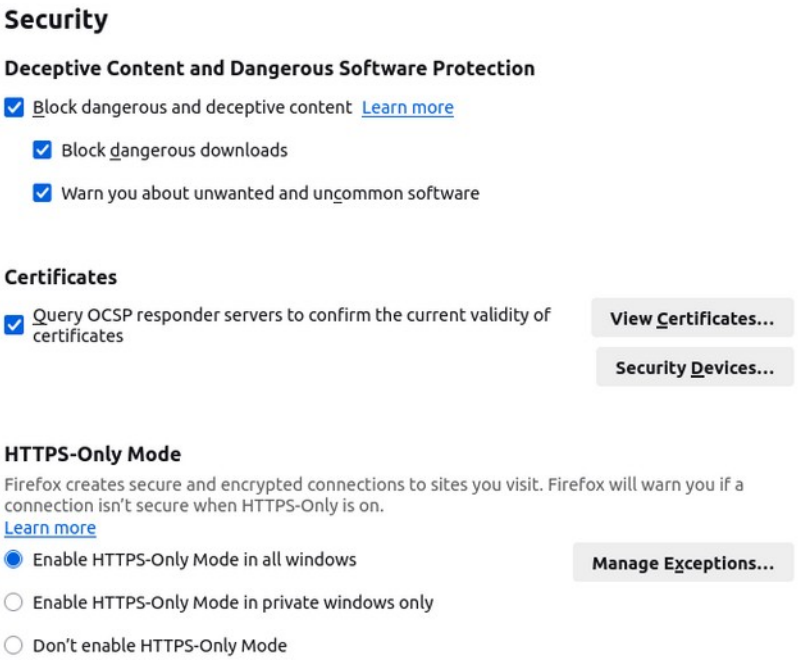
For each kind of permission, click on the settings button, then select "Block new requests asking to access your location", except for permissions you may allow on request.

Pop-ups and add-ons:

Firefox telemetry:



Security:



A note about Safe Browsing, or "Deceptive Content and Dangerous Software Protection".

Firefox uses Google Safe Browsing API against malware; the way its use this API is explained here, https://support.mozilla.org/1/firefox/131.0.3/Linux/en-US/phishing-malware

In summary:

* Every 30 minutes, the list of URLs to avoid is downloaded from Google servers and installed locally, on your computer, in Firefox configuration.

* The comparison between URLs and the list is done locally, on your computer.

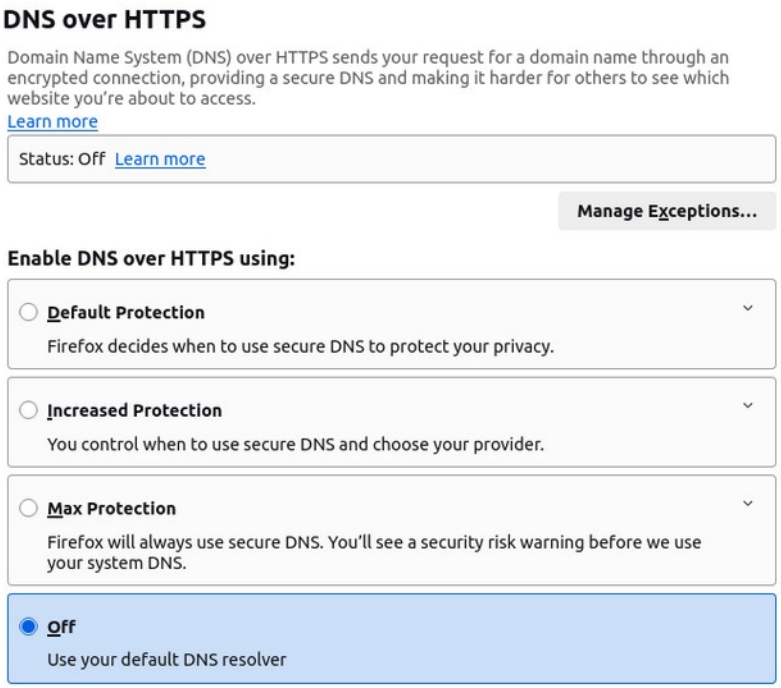
* Firefox sends a URL to Google servers very rarely, in two occasions: when a URL matches the list, in order to perform a double check; when user wants to suggest and URL to be added to the list.

* All data exchanges with Google servers are done anonymously.

→ User can use "Deceptive Content and Dangerous Software Protection" without any privacy risk.

DNS over HTTPS:

I recommend to disable DNS over HTTPS and, instead, use DNS over TLS, more effective since it is a system-wide setting (see Reduce what your ISP can know):

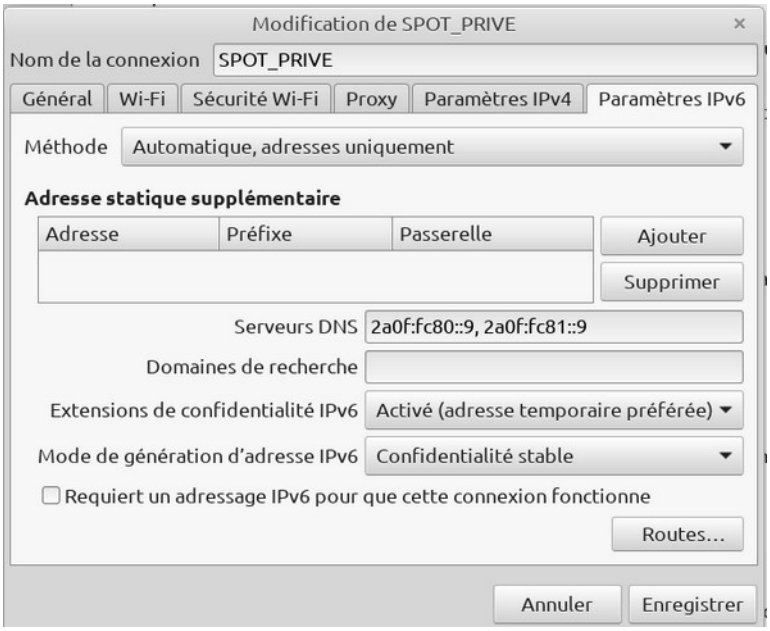


3) Extra protection against ads, malware and anti-tracking:

* Preliminary notes:

- Some Internet Service Providers attribute a fixed IPV4 public address, that never changes even when the modem-router and the computer are switched off and on. In that case, anti-tracking techniques alone are useless, and the use of a VPN is mandatory. See Annex 10: Install and Set Up Free Proton VPN.

- IPV6 addresses are built from your computer MAC address and are not changing. Ubuntu and Linux Mint have a way to reduce this impact by creating a temporary IPV6 address, in the network Manager. Activate "IPV6 confidentiality extensions", with "Default", "preferred temporary address" or "preferred public address" setting of your choice.



Even with activated IPV6 confidentiality extensions, VPN use is preferable. See Annex 10: Install and Set Up Free Proton VPN.

* "uBlock Origin" extension:

See its settings, 4.7) paragraph.

Two mandatory settings:

"Block Outsider Intrusion into LAN" filter list protects against a Firefox vulnerability, allowing a malformed web page to access your LAN (your localhost, the servers it may host, your LAN devices). This vulnerability is under correction, but no schedule is available.

"Easy Privacy" filter list protects against tracking. It works by filtering keywords.

* "Privacy Badger" extension:

It is an anti-tracking extension, with self-learning capability. It works by blocking tracking cookies.

* Or "DuckDuckGo Search and Tracker Protection" extension: it installs DuckDuckGo search as home page, DuckDuckGo search engine by default and DuckDuckGo Privacy Essentials extension.

* "ClearURLs" extension:

This extension will automatically remove tracking elements from URLs to help protect your privacy when browsing through the Internet. Many websites use tracking elements in the URL (e.g. https://example.com?utm_source=newsletter1&utm_medium=email&utm_campaign=sale) to mark your online activity. All that tracking code is not necessary for a website to be displayed or work correctly and can therefore be removed, that is exactly what ClearURLs does.

* "Decentraleyes" extension:

Decentraleyes is a free and open-source browser extension used for local content delivery network (CDN) emulation. Its primary task is to block connections to major CDNs such as Cloudflare and

Google, for privacy and anti-tracking purposes, and serve popular web libraries (such as JQuery and AngularJS) locally on the user's machine.

* Fingerprinting protection:

Firefox does include some fingerprinting protection, available in "about:config" settings. Changing the value of "privacy.resistFingerprinting" from "false" to "true" has the following effects: user-agent is changed (example, from "Mozilla/5.0 (X11; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0" to "Mozilla/5.0 (Windows NT 10.0; rv:131.0) Gecko/20100101 Firefox/131.0"; browser size is reduced and set to a current "standard" value. However, as tested with Browserleaks, https://browserleaks.com/, this protection is very few effective; particularly, user-agent spoofing is defeated by TCP/IP fingerprinting, and Linux OS is recognized; and, having a browser user-agent saying Firefox runs on Windows, while operating system is identified as Linux just increases the browser fingerprint.

One could change "Time To Live" (TTL) and "TCP Window Size" values with sysctl command to simulate another operating system:

Operating System	Time To Live	TCP Window Size
Linux (Kernel 2.4 and 2.6)	64	5840
Google Linux	64	5720
FreeBSD	64	65535
Windows XP	128	65535
Windows Vista and 7 (Server 2008)	128	8192
iOS 12.4 (Cisco Routers)	255	4128

With that, and the user-agent set to "Mozilla/5.0 (Windows NT 10.0…", Firefox has chances to lure web servers using passive TCP/IP fingerprinting and be detected as running on a Windows operating system (Windows is more common than Linux Mint / Ubuntu, so Firefox fingerprint would give less identifying information; moreover, potential attacks might target Firefox running on Windows and not Firefox running on Linux).

But changing those settings do not prevent all the ways to identify Linux OS. Some web servers use active OS fingerprinting specialized tools to identify the operating system and are not lured by TTL and TCP Window Size changes.

→ Spoofing the browser user-agent is never a good idea, it just increases the browser fingerprint, and "privacy.resistFingerprinting" should be let to "false".

Better protection is reached with a set of two Firefox extensions: "CanvasBlocker" (Protected fingerprinting APIs: canvas 2d, webGL, audio, history, window, DOMRect navigator, screen) and "Font Fingerprint Defender". The extensions work by randomizing audio context, canvas, fonts and WebGL fingerprints and reduce the identification information got from these fingerprints *[note that randomizing a fingerprint is in itself a fingerprint, but gives less identifying information than without randomization]*.

WebGL fingerprinting protection can further be improved by a setting in "about:config": set "webgl.enable-debug-renderer-info" to false, your GPU will not be identified.

Fingerprinting can still be reduced by using rounded length and height of your browser size.

My laptop has a 17.4" screen, with 1920x1080 resolution and 129 dots per inch (dpi); by default, in Firefox, the browser screen size is 1440x810 and the browser size 1440x843, as checked at https://browsersize.com/.

To have rounded values, in "about:config", look for "layout.css.devPixelsPerPx"; change default value (-1) for a positive value; with a value of 0.9 for "layout.css.devPixelsPerPx"; in Firefox, for my computer, the browser screen size is now 1600x900 and the browser size 1600x730. Test different positive values (0.9, 0.8…) and see the result.

[A final comment about fingerprinting protections: browser fingerprinting can be reduced, but not totally prevented].

* WebRTC Leak:

The WebRTC Leak is critical for anyone using a VPN or a browser on Tor Network, as it leverages the WebRTC API to communicate with a STUN server and potentially reveal the user's real local and public IP addresses, even when using a VPN, proxy server, or behind a NAT.

To disable WebRTC in Firefox: type "about:config" in the address bar and press Enter. In the search bar, type "media.peerconnection.enabled" and double-click the preference to set its value to false.

* First Party Isolation:

In short, First-Party Isolation (FPI) restricts access to cookies, the cache, and similar data to domain level only. This means that when you enable FPI, advertising agencies cannot use cookies to track your Internet activity as you browse the Web.

It is the strongest Firefox intrinsic protection against cookies tracking. The caveat is that some websites may not load correctly.

To enable it, in "about:config" change "privacy.firstparty.isolate" value from false to true.

4) Protection against JavaScript and CSS attacks:

JavaScript and CSS can be used as vectors attacks:

* "NoScript Security Suite" extension:

It allows to selectively enable JavaScript on websites you trust. It allows JavaScript, and other executable content (such as script, object, media, frame, font, webgl, fetch, ping, noscript, unrestricted CSS, rendering of plain HTML frames...) to run only from trusted domains of your choice (e.g. your banking site), thus mitigating remotely exploitable vulnerabilities, such as Spectre and Meltdown.

It protects your "trust boundaries" against cross-site scripting attacks (XSS), cross-zone DNS rebinding / CSRF attacks (router hacking), and Clickjacking attempts, thanks to its unique ClearClick technology. Moreover, without JavaScript, websites cannot gather much information about your browser and computer, and NoScript improves your privacy.

Though it includes a list of websites with proposed settings, it requires that the user chooses settings for all the websites not in the list, and so it needs time and the extension learning curve is slow.

* "CSS Exfil Protection" extension:

It protects against the "CSS data exfiltration attack", an attack that might gather, as an example, your username and password when connecting to a website. Though this attack has been described in 2020, browsers, including Firefox, have not yet fixed against this attack in 2024! A CSS Exfil vulnerability tester is available here: https://www.mike-gualtieri.com/css-exfil-vulnerability-tester.

* Just in time compilation, JIT:

It is used to speed-up web pages uploading, when they include large amount of JavaScript, by compiling JavaScript Code. But it increases JavaScript attack surface. Disabling it will reduce the attack surface and increase security, JavaScript will be interpreted and not compiled, and web pages making a large use of JavaScript may appear slower.

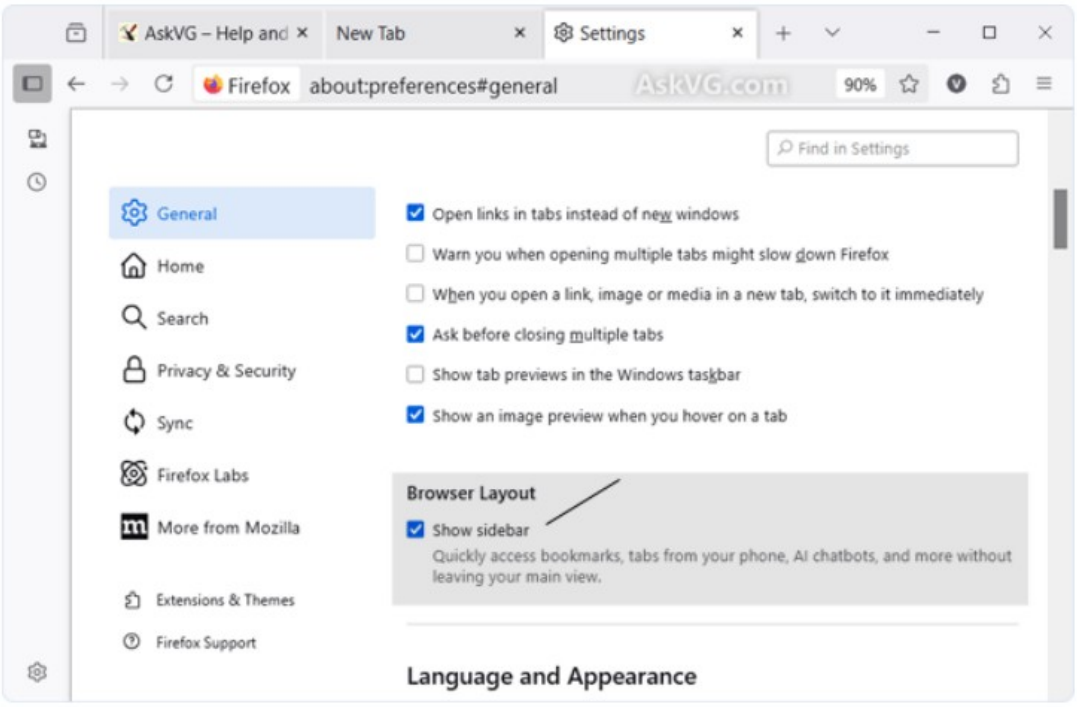
JIT can be easily disabled in Firefox: in "about:config", set "javascript.options.baselinejit" to false.

* Firefox PDF reader scripting:

It is enabled by default in Firefox PDF reader; scripting is used to enhance PDF reader capability, but could also be used to perform an attack with a malformed PDF document. You can disable scripting in PDF readers, with "about:config", change the value of "pdfjs.enableScripting" from true to false.

5) Disabling Firefox integrated AI Chatbot

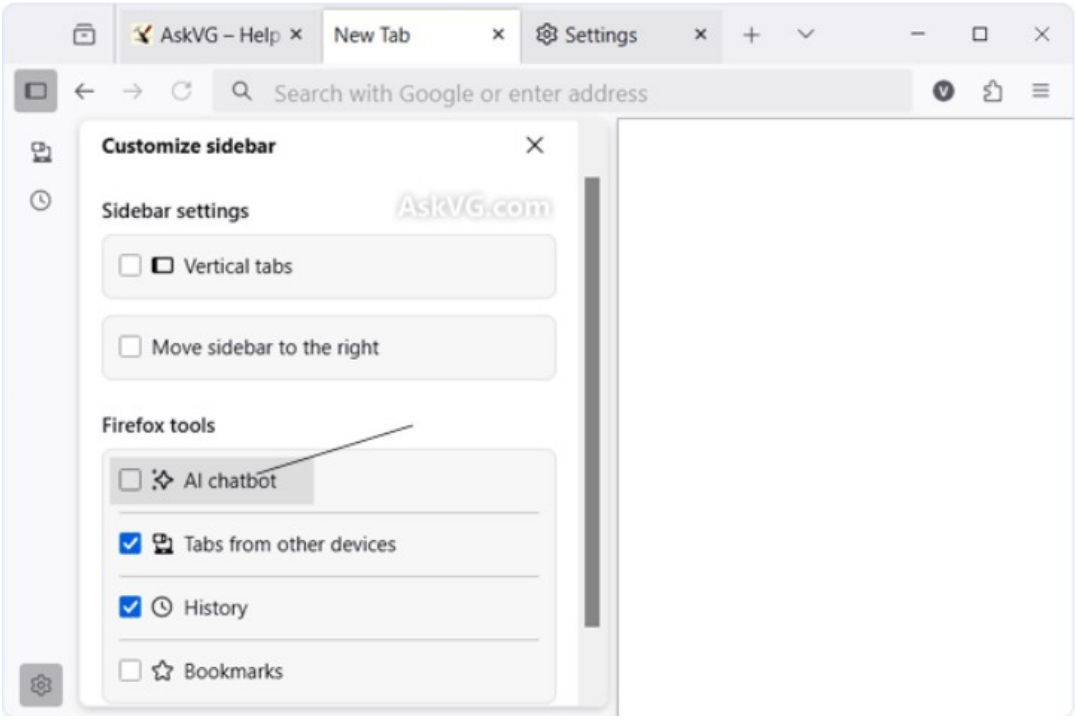
Firefox versions 134 and above integrate an AI Chatbot. It may be selected in the sidebar (when the "show sidebar" option is selected in Firefox parameters):



Normally, it is just there like an url: unless you use it, it should not affect your privacy or Firefox speed.

Some users might prefer to disable AI. Chatbot. There are two ways to do it:

- Click on the cog wheel (customize sidebar) icon present at the bottom of the sidebar. It will open Sidebar Settings panel. Now uncheck or deselect "AI Chatbot" option listed under the Firefox Tools section; it will immediately remove all AI chatbot features from your Firefox web browser.



- The second possibility is to use "about:config", then look for "browser.ml.chat.enabled" and change its value from "true" to "false".

[Note that, if you changed the "browser.ml.chat.enabled" value to false, it is not possible to select again "AI Chatbot" in sidebar settings, you need first to change the value back to "true".]

Annex 12: On-Demand Scan of Confidential Files

This annex explains how to scan your confidential files using ClamTk + ClamAV and Raspirus on your computer. Since the detection rate is probably much lower than VirusTotal one, this method should be reserved to confidential files that cannot be submitted online to VirusTotal.

Precautions:

- make a system backup or a system snapshot before to execute what follows,

- if unsure, make a system scan, see §5.1 Malware and viruses detection.

A – Use of ClamTK + ClamAV

1) Downloading and preparing debs

We will use:

- ClamTk, in its latest (and last) version, 6.18, found at https://github.com/dave-theunsub/clamtk. Direct download link: https://github.com/dave-theunsub/clamtk/releases/download/v6.18/clamtk_6.18-1_all.deb.

Note that ClamTk is no longer maintained (6.18 version is dated 2024-01-27), but still works; it is a GUI for ClamAV and it will work until its dependencies are no longer available, or until ClamAV makes changes in the way "clamscan" and "freshclam" executable binaries are called.

- ClamAV, latest stable version, downloaded from its website: https://www.clamav.net/downloads.

At the time of writing this annex, the latest stable LTS version is 1.4.3, and the package to download is "clamav-1.4.3.linux.x86_64.deb".

- ClamTk is designed for Debian or Ubuntu packaged ClamAV, with two different deb packages "clamav" and "clamav-freshclam". But ClamAV deb package, as downloaded from its website, is an "all-in-one" package, with all the executable binaries and required libraries (and no external dependency). A small trick is needed to have ClamTk working with ClamAV.

* Extract the content of "clamtk_6.18-1_all.deb"; in your file manager, select the deb, then right-click and "Extract here".

* In the extracted directory "/clamtk_6.18-1_all/", you will find to sub-directories, "/DEBIAN/" and "/usr/". In "/DEBIAN/" there is a text file called "control". Open it with a text editor.

Original content:

> Package: clamtk
>
> Version: 6.18-1
>
> Architecture: all
>
> Maintainer: Dave M <dave.nerd@gmail.com>

Installed-Size: 1025

Depends: perl:any, clamav (>= 0.95), clamav-freshclam (>= 0.95), libgtk3-perl, libtext-csv-perl, libwww-perl, libjson-perl, liblocale-gettext-perl, liblwp-protocol-https-perl, gnome-icon-theme, cron

Suggests: cabextract

Section: utils

Priority: optional

Homepage: https://gitlab.com/dave_m/clamtk/wikis/Home

Description: graphical front-end for ClamAV

ClamTk is an easy to use graphical front-end for Clam Antivirus.

Make the following change, delete "clamav-freshclam (>= 0.95),". Edited content:

Package: clamtk

Version: 6.18-1

Architecture: all

Maintainer: Dave M <dave.nerd@gmail.com>

Installed-Size: 1025

Depends: perl:any, clamav (>= 0.95), libgtk3-perl, libtext-csv-perl, libwww-perl, libjson-perl, liblocale-gettext-perl, liblwp-protocol-https-perl, gnome-icon-theme, cron

Suggests: cabextract

Section: utils

Priority: optional

Homepage: https://gitlab.com/dave_m/clamtk/wikis/Home

Description: graphical front-end for ClamAV

ClamTk is an easy to use graphical front-end for Clam Antivirus.

Save the edited "control" file. ClamTk will now not need "clamav-freshclam" any longer.

Delete the original "clamtk_6.18-1_all.deb"

Open a terminal in the directory where is found "/clamtk_6.18-1_all/". Enter the following code:

[code]

dpkg-deb -b clamtk_6.18-1_all

A new "clamtk_6.18-1_all.deb" with edited "control" file has been created.

2) Install ClamTk and ClamAV

Uninstall previous versions of ClamAV and ClamTk:

[code]

sudo apt-get autoremove clamav --purge

Install ClamTk dependencies:

[code]

sudo apt install libgtk3-perl libtext-csv-perl libwww-perl libjson-perl liblocale-gettext-perl liblwp-protocol-https-perl gnome-icon-theme cron

Open a terminal in the directory where you saved the two packages "clamtk_6.18-1_all.deb" and "clamav-1.4.3.linux.x86_64.deb" (or a more recent version); check that there is no other deb in this directory. Installing ClamTk and ClamAV:

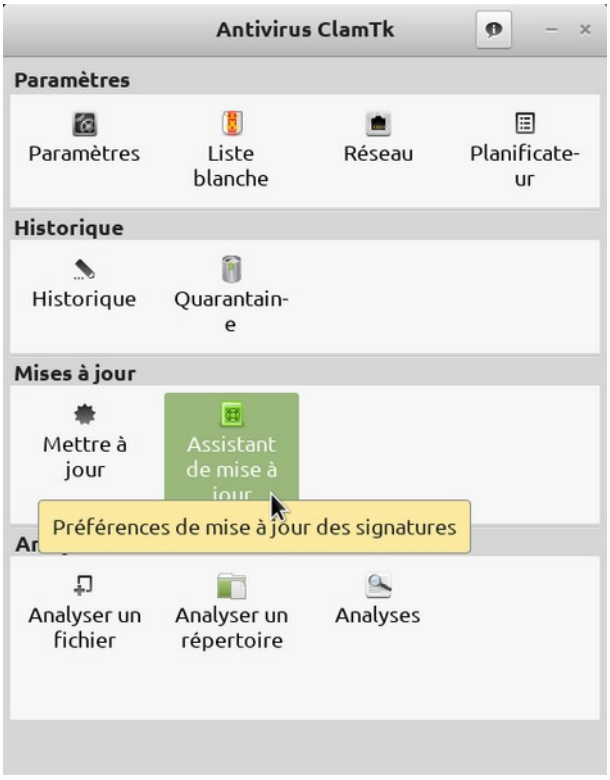
[code]

sudo dpkg -i *.deb

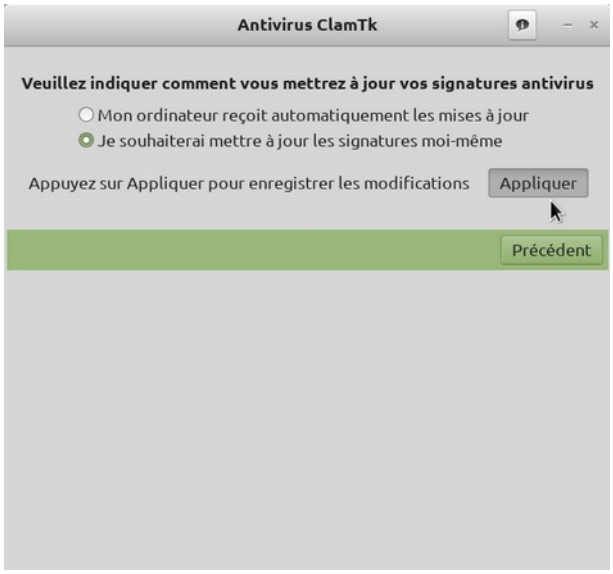
Installation is complete.

3) ClamTk signatures update settings

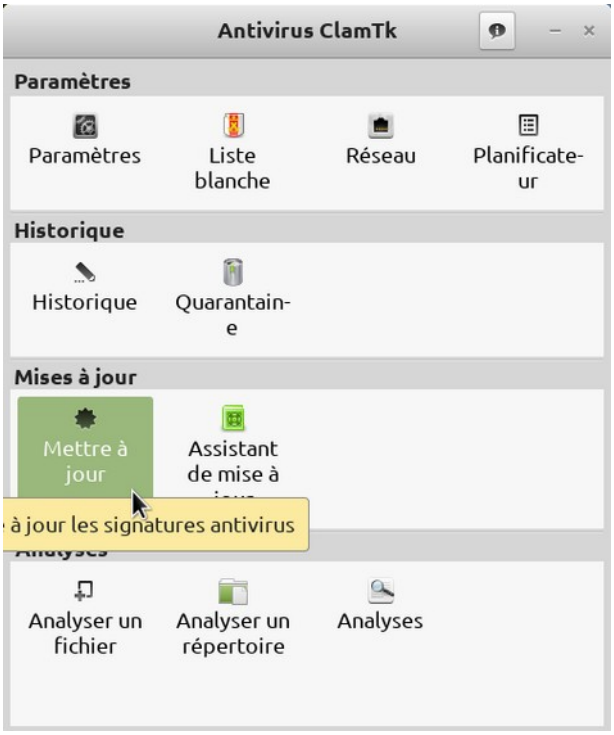
Launch ClamTk and click on "Update Assistant" (during installation, a shortcut has been created in your menu, in the "Accessories" category) :



Select "I would like to update signatures myself", then click on "Apply" and "Previous" buttons:



Now, update signatures:



Once done, click on "Previous" button. Quit ClamTk.

4) Improving ClamAV detection rate

As wee have seen, Malware and viruses detection, ClamAV detection rate, using official signatures only, has been between ~60% to ~75% on the period 2011-2020. The detection rate can be

improved by using extra unofficial signatures, though without reaching the detection rate of commercial software.

Content of "local.conf":

[code]

Local config

DatabaseMirror database.clamav.net

With this option you can provide custom sources for database files.

This option can be used multiple times. Support for:

http(s)://, ftp(s)://, or file://

#

####################################

Linux Malware Detect signatures

####################################

See https://www.rfxn.com/projects/linux-malware-detect/

DatabaseCustomURL http://www.rfxn.com/downloads/rfxn.hdb

DatabaseCustomURL http://www.rfxn.com/downloads/rfxn.ndb

DatabaseCustomURL http://www.rfxn.com/downloads/rfxn.yara

#

####################################

Sanesecurity hosted signatures

####################################

See https://sanesecurity.org/usage/signatures/

Warning, "http://mirror.seichter.de/sanesecurity/" is an unofficial mirror and might be disabled without notice.

If that occurred, try "https://mirror.ihost.md/clamav/sanesecurity/" or "https://mirror.rollernet.us/sanesecurity/".

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/badmacro.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/blurl.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/bofhland_cracked_URL.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/bofhland_malware_attach.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/bofhland_malware_URL.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/bofhland_phishing_URL.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_filename.cdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_generic.cdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_js.cdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_js.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/hackingteam.hsb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/junk.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/jurlbl.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/jurlbla.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/lott.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malware.expert.fp

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malware.expert.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malware.expert.ldb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malware.expert.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malwarehash.hsb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/MiscreantPunch099-Low.ldb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/phish.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/phishtank.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/porcupine.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/rogue.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/scam.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/shelter.ldb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/sigwhitelist.ign2

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/spamattach.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/spamimg.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/spear.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/spearl.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/sanesecurity.ftm

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow.attachments.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_bad_cw.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow.complex.patterns.ldb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_extended_malware.hdb

DatabaseCustomURL
http://mirror.seichter.de/sanesecurity/winnow_extended_malware_links.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_malware.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_malware_links.ndb

DatabaseCustomURL
http://mirror.seichter.de/sanesecurity/winnow_phish_complete_url.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_spam_complete.ndb

#

High False Positive risk, disabled by default, uncomment to enable

#DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_all.cdb

#DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_all.ndb

#DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_mail.cdb

#

##############

Ditekshen

##############

see https://github.com/ditekshen/detection/

DatabaseCustomURL
https://raw.githubusercontent.com/ditekshen/detection/master/clamav/clamav.ldb

DatabaseCustomURL
https://raw.githubusercontent.com/ditekshen/detection/master/clamav/indicator_rmm.ldb

#

##############

Twinclams

##############

see https://github.com/splunk/twinclams

DatabaseCustomURL
https://raw.githubusercontent.com/splunk/twinclams/master/twinclams.ldb

#

################

Securiteinfo

################

See http://www.securiteinfo.com/services/clamav_unofficial_malwares_signatures.shtml

Create an account, select basic (free) or pay account, you will get your personal signatures links; copy them in this file and uncomment the corresponding lines

#DatabaseCustomURL https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.ign2

#DatabaseCustomURL https://www.securiteinfo.com/get/signatures/xxxxxx/javascript.ndb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfohtml.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfoascii.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfopdf.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfoandroid.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/spam_marketing.ndb

#

Very large database of old malware, older than one year, disabled by default, uncomment to enable

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfoold.hdb

#

Below this line, the downloads will not work if you have free account; uncomment if you have pay account

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo0hour.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.mdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.yara

#DatabaseCustomURL https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.pdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.wdb

#

###############

Interserver

###############

DatabaseCustomURL http://sigs.interserver.net/interserver256.hdb

DatabaseCustomURL http://sigs.interserver.net/shell.ldb

DatabaseCustomURL http://sigs.interserver.net/interservertopline.db

DatabaseCustomURL http://sigs.interserver.net/whitelist.fp

#

###########

Urlhaus

###########

DatabaseCustomURL https://urlhaus.abuse.ch/downloads/urlhaus.ndb

#

LogSyslog no

Copy those lines in a text editor.

Create an account, free or pay, at https://www.securiteinfo.com/clients/customers/account, and modify / uncomment the corresponding lines with your own download links.

Then save the file as "local.conf" in "~/.config/clamtk/db/" (where "~" is a system shortcut for "/home/**your_username**").

ClamAV detection rate is now improved. Note the presence of Linux Malware Detect signatures, with more than 11,061 signatures of malware that specifically target Linux.

Subscription of pay account at Secureinfo will increase signatures by some 4,000,000 more.

5) Security

- ClamAV is not fully installed; there are no such files as "clamd.conf" and "freshclam.com"; no service is installed and running. ClamTk calls for "freshclam" when signatures are updated, and for "clamscan" when files or directories are scanned.

- There is no file written on your operating system "/", not even logs.

→ Those two precautions decrease ClamAV attack surface.

- Security can be improved by running ClamTk and ClamAV in sandboxes, using Firejail.

Download the latest version of Firejail from https://sourceforge.net/projects/firejail/files/firejail/. At the time of writing this document, the latest version is "firejail_0.9.74_1_amd64.deb".

* Install Firejail with:

[code]

replace firejail deb name by the one you downloaded in the following command

sudo dpkg -i firejail_0.9.74_1_amd64.deb

* ClamTk executable "clamtk" is installed in "/usr/bin/". But ClamAV "clamscan" and "freshclam" are installed in "/usr/local/bin/", and this prevents linking them to Firejail. It is normally possible to link "clamtk" only to Firejail.

[code]

sudo ln -s /usr/bin/firejail /usr/local/bin/clamtk

* But a trick allows linking "clamscan" and "freshclam" to Firejail:

[code]

rename clamscan and freshclam

sudo mv /usr/local/bin/clamscan /usr/local/bin/clamscan1

sudo mv /usr/local/bin/freshclam /usr/local/bin/freshclam1

link clamscan1 and freshclam1 to /usr/bin/clamscan and /usr/bin/freshclam

sudo ln -s /usr/local/bin/clamscan1 /usr/bin/clamscan

sudo ln -s /usr/local/bin/freshclam1 /usr/bin/freshclam

finally link usr/bin/clamscan and /usr/bin/freshclam to firejail

sudo ln -s /usr/bin/firejail /usr/local/bin/clamscan

sudo ln -s /usr/bin/firejail /usr/local/bin/freshclam

From now, "clamtk", "clamscan" and "freshclam" will always be executed in Firejail sandboxes, and security is greatly improved.

* The default "/etc/firejail/clamtk.profile" needs to be edited, when signatures are downloaded manually. Create a "~/.config/firejail/" directory in your home. Copy "clamtk.profile" to this directory:

[code]

cp /etc/firejail/clamtk.profile ~/.config/firejail/clamtk.profile

Open "~/.config/firejail/clamtk.profile" in a text editor:

[code]

Firejail profile for clamtk

Description: Easy to use, light-weight, on-demand virus scanner for Linux systems

This file is overwritten after every install/update

Persistent local customizations

include clamtk.local

Persistent global definitions

include globals.local

include disable-exec.inc

Add the below lines to your clamtk.local if you update signatures databases per-user:

ignore net none

netfilter

protocol inet,inet6

caps.drop all

ipc-namespace

net none

no3d

nodvd

nogroups breaks scanning

#nogroups

noinput

nonewprivs

noroot breaks scanning

#noroot

nosound

notv

nou2f

novideo

protocol unix

seccomp

private-dev

dbus-user filter

dbus-user.talk ca.desrt.dconf

dbus-user.talk org.gtk.vfs.UDisks2VolumeMonitor

dbus-system none

restrict-namespaces

Just uncomment the following three lines, after "# Add the below lines to":

ignore net none

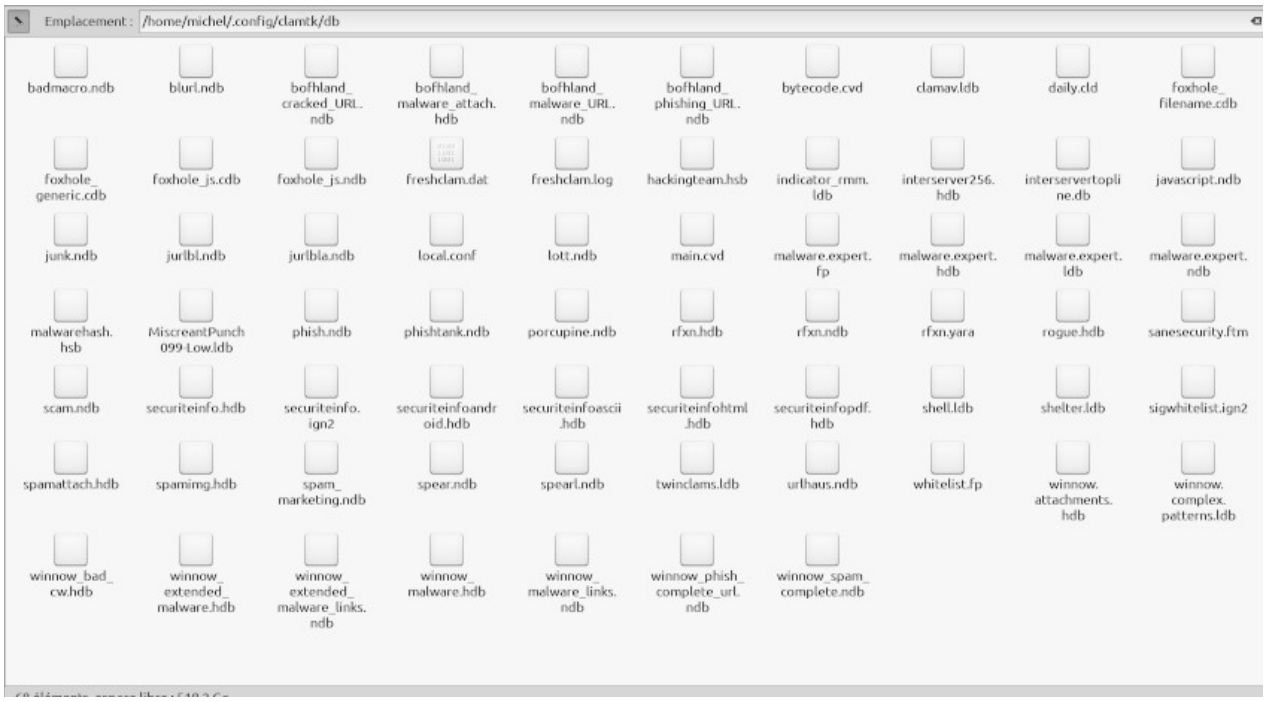
netfilter

protocol inet,inet6

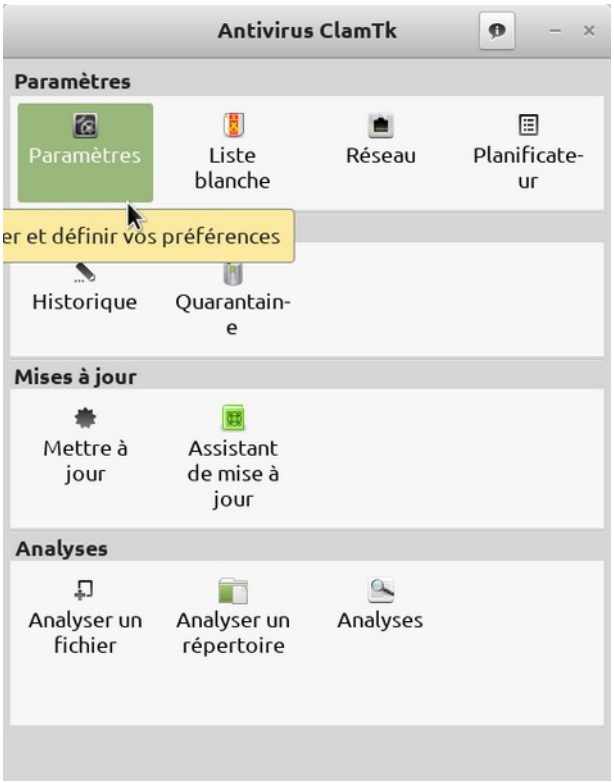
and save "~/.config/firejail/clamtk.profile".

6) Use

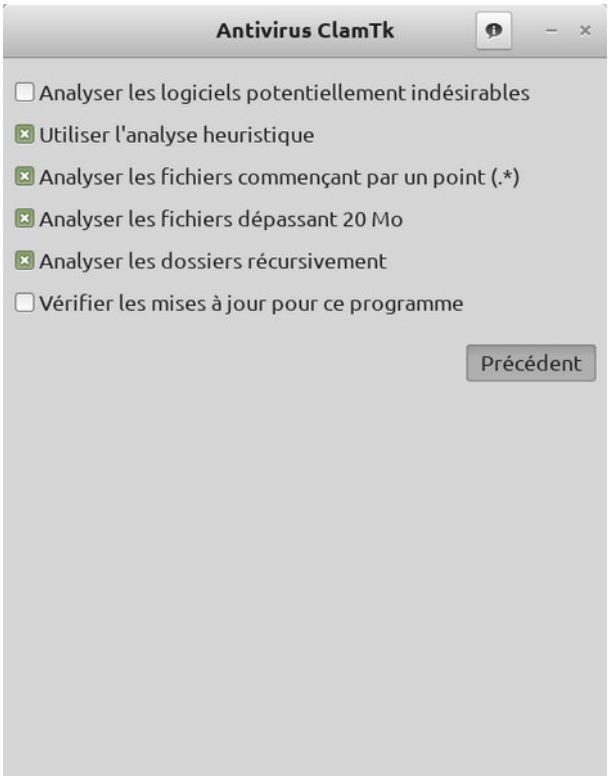
You can now launch ClamTK and update the signatures, you will receive all the updated signatures, official and unofficial ones, as mentioned in your "~/.config/clamtk/db/local.conf". Here is a screen capture of my "~/.config/clamtk/db/" directory:



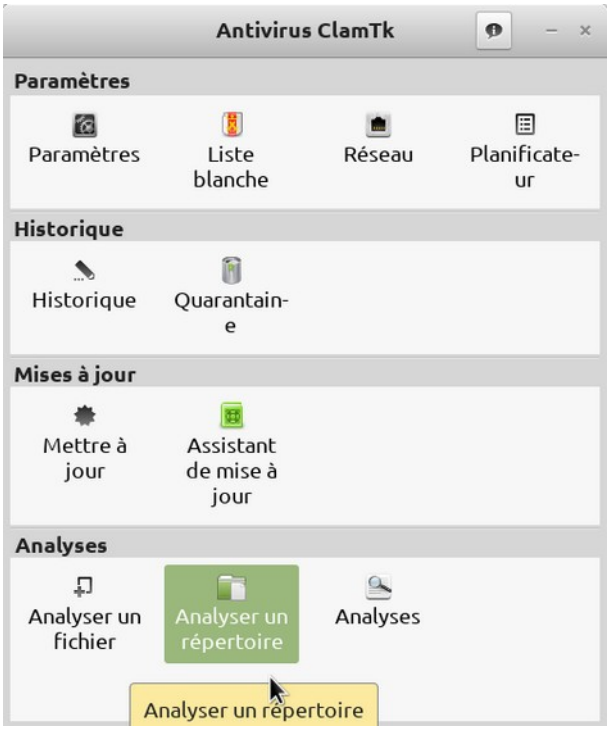
Set ClamTk parameters:

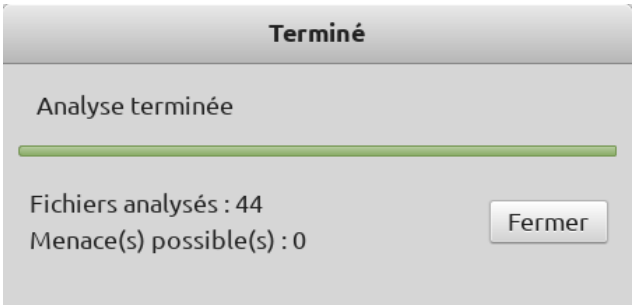


Select the scan parameters you want and click on "Previous" button:



You can now scan a file, or a directory:



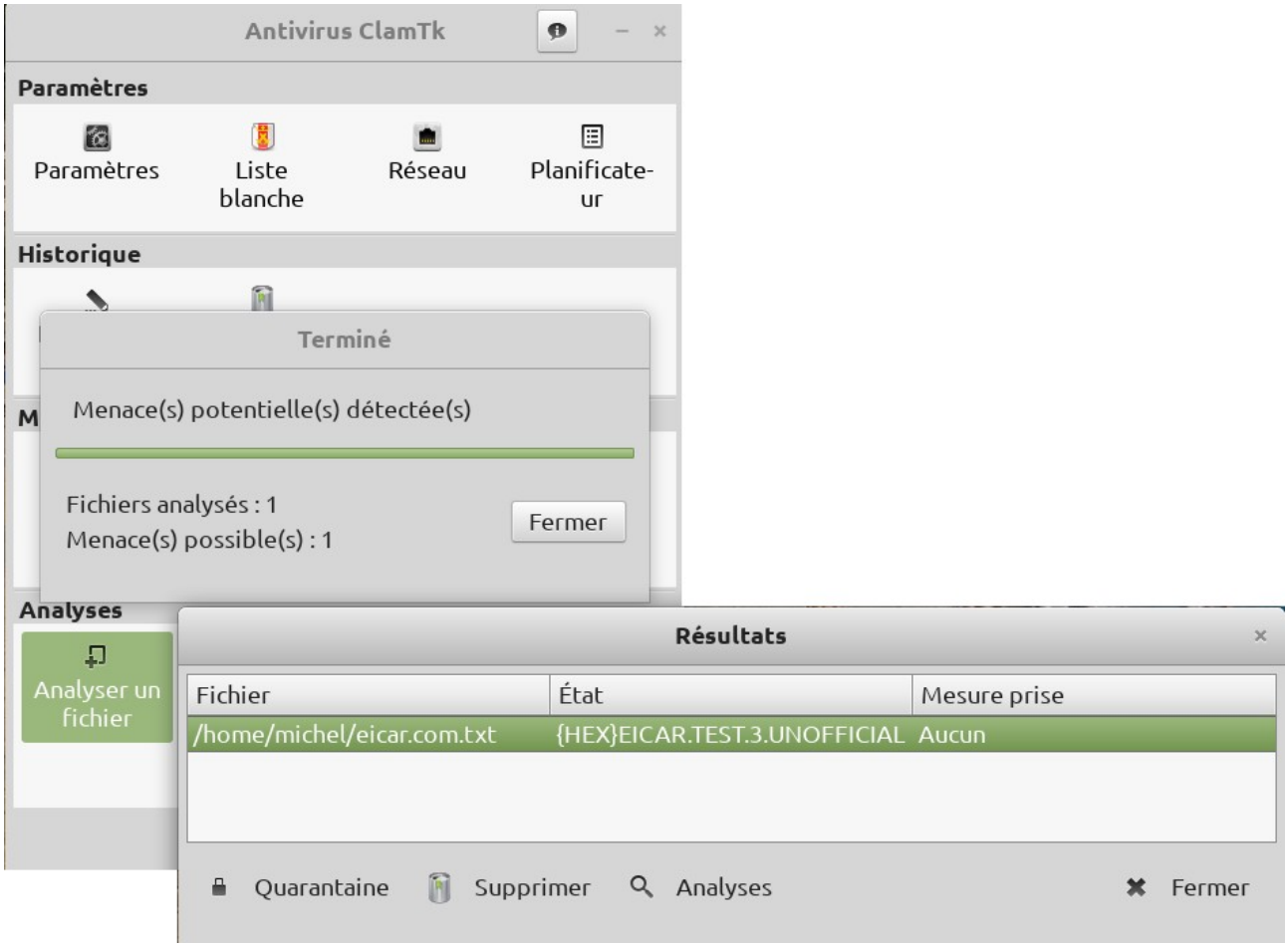


You can test your installation. Download "eicar.com.txt", a benign text file used to test virus scanners:

[code]

wget https://secure.eicar.org/eicar.com.txt

Once done, scan "eicar.com.txt". If the result you get conforms to what follows, your ClamTk + ClamAV installation is working:



7) Updating ClamAV

When a new stable version of ClamAV is available, you can follow this procedure to install it.

Before ClamAV new deb installation:

[code]

removing clamav links in "/usr/bin/"

sudo rm -f /usr/bin/clamscan

sudo rm -f /usr/bin/freshclam

removing clamav links in "/usr/local/bin/"

sudo rm -f /usr/local/bin/clamscan

sudo rm -f /usr/local/bin/freshclam

renaming clamscan1 and freshclam1

sudo mv /usr/local/bin/clamscan1 /usr/local/bin/clamscan

sudo mv /usr/local/bin/freshclam1 /usr/local/bin/freshclam

You can now install the new ClamAV deb package.

Once done:

[code]

```
# rename clamscan and freshclam
sudo mv /usr/local/bin/clamscan /usr/local/bin/clamscan1
sudo mv /usr/local/bin/freshclam /usr/local/bin/freshclam1
# link clamscan1 and freshclam1 to /usr/bin/clamscan and /usr/bin/freshclam
sudo ln -s /usr/local/bin/clamscan1 /usr/bin/clamscan
sudo ln -s /usr/local/bin/freshclam1 /usr/bin/freshclam
# finally link usr/bin/clamscan and /usr/bin/freshclam to firejail
sudo ln -s /usr/bin/firejail /usr/local/bin/clamscan
sudo ln -s /usr/bin/firejail /usr/local/bin/freshclam
```

8) Uninstalling

Follow this uninstall procedure:

[code]

```
# removing links in "/usr/bin/"
sudo rm -f /usr/bin/clamtk
sudo rm -f /usr/bin/clamscan
sudo rm -f /usr/bin/freshclam
# removing links in "/usr/local/bin/"
sudo rm -f /usr/local/bin/clamtk
sudo rm -f /usr/local/bin/clamscan
sudo rm -f /usr/local/bin/freshclam
# renaming clamscan1 and freshclam1
sudo mv /usr/local/bin/clamscan1 /usr/local/bin/clamscan
sudo mv /usr/local/bin/freshclam1 /usr/local/bin/freshclam
# uninstalling clamav and clamtk
sudo apt-get autoremove clamav --purge
# removing clamtk firejail profile
rm -f ~/.config/firejail/clamtk.profile
# removing clamtk configuration
rm -rf ~/.config/clamtk
```

B – Use of Kapitano + ClamAV flatpak

Kapitano is a recent GUI for ClamAV, with similar functions as ClamTk. The flatpak includes Kapitano and ClamAV executables. You can use it instead of ClamTk + ClamAV.

Pros:

- Flatpak offers the maximum security, and unattended write can be controlled with Flatseal (see Annex 6: Flatpak Tutorial).

- Easier to install and update then ClamTk + ClamAV

Cons:

- At the moment the GUI, Kapitano itself, and the flatpak are maintained, and ClamAV executables are updated; but will the author be able to do this for long?

- You need to accept the humorous, childish GUI: you are the captain of a ship (your computer) fighting against pirates (malware and viruses), using your radar (virus scanner) to detect them…

- If it is your only flatpak, you will have to download a heavyweight runtime.

1) Installation

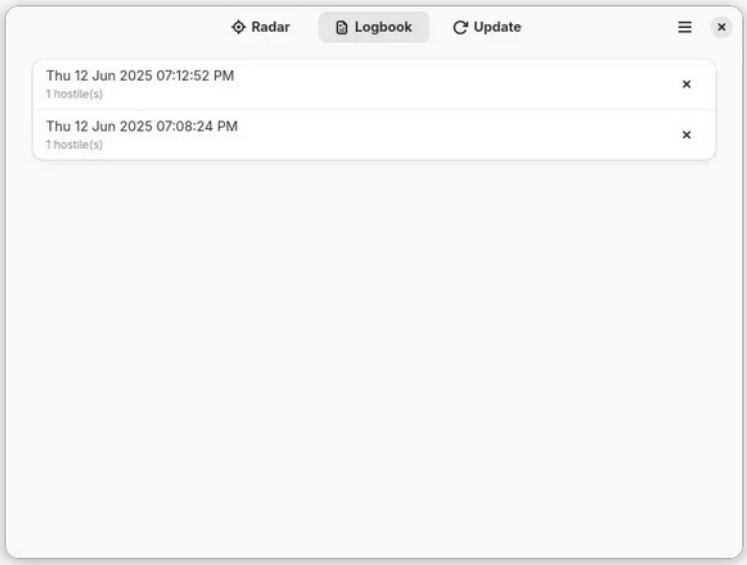
[code]

flatpak install page.codeberg.zynequ.Kapitano

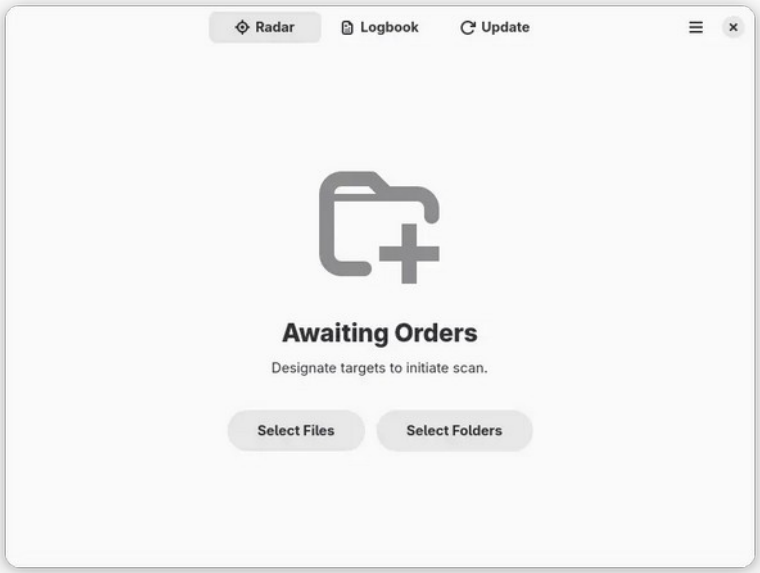
2) Some screenshots

(From https://flathub.org/apps/page.codeberg.zynequ.Kapitano)

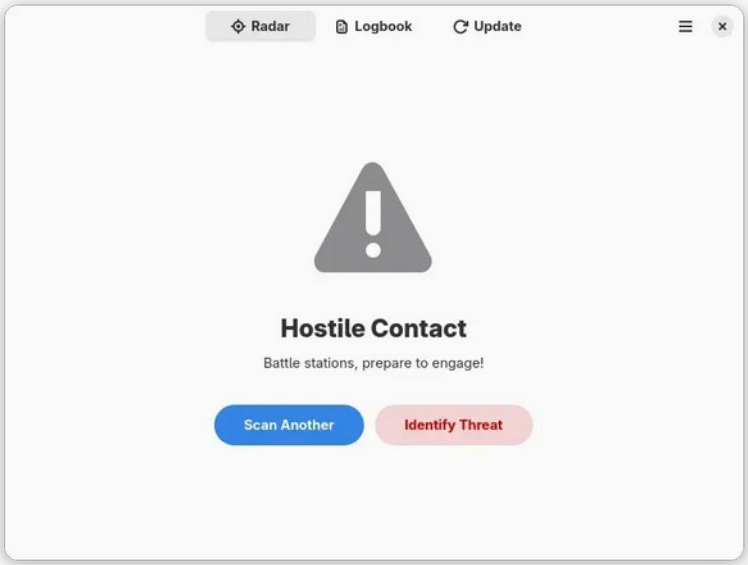
View scan reports:



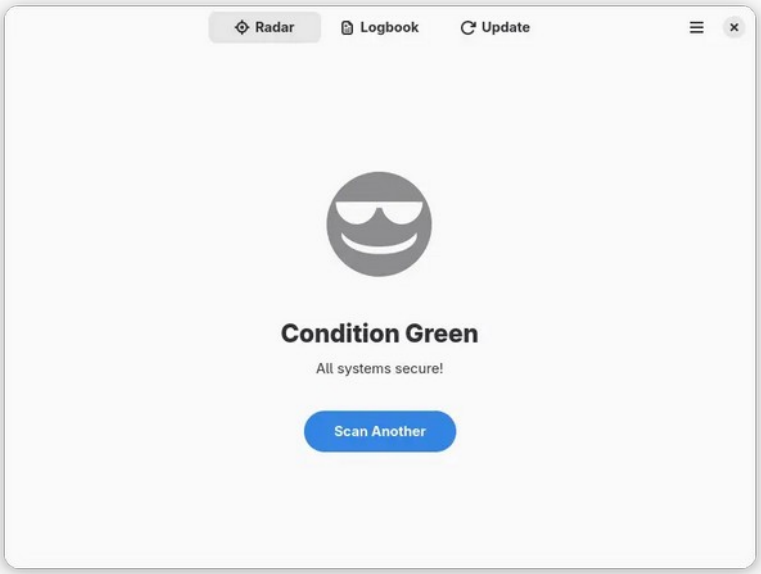
Scan files and folders:



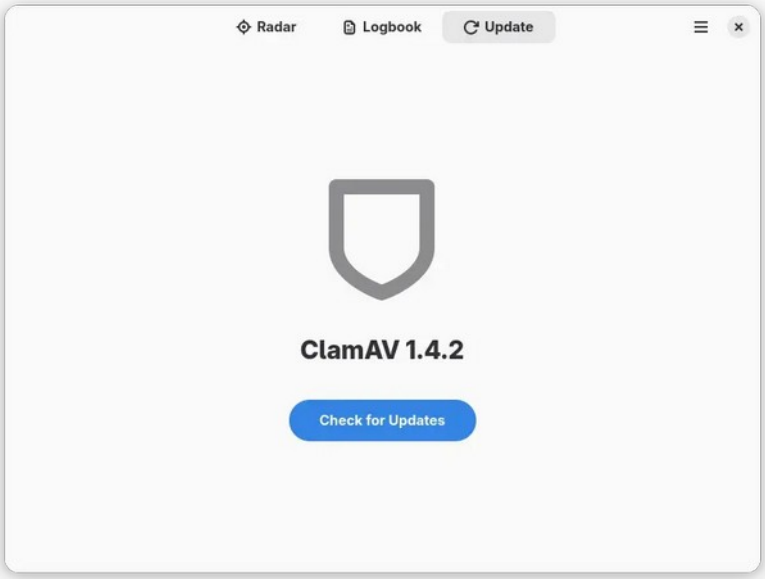
Review suspicious finds:



No dangers spotted:



Update malware definitions:



3) Improving ClamAV detection rate

As wee have seen, Malware and viruses detection, ClamAV detection rate, using official signatures only, has been between ~60% to ~75% on the period 2011-2020. The detection rate can be improved by using extra unofficial signatures, though without reaching the detection rate of commercial software.

Content of "freshclam.conf":

[code]

DatabaseMirror database.clamav.net

####################################

Linux Malware Detect signatures

####################################

See https://www.rfxn.com/projects/linux-malware-detect/

DatabaseCustomURL http://www.rfxn.com/downloads/rfxn.hdb

DatabaseCustomURL http://www.rfxn.com/downloads/rfxn.ndb

DatabaseCustomURL http://www.rfxn.com/downloads/rfxn.yara

#

####################################

Sanesecurity hosted signatures

####################################

See https://sanesecurity.org/usage/signatures/

Warning, "http://mirror.seichter.de/sanesecurity/" is an unofficial mirror and might be disabled without notice.

If that occurred, try "https://mirror.ihost.md/clamav/sanesecurity/" or "https://mirror.rollernet.us/sanesecurity/".

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/badmacro.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/blurl.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/bofhland_cracked_URL.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/bofhland_malware_attach.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/bofhland_malware_URL.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/bofhland_phishing_URL.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_filename.cdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_generic.cdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_js.cdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_js.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/hackingteam.hsb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/junk.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/jurlbl.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/jurlbla.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/lott.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malware.expert.fp

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malware.expert.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malware.expert.ldb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malware.expert.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/malwarehash.hsb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/MiscreantPunch099-Low.ldb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/phish.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/phishtank.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/porcupine.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/rogue.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/scam.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/shelter.ldb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/sigwhitelist.ign2

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/spamattach.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/spamimg.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/spear.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/spearl.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/sanesecurity.ftm

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow.attachments.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_bad_cw.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow.complex.patterns.ldb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_extended_malware.hdb

DatabaseCustomURL
http://mirror.seichter.de/sanesecurity/winnow_extended_malware_links.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_malware.hdb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_malware_links.ndb

DatabaseCustomURL
http://mirror.seichter.de/sanesecurity/winnow_phish_complete_url.ndb

DatabaseCustomURL http://mirror.seichter.de/sanesecurity/winnow_spam_complete.ndb

#

High False Positive risk, disabled by default, uncomment to enable

#DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_all.cdb

#DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_all.ndb

#DatabaseCustomURL http://mirror.seichter.de/sanesecurity/foxhole_mail.cdb

#

##############

Ditekshen

##############

see https://github.com/ditekshen/detection/

DatabaseCustomURL
https://raw.githubusercontent.com/ditekshen/detection/master/clamav/clamav.ldb

DatabaseCustomURL
https://raw.githubusercontent.com/ditekshen/detection/master/clamav/indicator_rmm.ldb

#

##############

Twinclams

##############

see https://github.com/splunk/twinclams

DatabaseCustomURL
https://raw.githubusercontent.com/splunk/twinclams/master/twinclams.ldb

#

################

Securiteinfo

################

See http://www.securiteinfo.com/services/clamav_unofficial_malwares_signatures.shtml

Create an account, select basic (free) or pay account, you will get your personal signatures links; copy them in this file and uncomment the corresponding lines

#DatabaseCustomURL https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.ign2

#DatabaseCustomURL https://www.securiteinfo.com/get/signatures/xxxxxx/javascript.ndb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfohtml.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfoascii.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfopdf.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfoandroid.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/spam_marketing.ndb

#

Very large database of old malware, older than one year, disabled by default, uncomment to enable

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfoold.hdb

#

Below this line, the downloads will not work if you have free account; uncomment if you have pay account

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo0hour.hdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.mdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.yara

#DatabaseCustomURL https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.pdb

#DatabaseCustomURL
https://www.securiteinfo.com/get/signatures/xxxxxx/securiteinfo.wdb

#

################

Interserver

################

DatabaseCustomURL http://sigs.interserver.net/interserver256.hdb

DatabaseCustomURL http://sigs.interserver.net/shell.ldb

DatabaseCustomURL http://sigs.interserver.net/interservertopline.db

DatabaseCustomURL http://sigs.interserver.net/whitelist.fp

#

###########

Urlhaus

###########

DatabaseCustomURL https://urlhaus.abuse.ch/downloads/urlhaus.ndb

#

DatabaseDirectory ~/.var/app/page.codeberg.zynequ.Kapitano/data/clamav

Copy those lines in a text editor.

Create an account, free or pay, at https://www.securiteinfo.com/clients/customers/account, and modify / uncomment the corresponding lines with your own download links.

Then save the file as "freshclam.conf" in "~/.var/app/page.codeberg.zynequ.Kapitano/data/" (where "~" is a system shortcut for "/home/**your_username**").

ClamAV detection rate is now improved. Note the presence of Linux Malware Detect signatures, with more than 11,061 signatures of malware that specifically target Linux.

Subscription of pay account at Secureinfo will increase signatures by some 4,000,000 more.

4) Uninstalling

[code]

uninstalling kapitano and clamav

flatpak uninstall page.codeberg.zynequ.Kapitano

uninstalling unused flatpak runtimes

flatpak uninstall –unused

uninstalling kapitano and clamav data

rm -rf ~/.var/app/page.codeberg.zynequ.Kapitano/

C – Use of Raspirus virus scanner

Raspirus is a small flatpak virus scanner using Yara rules. It can be used as a complement to ClamTk + ClamAV, or to Kapitano + ClamAV flatpak.

Like ClamTk + ClamAV, it is an on-demand scan program, and its signatures need to be manually updated by the user. Flatpak sandboxing guarantees its security.

1) Installation

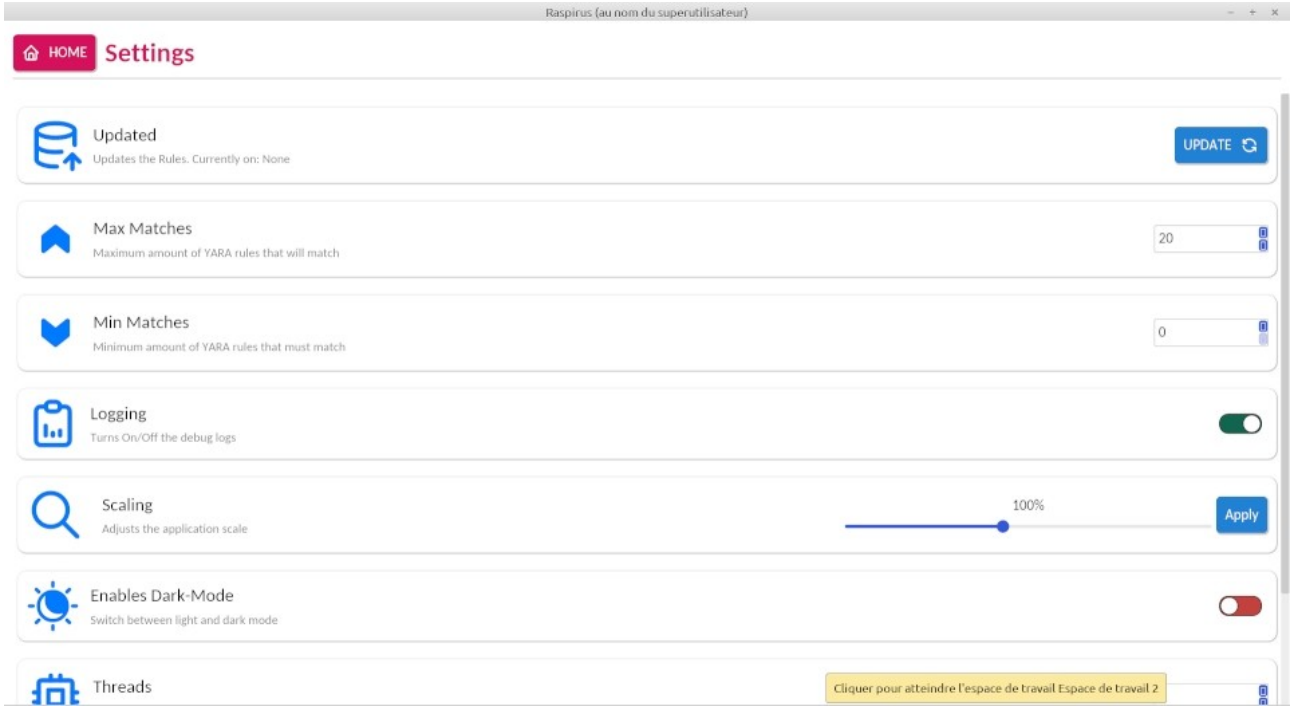
[code]

flatpak install io.github.raspirus.raspirus

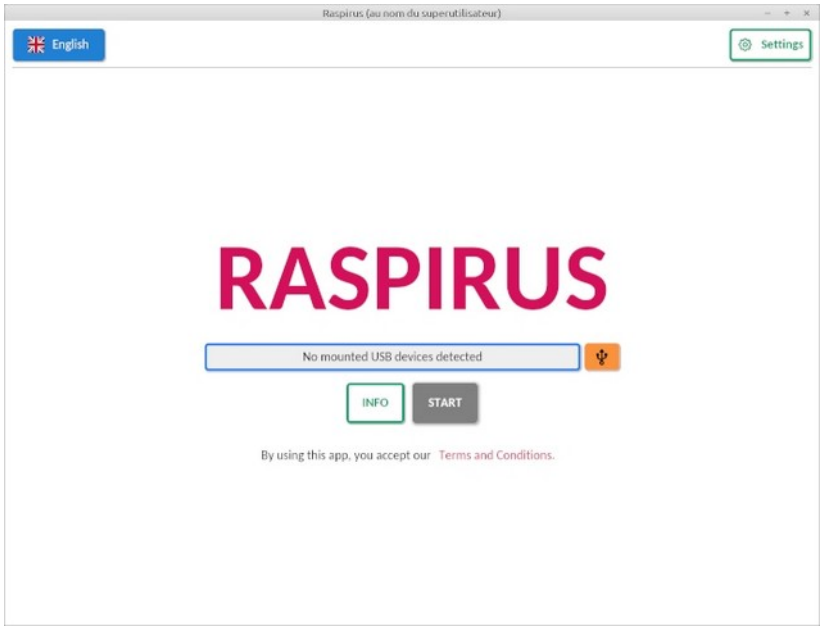
2) Settings

Launch Raspirus (during installation, a shortcut has been created in your menu, in the "System Tools" category).

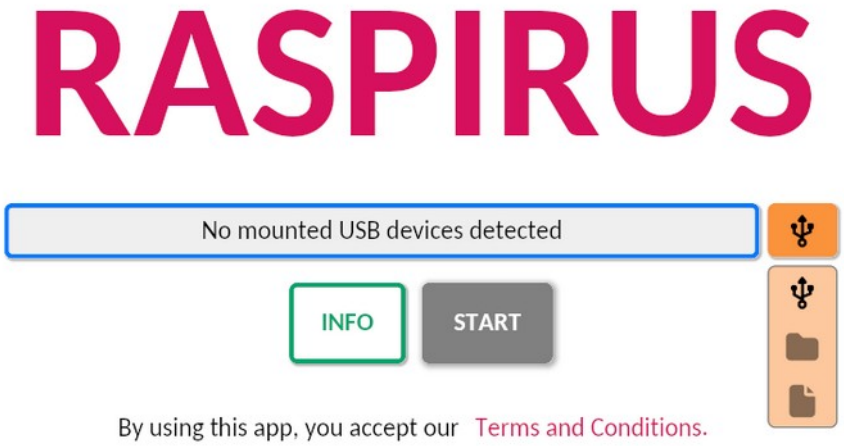
At the first launch, Raspirus proposes to be updated, accept. The following window opens:



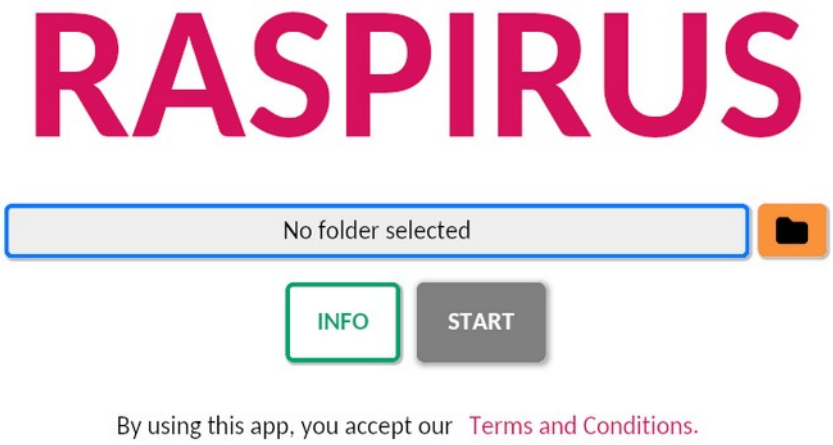
Click on the "UPDATE" button. Once updated, at your convenience turn off debug logging, enable dark mode and adjust scaling. Click on "HOME" and quit this window:



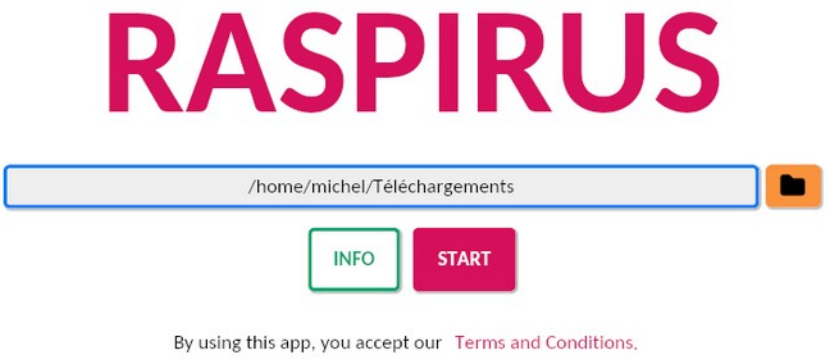
Click on the orange button, and select USB, Folder or File:



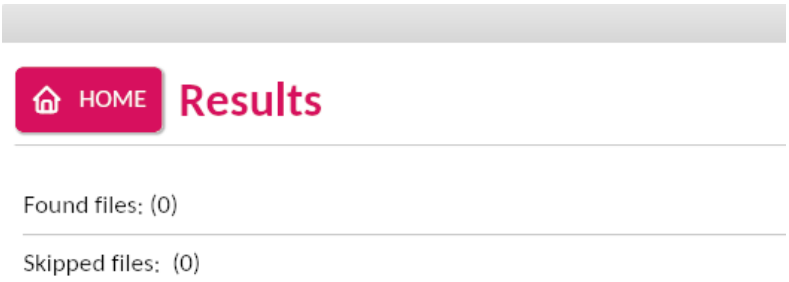
As an example, select Folder, and click on "No folder selected":



This will open a menu where you can select the folder to scan; once done, click on "START" button to start the scan.



In the next window you will get the scan result:



3) Uninstalling

Follow this procedure:

[code]

uninstalling raspirus

flatpak uninstall io.github.raspirus.raspirus

uninstalling unused flatpak runtimes

flatpak uninstall --unused

uninstalling raspirus data

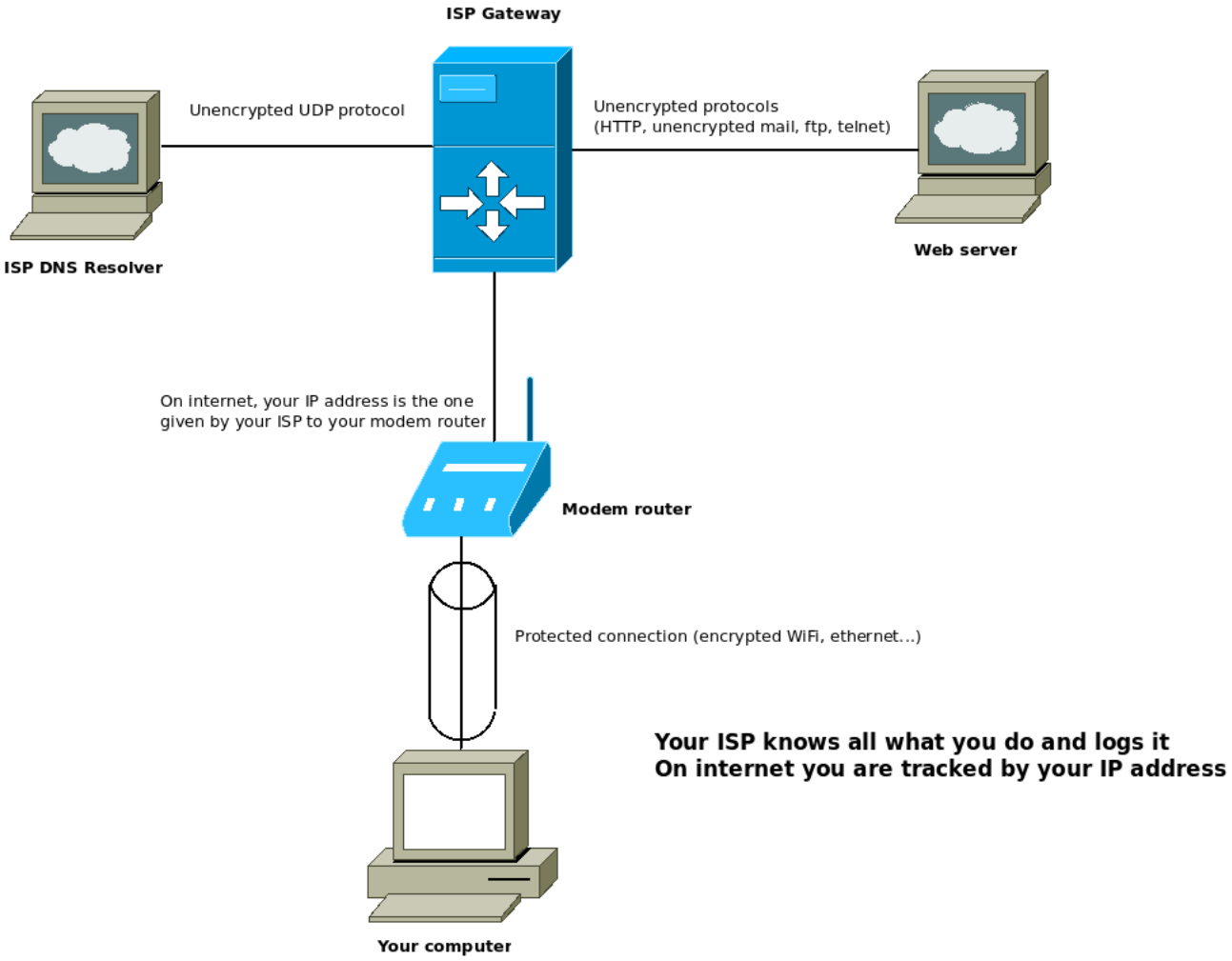
rm -rf ~/.var/app/io.github.raspirus.raspirus

Annex 13: Internet Connection Diagrams

These internet connection diagrams show different ways to connect to internet, with increasing levels of privacy and anonymity.

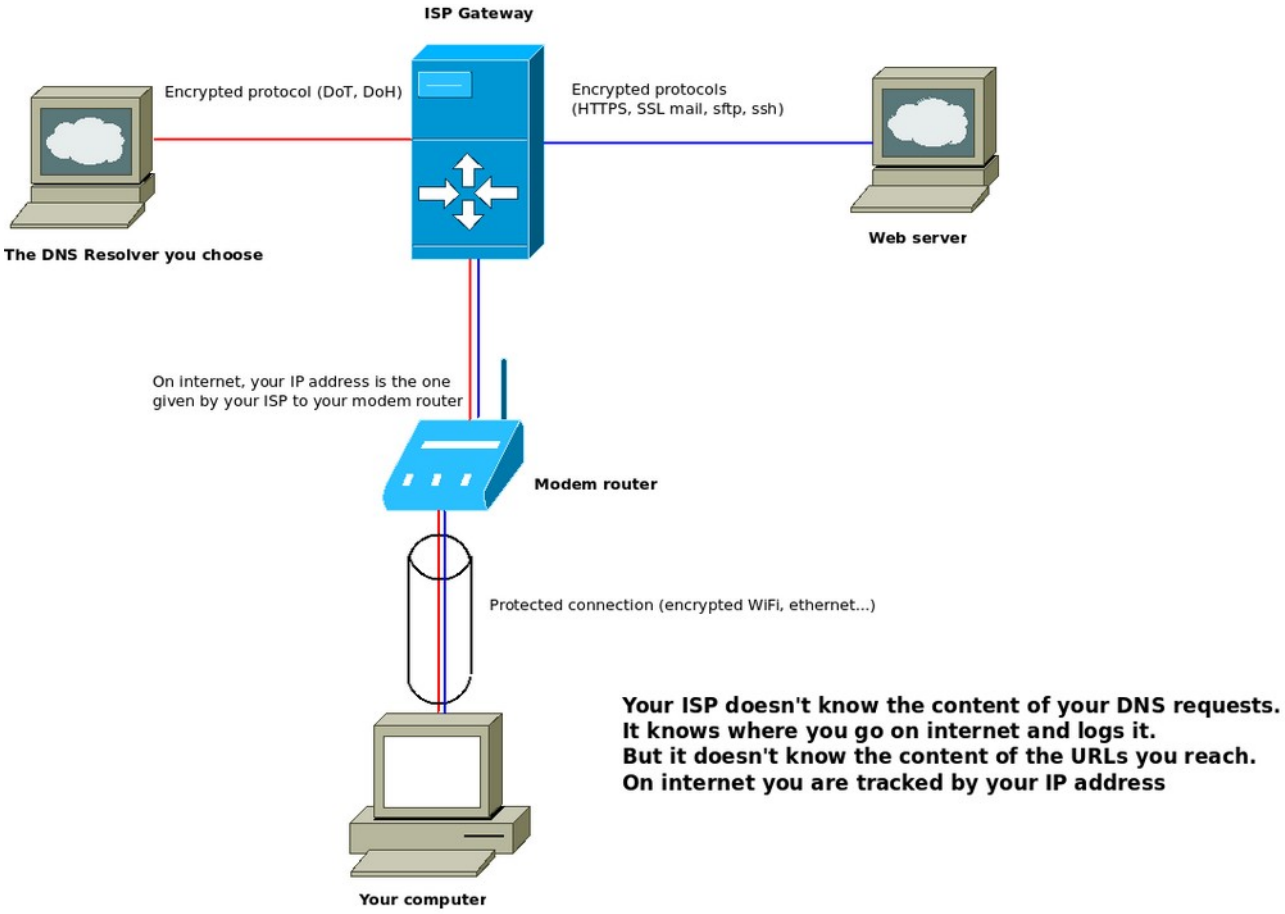
1) No privacy, no anonymity

Use of your ISP DNS resolver, unencrypted protocols.



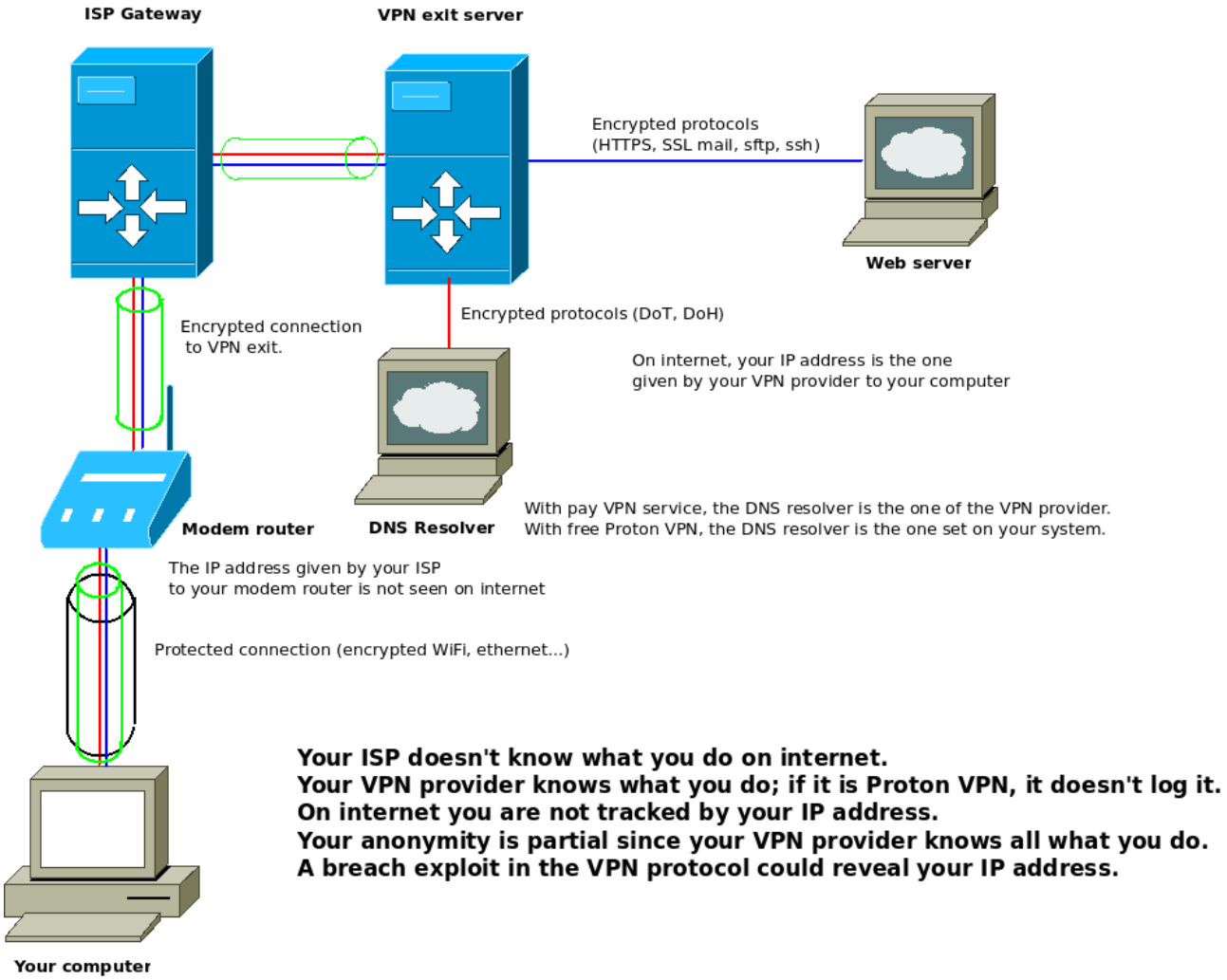
2) Privacy, no anonymity

Use of independent DNS resolver, encrypted protocols.



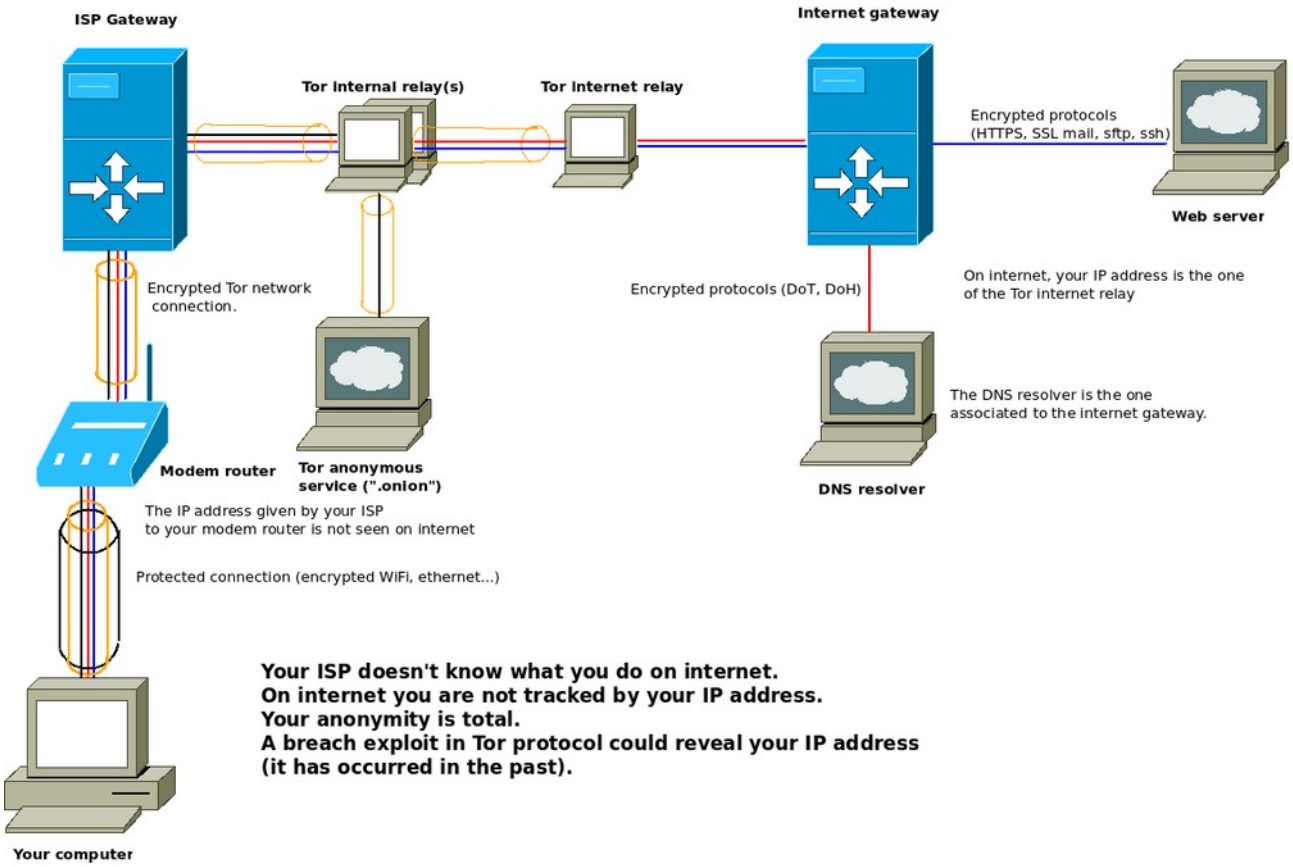
3) Partial anonymity

Use of a VPN.



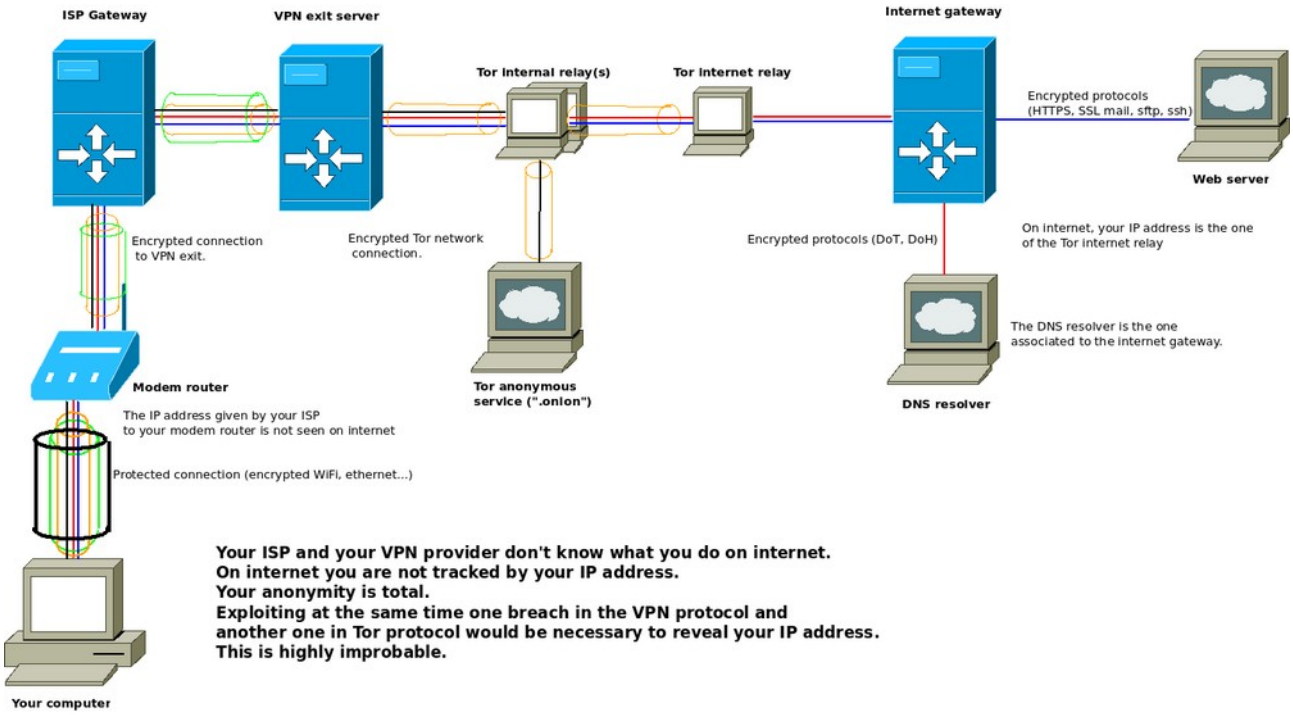
4) Total anonymity

Use of Tor network.



5) Total anonymity, no IP address leak risk

Use of a VPN and of Tor network.



Threats / Prevention Means correspondence matrix

Security threats

	LSA1	LSA2	LSA3	LSA4	DSA1	DSA2	DSA3	DSA4	DSA5	VST1	VST2
4.1)	X	X	X	X							
4.2)	X	X			X	X	X	X	X		
4.3)	X	X			X	X	X	X			
4.4)	X	X			X	X	X	X			
4.5)					X	X					
4.6)						X	X	X			
4.7)					X		X	X			
4.8)								X			
4.9)										X	
4.10)											X
4.11)											
4.12)											
4.13)											
4.14)											
4.15)											
4.16)											
4.17)											
4.18)									X		

Privacy and anonymity threats

	PT1	PT2	PT3	PT4	PT5	PT6	PT7	PT8	PT9	PT10	AT1
4.1)											
4.2)											
4.3)											
4.4)											
4.5)											
4.6)											
4.7)		X								X	
4.8)											
4.9)											
4.10)											
4.11)	X										
4.12)			X		X	X					
4.13)				X							
4.14)							X				
4.15)								X			
4.16)									X	X	
4.17)											X
4.18)											

Table of Contents