



hackLOG

IT Security & Ethical Hacking Handbook

VOLUME 1 ANONYMITY





hackLOG

IT Security & Ethical Hacking Handbook

Volume 1 Anonymity

Stefano Novelli

Warning

Violation of computers or networks of others is a criminal offense punishable by the law. Some of the procedures hereby outlined are only for educational/explanatory/informational purpose and only executed on devices under our possession or within controlled test environments, therefore you hold harmless the authors of this document for what you'll learn during this course and against any verifiable consequence.

Notes on this work

The content of Hacklog: Volume 1 is issued free of charge for the whole net, and is available in different formats, according to the Ethical Hacking self-regulation and respecting the different cultures practicing it.

You're free to use parts of this document for any work, properly quoting the source (Hacklog by inforge.net) and, including a footnote link, when possible. Since this project required a high amount of time, if this document has been useful for third party projects, we think it should be shared, out of respect for its author, his coworkers and who believed in it. The original text was written in 2017, in Italy.

Copyright

The textual content and the images of Hacklog: Volume 1 ebook are released under *Creative Commons 4.0 license* – non-replicable, no derived works, commercialization. The owner of the rights for this document is Stefano Novelli, and its distribution is by inforge.net.

For my friends, my loved ones,
and all who made this possible.

For all the hackers,
or aspiring hackers, worldwide.

Stefano Novelli

GLOSSARY

Translator's Foreword

Foreword

Anonymity

1. Operative System

1.1 Which distro?

1.1.1 Virtual Machines

1.1.2 Live Distros

1.1.3 The Terminal

2. Data Traces

2.1 MAC Address

2.1.1 Identifying the MAC Address

2.1.2 MAC Spoofing

2.2 Hostname

2.2.1 Changing the Hostname

2.3 Domain Name System

2.3.1 Choosing DNS

2.3.2 Changing DNS

2.3.3 Cache DNS

2.4 IP Address

2.4.1 Determining the IP in use

2.4.2 Proxy

2.4.2.1 Proxy types

2.4.2.2 Where you can find Proxies

2.4.2.3 How to use Proxies

2.4.2.4 How safe are Proxies?

3. Secure communications

3.1 VPN (Virtual Private Network)

3.1.1 VPN Types

3.1.1.1 PPTP, for the speed seekers

3.1.1.2 L2TP/IPsec, for the security and responsiveness enthusiasts

3.1.1.3 OpenVPN, for top security users

3.1.1.4 SSTP, for Windows users

3.1.2 Which VPN?

3.1.3 How to choose a VPN

3.1.3.1 Avoid Free VPNs

3.1.3.2 No Logs Policy

3.1.3.3 If they haven't got your data, they can't catch you

3.1.3.4 International Data Retention Laws

3.1.3.5 Payment Methods

3.1.3.6 DMCA Notices

3.1.4 VPN List

3.1.4.1. Multi Hop (cascading) VPNs

3.1.5 Using the VPN

3.1.6 Testing the quality of a VPN

3.1.6.1 Torrent Test

3.1.6.2 DNS Leak Test

3.1.6.3 Kill Switch (protection against disconnections)

4. Clearnet and Deep Web

4.1 TOR

4.1.1 What's the TOR network

4.1.2 TOR Projects

4.1.3 TOR installation

4.1.4 TOR use cases

4.1.4.1 TOR as a Browser

4.1.4.2 TOR as a P2P

4.1.4.3 TOR as Chat

4.1.4.4 TOR as a Proxy Software

4.1.5 TOR Relay

4.1.6 TOR Bridges

4.1.6.1 Bridges advanced use

4.1.7 Pluggable Transports

4.1.7.1 MEEK & Scramblesuit Protocols

4.1.8 Testing the quality of TOR

4.1.8.1 TOR Test via Browser

4.1.9 TOR and Deep Web

4.1.9.1 Where to find .onion sites?

4.1.10 Is the TOR network really safe??

4.1.10.1 TOR and HTTP protocol

4.1.10.2 TOR and compromised exit-nodes

4.1.10.3 TOR Browser and the issues with “pre-built” products

4.1.10.4 TOR, Google & CO.

4.1.10.5 TOR is not idiot-proof

4.2 I2P

4.2.1 Using I2P

4.2.1.1 Installing I2P

4.2.1.2 First launch of I2P

4.2.1.3 Configuring a Browser with I2P

4.2.1.4 I2P useful resources

4.2.1.5 Anonymous navigation in Clearnet

4.2.1.6 Where to find I2P sites?

4.2.1.7 Difficulties with I2P

4.3 Freenet

4.3.1 Freenet installation

4.3.2 Configuring Freenet

4.3.3 Using Freenet

4.3.4 Freenet useful resource

4.3.5 Security in Freenet

5. Combo Network

5.1 TOR via VPN

5.1.1 How to perform TOR via VPN

5.2 VPN via TOR

5.2.1 How to perform VPN via TOR

5.3 TOR over TOR

5.3.1 Tortilla

5.3.2 Is TOR over TOR helpful?

6. Local Resources

6.1 Private browsing

6.1.1 How to enable the Private or Incognito mode

6.1.2 What the Private/Incognito mode does (and doesn't do)

6.2 HTTPS

6.2.1 Controlling HTTPS protocols

6.3 Cookies

6.3.1 Cookies impact over security

6.3.2 Controlling cookies

6.4 “Special” Cookies

6.4.1 “Special” Cookies impact over security

6.4.2 How to block Flash Cookies

6.4.3 How to block DOM Storage

6.5 Javascript

6.5.1 JavaScript impact over security

6.5.2 Controlling JavaScript

6.6 Flash

6.6.1 Flash impact over security

6.6.2 Controlling Flash

6.7 Java

6.7.1 Java impact over security

6.7.2 Controlling Java

6.8 ActiveX

6.8.1 ActiveX impact over security

6.8.2 Controlling ActiveX

6.9 WebRTC

6.9.1 WebRTC impact over security

6.9.2 Controlling WebRTC

6.10 Browser Fingerprinting

6.10.1 Defining the Browser Fingerprinting

6.10.2 Defending yourself from Browser Fingerprinting

6.11 File Downloading

6.12 Browser Security Test

7. Data Security

7.1 Data Integrity

7.1.1 Checksum & Hash

7.1.1.1 Hash Types

7.1.1.2 Calculating a Checksum

7.1.1.3 Checksum in common use

7.2 Data Encryption

7.2.1 PGP, Pretty Good Privacy

7.2.2 GPG, GNU Privacy Guard

7.2.2.1 Understanding the public/private key

7.2.2.2 Creating your own PGP key

7.2.2.3 Importing, exporting and revoking a PGP/GPG key

7.2.2.4 PGP/GPG to encrypt and decrypt a file

7.2.2.5 PGP/GPG for data signature

7.2.2.6 PGP/GPG for data integrity

7.2.2.7 PGP/GPG for email encryption

7.2.3 Where to store the PGP/GPG keys

7.3 Disk Encryption

7.3.1 TrueCrypt

7.3.2 Veracrypt

7.3.2.1 Installing Veracrypt

7.3.2.2 Using Veracrypt

7.3.3 Zulucrypt, LUKS and family

7.4 Steganography

7.4.1 Steganography with LSB method

7.4.1.1 LSB Steganography Tools

7.4.1.2 Steghide

7.4.2 Cover Generation Steganography

7.4.2.1 Pure Steganography with SPAM method

7.4.2.2 Pure Steganography with PGP method

7.5 Data Backup

7.5.1 How many Backups do you need?

7.5.2 Rsync

7.5.2.1 Rsync installation

7.5.2.2 Local copy with Rsync

7.5.2.3 Remote copy with Rsync

7.6 Cold Boot RAM Extraction

7.6.1 How to perform CBRE

7.7 Metadata & EXIF Data

7.7.1 How to view the EXIF Data

7.7.1.1 MAT: Metadata Anonymisation Toolkit

7.7.1.2 Alternate software for Metadata

7.8 Camera sensors

7.9 Data Shredding

7.9.1 How to perform Data Shredding

7.9.1.1 Disk Cleaners

7.9.1.2 File Shredding

7.9.1.3 Physical Drive Destruction

8. Data Recovery

8.1 Post-Mortem Forensics

8.1.1 Which OS for P.M. Forensics?

8.1.2 Caine OS

8.1.2.1 TestDisk or PhotoRec, which one?

8.1.2.2 PhotoRec Mini Use Guide

9. Vulnerability

9.1 General Precautions

10. Enhanced OSs

10.1 Live OS

10.1.1 Tails OS

10.1.2 Live OS & Persistence: the risks

10.1.3 Live OS & Virtual Machines: the risks

10.2 Virtualized environments

10.2.1 Qubes OS

10.2.1.1 Virtualization logic

10.2.1.2 Network and Storage Domains

10.2.1.3 Why use Qubes and not Tails OS?

10.2.2 Qubes OS + Tais

10.2.3 Qubes OS + Whonix

10.2.4 Subgraph OS

10.2.4.1 Hardened like few others

10.2.4.2 Network and Anonymity

10.3 Pentest Distros

11. Online Identity

11.1 NEVER combine your identities

11.2 NEVER use the same data

11.3 Watch Out for your Habits

11.4 Disposable email

11.5 If you manage a Site/Blog/Forum

11.6 Things you should NEVER do

12. Online Payments

12.1 Buying in the Dark Net

12.1.1 Dark Net Markets

12.1.1.1 Types of Dark Net Markets

12.1.1.2 Where to find the Dark Net Markets?

12.2 Crypto-currencies

12.2.1 Precautions with Crypto-currencies

12.2.2 Bitcoin

12.2.2.1 How Bitcoins work

12.2.2.2 How to obtain Bitcoins

12.2.2.3 Making Bitcoins untraceable

12.2.3 Beyond Bitcoin

13. Be Free

Acknowledgments

Authors and Collaborators

Sources & Resources

Special Thanks

Donors

Translator's Foreword

Marco S. Doria is a professional translator and proofreader, working in the IT, Media and Marketing translation Industries since 2013. He loves computers, music, books, technology and, especially, his wife Laura and his daughter Penelope. He also wrote two short novellas in Italian. Contact: marcostefanodoria@gmail.com

I first came across the Hacklog Project by chance. I was talking with a colleague about how I wished to further explore the IT Security world, and he mentioned the Hacklog Volume 1, a very interesting handbook written by Stefano Novelli.

Since I was looking for new materials to improve as a Technical Translator, I immediately got my digital copy and started reading it.

I felt like captured! I couldn't stop reading... every chapter ran away so fast that I immediately felt to start over again.

Hacklog Volume 1 really opened my eyes about topics like Anonymity, Navigation Safety... Freedom! Yes, freedom! Because I learned how to use the Web more consciously; I learned how to be free from the control of big data companies dwelling on our personal information and habits.

I felt I had to contribute to this incredible, open project! So why not translate it?

Immediately, I mailed Stefano about this idea, and we started this adventure quite soon!

Being the son of one of the first IT Consultants in my area, I was close enough to the computer world to know the basic bits-and-bolts; therefore I really can say that translating Hacklog Volume 1 has been my best professional experience to date.

I really hope you enjoy it as much as I did working on the English version. I would like to thank Stefano for this incredible opportunity and Marco Silvestri,

who really helped me out reviewing the whole translation and adding true value to it.

Now, don't wait any further: enjoy your reading and... be free!

Marco Silvestri. Contact: marco.silvestri777@gmail.com

I already had the chance to work with Stefano as text reviewer for the Italian edition of Hacklog and while I was doing that I felt it was a good opportunity for me to learn something about the IT world I barely knew about. Internet security is extremely useful even if you don't work with computers and I think it's really important to have an idea of what happens every time you connect to the network and what lies underneath it.

When Stefano told me he wanted to publish an English version of the book I was really enthusiast cause I thought a lot of people could have enjoyed this book as much as I did.

I had the chance to help Marco, the translator that made this English version possible and Stefano Novelli, the mind behind the project, and I would like to thank them both for giving me the opportunity to help with the book.

Foreword

Welcome to Hacklog, the Cyber Security and Ethical Hacking course. My name is Stefano Novelli, and I am the author of this course – I decided to write this document to give anybody the chance to approach cyber security in a more accessible way, compared to traditional channels.

Hacklog is the result of many years of study in the Hacking and IT Security fields: it encompasses testimonies, techniques and considerations, collected from documents, training courses and first-hand experience in the Security industry. As a course, Hacklog is designed for who wishes to learn and have an insight over Cyber Security; this manual is not aimed to offer professional training to IT Security experts, and is not intended to replace any University-grade guide book. This course has been designed for you – as a student or a self-taught reader – who wish to familiarize with Ethical Hacking and Cyber Security, learn the main techniques to run security tests on your machines and protect yourself from the intruders buzzing in the dark world of cyber-crime.

I would be a liar if I told you that you can start over without any IT knowledge. However, I don't mean to discourage you, but it's quite the contrary: the fact that you're here is a *very good start!* This means you want to learn, and I can tell you this is a very important, if not crucial, fact.

While you read this document, I will demand you to:

Have a positive attitude towards the course, don't get discouraged soon!

Learn more about what is not too clear for you.

Take notes, with pen and paper if you wish!

Get in touch with other people if you can't understand any part of it.

Please, keep in mind that the IT basics will be taken for granted, such as the difference between hardware and software, what is an operative system, how to download programs, and so on. Let's begin already! Enjoy your reading.

Anonymity

Over the years, anonymity on Internet became one of the most crucial issues, to the point that nowadays a huge range of tools is out there to help us leaving no traces around. The need for being invisible online is not only a prerogative of cyber-criminals: in some parts of the world (such as *China, Saudi Arabia, Iran* or *North Korea*), government censorship is so strong that anonymity is necessary not to be tracked by public or private spy services and to avoid penalties in those country where Death Penalty is still inflicted. In the rest of the world, anonymity can be useful for other scenarios, i.e. to report poor working conditions or questionable internal policies of a given company, as well as to be free to use the net outside a strongly analytical system, refraining from sharing information about what we buy or sell, what we like or dislike with the Internet Big Companies, thus escaping the mass social experiment run by the major global powers.

Anonymity is also a fundamental feature for hacktivists, namely those who practice digital activism. One example is the Anonymous movement, and such name clearly reflects the need to be untraceable during online protests.

If you need to secure your IT structure, you should actually consider another good reason: to be anonymous as a means of prevention, avoiding any exposition to the Internet, where you can potentially be attacked by anyone.

Instead, if you work in the IT investigation field, you may be interested in knowing the tools used by cyber-criminals to execute their attacks staying anonymous and avoiding controls.

1. Operative System

When you use a computer or a smartphone, actually you're using the operative system installed on such device: without it, your machine would be a lifeless box filled with cables, capacitors and electronics. The Operative System is the software managing everything within a computer: its role is to understand what the user is typing, what to show on screen, run programs and so on.

There are different Operative System families available for the Desktop environment; the main three are: Windows, macOS (formerly OSX) and GNU/Linux. If you are familiar with them, you'll know that GNU/Linux is the most frequently recommended Operative System: maybe I'll say something unpopular, but this shouldn't be the only option, instead I think that every OS has pros and cons, and it may be more or less fit for any given scenario it has to be used in. Surely, at least for what concerns Anonymity, *GNU/Linux* Operative System is the ideal choice for who wants to be anonymous.

GNU/Linux is an open source project, therefore it's free, it can be modified, and it doesn't contain any intentionally malicious code. It's the best choice for users who need to stay anonymous: this system is built without any distortion, it has not been manipulated and it can hardly be tracked by spy services, governments, companies in the industry, intruders and so on. A great advantage of GNU/Linux is its flexibility, allowing anyone to build their own distro: such principle gave birth to big communities, even entire companies – such as the most popular *Red Hat*, *Novell* or *Canonical* – that get their revenue from the Penguin ecosystem, ensuring thousands of jobs every year. Trust me if I told you that distros are really unlimited in number: from the historical *Debian* or *Slackware* to the most user-friendly ones like *Linux Mint* or *Ubuntu*, to platforms designed for gamers, such as *Steam OS*, for audio/video productions, for microcomputers, *Servers*, *Firewalls*, *Routers*, etc. Among these infinite distros, we can also find some specifically designed for Anonymity.

1.1 Which distro?

I've always believed that there is no one-fit-for-all distro. In my opinion, choosing a given GNU/Linux distro should not be a simple matter of pre-installed software. Firstly, it must be related to what a user needs and their level of *knowledge* (also keeping in mind to what extent they adhere to the project philosophy).

If you are unfamiliar with a GNU/Linux distro, this may be the *best moment* to familiarize with it! In some cases, you could have to use Windows or macOS nevertheless: we will also marginally cover such Operative Systems.

During the course, we will mainly use Debian, a primary distro used to develop the most popular distros available online, such as *Ubuntu*, *Linux Mint*, *Elementary OS*, *Kali Linux*, *Parrot Security OS*, *Backbox*, *Tails* and many more. If this is your first approach, I suggest you to start directly from *Debian* – you'll learn much more and it will be easier for you to shift from a distro to another, once you'll familiarize with it.

In this document, we won't cover how to install and run Debian: you can read a doc (in Italian version) available for free at www.hacklog.net, explaining how to install a functional version of Debian and how to overcome the most common issues. If you don't feel comfortable with Debian for any reason, or if you're having trouble installing your peripherals, try with *Ubuntu* or *Linux Mint*, since they are more user-friendly and have built-in, proprietary drivers. Besides the positioning of some elements, the commands we are going to use will work on these distros as well.

Instead, if you are confident with a given distro, you won't have any problems in using another of these sub-distros, even if they belong to other families. In the final part of this book, you'll find a complete overview of all Linux distros designed for anonymity (and partially for pentest as well), so you will be able to repeat the different tests using working environments specifically designed for anonymity and not.

1.1.1 Virtual Machines

If you followed previous Hacklog courses, you know there is a faster and painless way to have Linux installed in your computer, without any partitions, using a Virtual Machine. Virtual Machine is a kind of machine acting as a complete computer, but actually residing within a different Operative System: this ensures a stronger software compatibility and a better System usability, however it may compromise performance and, specially, expose the user to serious risks in terms of security and privacy. You'll find the reasons behind the latter statement in the "Live OS" chapter, at the end of the course.

Finally, since the environment is virtualized, the System will have to follow the rules enforced by the main Operative System, then you may encounter some issues with anonymity software. For these reasons, using Virtual Machine is not advisable if you wish to apply most of the techniques explained here.

1.1.2 Live Distros

During this course, we will see why a safer approach is to use some types of Linux distros, distributed only to be used Live, namely running without being installed on your PC. Although they are extremely useful, we will only cover them at the end of the technical arguments, because they won't allow to apply some anonymity techniques, as for Virtual Machines.

1.1.3 The Terminal

One of the most important features covered in this course will be the use of the terminal, a software installed in all Operative Systems by default. Although we will commit to avoid any possible issue, the terminal behavior may be different according to the type of Operative System in use. This is one of the reasons why we suggest to use *only certain distros (based on Debian GNU/Linux)*, so we will be able to anticipate each Operative System response, preventing any fatal issue.

When we use the command line, we will use a program called Terminal. The terminal looks like this:

```
$ ping www.inforge.net
```

```
PING inforge.net (192.124.249.10): 56 data bytes
```

```
64 bytes from 192.124.249.10: icmp_seq=0 ttl=51 time=32.630 ms
```

```
--- inforge.net ping statistics ---
```

```
1 packets transmitted, 1 packets received, 0.0% packet loss
```

```
round-trip min/avg/max/stddev = 32.630/32.799/33.073/0.195 ms
```

From this screen, you only have to type “ping www.inforge.net”, excluding any data, which will change according to situations we cannot determine. Ignore the initial Dollar symbol (\$), this only shows the beginning of a new line.

Keep have this page available, whenever you get lost in the Operative System!

In order to know which files and directories are contained in the path we are in:

```
$ ls
```

To access a folder:

```
$ cd {foldername}
```

To go back to the previous folder:

```
$ cd ..
```

To copy a file:

```
$ cp {filename} {newfilename}
```

To move or rename a file:

```
$ mv {filename} {newfilename}
```

To create a folder:

```
$ mkdir {foldername}
```

To use a text editor (we will use the CTRL+X key combination to close the editor and Y/N to confirm a possible overwrite action):

```
$ nano {filename}
```

And so on. Using the terminal, we will run programs also requiring some parameters, specified by the - (minus) character: if we wish to know how the ls command works and its allowed parameters, we must use --help:

```
$ ls --help
```

Or, we can use the man tool:

```
$ man ls
```

Furthermore, remember that we will use the apt commands to install new programs on Debian:

```
$ apt-get install [packagename]
```

Although not officially supported by this document, it may be possible to install the same package on Red Hat-based distros (Fedora, CentOS, etc.) using the command:

```
$ yum install [packagename]
```

or also on Arch Linux-based systems, using the command:

```
$ pacman -S [packagename]
```

You'll always have to launch these and other commands as root (administrator). In such cases, you should use the prefix:

```
$ sudo apt-get ...
```

If the latter is not present, you'll have to login as root first, using the command:

```
$ su
```


2. Data Traces

Now that we have installed Debian, it's time to learn which traces we may leave on the net. With "data traces", we mean all the digital values that can help revealing our identity somehow. Such traces may identify your computer or your network adapter, as an evidence of your connection to unsafe networks.

In the worst case scenario, if you use your Internet contract, it's quite possible to expose the first and last name of the *connection* owner. There are many techniques to identify someone who surfed anonymously: later on, we'll cover how it can happen and the related countermeasures to avoid that situation.

2.1 MAC Address

MAC (Media Access Control) address is a unique 48bit code assigned by network adapters manufacturers to their 802.x models; the code is directly written in the adapter *EEPROM* memory and is used for the first authentication stage to a local network by a network device, such a router, a switch and whatnot, which will later specify a local IP.

MAC Address is composed by 6 couples of alphanumeric characters, including numbers from 0 to 9 and letters from A to F (the so called hexadecimal notation, or base 16) and is represented as follows: ab:bc:cd:de:ef:f0. The first three sets of numbers (ab:bc:cd) are related to the manufacturer; check the IEEE2 standard list for the manufacturers index^[2].

Imagine to connect to a hotel or a public plaza WiFi connection: in this case, a network structure will manage the DHCP protocol, a system which automatically assigns the MAC Address a local IP address, allowing you to freely surf the web! The importance of leaving no traces of a MAC Address is that data is stored in the network device, and the latter may not allow to remove logs, not even to its owner. Furthermore, this MAC Address will be probably shared by the router/switch with the ISP (Internet Service Provider), which could store it into their own databases.

2.1.1 Identifying the MAC Address

In order to test the upcoming techniques – allowing us to change our MAC

Address – we need to be able to identify our MAC Address first. To achieve this, we can use a command line tool available in any operating system (on Windows it's known as Command Prompt, while on Linux and macOS it's called Terminal).

On Windows, launch the command `ipconfig`; on macOS and Linux, use `ifconfig`; actually, the latter one is going to be deprecated and replaced by the `iproute2` software (evoked using the `ip` command). Please, keep in mind that commands have to be run as root, therefore you must use the `su` command to be granted with admin access. However, each command may show the configuration of all the network interface controllers in the computer:

```
$ ip link show {interface}
```

```
en1:  
flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST>  
mtu 1500
```

```
ether 61:a8:5d:53:b1:b8
```

```
inet6 fe80::6aa8:6dff:fe53:b1b8%en1 prefixlen 64 scopeid 0x4
```

```
inet 192.168.0.12 netmask 0xfffff00 broadcast 192.168.0.255
```

```
nd6 options=1<PERFORMNUD>
```

```
media: autoselect
```

```
status: active
```

Where *{interface}* is the name of our network adapter. Usually, *eth0* represents the Ethernet adapter, while *wlan0* is the WiFi controller. It may happen that identifiers are different, according to the number of interfaces installed on your PC. If you wish to verify it, you can see which interfaces are enabled using the command:

```
$ ip link show or ip a
```

We need to identify our MAC Address which is composed by 6 couples of hexadecimal characters, as mentioned above, separated by colons. In our case,

the MAC Address will be *61:a8:5d:53:b1:b8*.

2.1.2 MAC Spoofing

Fortunately, in (almost) all cases, we can hide our MAC Address – doing the MAC Spoofing in IT jargon – in a very easy and effortless way. On *GNU/Linux*, you just have to execute a couple of commands from the terminal:

```
$ ip link set dev {interface} down
```

```
$ ip link set dev {interface} address 00:00:00:00:00:01
```

```
$ ip link set dev {interface} up
```

Please note that when you set this MAC Address, your computer won't be able to access the net anymore. You'll have to generate a valid MAC Address, but I won't cover this here, due to the complexity of the topic^[3]. You can restart your network manager using the command:

```
$ service network-manager restart
```

Instead, we'll use a tool available in most of the GNU/Linux distros repositories to generate a random MAC Address. This program is *macchanger* and you have to install it first. In order to install it, use the command:

```
$ apt-get install macchanger
```

You will be prompted to change your MAC Address immediately. If you select *No*, you can do it anyway using three commands:

```
$ ifconfig {interface} down
```

```
$ macchanger -r {interface}
```

```
$ ifconfig {interface} up
```

On Linux, the *ifconfig* command allows you to check your configurations and control your network interfaces. As you have seen, with *{interface} down*, you told your network adapter (in this example identified as *eth0*) to shut off. In this way, you can use the *macchanger* command to generate a random value

(using the `-r` parameter) and apply it to the `eth0` network adapter. Once you complete these steps, reactivate your adapter using the `ifconfig {interface} up` command. Feel free to replace the `ifconfig` command with the newer `ip` (`iproute2`) one. In case of connectivity issues, you can also restart using the command:

```
$ service network-manager restart
```

Although this operation is quite easy, you can find different scripts online to automatize the entire process. Here are some:

- SpoofMAC (<https://github.com/feross/SpoofMAC>)
- spoof (<https://github.com/feross/spoof>)

In the Windows environment (Figure 2), you can choose among different options, such as directly changing the settings through the following path: *Control Panel -> System -> Hardware -> Device Management -> Network Adapters -> Adapter name -> *right-click* -> Properties -> Advanced -> Net Address -> Value:*

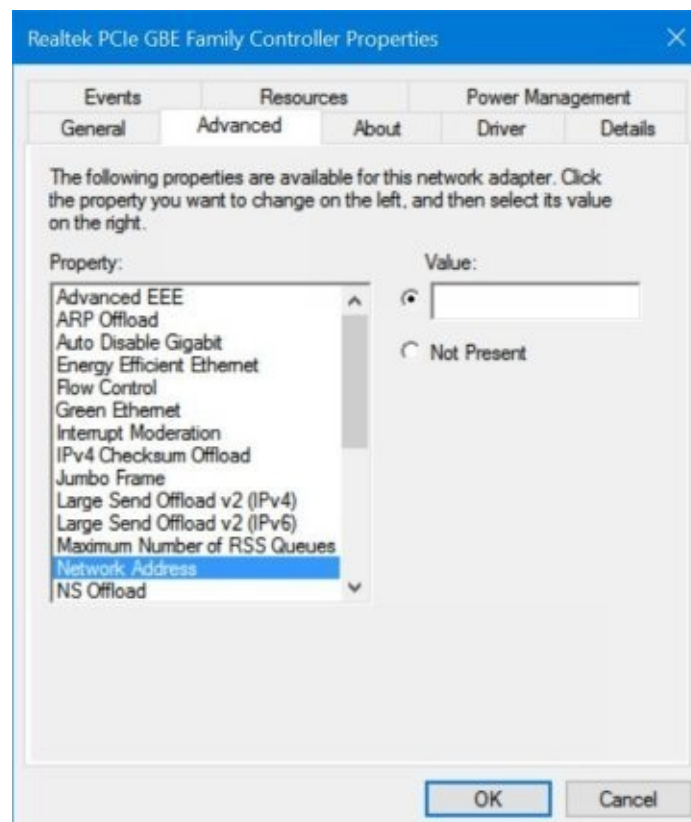


Figure 2: here you can change the vaule on Windows OS.

NB: this feature may be missing on some network adapters, because it is made available at the discretion of the manufacturers and according to the existing drivers.

However, the above feature can be used through many tools available online. If you have some spare time, you may want to try one of the following:

- MacMACs (<https://www.irongeek.com/i.php?page=security/madmacs-mac-spoofers>)
- Win7 MAC Address Changer (<http://www.zokali.com/win7-mac-address-changer/>)
- Technitium MAC Address Changer (<https://technitium.com/tmac/>)
- Change MAC Address (<http://lizardsystems.com/change-mac-address/>)
- mac-spoofers (<https://github.com/angusshire/mac-spoofers>)

On *macOS*, changing the MAC Address of an Ethernet network adapter is a relatively easy task. You only have to run the following commands:

```
$ sudo ifconfig en0 ether aa:bb:cc:dd:ee:ff
```

```
$ sudo ifconfig en0 lladdr 00:11:22:33:44:55
```

When it comes to a *WiFi* adapter, things get more complex. In this case, you have to patch your kernel^[4], but you should try it only if you feel confident enough with the Apple world.

If such practice is too complicated for you, but you still wish to work in the Mac OS environment, consider purchasing an external USB device, and you will be able to do the MAC Spoofing easily, as you'd do with the Ethernet adapter.

2.2 Hostname

Usually, a hostname is configured during the installation process. However, in some cases such possibility is hidden from Windows, MacOS and Linux Operating Systems.

The hostname is an identification name we can specify for a device, to see its role within a network: however, this piece information is often left to chance, so we can usually find the username (on Mac, we will read something like Stefano-MacBook-Pro.local), the Operating System in use or other data we may want to keep safe from other people in the LAN.

2.2.1 Changing the Hostname

In any Linux and MacOS distro, the command we can use to know our hostname is... hostname!

```
$ hostname
```

The latest release of Debian has integrated the *systemd* command, then you can also use:

```
$ hostnamectl
```

and you will learn more about your Operating System, including static hostnames, machine IDs, the GNU/Linux version in use, your architecture, etc. *The hostname command also allows you to temporarily change the value by typing:*

```
$ su
```

```
$ hostname [newhostname]
```

You can *verify* such change running the aforementioned commands again, or closing and reopening the terminal session. If you need to *permanently change the computer hostname*, on Linux you can use the command:

```
$ su
```

```
$ sysctl kernel.hostname=[newhostname]
```

while on macOS it will be:

```
$ sudo scutil --set HostName "[newhostname]"
```

As we will often see throughout the course, Windows manages information in its own way. In this case, you have to *right-click* the Computer icon and select *Properties*. You'll find all the hostname information in the screen under "*Computer Name, Domain and Workgroup Settings*".

2.3 Domain Name System

Before the invention of DNS, if you had to access to a computer in a network, you had to know its IP address, namely a set of numbers identifying an IT device in a network (we'll cover it shortly). As devices increased in number, it became impossible to remember such sets, therefore, in 1983, a new system was created in order to help memorizing them, using a unique name (i.e. *inforge.net*) instead of the reference *IP address* (such as 192.124.249.10).

It was the birth of DNS logic and the related DNS servers, which translate a *domain name* into the matching *IP address*. Domain is not referring to the website only, but to the entire network – we use the term domain to represent an entire network consisting in computers that share the same logic and the same rules enforced by their admins. For now, you only need to know that your computer or Internet network will only respond to the DNS given by your *ISP*, unless it is explicitly expressed.

As we are going to see, one of the major threats to users privacy online are *Internet Service Providers* (ISPs), for this reason you should replace the DNS you are using, whether you want to be 100% anonymous or not. Furthermore, you should consider the huge advantage in terms of response offered by more efficient and trusted alternate services. As you may have noticed for many websites, in order to accelerate governmental takedown operations, the competent bodies commit to directly censoring a domain resolution from a DNS, rather than blocking the related serves: it already happened for many sites (like *The Pirate Bay*) and it will keep on happening in the future. Using unfiltered DNS, you'll be a little more anonymous and automatically access to a complete, unfiltered list of all the websites actually available on Internet.

2.3.1 Choosing DNS

You can use two types of DNS: public and private ones.

Using public DNS you will improve your anonymity and privacy, and your DNS requests will be faster, with a safer navigation (if you are concerned about malware-filled sites). Public DNS usually leverage two IP addresses, known as *primary and secondary DNS*. Think of the secondary DNS as a backup, when the primary is temporarily unavailable or busy.

Currently, there are many DNS provided by a high number of companies online: I will not list the IP addresses, since they can constantly change, but I suggest you to follow the official links and choose the best fit for you:

DNS name	Official Website
Comodo Secure DNS	https://www.comodo.com/secure-dns/
DNS Advantage	https://www.neustar.biz/services/dns-services/dns-advantage-free-recursive-dns
FoeBuD e.V.	https://digitalcourage.de/support/zensurfreier-dns-server
German Privacy Foundation e.V.	http://www.privacyfoundation.de/service/serveruebersicht/
Google Public DNS	https://developers.google.com/speed/public-dns/?csw=1
<u>OpenDNS</u>	https://www.opendns.com

<u>OpenNIC</u>	https://www.opennicproject.org
PowerDNS	https://www.powerdns.com
Validom	http://validom.net/

Please, consider the highlighted ones as the most recommended for your digital raids; if possible, avoid DNS provided by Big Companies like Google (if you know how they operate, you should avoid them).

Alternatively, you can create your own private DNS on a Dedicated Server or VPS. This is an extremely complex system administrator task, then I'd only recommend it to networking veterans, using one of the many guides available online^[5].

2.3.2 Changing DNS

In most cases, you can use alternate DNS following two procedures:

1. Changing DNS on your router/modem (recommended)
2. Changing DNS on your Operating System

The first case directly applies to the Router or Modem you're using, through the web interface provided by your network device. Just access the gateway address (obtained by the commands we used for Mac Spoofing via *iproute2*, *ifconfig* or *ipconfig*) from your web browser, type the admin password and enter the IPs under the DNS change section.

On the OpenDNS^[6] forum, you can find a good list, including almost all products available in the market, and how to change their values. If you are working on an Operating System, it's easy as well.

In example, on Windows (Figure 3), just follow this path: *Start -> Control*

Panel -> Network Center -> *right-click the network you are using -> Properties -> Internet protocol (TCP/IP) -> Properties -> Check the “Use the following DNS server addresses” option.

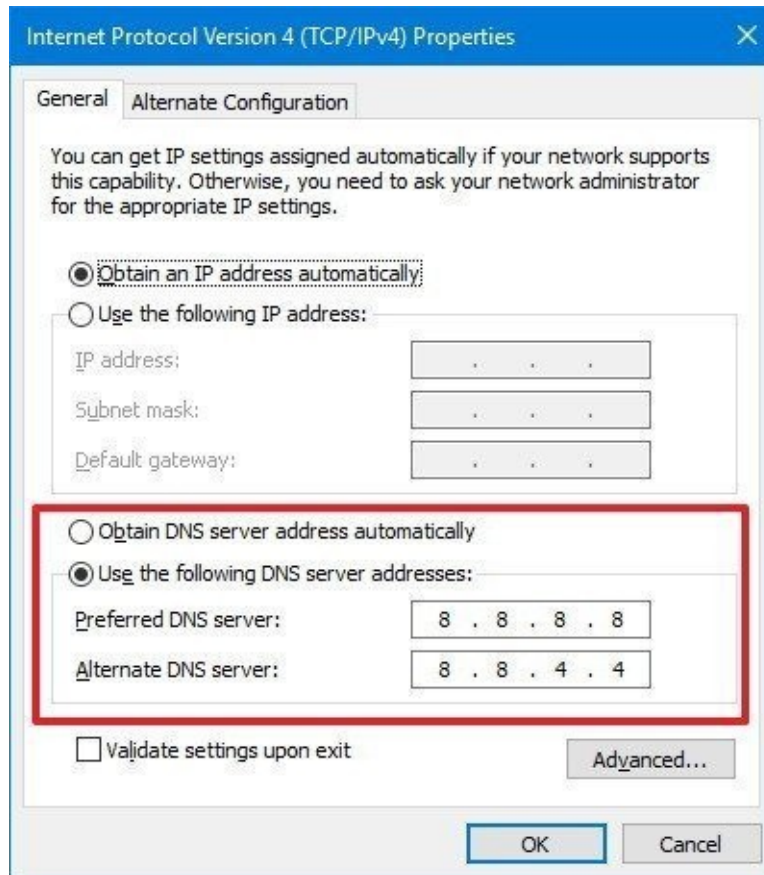


Figure 3: changing DNS on Windows

In our example, we changed our OS DNS, pointing to Google (Windows refers to the primary DNS as “preferred” and to the secondary as “alternate”).

On macOS operating systems (Figure 4), just follow this path: *Apple -> System Preferences -> Network -> Advanced -> “DNS” tab -> Complete the fields as shown and click the + button.*

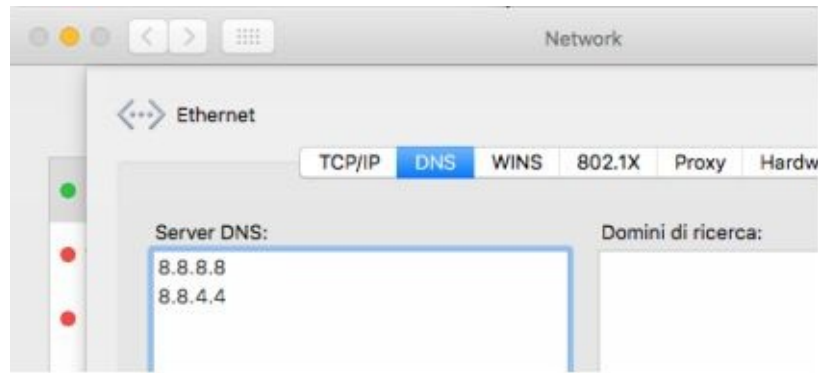


Figure 4: changing DNS on OSX/macOS Operating Systems

On GNU/Linux, naturally, it depends on the type of distro and the Desktop Manager in use. In our case, using Debian with GNOME 3 (Figure 5), you can change DNS under *Network Manager* (top right button) -> Choose the network (*eth0*) -> *click the wheel icon* -> IPv4 -> DNS -> add DNS with the + button.

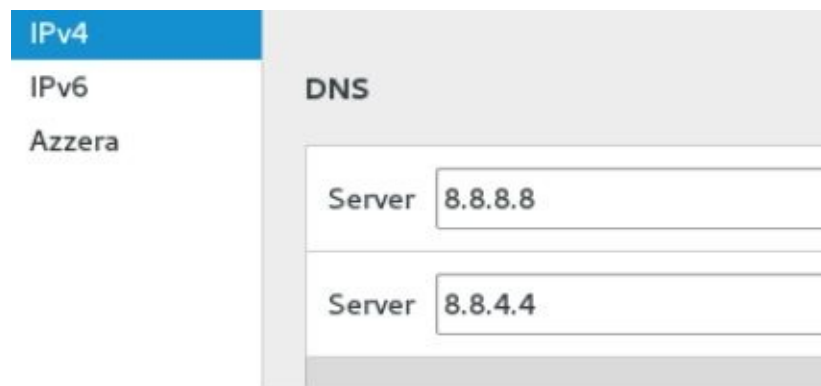


Figure 5: changing DNS on Debian Operating Systems with GNOME 3

Luckily, Linux users can do almost everything via terminal (changing DNS included). You can edit the “*resolv.conf*” file using *nano*.

```
$ su
```

```
$ nano /etc/resolv.conf
```

Within the file, enter the following (if you find any value, replace it or comment using #):

```
nameserver {DNS}
```

```
nameserver {DNS}
```

Remember that you'll save your files on nano using *CTRL+X*, then “Y” to confirm you changes and *ENTER* to apply the final change. Now, restart your *network-manager*:

```
$ service network-manager restart
```

You can verify your DNS by entering:

```
$ nmcli device show eth0 | grep IP4.DNS
```

2.3.3 Cache DNS

In time, operating systems introduced many features to improve general performance. One of the most important is DNS caching, a process which memorizes a domain resolution on a list stored in the computer, since domains rarely change their target IP addresses, making pointless the resolution of a domain IP address. However, this creates a *privacy* issue: *DNS caching* exposes the full list of domains visited by the final users, although they commit to stay anonymous (including private navigation).

Fortunately, clearing the DNS cache is quite simple, even because system admins must run maintenance on their network infrastructure quite frequently. Once we reached this stage, we have to wipe the cache for all our old local DNS.

On *Windows*, you can run the command:

```
$ ipconfig /flushdns
```

Furthermore, you may want to experiment without having to clear the damn cache every single time. On *Windows*, you may temporarily toggle this feature on/off from the command line:

```
$ net stop dnscache
```

```
$ net start dnscache
```

On *macOS*, we may find different variants, since some tools from certain versions are not available on the newer ones anymore (and vice versa). The

following seems to be the most functional one:

```
$ sudo dscacheutil -flushcache; sudo killall -HUP mDNSResponder
```

On *GNU/Linux*, we can install `nscd` first:

```
$ su
```

```
$ apt-get install nscd
```

then flush the cache:

```
$ /etc/init.d/nscd restart
```

You can learn more online^[7].

2.4 IP Address

The IP address is a *unique set of numbers* identifying an IT device connected to a network. IP addresses as we know them are in IPv4 format, composed by four sets of numbers evaluated from 0 to 255, for example 192.168.1.1. In the coming years, Internet will gradually shift to a new format – IPv6 – allowing more devices to have a unique identification code. Until then, this course will include examples with IPv4. Furthermore, many people mistake the public IP with the local one: an IP address is assigned by a *network* and the latter can be local or Internet-based, just like IPs.

The local IP address is then assigned by an internal network device, such as a Modem or a Router, to identify a device within a network (i.e. a computer within a local network). In the most common cases, IP addresses are specified with values like 192.168.0.x or 192.168.1.x.

The public IP address, instead, is assigned by the provider or ISP offering the Internet service: such address identifies a network or an IT device. Since public IPs are assigned by ISPs, they cannot be changed by final users, but only hidden. Finally, public IP addresses can be static or dynamic, therefore they can remain unaltered or change every time the modem is restarted (according to the customers Internet service agreement).

2.4.1 Determining the IP in use

In order to identify the public IP in use, we can rely on different online services. Most simply, we can visit one of the following portals via browser:

- <https://www.whatismyip.com>
- <http://whatismyipaddress.com>
- <http://whatismyip.org>
- <http://mxtoolbox.com/whatismyip/>
- <http://ip4.me>

If you wish to familiarize with the *Linux embedded terminal*, use the *wget* program:

```
$ wget https://ipinfo.io/ip -qO -
```

In order to learn how the *-qO-* parameters works, run the command:

```
$ wget --help or man wget
```

2.4.2 Proxy

Cyber criminals will aim to hide their IP public address – the one that can identify them on Internet – while they won't care too much about the local address, since they will have already wiped their *MAC Address*, so any data within the local network will not betray them. As you already know, the *local IP address* is assigned by a router and is not enough to identify the computer owner, unlike the *MAC Address*.

It's worth mentioning that experienced cyber criminals will mostly never work from their home or a nearby network: despite all precautions put in place, they perfectly know they must hide every single trace or evidence, including the “borrowed” network connection used for their attacks. Therefore, they will rely to one of the oldest IT tools: Proxies. Proxies (technically, open proxies) are servers – called *proxy servers* – which can perform different operations:

- Provide anonymous navigation

- Copy web pages
- Run software-level filtering, acting like a Firewall

We must consider that, nowadays, proxies are less and less used for anonymous navigation, since they have been replaced by more effective methods; however, they are still useful in certain scenarios – mainly in programming – therefore you have to know them Basically, proxies lay between a *client* and a *server*, acting as in intermediary between them.

2.4.2.1 Proxy types

As we mentioned above, there are many types of proxies, according to different purposes and design specifications. Although it would be useful to understand how they can be smartly used in *server infrastructures*, here we will only explain the differences in the scope of anonymous navigation.

Proxy HTTP/HTTPS

As we can tell, HTTP/HTTPS proxies can filter information within the *HTTP* protocol and its secure form, *HTTPS*. In short (at least, for now) let's say that HTTP is a communication protocol designed to parse information at the *World Wide Web level*. It's the most popular protocol and has two forms:

- HTTP (not encrypted)
- HTTPS (SSL/TLS encrypted)

When it comes to HTTP proxies, they are the most popular and easy to find, since servers only have to manage such protocol, and then optimize their machines for that single task. Compared to *SOCKS* (that we will cover shortly), they are usually more responsive but, naturally, restricted to their protocol. In turn, such types of proxies are broken down into sub-categories according to their “quality”. Although each agency distributing proxies use their own evaluation criteria, we conventionally distinguish 3 levels:

- Non anonymous proxies: they don't mask the original IP and usually add a single string to headers (data sent in packages) to the recipient server.
- Anonymous proxies: they mask the IP address but alternate headers to the

recipient server.

- Elite proxies: they mask the IP address and don't alternate headers.

SOCKS4 Proxies

Using a proxy supporting the *SOCKS4* protocol instead of *HTTP/HTTPS*, you can reroute any TCP-based data, and it is a huge benefit. This essentially means that you can filter the *World Wide Web* services – naturally based on *TCP* as well –but also the whole range of protocols supporting this kind of service. You can also find a variant named *SOCKS4a*.

SOCKS5 Proxies

Quite identical to the previous one, *SOCKS5* can also reroute data on the *UDP* protocol, making it the safest proxy de facto. Furthermore, *SOCKS5* protocol allows the proxy owners to enable an internal authentication system as well as the *IPv6* support. Then, you can use *SOCKS5* proxies with any type of software that uses an Internet connection, such as mail, chat, p2p programs, etc. It is the direct evolution of *SOCKS4* protocol.

Web Proxies (or CGI Proxies)

Web Proxies are actual websites that don't require any configuration or particular tools on the computer, allowing to directly navigate in anonymity. You can find dozens of these tools online. Here are some we tested for you:

- whoer.net
- hide.me
- proxysite.com
- vpnbook.com
- hidemyass.com
- kproxy.com
- hidester.com

- filterbypass.me

You can find a complete list at www.proxy4free.com .

2.4.2.2 Where you can find Proxies

Once we understood what's the use of proxies, we also need to know where we can find them!

Through lists

Probably, a novice user would use Google, typing keywords like “*proxy list*” but they may not be aware they are the last in a line of millions of people thoughtlessly using proxies. This means that, in 99.9% of cases, they would obtain *compromised proxies*, marked as abused and banned, filtered or even inactive, being closed by their host, while the active ones would be slow and unstable.

For your information, the most active and popular sites where you can find proxies are:

- [Hidemyass](#) (Proxy list) - HTTP/HTTPS/SOCKS
- [Proxy4free](#) - HTTP/HTTPS
- [samair.ru](#) - HTTP/SOCKS
- [inCloak](#) (Proxy List) - HTTP/HTTPS/SOCKS
- [Cool Proxy](#) - HTTP
- [GatherProxy](#) - HTTP/SOCKS
- [SSLProxies](#) - HTTP/HTTPS/SOCKS

Then, you will need to constantly find new proxies that are fast enough, not partially blocked by websites and services, and offer a fair, general compromise in terms of anonymity.

Through Proxy Scrapers

Proxy Scrapers are software programs designed to scrape, or collect proxies over the web, in order to obtain the latest proxies, faster and with no efforts. Once again, we suggest you to use any search engine; we found some for you,

hoping it can be helpful:

- [Net Ghost](#)
- [GatherProxy Scraper](#)
- [Proxy Harvester](#)
- [Holy SEO Proxy Scraper](#)

Beware using those programs. Almost all are poorly programmed or, worse, they may contain harmful code for your Operative System (Proxy Scrapers are not always designed for fair purposes, and you make your bed...). The best thing you can do, then, is to make your own scraper, using a programming language and being very, very patient.

Through Premium Lists

Premium Lists are sites or newsletters/mailling lists containing an index of proxies not publicly shared yet. Such lists are mostly paid or reserved to elite groups. Actually, there are very few public services offering paid proxy lists, and the last remaining ones are not that exclusive, after all:

- [Hidemiyass](#) (around 24€/lifetime)
- [Premium Proxy Switcher](#) (around 9€/month)
- [ProxySolutions](#) (around 18€/month)
- [SharedProxies](#) (around 8€/10 proxyes)
- [Coolproxies](#) (around 10€/month)

2.4.2.3 How to use Proxies

On the whole Operating System

At this point, every user should be able to connect to a proxy, with more or less difficulties, according to the Operating System in use and its version.

In example: on Windows (Figure 6) you can set a proxy for the entire

computer following this path: *Control Panel -> Internet options -> Connections -> LAN settings -> Proxy Server*, while you can try different methods at the same time on *Windows 8* (but it's quite confusing). Once you reached the location, you can enter the proxy address and port into the relevant fields.

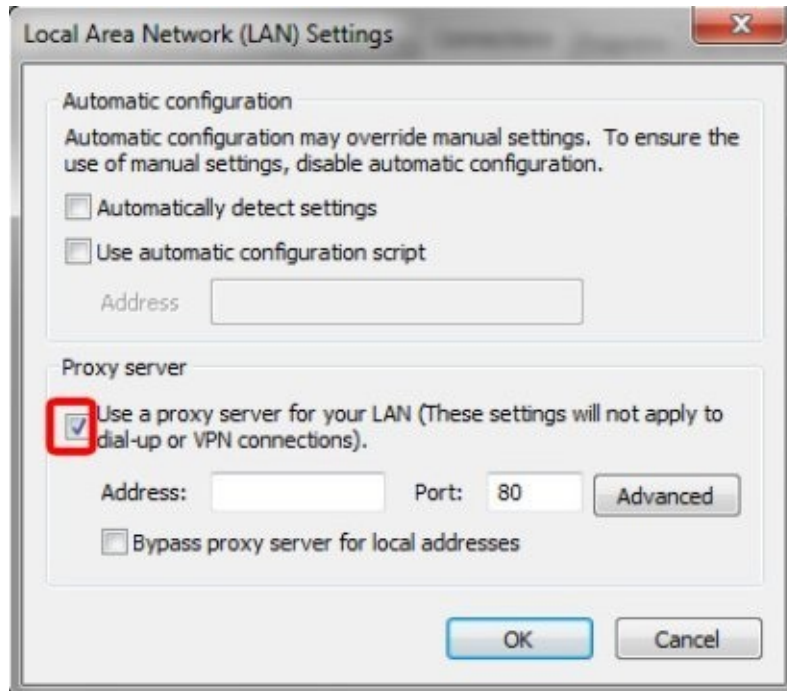


Figure 6: using a proxy in the Windows OS environment

Fortunately, the graphical environments of *GNU/Linux* (Figure 7) make things simpler: on Debian with GNOME 3, you can go to *Settings -> Network -> Network Proxies*.

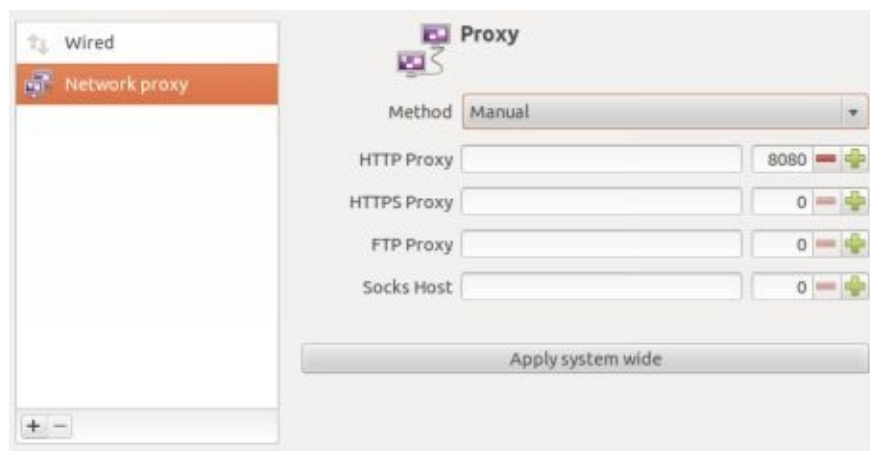


Figure 7: using a proxy in the Debian environment with Unity

To close this little chapter, we are mentioning the Apple operating system: in *macOS* (Figure 8) you can reach the option through *System Preferences -> Network -> Advanced -> Proxy*. Here you can also specify the different service types to filter and assign each one the related authentication data (in the case of *SOCKS5*).



Figure 8: using a proxy in the OSX/macOS environment

Once again, you can run the same process on Linux command line, using a text editor (*nano*) and editing the file you can find in */etc/environment*:

```
$ su
```

```
$ nano /etc/environment
```

As stated previously, regarding the graphical configuration, we will compile the file pasting the following lines:

```
http_proxy="http://myproxy.server.com:8080/"
```

```
https_proxy="http://myproxy.server.com:8080/"
```

```
ftp_proxy="http://myproxy.server.com:8080/"
```

```
no_proxy="localhost,127.0.0.1,localaddress,.localdomain.com"
```

Please keep in mind, however, that some internal programs (such as *APT* for Debian/Ubuntu-base distros) will bypass that reading^[8].

Using programs

Some software, such as sharing, chat, and other programs, allow the final user to use *internal proxy configurations*. The reasons may vary (enterprise grade, university, and other types of proxies) and this allows to use a proxy to anonymize incoming and outgoing connections as well. In order to know whether a program provides proxy functionalities or not, please see the official documentation.

Proxychains

The *proxychains*^[9] software is one of the best for using proxies in a targeted way; perhaps it is the best proxifier currently available. Unfortunately, its development halted in 2013: the good news is that a fork has been introduced, *proxychains-NG*^[10]. As an advantage, proxychains ensures that any program – and all its dependencies – use only *SOCKS4*, *SOCKS5* or *HTTP/S* protocols for outbound communications. A word of warning: *proxychains* is officially available for UNIX systems only, i.e. *GNU/Linux*, *macOS*, (via *Brew*) and *BSD*. In order to limit any possible issues, we will use the historical version (the deprecated one), still 100% working and available from Debian 8 repositories. Here's how to install it:

```
$ su
```

```
$ apt-get install proxychains
```

```
$ exit
```

It can also be executed by normal users, so you should get back to your default user via the exit command. To use the program:

```
$ proxychains wget http://ipinfo.io/ip -qO -
```

As you can see, it's very simple: just type "proxychains" before the command you wish to use. However, when you launch the command, you'll get the following error:

```
$ proxychains wget http://ipinfo.io/ip -qO -
```

```
ProxyChains-3.1 (http://proxychains.sf.net)
```

```
|DNS-request| ipinfo.io
```

```
|S-chain| -<>- 127.0.0.1:9050-<--timeout
```

```
|DNS-response|: ipinfo.io does not exist
```

This happens because proxychains reads a proxy configured over the 9050 port of your computer by default, i.e. in local with 127.0.0.1 as the IP. To let it read a list of your proxies, you have to create a config file. Using the terminal, launch the following:

```
$ mkdir $HOME/.proxychains
```

```
$ nano $HOME/.proxychains/proxychains.conf
```

This way, we firstly created the folder containing the config file, then we launched the nano text editor over the “secret” *path* in our user folder. Please note the first call of *\$HOME*, a variable allowing you to immediately access the absolute folder of your user. In my case, the user is named *stefano9lli*, so the folder path will be */home/stefano9lli*, which is in turn appended to the rest of the string.

You may also notice that the *.proxychains* folder is called after *\$HOME* variable. In the UNIX world, a period before a folder name means that it must be *hidden* when you use a file manager. Now, we will create the *proxychains.conf* file within the folder. And we are ready to add some values:

```
strict_chain
```

```
proxy_dns
```

```
[ProxyList]
```

```
http proxy port
```

Save using the *CTRL+X* combination, the *Y* key and pressing *ENTER*. Remember: you can use one of the following configurations to use different protocols:

```
strict_chain
```

```
proxy_dns
```

```
[ProxyList]
```

```
http proxy port
```

```
socks4 proxy port
```

```
socks5 proxy port
```

Launch again *wget* with the new configuration (in this example, the proxy is

177.73.177.25 and the port is 8080):

```
$ proxychains wget http://ipinfo.io/ip -qO -
```

```
ProxyChains-3.1 (http://proxychains.sf.net)
```

```
|DNS-request| ipinfo.io
```

```
|S-chain|-<>-177.73.177.25:8080-<><>-4.2.2.2:53-<><>-OK
```

```
|DNS-response| ipinfo.io is 54.164.157.29
```

```
|S-chain|-<>-177.73.177.25:8080-<><>-54.164.157.29:80-<><>-OK
```

```
177.73.177.25
```

```
|DNS-response|: ipinfo.io does not exist
```

As we can see, this time the ipinfo.io website returns the proxy IP instead the one of yours, indicating that proxychains worked successfully. Keep in mind that, actually, many proxies could be already configured to work under certain conditions only, so they could refuse to respond to some types of domain requests or to programs without user-agent. Naturally, trying is the best option. All the use cases for proxychains are explained in the following manual:

```
$ man proxychains
```

Proxycap

Maybe the most popular Windows counterpart to proxychains is *Proxycap*^[11] (Figure 9), a program developed by the Initex team for over 10 years. Just like *proxychains*, *Proxycap* can reroute all Internet communications, and is also equipped with a *GUI*. Unfortunately, it is also a paid app. Furthermore, you should also consider more alternatives. Wikipedia^[12] is hosting a page where you can compare the different proxifiers online.

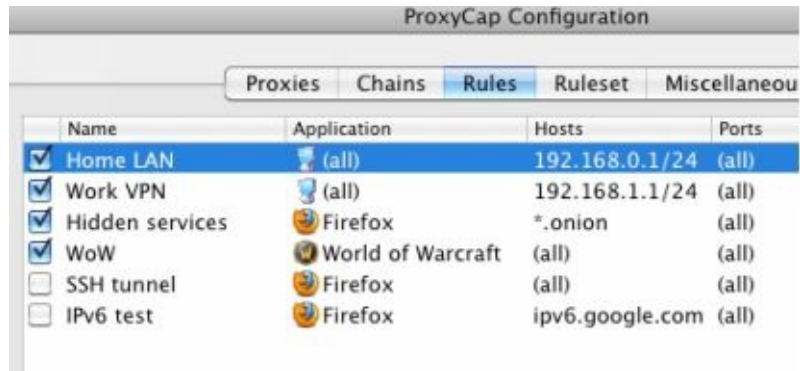


Figure 9: screen showing how ProxyCap works

During web navigation

Here, we will cover the proxy configuration via browser, however, you should keep in mind that screenshots and menus may slightly vary according to *Operating Systems* and *Browsers* versions. In our case, we will only cover the main navigation Browsers. *System browsers* (*Safari, Internet Explorer, Edge, etc.*) always rely on the system configuration.

Keep in mind that each Browser supports extensions as well, and you can always find a GUI to accelerate proxy use. Go to each Browser store and you will surely find the best extension for your needs.

Google Chrome / Chromium

From the Google browser, go to *Settings* using the *top right button*. On the tab that appears, click “*Show advanced settings...*”, then click “*Change Proxy Settings*”. Thus, you will directly go to the *Proxy configuration* prompts of your *Operating System*.

Mozilla Firefox

From the Firefox browser, select the *top right Settings* button, then open the *Advanced tab* and select the top *Network tab*. Here, you will find “*Connection*” and the “*Settings*” button next to it. Once opened, you can configure your browser with its own proxy settings, using “*Proxy manual configuration*”, or relying on the *system settings*.

Opera Browser

On Opera, it is a very easy task. Open the *menu* and select *Settings*, then go to *Preferences*. From the *Advanced tab*, go to *Network*, then click on *Proxy server*. Now, edit the *client settings*.

Beware the Blacklists

Proxies may be often included in Blacklists, online databases storing the IP addresses used for web abuse, fraud, spam and so on. Such lists are stored by free and paid services (the so called Honeypots), to help web portals, *Firewalls*, *CDNs*, and whatnot to rapidly search the visitor IP in the harmful IP database. The most popular are *Spamhaus*^[13] and *Barracuda*^[14], but there are more around. In order to check whether your IP is blacklisted or not, you can use all the IP verification services supporting it, or use the specific service offered by WhatIsMyIPAddress.com.

2.4.2.4 How safe are Proxies?

Now, the question we should ask to ourselves is: are proxies really ensuring full anonymity? In most case, the answer is: *not at all*. No matter how safe they may seem, proxy servers are managed by external services, paying to maintain servers capable to host our Internet requests.

Being external services, we ignore who manages them; furthermore, we don't know their business and why they should be such benefactors. Sometimes we may encounter entities fighting against censorship, or university proxies for research purposes; however, in some cases, we may find companies making money over our navigation (e.g. to run marketing surveys) or, in the worst case scenario, honeypots managed by governmental organizations like *NSA* or *FBI*, monitoring traffic.

Without mentioning the fact that a proxy server may store almost everything you do (sites visited, logins, operations done, etc.) and all the data you release online (IP address, browser, operating system, etc.), actually making it a double-edged sword. This doesn't mean that IP spoofing done via *proxy* is pointless; instead, its popularity and ease of use allowed to create lots of libraries for every programming languages, thus to define new use cases, e.g. many *bruteforcers/bots/stressers* and whatnot still use *proxy lists*.

3. Secure communications

So far, we have seen how an IP address can become a dangerous track to leave behind when you surf the web; any server in the world can log and store visitors IP addresses and match them to any action performed. Hiding an IP Address (technically, spoofing an IP) is not enough to mask Internet users activities online: for example, non encrypted requests can be tracked down by governments and *ISPs (Internet Service Providers)*, as well as other services and intruders lurking the web.

We previously introduced the *HTTPS* protocols as a new communication method for the computers connected to the web. As data encryption is getting more and more crucial in the IT scenario, secure protocols are replacing the weaker ones (*SSH -> TELNET, SFTP -> FTP, HTTPS -> HTTP* and so on). Unless the program or the proxy we are using is not explicitly encrypted, all our Internet operations will be easily traceable.

One note about WWW navigation: if privacy and anonymity are your priorities, just forget Google and similar forever; choose search engines that won't track you, like DuckDuckGo^[15] or StartPage^[16], instead. Why? Let's see an example: *YouTube* is a service acquired and managed by Google and, as we know, it tracks everything. *YouTube* reads your IP and the video you're watching and creates a user profile, called fingerprint, making predictions about what you'd like to watch next or, perhaps, what you're going to purchase while you visit websites with *Google Ads*. A pain chain.

3.1 VPN (*Virtual Private Network*)

We've seen that proxies are useful tools, but can't ensure a proper balance between safety and speed for several problems. Furthermore, navigating with an open proxy is virtually impossible, and dangerous as well! Unbelievable lag and sudden downs make it unsuitable for operations requiring more than 5 minutes! VPNs are considered by many as the tomorrow's proxies. Is it true? Let's find out!

VPNs (acronym of *Virtual Private Network*) are encrypted "tunnels" acting like an intermediary between a client and a server, just like proxies, so that all the Internet traffic passes through that *encrypted tunnel*, blocking anyone from monitoring your connection.

VPNs were originally designed to create a LAN network of computers connected via Internet, exactly like a physical network, but without all the related costs (device location, real world connections, etc.) and with all the proper arrangements, like Firewalls, Proxies and so on. Using a VPN, you won't have to concern about finding active lists or certain protocol types: all the traffic passing through a VPN is usually routed and encrypted with a 128bits quality standard at least.

Compared to a *proxy*, VPN ensure higher responsiveness; its architecture and server *geolocation* allow to optimize Internet network requests. Furthermore, you don't need to reconfigure browser and tools to stay anonymous, because the *tunneling* is generally provided for the entire system.

3.1.1 VPN Types

We can find at least three types of VPN in the market: *Trusted VPN*, *Secure VPN* and *Hybrid VPN*.

In this chapter, we will cover Secure VPNs, because the Trusted ones require special agreements with ISPs and are not easily applicable to common cases – they are almost exclusively designed for enterprise networks where information delivery to recipients must always be guaranteed.

Hybrid VPNs, instead, are the combination of Trusted and Secure ones and, since we are not going to cover the former, we will exclude the latter as well.

The quality of a VPN in terms of security is determined by the types of protocols and the safety of the keys provided – in addition to the policies and the service stability, as we will see at the end of this chapter.

Most commonly, a VPN alone doesn't ensure *security*: i.e. until a couple of years ago, the popular VPN provider, iPredator^[17] offered connectivity only via PPTP protocol: this kind of protocol was already considered as not fully secure, since dismissed by Microsoft (which invented and patented it), and now we are almost certain that governmental spy services can *crack* it in short times. This is just one example of what we found out. Now we're going to review each protocol and sum up their features and quality.

3.1.1.1 PPTP, for the speed seekers

PPTP (acronym of Point-to-Point Tunneling Protocol) was developed by *Microsoft* in order to create enterprise VPNs via telephone dial-up. It was exclusively designed for VPNs, and generally relies to *MS-CHAP* to manage authentication. Due to the popularity gained in the years, this tool can be easily installed (or found pre-installed) on any device in the market; furthermore, it is fast and requires limited resources to run. The PPTP protocol supports *128bit* keys only and started to lose ground to vulnerabilities, to the point that Microsoft declared it unsafe in 2012, despite the dozens of patches released to ensure its efficiency. The protocol is considered unsafe and, quite probably, it has already been violated by *NSA*. Nevertheless, PPTP is still useful for low latency activities, such as *online gaming, torrent, streaming, etc.*

3.1.1.2 L2TP/IPsec, for the security and responsiveness enthusiasts

L2TP (acronym of *Layer 2 Tunnel Protocol*) is a VPN-type protocol that doesn't ensure any data protection alone; for this reason, it's often integrated with the *IPsec suite*. L2TP/IPsec combines a tunneling protocol with encryption and is already implemented on the next generation operating systems, allowing an easy setup via client and a good overall speed. At the moment, no critical vulnerabilities have been identified for this protocol, so I can recommend it if you wish to maintain a good privacy and security layer. A research conducted by some industry experts^[18], however, suggests that *NSA* is involved in an ongoing effort to violate it. Although it has not been proved yet, some sources^[19] confirmed that IPsec is one of the main *NSA* targets, and an attack would be theoretically possible. However, L2TP/IPsec features dual stage data

encapsulation with *256bit* encryption keys; although it's slower than PPTP, the multi-threading support implemented in next generation *kernels* allows to leverage *multi-core processors architecture* for encryption and decryption operations. The only minor downside is that L2TP uses the *UDP 500* port by default, which is often blocked by business firewalls and requires *port-forwarding* on the most enhanced *routers* and *access points* (disrupting navigation, especially in open networks).

3.1.1.3 OpenVPN, for top security users

OpenVPN is an open source software specifically developed to create encrypted tunnels between two IT systems, leveraging the *SSLv3/TLSv1-based* encryption protocols as well as the *OpenSSL* library. The system is totally open and transparent enough to be considered as the most reliable and secure solution; at the moment, the risk of being violated by any governmental spy service is minimal. Due its open nature, you can configure this product as you wish and use it on any port without any port forwarding (i.e. also leveraging the *TCP 443* port to address *HTTP* requests through *SSL*) on your network device. The library in use (*OpenSSL*) can rely on different ciphers (like *Blowfish*, *AES*, *DES*, etc.), however, most VPNs use *AES* or *Blowfish* ciphers almost exclusively. The latter is *128bit*-based and the default cipher in *OpenVPN*. *AES*, instead, is a relatively new cipher and is currently used by different governments around the world for data protection purposes: since it's able to manage *128bit* blocks, it can manipulate data up to *1GB* in size, unlike the *64bit*-based *Blowfish*, which can manage only the half. Slower than *IPsec*, this protocol can negatively impact devices with limited processing power due to the lack of native multi-threading support; for this reason, it cannot leverage the next generation *CPUs*. Although it is not a standard de facto, like the aforementioned *PPTP* and *L2TP/IPsec*, *OpenVPN* has been welcomed in the VPN provider market, and the developer community released its client for all the most popular *Operative Systems*, including mobile devices.

3.1.1.4 SSTP, for Windows users

SSTP (acronym of Secure Socket Tunneling Protocol) is a tunneling protocol introduced by *Microsoft* and native for all *Windows* versions – *Vista* and later – and available, but not pre-installed, on *Linux* and *BSD* systems. Currently, there are no certain plans for mobile and the most popular *router firmware* (except *Router-OS*^[20], currently the only Operating System for routers supporting it).

Just like OpenVPN, it uses the SSLv3-based encryption, allowing to use encrypted tunnels even behind firewall protected networks; the SSTP protocol can be used together with *Winlogon* or *smartcard* authentication. It's the security protocol currently used within the Microsoft Windows Azure cloud. Unlike OpenVPN, however, it's a closed protocol, and the *PRISM*^[21] scandal, that revealed a collaboration between *Microsoft* and *NSA*, is not very reassuring.

3.1.2 Which VPN?

Time to sum up: what type of VPN is the *best choice for you*? Personally, I would recommend an OpenVPN, because it encompasses all the features you may want from a VPN, namely the best compromise among speed, safety and development transparency. The (minor) downside is that it's difficult to install and used, compared to other types of VPN (due to the lack of a built-in feature in almost every OS); most companies, however, provide documentation you can refer to for setup and utilization troubleshooting. L2TP/IPsec is quite popular too and unless you are utterly paranoid, ensures high speed and a good overall security. Honestly, I cannot recommend PPTP and SSTP: the former is obsolete and may be very harmful, the latter is focused to the enterprise world, rather than anonymity.

3.1.3 How to choose a VPN

Listing the top online VPNs and electing the best one wouldn't be wise, due to their ever changing market; as we have done for proxies, we will only provide some guidance on how to choose the best VPN for your needs. Then, we will summarize the most popular VPNs you can find.

3.1.3.1 Avoid Free VPNs

Perhaps you wondered: *are VPNs free or paid*?

The answer is: both. However, I want to clarify that from now on, I will only refer to paid VPNs. Why?

Reason #1: maintaining a VPN services has some costs

Some of the best VPN services, like HideMyAss, NordVPN or ExpressVPN offer more than 1000 servers around the world. And, unsurprisingly, those servers have a cost! There's a cost for maintaining them, a cost for replacing the broken ones, a cost for managing them. Unless you believe this world is filled

with benefactors spending hundreds of thousands of dollars each month to maintain them, you should never trust free VPNs!

Reason #2: providers may sell your data

How does a VPN monetize? Simply put, the providers may sell your data. I am not referring to usernames and passwords (but one never knows!), but to actual honeypots used for analytics purposes and to sell data to the highest bidders.

Reason #3: providers may reuse our bandwidth

Once you are in the circuit, you become part of the virtual network, so you are an “accomplice”; your Internet connection will be slower (quite obviously) and you may get to the “end of the line” and be deemed responsible for illegal practices performed by other users.

Reason #4: providers may bomb you with advertising

This is a quite common practice, both for the free proxies and the free VPNs. Adware in the Free VPNs may be embedded in the client, or showing up during navigation, changing the HTML code of the web pages you’re going to view.

Reason #5: you are not protected

When you purchase a service, you are protected by a document that you and the seller company automatically agree to, the “Terms of Service”. Together with the Privacy Policy, it is the legal document regulating the relationship between two parties. When it comes to Free VPNs, however confusing the documents may be, users tend to think: “as long as it’s free, who cares!” Actually, as we’re going to learn soon, ToS and Privacy Policies guarantee the VPN quality and ensure efficiency and safety during navigation.

3.1.3.2 No Logs Policy

Logs are files generated for each activity performed within an IT system: in the case of VPNs, logs may store IPs, access information and other data not encrypted before the *handshake* (taking to the actual *tunneling* and then to the total *encryption*). A short story before we go further.

Do you know the LulzSec group? Exactly, the guys who violated Sony and CIA.

Did you know that a LulzSec member, Cody Kretsinger – aka recursion – was arrested after he had been identified by the Feds, who required the access logs from the VPN provider, HideMyAss, used by the hacker to violate Sony Pictures?

If you're choosing a logless VPN, don't trust the advertising and go check the *Privacy Policies* declared by the provider.

3.1.3.3 If they haven't got your data, they can't catch you

Imagine you are the owner of a VPN company and, in the middle of the night, the FBI (or CIA, the police or whatever) rings at your door with a warrant to search data across your *servers*. Would you feel like being a justice warrior and protect someone you don't know, who possibly played with the mainframe of some corporation in the other corner of the planet? Needless to say, the answer is no! There are *no VPN providers* who would risk years in jail for you. There are no such benefactors; remember that providers will always mend their fences and, under the right pressure, they may sell you out (like HideMyAss).

Then, keep in mind that a VPN provider *cannot disclose information about you they haven't got* – therefore, they cannot be prosecuted for having failed to hand off data actually not in their possession. Usually, a VPN provider requires personal information to create an account and process payments (ex. name, email, payment information and billing addresses). Recently, the best VPN providers realized they can ensure better anonymity to their users offering payments in crypto-currencies (we'll cover them later), allowing, with the proper precautions, to anonymize the transactions, freeing the sellers from the risk of storing billing data.

3.1.3.4 International Data Retention Laws

Each country has *specific laws* about data protection and privacy, among others. The map below (Figure 10) shows countries in the world with a color code from *red* to *green*: countries in red have strict data retention laws, while green ones are quite flexible (states in white haven't any laws of this kind).

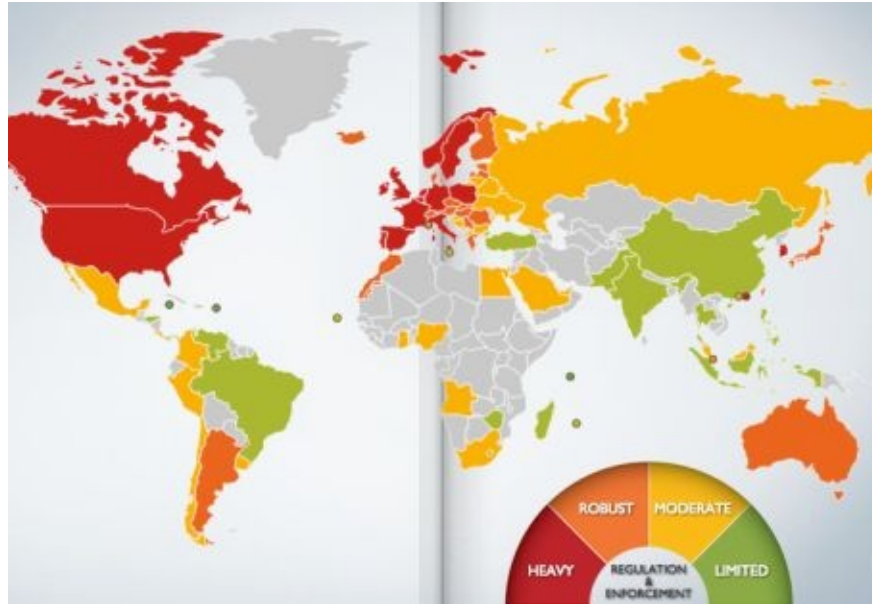


Figure 10: the following map and the related information are available online at dlapiperdataprotection.com

Just a real world example: NordVPN is a company located in *Panama*, a nation with almost no restraints in terms of data retention laws. Unsurprisingly, it is also defined as a tax haven, where 120 banks secretly serve rich entrepreneurs (including tax evaders) and offshore companies. Here, companies have no obligation to produce financial statements and residents may decide not to file their income tax declarations, so why would local VPN resellers bother storing customers tax data?

Similarly, let's think of HideMyAss, located in the United Kingdom: online trading requires the submission of documents, traceable payments, financial statements and, most importantly, is subject to cyber-crime laws as regulated by the *Computer Misuse Act*, allowing the Government to enforce searches wherever they want.

3.1.3.5 Payment Methods

Features that distinguish a secure VPN from a non secure one include the supported payment methods. If you are considering to rent a VPN with payment services like PayPal, credit card or bank transfer (using your name), be aware that you will leave behind significant traces. No matter how strong a VPN privacy policy is, banks will store your payment data (and we know they get quite along with governments).

A VPN accepting only traceable payments – *credit card, bank transfer, money order and so on* – cannot be defined as a secure VPN; unlike free VPNs that can only get your IP and any registered accounts, paid VPN may store data potentially threatening your anonymity, like the billing information of a credit card or a bank account.

In this case, you should prefer a VPN offering payments in crypto-currencies, like *Bitcoin, Litecoin*, etc., and take the proper precautions in order to avoid exposing your wallets to traceability risks (we will cover the safe use of crypto-currencies later).

3.1.3.6 DMCA Notices

DMCA (acronym of Digital Millennium Copyright Act) is a collection of American laws against the illegal distribution of copyrighted materials. Although it is the legislation of a foreign overseas country, is quite similar to the UE^[22] Copyright Law and may be somehow applicable to our country as well. We won't cover this issue any further due to its highly technical legal nature. The only thing we can be sure of is that any *DMCA* violation may compel your VPN to block your account, in order to avoid any legal issue.

3.1.4 VPN List

The following list includes some of the most popular VPNs I found online: you can find a more complete index at vpndienste.net.

I underlined the best VPNs I think you should use to avoid being traced back during your navigation. According to their Privacy Policies, they won't store any IP when you use their services; furthermore, their offerings (protocols, data, nation, tolerance and payment methods) are clearly outlined.

VPN Name	State	Data collected	Log IP	DMCA
AIRVPN	Italy	Personal information	✓	-

<u>BTGuard</u>	Canada	Personal information		-
Boxpn	Turkey	Personal information	✓	?
ExpressVPN	USA	Name Email Address Credit Card	✓	✓
HideMyAss	UK	Email Address Billing Information IP Address	✓	✓
iPredator	Sweden	Email Address	✓	?
<u>MULLVAD</u>	Sweden	-	-	-
<u>NORDVPN</u>	Panama	Email Address Username/Password	-	-

		Billing Information		
<u>PRQ</u>	Sweden	Username/Password	-	-
<u>Private Internet Access</u>	USA	Billing Information	-	✓
PureVPN				
Security Kiss	UK	Email Address Name Billing Information	✓	?
<u>SHADEYOU</u>	Holland	Username/Password	-	✓

TorGuard	USA	Personal information	-	✓
<u>OCTANEVPN</u>	USA	Personal information Email Address Payment information	-	✓
<u>SLICKVPN</u>	USA	Email Address Username/Password Payment information Google Analytics Temporary Cookies Webserver Data	-	✓
<u>SECUREVPN.TO</u>	Multiple	Personal information	-	✓
Steganos	Germany	Name Address Telephone Number	✓	?

VyprVPN	USA	Personal information	✓	✓
WiTopia	USA	Name Email Address Telephone Number Credit Card	✓	✓

Be particularly careful with VPN reviewing sites. They have the bad habit to create fake portals sponsoring their services and giving them 5 stars to alter any kind of result. Please, choose carefully and discuss with real people.

3.1.4.1. Multi Hop (cascading) VPNs

When a user connects to a VPN service, their Internet traffic is protected towards a single VPN. Multi Hop is a connection from a VPN from another VPN (and so on). Multi Hop offers huge benefits in terms of privacy and anonymity, ensuring different data protection layers as well as different jurisdictions for the inter-linked VPNs. However, “hopping” may cause significant slowdowns and I think no further explanation is needed. Otherwise, they exactly work as the direct connection VPNs (*client->VPN*) with the sole difference that one or more additional VPNs lay between the two (*client -> VPN -> VPN and so on*). Currently, the only VPN providers (that I found) offering this kind of solution are:

- NordVPN (<https://nordvpn.com>)
- IVPN (<https://www.ivpn.net>)
- Perfect Privacy (<https://www.perfect-privacy.com>)

3.1.5 Using the VPN

Using a VPN on any given Operating System is an extremely easy task, considering that all the major producers offer ready-to-use configurations to be passed to tunneling clients or, better, offer pre-made VPNs you can activate in one click. This applies to all Operating Systems, except the GNU/Linux scenario: the difficulties developing a single tool for the fragmented Penguin brand market convinced the producer to ignore it, only providing protocols connectivity. However, this problem offers an advantage: the Linux community can rely on a single client to manage all VPN connections, so we will have a single path to follow. In the test configuration, we are going to use NordVPN as the provider together with the OpenVPN protocol.

From the terminal, download and install the OpenVPN client:

```
$ su
```

```
$ apt-get install openvpn
```

Go to the program installation folder:

```
$ cd /etc/openvpn
```

Each provider offers a list of VPNs you can immediately pass to the client. Download this file:

```
$ wget https://nordvpn.com/api/files/zip
```

You have just downloaded a zip file (with no extension). Extract it using the unzip command:

```
$ unzip zip
```

Now all the files are extracted. View them using the ls command

```
$ ls -al
```

Once you chose the server to connect to, launch the openvpn command:

```
$ openvpn [filename]
```

for example:

```
$ openvpn it3.nordvpn.com.udp1194.ovpn
```

Enter the Username and Password Now you are connected to the VPN and ready to use the network tunnel. You can verify it downloading your IP in the network:

```
$ wget http://ipinfo.io/ip -qO -
```

To close the VPN connection, use the CTRL+C key combination. Then, verify your IP again.

3.1.6 Testing the quality of a VPN

Finally, you rented your VPN – or still using the free trial – but you’re uncertain about your choice, aren’t you? Well, you can’t be blamed, especially because you know there are very complex dynamics behind the Internet. For example, in case of a VPN misconfiguration, you will still hide your IP from the final site, but DNS resolution may be not encrypted, allowing your ISP to log your domain requests, compromising any encryption in place.

On the Testers we are going to introduce shortly, you will see some alerts related to JavaScript, Apple-X, Cookies, WebRTC, Java... all these vulnerabilities will be covered in a separate chapter, “*Local Resources*”.

3.1.6.1 Torrent Test

The tests we are going to run will check if VPN is working correctly, even with P2P protocols (especially Torrent). Unfortunately, you just cannot visit the usual “what is my ip address”; instead, you have to use the same Torrent client and a set of mini-hacks. Let’s drill down; firstly I wish to introduce you to other three web services offering that check:

- TorGuard (<https://torguard.net/checkmytorrentipaddress.php>)
- IPLeak.net
- ipMagnet (ipmagnet.services.cbcdn.com)

How to run the VPN test on Torrent

Get your preferred Torrent client first, then download a special .torrent file (or use a magnet link) and open it in the Torrent client (Figure 11).



Figure 11: the torrent is downloading

Now, each service has a specific way to run the test: in the case of TorGuard, you only have to download the torrent and view the active trackers page; to verify the outbound IP, you will see the VPN-assigned IP on the tracker status (Figure 12).



Figure 12: from the ipMagnet site, you can check how your IP is seen on Internet

The others work in a similar way, just refer to the instructions on each web page.

3.1.6.2 DNS Leak Test

There are different online services to test and verify any “leaks” between you and DNS. We already covered them early in the manual; if for any reason you still have doubts, go back and review those topics! In some cases, your operating system may still use the default DNS provided by the ISP, although your network looks 100% anonymous, thus utterly compromising your anonymity. You shouldn’t underestimate this problem: the normal IP retrieving services give a false sense of safety to VPN users, who are unaware that *hiding just the IP Address isn’t enough*. Plus, there is a second problem: imagine you’ve just changed your DNS using Google, OpenDNS, Comodo and whatnot. You may think your ISP cannot read your requests anymore. Well, that’s wrong. Some ISPs can re-read the DNS connections using transparent DNS proxies.

3.1.6.3 How to defend yourself against DNS Leaks

If you want to defend yourself against DNS Leaks from your ISP, you must set your system to use the VPN DNS or alternate DNS. Before going mad with your operating system setup, ensure your default VPN has not the DNS Leak Prevent feature available. The existing VPNs offering this service are quite rare.

- Mullvad (<https://mullvad.net/en/>)
- Private Internet Access (<https://ita.privateinternetaccess.com>)
- TorGuard (<https://torguard.net>)
- LimeVPN (<https://www.limevpn.com>)

- PureVPN (<https://www.purevpn.com>)

Currently, the software solutions are:

- VPN Watcher (paid / available for Windows, Mac, Android, iPhone, iPad / www.ugdsoft.com/products/vpnwatcher/)
- VPNCheck (paid / available for Windows, Linux / www.guavi.com/vpncheck_free.html)
- VPN Lifeguard (open source / available for Windows / <https://sourceforge.net/projects/vpnlifeguard/>)
- TunnelRat (open source / available for Windows / www.tunnelrat.net)
- VPNNetMon (free / available for Windows / vpnetmon.webs.com)

These programs check if the DNS match the specified ones and, in case of trouble, disconnect the Internet connection.

3.1.6.3 Kill Switch (protection against disconnections)

A Kill Switch (Figure 13) is an important – if not crucial – feature integrated with many VPN clients, allowing to cut off the network when the tunnel stops working. We can say it is some kind of network detonator, triggered when a VPN turns the tunneling off and is no more available.



Figure 13: Kill Switch feature, integrated with the NordVPN client

Without this feature, in case of VPN disconnection, your device will try to reconnect to Internet, leaving you exposed. You really should enable it, especially if you use *background* applications (e.g. when you download from Torrent) or if you need to go away from the device (ex. when a scan requires more time than expected). It is not easy to tell which *VPN provider* offers such solution; each calls the “Kill Switch” with a proprietary name, therefore I can only suggest you to make a deep search for each system and evaluate carefully.

4. Clearnet and Deep Web

So far, we only discussed about how to safely and anonymously navigate the Clearnet, the portion of Internet you can access through any device and search engine capable of communicating with TCP/IP protocols according the most common standards. During the years, however, Internet users needed to create a new kind of network, only accessible with the due precautions. Today, such network is known as the Deep Web.

Some people unconsciously believe the Deep Web is the “evil” part of Internet, conversely the Clearnet (or Surface Web) is the legit one. Truth is, Deep Web is the part the World Wide Web cannot index, a circuit accessible only with the due precautions (ex. using specific software). When, instead, we refer to the “twisted” world of arms/drugs trafficking and child pornography, the proper term is Dark Net (or Dark Web for web navigation). If you’re interested in this topic, read this interesting article^[23] and learn more about the related terminology.

Besides etymology, you shouldn’t underestimate the possibility of an alternative to the common Internet. Accessing the Deep Web may be useful, if not crucial, for tasks like engaging your coworkers, getting info removed from the Clearnet, obtaining exploits before the public roll-out and so on.

Ok, but why this whole premise? Now that we know the basics of anonymous navigation in the Clearnet (although we still have to further explore it in the next chapters), we will also cover the Deep Web, shortly, and how to engage with this particular world, considering each software/network.

4.1 TOR

Time to discuss about TOR^[24]: I am aware that some people is not missing that, and they may be right, since it's getting quite redundant! I'll try to make this part the least tedious possible, skipping the obvious things and getting straight to the point. Let's begin with a little review!

4.1.1 What's the TOR network

TOR is an anonymous network created to allow secure navigation and protect users privacy. The software is maintained by The Tor Project, an association funded by a U.S. governmental department for TOR network development and research. The project is represented by an onion icon, perfectly conveying how the network operates: TOR servers act like a router, building a virtual, private network, layered like an onion. Such stratification includes the following:

- Client: users
- Middleman: servers bouncing data in the network
- Exit routers: final servers on the chain, that "exit" towards Internet
- Bridge routers: similar to exit routers, with the exception that their identifier is private, allowing to bypass the block against TOR users.

4.1.2 TOR Projects

To facilitate TOR network access, TOR Project started developing different projects for many navigation scenarios, including:

- Tor Browser (<https://www.torproject.org/projects/torbrowser.html.en>): a package with a browser (Firefox), the HTTPS Everywhere plugin (forcing SSL connections), the NoScript plugin (blocking JavaScript) and, obviously, the Tor client. It's available both in installer and portable versions for all Operating Systems.
- Orbot (<https://guardianproject.info/apps/orbot/>): client allowing to connect to the TOR network and protect the traffic of all the apps on Android devices.

- Tails (<https://tails.boum.org>): a GNU/Linux distro designed for anonymous navigation, allowing to route connection to the TOR network. It also features encryption and anonymity tools.
- Arm (<https://www.atagar.com/arm/>): command line tool allowing to monitor and configure the TOR network.
- Atlas (<https://atlas.torproject.org>): web tool allowing to check the status of the TOR network relays.
- Pluggable Transports (<https://www.torproject.org/docs/pluggable-transports.html.en>): here, you can find supported third-party software designed for anonymity.
- Stem (<https://stem.torproject.org>): Python library allowing to interact with TOR.
- OONI (<https://ooni.torproject.org>): software used by governments to detect traffic manipulation and monitor our connection.

Speaking of Tor Browser, you should know that the legacy instances included Bundle (who remembers Vidalia and Privoxy?) and Browser versions.

4.1.3 TOR installation

Due to it's popularity, TOR is available in almost all existing repositories. In fact, you can use the command:

```
$ su
```

```
$ apt-get install tor
```

In Debian, however, we will rarely use the latest stable version; the Tor Project developers advise against using TOR in Ubuntu and related distros, since it's outdated and unreliable. As a best practice, enter the TOR official repositories directly to your Debian distro; firstly, use nano editor and open the `/etc/apt/sources.list` file:

```
$ nano /etc/apt/sources.list
```

Using Debian 8 Jessie, as recommended in the official website^[25], append the following lines to the file:

```
# TOR repository
```

```
deb http://deb.torproject.org/torproject.org jessie main
```

```
deb-src http://deb.torproject.org/torproject.org jessie main
```

save with *CTRL+X*, press “Y” and then *Enter*. You will be redirected to the terminal. In order to avoid any problem with file certification, you have to import GPG keys:

```
$ gpg --keyserver keys.gnupg.net --recv  
A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89
```

```
$ gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-  
key add -
```

Update your repositories, then install the TOR package:

```
$ apt-get update
```

```
$ apt-get install tor deb.torproject.org-keyring
```

Here you go! Now you’re ready to use TOR, which will appear as a local proxy listening to the 9050 port via SOCKS and the 9150 port for Tor Browser (we’ll cover that shortly). You can also verify the service status by typing:

```
$ service tor status
```

to stop it:

```
$ service tor stop
```

to start it:

```
$ service tor start
```

and to restart it:

```
$ service tor restart
```

To verify TOR operational status, we're going to use proxychains (see the Proxy chapter), configuring it to connect to TOR local proxies. First of all, check the actual location of TOR and the port listened:

```
$ netstat -tanp | grep tor
```

The netstat command allows you to obtain the entire list of active tasks using network resources; grep will allow to filter results only by the process you will specify. The | (pipe) operator concatenates the two programs. The expression will return 127.0.0.1:9050, where 127.0.0.1 is the local IP (our PC) and 9050 is the port being used. Before modifying the proxychains configuration, get back to the normal user:

```
$ exit
```

then, open the proxychains.conf file:

```
$ nano $HOME/.proxychains/proxychains.conf
```

and edit it as follows:

```
dynamic_chain
```

```
proxy_dns
```

```
[ProxyList]
```

```
socks4 127.0.0.1 9050
```

save using *CTRL+X*, the *Y* key and pressing *ENTER*. Note that we changed the strict_chain into dynamic_chain, because you may encounter non operational relays when using TOR. The dynamic_chain functions allows you to use proxy with more elasticity; strict_chain, instead, is strict to the point that it will block any modifications to the proxy structure.

Now, verify you current IP:

```
$ wget http://ipinfo.io/ip -qO -
```

```
82.51.116.171
```

alternatively, you can use a simpler command:

```
$ curl icanhazip.com
```

```
82.51.116.171
```

and compare it with the outbound one using proxychains:

```
$ proxychains wget http://ipinfo.io/ip -qO -  
ProxyChains-3.1 (http://proxychains.sf.net)  
|DNS-request| ipinfo.io  
|S-chain|-<>-177.73.177.25:8080-<><>-4.2.2.2:53-<><>-OK  
|DNS-response| ipinfo.io is 54.164.157.29  
|S-chain|-<>-177.73.177.25:8080-<><>-54.164.157.29:80-<><>-OK  
177.73.177.25
```

Of course, you can setup the entire system to pass all the traffic through the network-manager; alternatively, you can edit the `/etc/environment` config file as in the Proxy chapter. You should consider that, if you wish to use TOR for web navigation, you may need to use Privoxy, a web proxy service capable of changing HTTP requests, disabling ads and more. It is already integrated with TOR browser, and we encourage to continue if you need to navigate using TOR. Alternatively, visit the official web page^[26] and go to the dedicated FAQs.

4.1.4 TOR use cases

Once TOR is active in your operating system, you can use it in different ways. Here are the most common services and use cases.

4.1.4.1 TOR as a Browser

Perhaps, the Tor Browser Bundle is the most popular TOR Project. The browser is based on *Firefox ESR* and is pre-configured to connect to TOR internal *SOCKS* proxyserver at the `127.0.0.1:9150` address. It also comes with the following:

- *TorLauncher* starts the TOR network link in ghost mode;

- *TorButton* allows to control TOR client identities and settings;
- *NoScript* prevents JavaScript code to be executed (for more info, see the Local Resources chapter);
- *HTTPS Everywhere* forces the web connections to use the HTTPS protocol (see the Local Resource chapter again).

The client is available for Windows, OSX and Linux at the Tor Browser official web address^[27]; you can download three versions:

- *Stable*, la versione stabile
- *Experimental*, la versione nightly più aggiornata (ma meno testata)
- *Hardened*, la versione alpha del progetto disponibile solo per Linux x64^[28]

Installing TOR Browser

Windows and *MacOS* binaries can be launched with a double-click; on *GNU/Linux*, instead, you can have a little fun with the terminal to familiarize with it. Choose your preferred version for the available architecture and download it from the official website. If for any reason you are uncertain about which one to pick, always prefer the *32-bit* version. Once you downloaded the file, open the terminal and go to your downloads folder:

```
$ cd $HOME/Downloads
```

In our case, the file is “tor-browser-linux32-6.5a3_en.tar.xz”. We know it because we got the file list by using the command:

```
$ ls
```

Then, extract the compressed file:

```
$ tar -xvJf tor-browser-linux32-6.0.5_it.tar.xz
```

Pro tip: typing the name of a folder or a file every time can be quite annoying. UNIX-based terminals include an auto-complete feature: just type a portion of the name (ex. tor-), then complete it using the `[TAB]` key. Example:

```
$ tar -xvJf tor-[TAB]
```

The terminal will automatically complete the file name. The folder containing the executable will be extracted in tor-browser_en/. Access it with the command:

```
$ cd tor-browser_en
```

To launch the executable, you can use the start-tor-browser.desktop script. Run it with the command:

```
$ ./start-tor-browser.desktop
```

More about TOR browser

The TOR Browser Bundle can be used both in the clearnet and in the

deepweb. Besides its portability (you can use it via USB drivers or SD cards), this software conveniently features the pre-installed TOR core and TorButton (Figure 14), allowing you to handle connections without external GUIs (as it happened with the previous version). The entire TOR network, thus, is managed by the TorButton, by clicking the green onion next the browser URL bar.



Figure 14: TOR Button on FireFox

From the *Security & Privacy Settings* you can set four features already available in the Firefox preferences and use the Security Levels to choose from four user profiles, determining your “*paranoia*” level (Figure 15).

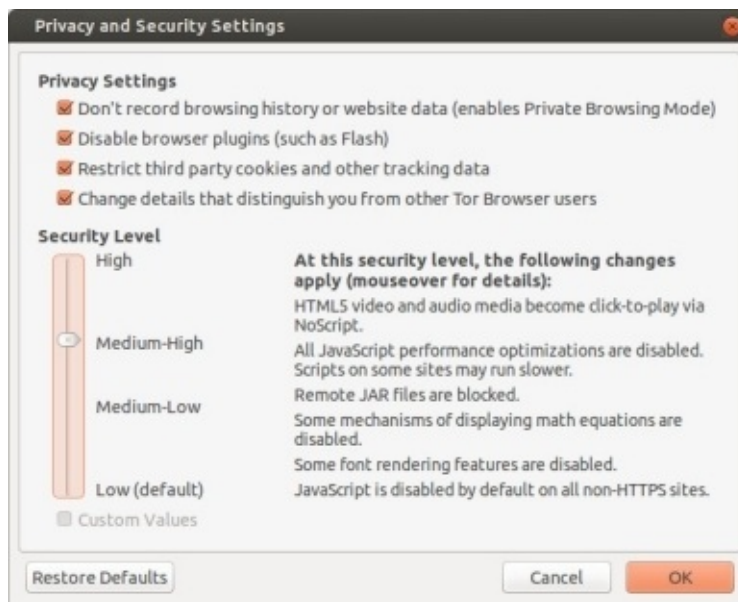


Figure 15: advanced settings in TOR Browser

4.1.4.2 TOR as a P2P

TOR Project advise against any P2P sharing^[29], with a particular reference to the most popular service, Torrent. There are two main reasons why you should never use Tor for P2P sharing:

1) Tor network cannot support bandwidth-consuming applications. If all Tor users shared files using the P2P technology, the Tor network would saturate.

2) The Torrent network may “sell you out”. As many other P2P networks, Torrent needs to pass your IP address to a public database, in order to connect you to trackers and then to peers. Therefore, the Torrent client may send your IP address directly to the tracker, thus exiting the Tor network for the download/upload stage and establishing a direct connection.

Actually, with proper precautions, you can use Torrent nevertheless, although it is not advisable. To anonymously share in the P2P networks, you should use VPN or I2P (we will discuss it later).

4.1.4.3 TOR as Chat

Services like Gmail, Hotmail, Skype, Facebook Messenger, as well as the old Yahoo! Messenger and MSN and any other form of communication over Internet can be tracked and stored for long times, even more than 5 years. We will also discuss how to encrypt messages within the network later; for now, let's only introduce the TorChat software.

TorChat^[30] is a decentralized and anonymous instant messenger that leverages the Tor network for Internet communications via the .onion meta-protocol. It allows to exchange end-to-end encrypted messages and multimedia. TorChat is natively available for *Windows*, *Linux* and next generation smartphones. You can also find unofficial version for OSX systems^[31]; use it at your own risk.

Installing TorChat

If you integrated the TOR Project repositories to install TOR, you can also install torchat. First of all, update your system:

```
$ su
```

```
$ apt-get update && apt-get upgrade
```

Note how we introduced the `&&` concatenation symbol. We can use it to run two discrete commands that shall not intercommunicate, unlike the `|` (pipe) symbol above. The `apt-get update` and `apt-get upgrade` update repositories and software in our system, respectively. At this point, you can install `torchat` with no efforts:

```
$ apt-get install torchat
```

After the installation, launch it directly from the terminal by typing:

```
$ exit
```

```
$ torchat
```

How TorChat works

In TorChat, each user has a unique alphanumeric ID with 16 characters. It is randomly generated by Tor at the client first launch and takes the form of a `.onion` address. Then you will obtain a code like this: `murd3rc0d310r34l.onion`, and your ID will be `murd3rc0d310r34l`. You can share it with other users who want to chat with you.

About TorChat security

The actual level of user security offered by *TorChat* is still a hot topic. A doubt arises from how the tool works: it creates a service within the host computer and simply transfers some data (just like `netcat`), exposing the computer to the same de-anonymization attacks already used in any other anonymous networks.

The second problem may relate to data transfer: there is no manual control over accepting a file transfer, and all the temporary part is written on the `/tmp` path: theoretically, an attacker may transfer random data to the Operating System `tmp`, causing a crash, since the OS is RAM-mounted. In the worst case scenario, we may also speculate a machine exploiting, after an overflow or other types of theoretically acceptable attacks.

The final critical issue is that everyone will always know when a *TorChat* ID is online, and you cannot prevent it. Then, if you want to end relations with other users, you will have to create a new *TorChat* ID. In conclusion, *TorChat* is a useful tool; however, you should use it only with trusted people and only when strictly necessary.

4.1.4.4 TOR as a Proxy Software

Just like Proxies and unlike VPN tunnels, you must configure your own tool to work within the TOR network. Once TOR is active, you can use an actual SOCKS proxy in your computer.

At this point, you can run your software, proxified with Proxycap (see the Proxy Servers chapter), connecting to the 127.0.0.1 address (or localhost) through the 9050 port. We already experienced this scenario when we installed and tested TOR (not TOR Browser), so please refer to the related paragraphs above to learn how to proceed.

4.1.5 TOR Relay

In the TOR universe, Relays give away free bandwidth to the network users. The torproject^[32] recommends TOR users to enable the Relay feature if they have more than 250kb/s both in upload and in download.

In the diagram showing the TOR elements list, Relays belong to the Middleman and Exit Node categories: anyone can run a Relay in their network and choose to act as a Middleman, an Exit Node or both. For the purposes of this guide, setting up a relay is not fundamental; if you wish to contribute to the TOR network development, however, you can create a personal relay.

4.1.6 TOR Bridges

TOR bridges – called *bridge relays* – are TOR network nodes that allow to bypass ISP and website filtering related to TOR network usage. To ensure the system works effectively, you won't find any complete list of bridge relays, since ISPs and websites honeypots would identify and block them at once.

You can instruct the TOR Browser client to use bridges, however, selecting “*My Internet Service Provider (ISP) blocks connections to the Tor network*”. Enable this option in *TOR Network Settings* (if you use TOR Browser, click the

top left green onion icon).

4.1.6.1 Bridges advanced use

If you wish to manually set your bridges, (e.g. to use Tor Expert Bundle, TOR-based Linux distros like Tails or TOR Browser through advanced configuration), you must firstly visit the Bridge page of Torproject (<https://bridges.torproject.org/bridges>), skip to step 2, complete the (impossible) captcha on top and then obtain a value like the following (***) have been added):

```
92.***.0.174:9001 65B2F8E594190A3*****59B0E32FC45720
```

```
194.***.208.26:27049 47063AFD4CB*****F16D6FE8DC68E6942DD6
```

```
107.191**.23:443 225A895211B179FDE2E8F8E3*****ECC0B0
```

You can launch TOR Browser and pass the newly obtained bridges (Figure 16 only).

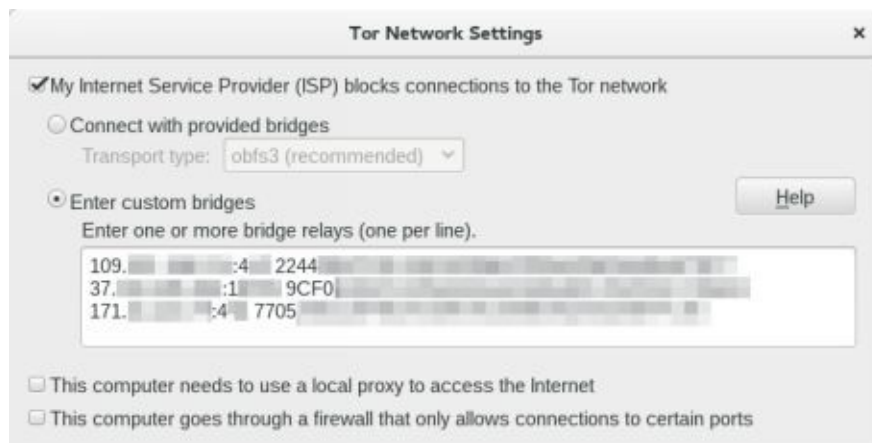


Figure 16: entering bridges on TOR

4.1.7 Pluggable Transports

Keep in mind, however, that bridges may be blacklisted as well, since everyone can access them, censors included. In order to bypass that check, TOR developers introduced a new feature, known as *pluggable transports*. PTs turn the TOR traffic flow into “clean” traffic between client and bridge that may otherwise be intercepted by the ISP with the *Deep Packet Inspection* (DPI) technique, classifying the IP traffic flows and, once the pattern is compared,

blocking them upstream.

At the moment, PT technology is under active development and requires operators and developers, in order to be efficiently integrated with the TOR Project. Learn more by visiting the official web page^[33]. The currently most common PTs are defined as obfuscated bridges, since they obfuscate traffic in order to make it hardly interpretable by ISPs. The underlying technology leverages algorithms, identified by protocols, that mix the inbound and outbound packets. There are three protocols of this kind: *obfs2*, *obfs3* e *obfs4*.

Obfs2 (version 2, also known as “Twobfuscator”) is the simplest one: the underlying technology allows to fetch inbound and outbound traffic data and resort them randomly. As shown by recent studies, this protocol can be cracked by intercepting the initial handshake (just like what happens with WEP security of WiFi networks), thus revealing the enclosed information. As a deprecated version, it’s out of development and unsupported by TOR.

Obfs3 (“Threebfuscator”) is quite similar to the previous protocol; however, it uses Diffie Hellman for keys swap during the handshake (we will explain this topic in “Encryption”).

Obfs4 is the fourth version of the protocol, although “it is closer to ScrambleSuite than obfs2/obfs3”, as its developer said. The latest version is seemingly the safest one, and is currently available in Tor Browser by default. You can learn more about the protocol on the official Github page^[34]. The Tor Project official page also includes an Obfs4 list^[35].

4.1.7.1 MEEK & Scramblesuit Protocols

TOR can communicate with many other protocols, besides the Obfs family* (Figure 17).

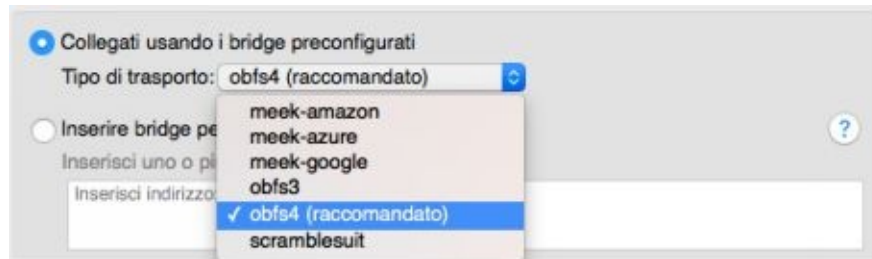


Figure 17: bridges selection on TOR

Meek-*

The protocols of the meek-* family have been created in 2014 to allow tunneling in a HTTPS circuit. Furthermore, a technique known as “*domain fronting*” hides TOR bridge communications to ISPs. As you can see, the meek-part is followed by a popular web service: if you choose Amazon, for example, the ISP will think you are communicating with the world-famous e-commerce (or with AWS cloud, more precisely), or Azure with Microsoft cloud and Google... well, with Google services of course. As explained by TOR Project, the meek-* based protocols are slower than the obfs-* ones and should be used only when the ISP blocks the latter. If needed, you can follow the TOR Project official guide; it explains how to configure your client to use this meek^[36]. In case of doubts, you can safely skip this protocol (or at least run some tests). Currently, they seem to be the only solid alternative in the occurrence of advanced censorship, like the late 2015 case in China; however, it is still an early version and the situation may change in years.

ScrambleSuit

The ScrambleSuit project – as reported on the Github official page^[37] – aims to solve two problems:

- Protect the user against monitoring attacks, requesting a “secret” shared by client and server and leveraging an out-of-band communication via BridgeDB (TOR bridge listing service).

- Protect from analysis attacks, alternating the data flow. ScrambleSuit can alter time and length of the package being communicated.

It was developed as an independent transport protocol from SOCKS protocols, e.g. HTTP, SMTP, SSH and so on. All the above should better explain how the Obfs4 protocol works as well, which is considered as faster and more stable^[38], so we recommend to use ScrambleSuit only when Obfs4 is unavailable. At the current stage, ScrambleSuit is no more under development.

The abandoned protocols also include *SkypeMorph*^[39], *Dust*^[40] and *FTE*^[41]. You can find the complete relays and P.T. documentation on the Tor Project official page^[42].

4.1.8 Testing the quality of TOR

In this part of the document, we will run some tests to check the TOR Browser safety.

As for VPNs, on the Testers via Browser you will see some alerts related to JavaScript, Apple-X, Cookies, WebRTC, Java... all these vulnerabilities will be covered in a separate chapter, “Local Resources”

4.1.8.1 TOR Test via Browser

The reference site for your tests will be TorCheck^[43], powered by xenobite. You can see the test results in Figures 18 and 19..



Figure 18: TorCheck without using TOR Browser

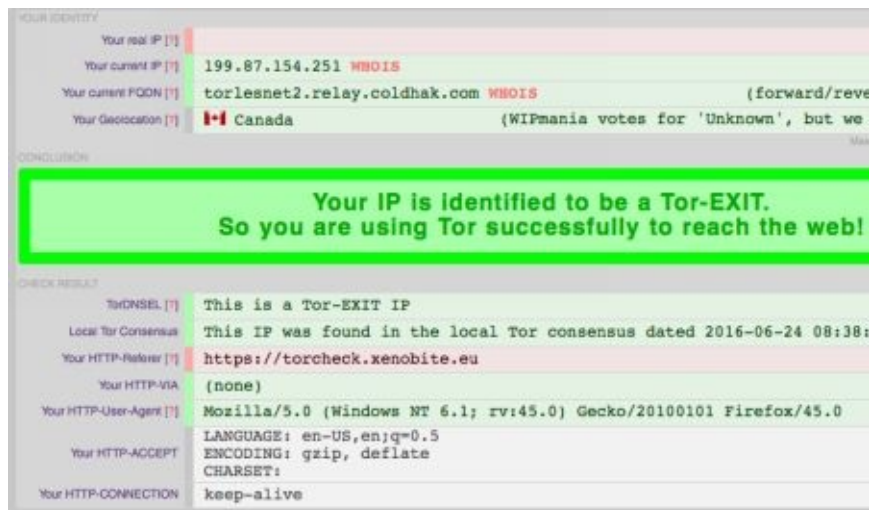


Figure 19: TorCheck using TOR Browser

From this page we get a report including different items and values. As you can see on the top screen, the fields with a *light green* background represent a good protection, the light red ones, instead, relate to issues to be fixed (please note, in this example you will receive the “Your real IP” item as red, probably due to a bug). Here’s the meaning of the items:

- *Your real IP*: your actual IP address. If you can see this, your security may be compromised.

- *Your current IP*: here you can see the IP address shown to the site you are viewing. If everything goes as planned, you will get a different IP than yours (it will be the exit node one).

- *Your current FQDN*: FQDN is the domain name specifying the DNS levels. That identifier warns you that your IP address is still logged by your ISP during the domain resolution.

- *Your Geolocation*: here you can see the geographical location obtained from the IP address. It's approximate and refers to the ISP switch, not to the real address of the connection user.

- *TorDNSEL*: here you can check if the "exiting" IP address is part of the Exit Node list. This is an important item, since it allows you to know whether the outbound connection has been manipulated or the exit node is identified as coming from TOR.

- *Local Tor Consensus*: no related documentation.

- *Your HTTP-Referer*: here you can verify if you are leaving Referer-type traces. The Referer value allows another website to see where the client comes from (ex., from a search, a site, a mail, etc.).

- *Your HTTP-Via*: shows the value informing the server about the type of request made via Tor proxy (ex., Via: 1.0 fred, 1.1 inforge.net (Apache/1.2)).

- *Your HTTP-User-Agent*: this is the lookup of your browser and operating system. HTTP-User-Agent can be manipulated, and we will see how to do it in the next chapter, related to Local Resources.

- *Your HTTP-ACCEPT*: here you can see the values accepted by your browser, in example information about language, cookies, cache, etc.

- *Your HTTP-CONNECTION*: reports the browser Connection value. Usually, you will find the keep-alive value here.

4.1.9 TOR and Deep Web

The TOR network is the most popular tool to access the Deep Web, or, better, the TOR Deep Web. Without it, your browser cannot resolve domains with .onion extension, e.g. websites hosted by servers and computers connected to TOR.

4.1.9.1 Where to find .onion sites?

Good question. Where do you go when you look for something? Google, of course! As I mentioned before, however, Google (as well as Bing, Yahoo and whatnot) is the black death for anyone wishing to stay anonymous. What to do, then?

The first step an aspiring deepnaut should take is getting the *The Hidden Wiki*, a Wikipedia-like page gathering some of the top .onion sites available. In order to find the Hidden Wiki, just googl... ehm... look up the keyword “The Hidden Wiki” on a search engine and get to some site – with a certain authority, if possible – so you can obtain a .onion address like the following: <http://zqktlwi4fecvo6ri.onion> (currently, it is the active one, but it may go down) or even websites in the clearnet.

About the Hidden Wiki: you can find many versions around. The most popular are the “ion” ones, although they are quite outdated. Alternatively to the official one, you can also find the “Mirror Version”, which is the most complete wiki. As a third choice, instead, you can get the HackBlock’s Hidden Wiki that can be updated by the community (careful what you look for, anyway). Dozens of wikis are created (and closed) everyday, therefore you have to be patient and use a good search engine.

4.1.10 Is the TOR network really safe??

In years, the TOR Project gained a certain popularity in the Internet world, and now is acknowledged as the premium anonymous network. Some newspapers acclaimed it as the “perfect tool for anonymous navigation”, slightly distorting the truth applicable to all software: “perfection doesn’t exist”

TOR is all but perfect: it’s still a piece of software, a program made by humans that can make mistakes. Furthermore, it has been violated: helped by external researches funded by the Department of Defense, FBI compromised the circuit between 01/30/20014 and 07/04/2014, monitoring hundreds of thousands of connections (and shutting off Silk Road too). One year earlier, FBI also arrested an Irish child-abuser by leveraging a bug in Firefox 17, the same version used by Tor Browser.

The latest attack was in 2015, when a university received 1 million dollars to sabotage TOR network^[44]. We will cover it shortly

About TOR: yay or nay, then? TOR is just a tool and can be very beneficial if you use it wisely. Of course, you have to master it, if you want to leverage its full potential.

4.1.10.1 TOR and HTTP protocol

As mentioned above, Tor Browser is included by default in HTTPS Everywhere, a Firefox add-on forcing HTTPS (HTTP + SSL/TLS protocol) connections to websites. *Why should you force the HTTPS connection?* Put simply, TOR is just a traffic router, and not a program capable of encrypting data in the network. The only task of the Tor network is to ensure the anonymity of the request source, encrypting the internal connection; however, encryption is not done outside Tor circuit. The latter operation is then executed by the HTTPS protocol, as long as the host site supports it (otherwise, it could be difficult to navigate it). Data interception attacks within a non encrypted network are called “eavesdropping”.

4.1.10.2 TOR and compromised exit-nodes

One of the highest risks you can take navigating the TOR network is finding a compromise exit node, the last node exiting to the Internet from TOR. Without proper precautions, the inbound and outbound traffic passing through an exit node may be not encrypted, meaning that an exit node owner (like a spy service) may monitor the network traffic. The mere connection to the TOR network, however, can't help identifying the request sender, since the intrinsic TOR structure prevents it (remember, TOR is built over multiple computer connections, this way the request source cannot be traced): however, you can identify clear data shared in the network, like personal information, emails, passwords, and so on. Exit-nodes may also re-reroute users to fake websites, in order to steal personal data from them, and this is one of the main reasons why you should always prefer HTTPS connections – in case of fake websites, you will get an incorrect certificate notification.

4.1.10.3 TOR Browser and the issues with “pre-built” products

The Tor Browser Bundle is developed by The Tor Project together with EFF, and is the first – and often the only – step to take if you wish to immediately interact with the Tor network. Using Tor Browser raises a problem, e.g. the nature of the bundle itself: because it's an All-in-One package, and then a Starter Pack, users may feel an illusion of safety, neglecting the next pages and thinking: “Hey! Why should I bother configuring everything from scratch? The Bundle is out there for that!” Do you know how the Freedom Hosting (the web service that hosted many darknet sites) guys got arrested? Because of the inner vulnerabilities of the Tor Browser Bundle. Word to the wise...

4.1.10.4 TOR, Google & CO.

In years, Google created a network across their services, capable of anticipating users demands and needs. Keep in mind that Google services are almost everywhere: Browser, Operating System (Android and Chrome OS), Account, Add-ons, Products and more. Once again, it's not impossible to stay fully anonymous using Google, but it's not recommended nevertheless: it would be better to use search engines that won't log any IP and search data, like DuckDuckGo or StartPage.

4.1.10.5 TOR is not idiot-proof

I beg your pardon for this part, but it was somehow necessary... How can one expect to be anonymous, if they purchase a new exploit on the Dark-Net staying connected to their Facebook account at the same time? No, that's not crazy at all, since it happens frequently: for example, some Tor users perform the two-step authentication of their accounts (using their mobile number!), some access their mail, some register using their personal information and so on. Now I will tell you a story of TOR abusing against Harvard University.

On December 18, 2013, a 20 years old man, Eldo Kim, was arrested. He was accused of having triggered a bomb alarm at Harvard University, in order to skip some final exams. For this purpose, Eldo used an anonymization software called TOR and a junk-mail service, Guerrilla Mail, that allows to create and temporarily send emails with no user data. TOR software worked successfully, hiding his operations both from the ISP and from the mail service, but not from his University. Righty-right, the good old Eldo made a mistake: he did all his tricks using the University WiFi connection, which can be accessed only by the username and password assigned to each freshman in order to prevent any abuse. A cross-check of the WiFi access data and the protocols and servers in use led to the identification of the guy, who later confirmed the charges. In that case, he was betrayed by his naiveness: he didn't realize or mind that he had to enter user and pass to access the network; like any other Hotspot, data are matched with a local IP address, which in turn stores any activity into the logs. The man was sentenced to five years of imprisonment, with a 250.000\$ fine.

I think this story is enough to make clear the message I wish to convey: it is not just about the "stupidity", you have to think of the possible lack of proportionality of the penalty to the offense. Just imagine the consequences of illegally purchasing something in the Dark Net or publishing an inconvenient message within a dictatorship where death penalty is still applied. Remember that TOR is no magic, it's just a program connecting many users to the same network. Whether you know the programming patterns or simply have a grasp of how it works, it's still a tool written by humans, and it can't guarantee full anonymity alone. Be thoughtful.

4.2 I2P

In the Internet world, you may often hear about I2P, the alternative network to TOR. By default, I2P won't allow you to navigate the clearnet, the "clean" part of Internet, being a project specifically developed to navigate within its own darknet. For this reason, it cannot be properly compared to TOR. First things

first. Unlike *TOR* requiring *Onion Routers* to survive, *I2P* (acronym of *Invisible Internet Project*) is a de-centered network, totally based on peer-to-peer technology. It's still in beta, but is consistently updated, with releases every 6/8 weeks; furthermore, developers consider it as a stable network, due to the high improbability of bugs. *I2P* is available and pre-installed in many *GNU/Linux* distros, including *iPrediaOS*, *Liberté Linux*, *Whonix* or the most popular *Tails*. Its worth mentioning that, unlike *TOR*, *I2P* won't force users to use the *HTTPS* protocol. The reason is that *I2P* already encrypts the connection, even before reaching the *HTTP*.

4.2.1 Using I2P

Being written in Java, *I2P* requires the Java Runtime Environment^[45], available for the most popular Operating Systems. Once installed, you have to launch a software executing the entire process, in order to connect to the peer-to-peer network. In most cases, you will only see one terminal, or nothing at all.

4.2.1.1 Installing I2P

Like *TOR Browser*, *I2P* can be easily installed on *Windows* and *macOS*, using the pre-compiled binaries. On *GNU/Linux* (Debian 8 “*Jessie*” in our case), instead, you have to add the official repositories from the *i2p2.de* site. To do this, edit the *sources.list* file:

```
$ su
```

```
$ nano /etc/apt/sources.list
```

Then, paste the provided sources in the file (if you use a different Debian version, paste the part with the proper codename, in this case “*Jessie*”):

```
deb https://deb.i2p2.de/ jessie main
```

```
deb-src https://deb.i2p2.de/ jessie main
```

Now you can update your repositories.

```
$ apt-get update
```

Something will go wrong, however. The reason is that you used the *https*

protocol, which is not in apt by default, as you can see in the above repositories. Before updating your software pool, install the apt-transport-https pack (as also suggested by the terminal):

```
$ apt-get install apt-transport-https
```

Now you can launch the update:

```
$ apt-get update
```

Here is coming another problem: you need certificates! Download them with the command:

```
$ apt-get install i2p-keyring && apt-get update
```

You will be prompted to confirm your choices in two moments; in both cases, press *Y* and confirm with *Enter*. Finally, install i2p:

```
$ apt-get install i2p
```

Like TOR, I suggest you not to use I2P as root. Logout from root:

```
$ exit
```

then, launch the service by using the command:

```
$ i2prouter start
```

4.2.1.2 First launch of I2P

Actually, the I2P is already 100% running; to verify, visit the I2P Router Console (Figure 20) at this address: <http://127.0.0.1:7657>. If you get a screen like the following, the I2P service is already running or, at least, the daemon managed to create a local web server, allowing you to use the I2P Router Console. The latter allows you to configure and monitor the status of the network. When I2P is running correctly, the Internet connection is “relatively” free, e.g., inbound and outbound rules, if any, must not be too restrictive. A generic firewall in the router is not a problem, but users in NAT will be negatively impacted (as for Fastweb users in Italy). Besides RTFM, I2P requires a couple of minutes to stabilize the p2p network pairing.



Figure 20: management console of I2P Network

4.2.1.3 Configuring a Browser with I2P

Once the I2P service started, you can configure your preferred browser to connect to services. To do this, point your browser to the following addresses:

- *HTTP: 127.0.0.1 (port 4444)*
- *HTTPS: 127.0.0.1 (port 4445)*

If you don't know how to change your browser proxies, please refer to the previous chapter, covering Proxy Servers. For the sake of clarity, however, you can see how to configure *Firefox* in Figure 21. This should be quite easy.

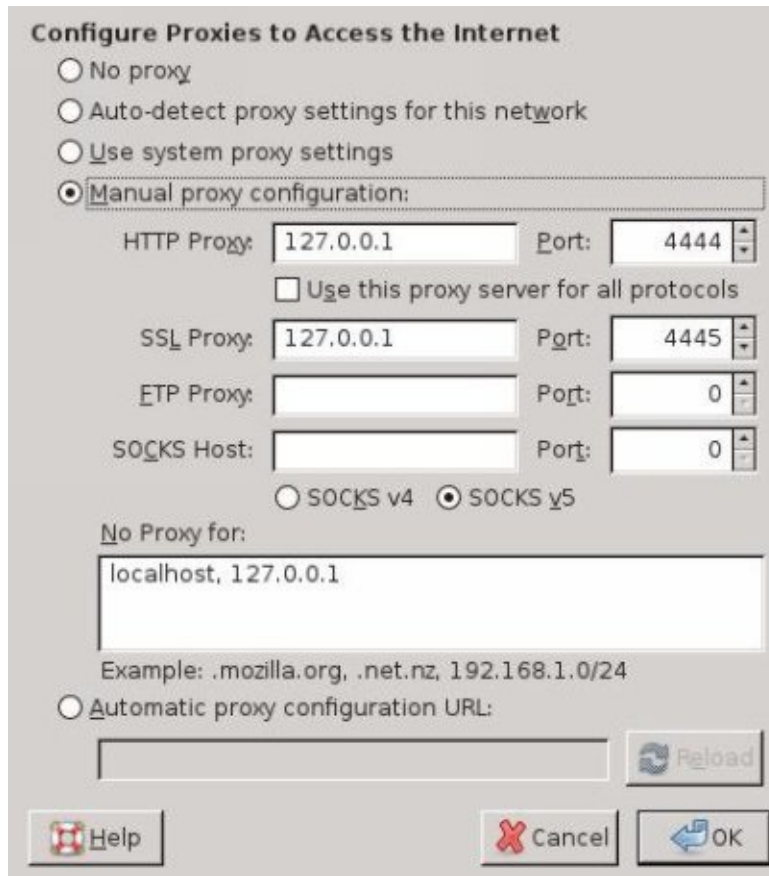


Figure 21: configuring Firefox with I2P

4.2.1.4 I2P useful resources

At this point, people always wonder: what now?

As we mentioned above, I2P will almost never allow to connect to the external network, being a “closed” or somewhat limited circuit in terms of services.

Eepsites

Eepsites are particular sites with the .i2p extension. They are only accessible via the I2P network (exactly like .onion nodes in the Tor network) and belong to the darknet world. We will cover that later in the “Deep Web” chapter. Please note, each user in the I2P network can immediately have their .i2p site for free and with no efforts. To do this, go to <http://127.0.0.1:7658>, to understand how to edit the site and request the official .i2p domain.

Irc2p

Irc2p is the name of the tunnel created during the first start of I2P. You can configure any IRC client and connect to the following address, then use it like an actual server. You can connect to I2P IRC server, connecting to 127.0.0.1:6668.

Blogging

The I2P circuit includes two eepsites offering microblogging services: <http://id3nt.i2p> and <http://jisko.i2p> . They are not responsive at the moment, however, and tend to load after some minutes.

Torrent

And what about an official client for downloading via torrent? Here you go! The project name is I2PSnark, and you can immediately reach it at <http://localhost:7657/i2psnark/>. Keep in mind that I2P torrenting can be as secure as terribly slow: the problem is caused both by the encryption process and the lack of I2P supporting peers. For this reason, you need to integrate the Torrent client with a dedicated tracker. The following ones may be interesting for you:

- <http://diftracker.i2p/>
- <http://tracker2.postman.i2p>

eDonkey

Perhaps the most popular I2P project, it has been deployed together with the I2P core in the latest versions, making it a stand-alone tool de facto. The reference eepsite address is <http://echelon.i2p/imule>. There's also a parallel project, Nachblitz^[46], acting as an alternate (but fully compatible) client to iMule, based on .MET Framework libraries and available for Windows and Linux via Wine emulation.

Mail

Susimail^[47] is a particular I2P service, acting as a mail interface for two mailing services (POP3 and SMTP) that can also be used with a local mail client. Email accounts are managed by an official router (hq.postman.i2p), offering mailing services both for dark and clear net. Such service allows you to have a mail address in the I2P network – username@mail.i2p – and to send and receive mail also on Internet – username@i2pmail.org.

NB: the service providers suggest to regularly check your mail, removing the unnecessary messages in order to make room for other users. Furthermore, according to its report abuse policy, your account may be removed, therefore you are advised to avoid any service abuse.

Learn more

This is just an approximate list of all services in the I2P network (it couldn't be any different!). The most important projects are listed at <http://echelon.i2p/>, including the development storyline of each project.

4.2.1.5 Anonymous navigation in Clearnet

Ok, before you think you're going insane, rest assured: I mentioned that I2P is a totally closed network... well it's not necessarily so. In years, the loving I2P community thought about a way to use their technology to anonymize navigation in clearnet; the most common method is using outproxies, e.g. routers – like TOR exit nodes – capable of connecting to the standard Internet as well. If you wish to try navigating the Internet with I2P, just follow the official website FAQs^[48]. Success, however, is not guaranteed (currently, there is only one very unstable outproxy); if you wish to navigate the Clearnet, maybe I2P is not the network for you.

4.2.1.6 Where to find I2P sites?

Unlike the TOR network, no Hidden Wikis are available, but you can find many search engines scattered across the Dark Net and the Clearnet. My suggestion is: use the clearnet to look for “i2p lists” with the search engines, then dig into the active eepsites (consider that many links may suddenly get broken). Furthermore, you can check a list of the active eepsites – the registered ones – in the I2P network, referring to the index available at <http://identiguy.i2p> .

4.2.1.7 Difficulties with I2P

In little more than a year, the I2P community seems to have left the project behind. Even one of I2P founders abandoned the Syndie and I2P projects, followed by many other developers and volunteers. The biggest problem was not only losing a programmer, but also the official i2p.net domain, announcing a bad ending for this project; however, the official geti2p.net team wants to remark that I2P is still alive, as proved by the new I2P releases^[49].

Currently, the project seems to be slightly resuming, but many eepsites – in particular the projects highlighted in the home section – have been abandoned; the official IRC channels is desert too. Perhaps one shouldn't complain too much, but it's something you have to consider if you wish to spend some time on this network. Another relevant factor may be the I2P foundation; it has been written in “Java”, a real CPU vampire. The group is at work on a fresh and better performing software, based on a I2P kernel recording in C++, namely the I2PD project^[50].

4.3 Freenet

Like I2P, the Freenet network is a peer-to-peer technology that leverages the users resources (bandwidth, space and processing power) to create an alternate communication tool with high security standard. It was designed 15 years ago, but has still much to say: just think that duckduckgo.com (the popular search engine considered as an anonymous alternative to Google) donated 25,000\$ to the project in May 2016. Freenet is not a proxying system for clearnet navigation, exactly as I2P, therefore we will only cover the Deep Web navigation. The Freenet project started as a free tool with no censorship; the first prototype was designed by Ian Clarke in 1999 and published for the first time in March 2000. It's the oldest project of the three, considering TOR and I2P. The system is entirely based on the peer-to-peer technology, then nobody can control what is

published and it's theoretically impossible to identify the author of a given content. The P2P network was created to connect users with their friends (if they wish to have a safer network) or geolocalize with other user (to have a faster and more stable network). Users (called nodes) are interconnected but nobody knows whether a message is actually sent by the sender or through an intermediary node.

4.3.1 Freenet installation

Freenet was created in Java, then if you wish to use it you must get the Java Environment Runtime^[51]. From *GNU/Linux* (Debian 8) install it using the command:

```
$ su
```

```
$ apt-get install default-jre
```

If you have been attending the course since the beginning, you should have already installed it, since I2P is distributed with the JRE installer. Now you can follow the Freenet official guide, as we used it to provide you with the recommended steps. Download the self-installing script and rename it as follows:

```
$ wget 'https://freenetproject.org/assets/jnlp/freenet_installer.jar' -O installer.jar
```

Launch it by using the command: If you are the root user, logout to a normal user with the exit command:

```
$ exit
```

```
$ java -jar installer.jar
```

An installation wizard will follow; proceed until you receive the successful setup message, and let your browser get completely loaded.

4.3.2 Configuring Freenet

The first launch of Freenet includes a short system configuration step. Here

you can choose between two pre-configurations:

* *Low security level:* you will be connected to random Freenet users. This options is the fastest. However, other users may monitor data traffic and identify you.

* *High security level:* you can only connect to friends already in Freenet. Although safer, this option requires a high users count in the circuit.

The further option is a custom configuration for your specific activity. You can also change options later, through the Freenet configuration panel. After the wizard prompts, you will be redirected to the initial dashboard, where you can monitor the network and use other features.

4.3.3 Using Freenet

Unlike TOR or I2P, the system is installed as an internal proxy through the following address: <http://localhost:8888>; you don't need to reconfigure your navigation browser. This way, the client is always on – if you wish to contribute keeping the network alive – and you can access it only when necessary, without using particular browsers. Go to the initial dashboard at <http://localhost:8888> (or <http://127.0.0.1:8888> according to your preferences) and you will see an initial list of links you can access to, just like I2P.

When you open a link, it may not load immediately, showing a timeout screen instead (Figure 22). The value shown is the remaining time before the page is resolved in your browser:

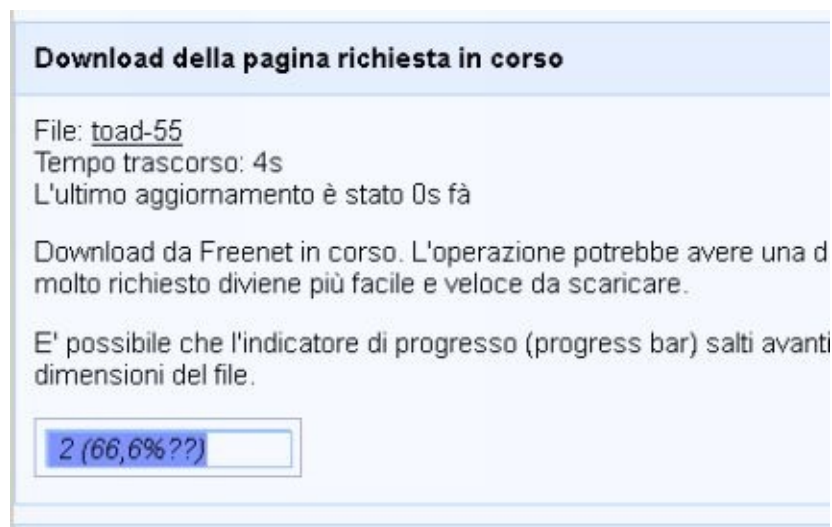


Figure 22: a page loading on Freenet

4.3.4 Freenet useful resource

The Freenet ecosystem is kept alive by the community, constantly creating new materials. Most of it is political propaganda, or whistle blowing against the abuse perpetrated by governments or public entities, however we cannot exclude the presence of marketplaces, pornography, and graphic content.

Freesites

Freesites are the essence of Freenet. They are user-created and loaded directly from the Freenet client dashboard.

How to create one is explained here: <http://localhost:8888/insertsite/>; you can also find tools like Sharesite and FlogHelper^[52] to facilitate the creation, while the list of the active sites is broken down to three tiers^[53]:

- Enzo's Index contains all sites, sorted by language, category, etc.
- The Filtered Index contains sites, excluding the disturbing ones.
- Nerdageddon mostly contains open source documents and informative sites.

Besides the above official ones, I wish to share some non official, but complete ones:

- Linkageddon is organized as Nerdageddon but without any site filtering.
- The Ultimate FreeNet Index is another index filled with freesites, you can choose the category.
- TPI: The Public Index, a website directory, autonomously managed by the community. The instructions to reach a freesite are available at the bottom of the page.
- AFKindex, a directory updated through Freenet crawling. Erotic and porno sites are filtered out.

Once loaded, the freesites stay within the network and are shared by peers, as long as they get regular views. If ignored for too long, they are automatically deleted.

Social Networking

Freenet has an internal suite of programs, allowing communications among users, something impossible with the static freesites, that allow only “one-to-many” communications. To learn more about these tools, visit the Communication page^[54].

Mail

Freemail is the official tool to communicate with others in the Freenet network. It is pre-installed in the Freenet package, but is not enabled by default. To enable, go to the plug-ins page^[55], select Freemail and click the “Load” button. This tool allows you to communicate only with Freenet user, or, better, those belonging to the Web of Trust. WoT is an additional plug-in (it must be enabled just like Freemail) allowing to have an identity recognized within the network. Once enabled, you have to setup a fist alias before you can use Freemail, then you will be able to create and use multiple aliases at the same time. In other words, Freemail can be used only by the registered users in the Web of Trust, and not outside the Freenet network.

4.3.5 Security in Freenet

We assume that Freenet is a very secure network, as long as it is used properly. Keep in mind that your privacy will depend on the number of “friends” you are connected to in Freenet: if you haven’t any, you will be connected to strangers. For this reason, Freenet can be configured for different security levers: you can change them in the dashboard menu under *Configuration -> Security Level*. The higher the level, the higher the network security (compromising speed).

Furthermore, Freenet has a second option, in case your computer is confiscated: from the Security Level page, you can set one of four encryption levels, from the weakest (no encryption at all) to the most paranoid one (everything related to Freenet is wiped at every restart). Personally, I recommend you to use the Medium-High level on Linux Live systems (we will cover them

later), while the Paranoia Level should be used only in Virtual Machines or on Computers you use frequently and consistently.

5. Combo Network

Now we are ready to discuss one of the hottest topics about anonymity: *TOR and VPN* – how to use them jointly? Or, better, *TOR via VPN* or *VPN via TOR*?

Conventionally, we will relate to TOR as an “anonymizing” network. We won’t cover the others, because their nature cannot be related to Clearnet.

Before we can answer such questions, we need to remember some key points: regardless of the use of TOR, you should always use VPNs not just to be anonymous, but rather to *protect yourself when you use an unsafe hotspot* (like airports, hotels, bars etc.) or if you’re afraid your home router is monitored. All networks, and especially the public ones, may be traffic-monitored. This reason should be enough to have an active VPN all the time, regardless of how much you feel safe about the hotspot you connect to.

Our navigation data, or at least the ones we want to safeguard (emails, passwords, credit cards and whatnot) must pass through a secure VPN, in order to avoid attacks in the local network. Also remember what we mentioned about VPN no-logs.

It’s true that you can do the same with TOR, as long as you take the proper precautions; it’s also true, however, that using the sole TOR network could probably make you a suspect. Of what, you may ask? Whether you are doing something illegal or not, the NSA/FBI/GCC and all the other governmental/private monitoring bodies want to know if you are using TOR or even just Linux. One of those blacklists is known as NSA watch list online, but we cannot exclude that others than NSA deal with this kind of business. How do they now if you are using TOR? They just ask the ISP. Then you must not only hide your activities to sites and online services, but also leave no visible traces to the ISP.

5.1 TOR via VPN

This configuration follows this pattern

Computer -> VPN -> TOR -> Internet

Assuming that your device connects to the Internet, when you use a VPN everything passing through the net will be encrypted, and from such tunneling you will connect to TOR routing, hiding the network access from the ISP. I mentioned no-logs above because, using a VPN you hide your activities from the ISP, but if the VPN doesn't enforce a serious privacy policy, it may store your activities, compromising all your efforts.

Connecting to TOR after the VPN tunneling, the VPN provider may know you're using the TOR network, just like an ISP. The latter may know you are connecting to a VPN, but not to the TOR network. Then, the VPN provider may store all the non encrypted activities. We may choose a TOR over VPN configuration for many reasons, including:

Pro

- You hide your TOR activity from your ISP, limiting the NSA watch list effect
- You can access the deep web (.onion addresses et similia)
- The TOR entry node won't get your IP but the VPN one

Cons

- You will tell your VPN provider you're using TOR
- If the exit node is compromised, your VPN provider will be exposed (but not your real IP), in this case, choose a really secure and logs free VPN
- Some TOR exit nodes reject connections from VPNs.

Some VPNs offer "TOR over VPN" services, simply performing a configuration in OpenVPN. If your VPN provider allows that, probably you can find a dedicated section in their website to learn how to do it.

5.1.1 How to perform TOR via VPN

As an Operative System-level solution, you can connect the Whonix Workstation to your VPN, while the Gateway will be already connected to the TOR network. We will discuss Whonix later, so if you wish, take a look at that part and come back when you're ready to continue.

Speaking of virtualization, you may perform this process with a Host computer which is completely connected to the TOR network (in this case, any Operating System will work fine) and connect to the VPN network on the virtualized machine.

As an interesting alternative, you may also use a hardware router like P.O.R.T.A.L. [\[56\]](#), allowing to directly connect a router to the TOR network, just like the connection we mentioned with the VPN. However, this method is only for the bravest.

5.2 VPN via TOR

This configuration follows this pattern:

Computer -> TOR -> VPN -> Internet

The user connects to the TOR network creating their secure network. From here, they connect to the VPN hiding the TOR exit node IP, making it appear just as a VPN user. If you perform a VPN network tunneling after you connected to TOR, you will let your ISP know you're using the TOR network. The latter creates a network apart, and won't make a direct tunneling to the VPN, which instead will be performed separately, thus informing the ISP that you are connecting both to TOR and to a VPN.

Pro

- You hide your identity from the VPN provider
- You can access websites that block TOR exit-nodes
- If you VPN is lost you will be protected by TOR nevertheless. You should always prefer a VPN with the Kill Switch feature anyway.

Contro

- You will tell your ISP you're using both a VPN and TOR
- You won't access the deep web (.onion addresses et similia)
- You will be in the NSA watch list spotlight

5.2.1 How to perform VPN via TOR

You can perform this type of connection semi-permanently with a router compatible with the OpenWRT^[57] or dd-wrt^[58] firmware, supporting VPN connections. For more information and a list of routers compatible with those firmware, visit the relevant official websites. Once you connected to the router, you may navigate with ease using your preferred Tor client. Once again, you can try some alternatives: you may want to use pre-configured (or self-configured) distros, allowing to use the TOR network only to make external connections; the correct procedure, i.e. on Tails, is described on the official page^[59].

5.3 TOR over TOR

On the TOR network chapter, we mentioned there is a very little chance that the exit node, namely the final "layer" of the network going to the clearnet, may track our actions. Although I may appear as paranoid, I want to introduce a method allowing to perform a TOR tunneling within another TOR network.

Even if this operation won't solve the exit node monitoring problem, it allows to change the game in the routing, drastically reducing the chances of identifying the request origin from the router itself.

5.3.1 Tortilla

Tortilla is a program which reroutes all TCP and DNS requests within the TOR node. The tool is deployed with pre-configured and independent binaries than the official ones, allowing to avoid any conflict. Using it with Tor Browser or a TOR standalone version, you can perform a double tunneling, exactly as we saw between VPN and TOR. The only “limit” is that it’s *only available for Windows*.

That won’t be a problem, however, since you can use Windows as a Host computer and run a Virtual Machine for your operations (we will discuss it a couple of chapters later). Tortilla is available in open source version through Github official channels^[60] and pre-compiled from the official website^[61].

It’s very easy to use: get the TOR Expert Bundle from the official site first, then install it on your Operating System (even better on a USB drive). Launch the tor.exe file and you’ll get to the command prompt; when the logs show Bootstrapped 100% Done, the connection to TOR circuit is complete, so you will be ready to launch the tortilla.exe client. Remember to run both programs with admin privileges; some Windows versions also require to enable unauthorized certificates – please refer to the Microsoft official guide^[62].

All the application methods for VPN over TOR circuits also apply to this type of configuration.

5.3.2 Is TOR over TOR helpful?

Personally, I think that connecting two cascading TOR networks won’t ensure any benefit in terms of privacy, compared to what you get from a good VPN. Using such configuration has to be considered as merely experimental and not in line with the anonymity quality standards ensured by other types of combo networks, therefore you should use it only for test environments and for educational purposes.

6. Local Resources

In this part of the document, we will cover *the Local Resources*, namely the pool of software and virtual objects on a computer that may somehow compromise the anonymity of a given web browser, a client and so on. When you use a browser to navigate the *web*, you may unknowingly leave thousands of data. Now we are going to provide you with an overview for each resource that may expose the final user to the digital unblinding.

6.1 Private browsing

Built-in in any next generation *browser*, the Private or Incognito mode is a special feature allowing you to temporarily disable all the data which may compromise your anonymity, such as *cookies, history, temporary files, sessions and saved passwords*.

6.1.1 How to enable the Private or Incognito mode

Each browser allow to enable the *Private or Incognito mode* using the proper combination of keys. For your convenience, the following list includes all the keyboard *shortcuts* for the most popular online browsers:

Browser Type	Shortcut (CTRL for Win/Linux, CMD for macOS/OSX)
Mozilla Firefox	CTRL + SHIFT + P
Google Chrome	CTRL + SHIFT + N
Opera Web Browser	CTRL + SHIFT + N

Safari	CMD + SHIFT + N
Internet Explorer	CTRL + SHIFT + P
Microsoft Edge	CTRL + SHIFT + P

6.1.2 What the Private/Incognito mode does (and doesn't do)

The Private/Incognito mode can be very useful to perform some operations without getting mad configuring menus and sweeping resources every time. For the sake of clarity, the Private/Incognito mode works over the following resources:

- Cookies: no previously created cookies are created; and the ones generated in Private/Incognito mode are erased once the mode is closed

- History: the websites you visit are not saved in the navigation history

- Cache: files stored in the PC are not used, and no file is saved in order to accelerate the page loading

- Extensions/Add-ons: no extension/add-on installed on the default browser is loaded (unless you automatically enable them)

As a benefit, you won't need to clear the history, flush the cache or remove cookies, and disable potential dangerous extensions/add-ons every time. The *Private/Incognito* mode hides some local activities, but you must remember that *it won't protect your data externally*.

6.2 HTTPS

Since we already covered the HTTPS protocol, we'll take just a quick look at it.

HTTP is a protocol specifically made to allow communicating information between server and client in the *World Wide Web*; HTTPS is a *HTTP* protocol using an encrypted connection through *TLS* or the older *SSL*. Connecting to a

website using HTTPS you safeguard the data passing through the network, preventing any possible spying actions through *man-in-the-middle attacks*. For now, then, you only need to know that HTTPS improves your safety and that failing to use it is a risk factor.

6.2.1 Controlling HTTPS protocols

In the web scenario, the top prevention software in this field is HTTPS Everywhere^[63], developed in partnership by *The Tor Project* and the *Electronic Frontier Foundation*. The tool is available for the most popular web browsers and is also distributed within the Tor Browser.

6.3 Cookies

In the IT world, a cookie is a text file stored within a computer. It is used by a web browser to keep track of data like logins, visited pages, user preferences (ex., graphics or language options) and so on. Cookies are created by a server using the HTTP protocol and can only be read or written by the relevant domain. A cookie is *composed of*:

- Name: an identifier to make it recognizable. This value is mandatory.
- Value: the cookie content. This value is mandatory (but can be blank, so that its value will be blank as well).
- Expiry: the life span of the cookie within the browser. This value is optional.
- Security: whether the cookie must be passed only using the HTTPS protocol or not.
- HttpOnly: whether the cookie must be passed only using the HTTP protocol or it can be also handled by client-side languages like JavaScript.

6.3.1 Cookies impact over security

Cookies are ranked according to the levels they can work on and to their technicals purposes. Considering the objectives of this chapter, however, you don't necessarily need to know them. What you have to know is that a cookie, among its many features, may also record analytics; in addition, the latter can also be created by *third parties* (the so-called third-party cookies). Furthermore, cookies may leave traces about the sites you have visited, since all websites virtually release at least one cookie into the browser (whether it is related to preferences, logins and whatnot).

A 2013 Washington Post article^[64] explains how the NSA is secretly monitoring the Internet users through third-party cookies, such as the ones stored when you visit a site that uses Google Analytics. If it is not clear yet, think about this: do you know when some ads seem to follow you or even get replicated on other sites? Blame it on the cookies (technically called *profiling cookies*) that store your interests and show you advertising campaigns in line with your persona.

6.3.2 Controlling cookies

As we explained, cookies are used to ensure that certain elements successfully work within the websites. Blocking them completely using the browser features is *inadvisable*, since it may cause website malfunctions or, in the worst case scenario, a lockout by the portal. *What to do, then?*

You can choose to use extensions/add-ons known as *cookie manager* that can block cookies per domain or just leave them on and use the Private/Incognito mode of your browser (see the previous chapter). You just have to keep in mind that third-party cookies may share data about the websites you visit. Such cookies may come from external analytics, advertisement and CDN services that could track the navigation.

Among the many cookie-blocking extensions/add-ons, one of the best is Ghostery^[65], which blocks all third-party *cookies* generating *scripts* upstream (you can find it on the relevant browsers stores). If it's not available for your browser, you can always use NoScript^[66] (see the JavaScript section).

6.4 “Special” Cookies

In addition to the list of regular cookie types in the web, other types are emerging from time to time, namely the proprietary cookies. *Adobe*, for example, created the “Local Stored Objects” (also known as “Flash Cookies”) which are embedded in Flash Player; *Mozilla* integrated the latest versions of *Firefox* with the DOM storage, allowing a faster rendering of web element.

6.4.1 “Special” Cookies impact over security

See “Cookies impact over security”.

6.4.2 How to block Flash Cookies

If you just can’t live without Flash – we will explain why it should be disabled later – you can deactivate the Local Shared Objects. To do this, you must set a value of “0” for the space that can be used by the LSOs. For more info, refer to the Adobe official guide^[67].

Once again: you’d better forget about Flash. If you can’t help downloading a video, just use some browser add-on, or a download manager like JDownloader^[68] and whatnot.

6.4.3 How to block DOM Storage

Disabling the DOM Storage on Firefox is quite an easy task. Type “[about:config](#)” on the browser address box, search “*storage*” by filtering the results; then click with the right mouse button on “*dom.storage.enabled*” and double-click to set it on “*false*”. You can also block Firefox DOM Storage using FireGloves (we will cover it in the Browser Fingerprint chapter).

6.5 Javascript

The JavaScript is one of the authorities of the web world. It is a scripting language, mainly used to fetch client events, or user actions (like hovering the mouse over a button, a live notification, a scroll, etc.), performing some operations the HTML alone won’t be able to execute.

Remember that *JavaScript IS NOT Java*: they are two discrete programming languages, and are used and work in a totally different manner. Without it, today

we wouldn't have any dynamic websites with live notifications and many features making the web faster and more attractive. Also consider that – according to a W3Techs^[69] research – 93.5% of websites use JavaScript, to date. A huge thing indeed.

6.5.1 JavaScript impact over security

Nevertheless, JavaScript can interact with user activities, i.e. it can gather what they type on a web page, working as an actual *keylogger*. Many *analytics/advertisement* companies, in example, use the JavaScript to analyze websites *keywords* and sell the most visited or interesting pages to their clients.

JavaScript allows to (partially) check if the user is using TOR and VPN, shows the browser plug-ins list, the installed font, your Time Zone (revealing your nationality), your user-agent (even if *spoofed* using a cross-check of pseudo-classes with CSS), pages history, some installed programs (like *OpenOffice*, *Adobe Reader*, *Microsoft Silverlight* and others) and other information.

Last but not least, the JavaScript can also be used as a “controller” after an attack defined as *XSS (Cross Site Scripting)* which allows an intruder to take possession of a web page and automatize some client-side operations (ex. copying cookies and sending them to another page) or reroute it to a fake login and fetch the access data.

6.5.2 Controlling JavaScript

We can identify the best choice for each browser. As usual, we will only cover the extensions/add-ons for the most popular browser:

- Mozilla Firefox: the most important extension for the red panda browser is NoScript. Such extension blocks JavaScript, as well as Flash, Java and any other external application. NoScript can intercept and block XSS and Clickjacking attacks as well.

- Google Chrome: unfortunately, the Google's counterpart cannot rely on the excellent NoScript suite; however, a really valuable alternative is available, [uMatrix](#), which is to some extent even more complete

- Opera Web Browser: once again, you can use the excellent uMatrix

- Safari: on the macOS/OSX browser, you can disable the JavaScript directly from *Preferences* -> *Security* -> *Enable JavaScript*

- Microsoft Edge: you can disable the JavaScript changing the Group Policies by using this path: User configuration -> Administrative Templates -> Windows Components -> Microsoft Edge.

6.6 Flash

Developed by Macromedia and later acquired by Adobe, Flash technology contributed to make the interactive media format accessible to all the Internet users in the last decade. Before going on, we must state that Flash is dead. Or, better, it is dying. According to statistical data, less than 1% of websites still use Flash in 2018, *Chrome* browser have disabled it since 2017, followed by *Firefox*, one year later. *Adobe* itself announced they will deprecate such technology in favor of *HTML5*, the new web standard. However, if *Flash* will end like *Windows XP*, we can surely expect to see this application installed at least for another 10 years!

6.6.1 Flash impact over security

Flash Player has been criticized by many researchers who deemed it as dangerous for users, instable and poorly performing. Since the latest version, released in January, there are hundreds, maybe thousands of vulnerabilities yet to

be fixed^[70], and this proves how it can become a time bomb when installed on a computer.

6.6.2 Controlling Flash

The only recommendation I can make is: *uninstall it completely*. If you need it, refer to the “Controlling JavaScript” section, since all the extensions/add-ons listed there can also block the *Flash Player*.

6.7 Java

Java is a popular programming language among the developers across the web (although if it recently lost some of its appeal, it gained new popularity thanks to *Android*) and has given birth to very good *web applications* for years. Recently, however, the HTML5 technologies and browsers in general are getting more and more popular, placing the Web Java as a niche language. It is still a valuable tool to date, of course, especially if used to fully leverage the hardware available in the market. Keep in mind, however, that most of the modern browsers are going to abandon it soon. Consequently, sooner or later Java may become deprecated in the web sector.

6.7.1 Java impact over security

The older versions of Java were under discussion, since it was impossible to setup a *SOCKS4/5* proxy externally, thus forcing users to completely disable it. The problem has been fixed in the latest versions with a new feature; hopefully, the development team will better document such new possibility. Nevertheless, we suggest you to completely disable the Java client, because a misconfigured browser may cause a *DNS leak* (which has been thoroughly explained in the VPN chapter).

6.7.2 Controlling Java

You can disable the Java client using the same tools already outlined in the “Controlling JavaScript” paragraph. However, if you still need it, we suggest you to use Orchid^[71], an experimental browser based on Tor Browser, which fully supports the Java libraries, even on Android devices.

6.8 ActiveX

ActiveX is an extension created by Microsoft to... extend the functionalities

of the Internet Explorer browser. Even if not too popular in Europe (differently from Far-Eastern applications, like IP Cameras), it permits to completely control the machine running it, allowing operations that can potentially compromise the whole user's system.

6.8.1 ActiveX impact over security

As you can imagine, ActiveX is an extremely dangerous tool, if used by criminals. Fortunately, it is not too popular, and has been deprecated by almost all the public services. You should keep in mind, however, that regardless of staying anonymous or not, an ActiveX can penetrate the host device and infect it with any kind of malware and trojans, compromising any anonymization effort.

6.8.2 Controlling ActiveX

Since we cannot know the nature of each single ActiveX application, we strongly discourage you from executing the ones with an untrusted source. If you run an application executing a separated Internet connection, you have to ensure that the entire System is configured to the external connection through *Proxy/VPN/Tor*. If possible, you should also verify the origin of each single application certificate (the digital signatures) and their integrity. In case of doubts, never allow the client-side executions (this option is only available on *Windows XP SP2* and later versions).

6.9 WebRTC

WebRTC is the new technology, established in 2011, allowing to make video chats with a browser, using the *HTML5* and *JavaScript* languages. Such technology is pre-installed on next generation browsers and OSs^[72] and can be currently used in services like *Firefox Hello*, *Google Hangouts*, *Skype (web version)*, *Facebook Messenger* and so on.

6.9.1 WebRTC impact over security

Being a relatively new technology (it's only 5 years old!), there's only a small number of *case-histories*. Actually, there is only one. According to a research conducted by TorrentFreak^[73], a remote site can leverage the WebRTC protocol to reveal the real IP address of a user, even if they are connected to a VPN or a TOR network. And it's not only limited to the public address, since it can also reveal the local one!

Are you paranoid? Well, you may be right; however, this vulnerability is (hopefully) leveraged only by a few portals. Nevertheless, try to connect to a VPN and visit the test address^[74]. If your real IP address is shown (whether it is the local or the remote one) – notwithstanding the VPN or other systems spoofing your IP – then you are vulnerable. You can further explore this particular vulnerability on the researchers GitHub page, which includes a proof-of-concept and a technical explanation of the attack.

6.9.2 Controlling WebRTC

To be quite honest, the WebRTC is not so good at all! Personally, I recommend everyone disable it directly from the browser, using extensions/add-ons like:

- WebRTC Network Limiter for *Chrome*^[75], ScriptSafe^[76] for *Opera* and *Chrome*
- Disable WebRTC Addon^[77] for *Firefox*

With *Firefox*, you can also disable the feature directly from the browser, just type “about:config” on the address bar, search the string “*media.peerconnection.enabled*” and double-click to set its value to *false*.

6.10 Browser Fingerprinting

All the technologies we covered so far have been analyzed to show how they can become a security problem for the user. Now we have to explain that all these technologies together form the so-called browser fingerprinting.

The term fingerprinting relates to a unique value that is assumed by the browser when the sum of all the related information take to a unique result. For the sake of clarity, imagine you can literally disassemble your *browser*. Each *part* belongs to a puzzle, and if such puzzle has an unique sorting in its structure, then it automatically assumes a unique identity; if you are matched to such identity, no *proxy/VPN/Tor* will ever protect you. But what those *parts* are?

6.10.1 Defining the Browser Fingerprinting

First of all, we must clarify that *fingerprinting* is an extremely complex operation, and is only performed by purpose-specific pieces of software. When

we navigate the web, our browser leaves a channel “open”, allowing any site to get the following information:

- Resolution, color depth
- Active plug-ins and the related versions
- Current time and Timezone
- WebGL Fingerprint
- List of fonts in the Operating System
- Current language
- Operating System and version
- User Agent, namely the browser and the underlying technology, and its version
- External devices, like a Touchpad
- Use of AdBlock
- ... and all what we have already discussed of.

You will be amazed by knowing the amounts of information we release over the websites we visit. If you wish, you can run a test on the Panopticlick site^[78], developed by EFF. Using Opera on a freshly formatted OSX 10.11.5, the result shows that the browser is unique across more than 139,000 tests (Figure 23).

Your browser fingerprint **appears to be unique** among the 139,655 tested so far.

Currently, we estimate that your browser has a fingerprint that conveys **at least 17.09 bits of identifying information**.

The measurements we used to obtain this result are listed below. You can [read more about our methodology, statistical results, and some defenses against fingerprinting here](#).

Browser Characteristic	bits of identifying information	one in x browsers have this value	value
Limited supercookie test	0.44	1.36	DOM localStorage: Yes, DOM sessionStorage: Yes, IE userData: No
Hash of canvas fingerprint	9.38	668.21	ab501a62bd46cb7a9ce878ad8ddd8940
Screen Size and Color Depth	4.51	22.79	1280x800x24

Figure 23: results of a conventional Opera browser on Panoptlick

6.10.2 Defending yourself from Browser Fingerprinting

If you accurately followed each single recommendation from the previous topics, your browser is probably quite secure. You can do more, however. The trick is changing the game, handling the aforementioned resources. Each browser allows some “covering-up”, such as changing the font list, disabling plug-ins, etc. However, this topic would require more than a single book! You can use some extensions/add-ons, though, for example:

- FireGloves^[79], available for *Mozilla Firefox*
- StopFingerprinting^[80], available for *Google Chrome*

6.11 File Downloading

This category includes all the files that are downloaded but, once opened, may reveal information about your online data. When you need to open any files, you should use tools like a *Virtual Machine* on a host computer not connected to Internet. Files downloaded from the Internet may contain executable code capable of communicating outside the anonymous network: for example, with the proper knowledge, some arbitrary scripting code can be inserted into *Word* or *PDF* files, not mentioning, of course, the classic executables available for your operating system (.exe, .dmg, .sh and so on).

6.12 Browser Security Test

Browser Security is a very complex and ever changing topic, and requires an extensive knowledge from multiple fields. Currently, the most complete and reliable tool to test your browser and its security is offered by BrowserSPY^[81], allowing to verify the existence, or rather the exposition, of any technology in the browser.

Using this tool is quite simple: each item on the left side of the screen will open a technology summary tab, and a list of values exposed to the network. You must ensure that all the items that may somehow compromise your anonymity are properly hidden, possibly exploring the ones that have not been covered in this document.

7. Data Security

If, despite all precautions, somebody is accused of a crime – something I would not want anybody to go through – all IT devices potentially leading to a crime may be confiscated.

Computer forensics is the IT branch that studies methods and approaches to find any data inside an IT device. Such field has been quite successful in recent years: just think about the number of cases solved thanks a phone call, a picture taken by a smartphone or recovered files from a criminal's computer. Furthermore, it has deeply changed and evolved: until a couple of years ago, everything was confiscated together with the computers: keyboards, monitors and mouse mats, and for no good reason!

Nowadays, labs and highly trained personnel are involved and results are often excellent. The forensic research practices may be used by law enforcement bodies – their actions are subject to the applicable laws – as well as by anyone skilled enough to perform them. As we will see, some of these skills can be easily learned and, except in rare cases, won't require any particular tool. In this part of the document, we will cover all the methods used to verify information and how to counter the *forensic research* eliminating any trace of your actions from the device you used.

7.1 Data Integrity

No matter how safe a connection is deemed, it cannot necessarily guarantee the integrity of the data passing through the network. With data integrity, we mean the original state of all information that can be sent and received: if, for example, we download a program from a developer and we need to be 100% sure that what we received is the same file or files originally distributed, we need to verify the program integrity.

Imagine you wish to download the .ISO of the latest version of Ubuntu, the popular GNU/Linux home distro: if it gets manipulated within the hosting server (i.e. if an intruder manages to violate Ubuntu servers and alter images using a backdoor) or, maybe, the download gets interrupted and you unknowingly use it for your daily operations, you may encounter several usage issues. Let's see a

very topical case-history about it:

On 20th February, 2016, the popular Linux Mint portal, currently the most appreciated Linux distro, as well as the latest release (Cinnamon 17.3) were violated. The attack on the disk image allowed the criminal known as *Peace*, to have the complete control over all users who downloaded and installed the .ISO during the previous 24 hours, through an IRC trojan called Tsunami.

If all users had run the data integrity verification (in this case over the .ISO), probably nobody would have been infected.

7.1.1 Checksum & Hash

In IT, Checksum is the sequence of bits resulting from a calculation over an information content. Such calculation is generated from a hash, a mathematical function that returns an alphanumeric value (namely, the checksum) in a non-reversible way: simply said, passing any information to a hash produces a checksum, (the result). This way, anybody can generate a checksum starting from a piece of information, but not vice-versa.

In addition, to be defined as good, a hash must be collision-resistant, e.g. it must produce unique checksums that cannot be applied to two different types of information. Due to their specific nature, hashes are commonly used in IT, especially in the scope of password memorization: when you enter a password on a portal, such password – as per security practices – is converted in the related checksum using a specific hash, in order to compare the user input with the password checksum in the database, avoiding any risk from storing it. Actually, passwords are “salted” first, but that’s another story.

7.1.1.1 Hash Types

In the IT world, you can commonly find three types of hash:

- MD5
- SHA-1
- SHA-2 (256 or 512-bit)

Each of them has its own characteristics, with pros and cons: for the purposes of this course, we'll only state that the safest ones to date are SHA-256 and SHA-512.

7.1.1.2 Calculating a Checksum

Often, in the macOS, Linux and BSD environments, you can find a very convenient command line tool, `shasum`. Use such tool as follows:

```
$ shasum [filename]
```

Just like the vast majority of UNIX programs, you can use different parameters to get the best out of it. If you wish to generate a checksum with 512-bit SHA, you must find the correct parameter in the documentation, by using this command:

```
$ shasum -h
```

or using the `man` command:

```
$ man shasum
```

Here you'll see that the `-a` parameter manages the algorithm type of "depth". Then, use the command:

```
$ shasum -a 512 [filename]
```

to generate the 512-bit hash. As the result, you'll get the generated checksum; feel free to try with your own files. Here's a sample output from a random file:

```
c568ac4df6aef33d887b0326c46d340196fe722f34d696bf7ab7ac9bd2cad933bdc9.  
stefano9lli.txt
```

Let's say you want to verify the latest Debian version (currently, 8.6.0 in standard version) you downloaded to your computer, in order to verify its integrity. Firstly, generate the local checksum:

```
$ shasum -a 512 debian-live-8.6.0-amd64-standard.iso
```

Now, compare it with the one provided by the developers on the official mirror^[82]. In this case, choose *SHA512SUM*, then find the portion of interest in the document:

```
e9506a3746e351203757599a8ce01ba4a84260a633177ee719fa6754b70151f8  
debian-live-8.6.0-amd64-standard.iso
```

If the two *checksums* are identical, you downloaded what the devs deployed. *Windows* users can perform this operation with an integrated software. The command is `certUtil`:

```
$ certUtil -hashfile [filepath] [algorithm]
```

then, in case of a random file on your Desktop through the *SHA-512* algorithm, the result will be:

```
$ certUtil -hashfile C:\Users\stefano9lli\Desktop\file.txt SHA512
```

Worthy of note is *Hashtab*^[83] (Figure 24), a freemium program that installs the checksum feature directly into *Windows Explorer*, integrating it in the "*Properties*" menu, when you right-click a file.

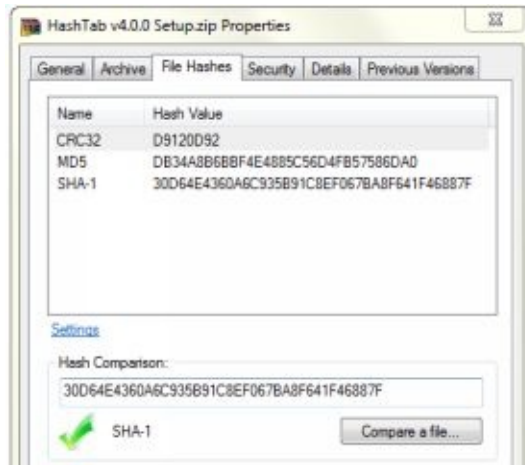


Figure 24: screenshot from the HashTab program for Windows

7.1.1.3 Checksum in common use

At the beginning of this paragraph, we explained how checksum ensure the integrity of a piece of information: this statement is true, however, as long as you're the only one who generated it or if the checksum origin for the counter-verification cannot be manipulated. The checksum, as we observed it, acts as a file fingerprint but doesn't guarantee the origin of the checksum you're going to use for the comparison. If someone manages to violate a website and the hosted files, they wouldn't mind too much changing the checksum in the page, would they?

Using hashes without any digital signature may be useful only to verify your personal data, since they can be only altered by external attacks. In this case, the owner will adopt the right measures to keep safe the original checksum and compare it when needed, but they cannot use it as a stamp or seal to guarantee the integrity verification of a file. For the time being, then, no tool is available yet to verify if what we want to download from the web is exactly what we expect – we will return to this topic on the chapter about PGP/GPG for data integrity.

7.2 *Data Encryption*

Now we should be informed enough about navigation and using anonymity tools on Internet. What we're still lacking is a good preparation of the workspace and a minimal knowledge of the tools we can use to leave no traces pointing to our activities on our computer. Imagine you are a Silk Road 3.0 (or the current version) user or a member of any community where the subscription itself could put you in serious trouble... you certainly don't want to be identified by anyone, aren't you? Keep in mind that NSA caught dozens of drug dealers and customers through their user names and passwords on Silk Road.

Curious is the fact that, after all the precautions taken, the computer formatted, *TOR* freshly installed, a brand new *Bitcoin wallet* and whatnot, one still gets caught, because their password contained their cellphone number. No joke. Since we already mentioned the secure protocols, we know the importance of message encryption. This operation is applicable both to the connection and the silent data, as well as to the messages we share with other users (friends, family, sellers, etc.).

7.2.1 PGP, Pretty Good Privacy

When it comes to data encryption, we cannot avoid mentioning PGP (*Pretty Good Privacy*) a tool that encrypts, decrypts and signs text, emails, files and directories to improve your documents safety. It works as follows: the user who wants to encrypt the message will create two keys – one public and one private. The *public* key allows anyone to send you an encrypted message, while the *private* one is the only key that unlocks the message created by the public key, allowing you to read it.

This is essentially the encryption behind most of the IT communications: the public/private system is also known as *Asymmetric Encryption (or Diffie-Hellman)*, while using a single key (using PGP anyway) is defined as *Symmetric Encryption*. If you lose the private key in PGP, consider the protected information as lost for good.

7.2.2 GPG, GNU Privacy Guard

The GNU Privacy Guard (from now on, GPG) tool suite is available for *Windows, macOS, Linux and BSD*. It was created as a free alternative to PGP, from which it inherited the *OpenPGP* encryption standard. Let's then consider the GPG as a free alternative to PGP, the software which created the standard used by GPG to work. Besides the *CLI version*, GPG^[84] is also available as:

- GPGTools^[85], a tool suite for macOS
- GPG4Win^[86], a client for Windows
- gpg4usb^[87], a version designed to run only on USB (Windows and Linux)
- ... and many more!

GPG is available by default in many GNU/Linux distros. If you prefer the UI mode, you can use seahorse (the same used by Tails). From now on, we will use the terminal quite often, since the UI is intuitive enough. All file operations can be done using the right button, then selecting the items available according to the situation. In case of doubt, you should first learn the command line procedure, then try with the UI mode.

7.2.2.1 Understanding the public/private key

We explained the difference between the private and the public key above, so

there's no need to reiterate it; that's enough to understand how they work. Summarizing:

- The private key must remain a secret, it's yours and you shouldn't share it with anyone.

- The private key must remain a secret, it's yours and you shouldn't share it with anyone.

To simplify, the relationship between the private and the public key is: *a public key can only be decrypted by the relevant private key.*

To simplify further, here's an example: *Andrea and Beatrice* are two friends wishing to exchange messages. However, they do not trust the communication channels, and decide to use PGP to text each other. In order to mutually encrypt and decrypt, they should have a common password, but the latter would have to be shared through a communication system they don't trust.

To solve this issues, PGP uses a type of encryption known as "asymmetric", where the messages are shared using public and private keys. *Andrea* has his own public key, as well as *Beatrice*. When *Andrea* wants to send a message to *Beatrice*, he will use her public key. If *Beatrice* wishes to decrypt that message, she will have to use her private key. Since *Beatrice* is the sole owner of that private key, she's the only one who can decrypt that message. Quite simple, right?

7.2.2.2 Creating your own PGP key

In this part of the guide, you'll learn how to create your public and private keys to allow other users to send you encrypted messages that only you can read. Assuming you're using Debian, launch the GPG GUI (Figure 25) launching the "seahorse" program from the Terminal, or more commonly, "Passwords and Keys" from the application list.

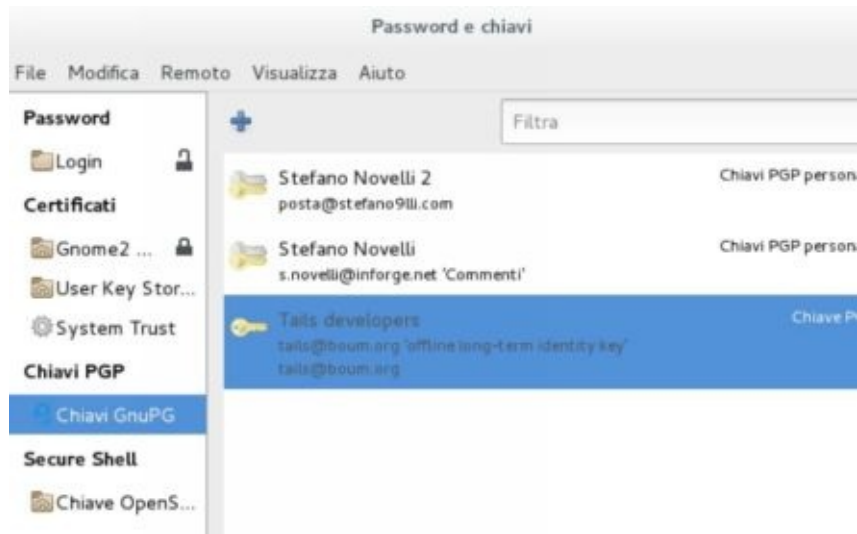


Figure 25: initial screen of the “seahorse” GUI on Debian GNOME 3

Now, click *File* -> *New* (or use the CTRL+N shortcut) and click *PGP Key*, then select *Continue*. Specify your *Full name* and *Email address*. From the advanced settings, you can change the *key type* (RSA is the recommended option) and the *encryption strength* (up to 4096, the strongest key you can use to date). You can also choose to assign an *expiration date* and an additional *comment*. You can now click *Create*. At this point, assign a password to your key. Then, you can see your newly created key under “GnuPG Keys”; conversely, you have to wait for the program to generate enough entropy for your key (I’ll explain it later). If you are a terminal enthusiast, you can do it from there too. In this case, run the command:

```
$ gpg --gen-key
```

You will be prompted to choose a key, then you’ll assign the key length and an optional expiration date. Just like the GUI version, you’ll now specify Name and Email and, finally, type the passphrase. Then, you’ll be prompted to “move around” with your keyboard and mouse: this operation allows to gather enough entropy to be associated to the key strength. Besides pressing random keys, you may want to do something to kill time, just like a Connect Four game!

7.2.2.3 Importing, exporting and revoking a PGP/GPG key

In order to send encrypted messages to other people, you must first import the recipient public key. The GUI dedicated button makes it a no-brainer: you

can find it on Debian under *File - > Import* or using the *CTRL+I* shortcut; if you wish proceed via command line, instead, just type:

```
$ gpg --import [filename]
```

If you wish to export a key, select *File -> Export*; alternatively, you can use the command line (replace [ID] with the User ID – the format is like AB1234567):

```
$ gpg --export [ID]
```

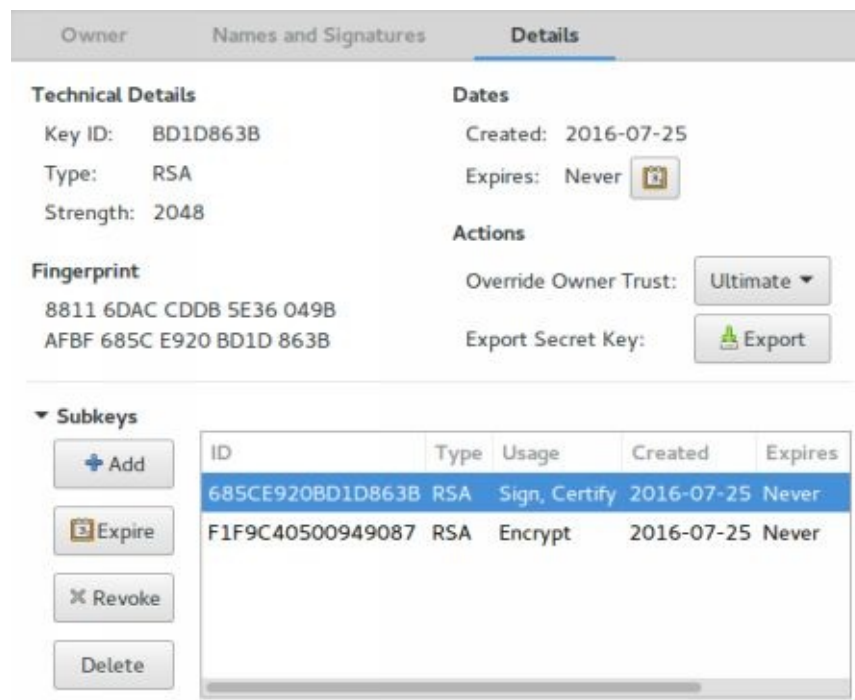
However, you’ll obtain an illegible output; you can format it in ASCII using the following parameters:

```
$ gpg --export -a [ID]
```

then, you can also save the output to a file, like the example:

```
$ gpg --export -a [ID] > my.key
```

Additionally, you can revoke a private key, in case you lost it or, worse, if it has been stolen. This item – as well as the next ones – is available under the “Details” tab for each key (Figure 26).



The screenshot displays the 'Details' tab of a GPG key management interface. It is divided into several sections:

- Technical Details:** Key ID: BD1D863B, Type: RSA, Strength: 2048.
- Dates:** Created: 2016-07-25, Expires: Never.
- Fingerprint:** 8811 6DAC CDD8 5E36 049B, AFBF 685C E920 BD1D 863B.
- Actions:** Override Owner Trust: Ultimate, Export Secret Key: Export.
- Subkeys:** A table listing subkeys with columns for ID, Type, Usage, Created, and Expires.

ID	Type	Usage	Created	Expires
685CE920BD1D863B	RSA	Sign, Certify	2016-07-25	Never
F1F9C40500949087	RSA	Encrypt	2016-07-25	Never

Figure 26: details of a GPG key on Seahorse

which translates into the command line as follows:

```
$ gpg --output revoke.key --gen-revoke [ID]
```

in order to generate a key revocation, populate the fields as required, then import the revocation certificate:

```
$ gpg --import revoke.key
```

If for some reason you synced your key with PGP keyservers, you must request the resync as follows:

```
$ gpg --send-keys --keyserver hkp://subkeys.pgp.net [ID]
```

Finally, update your keyring as follows:

```
$ gpg --refresh-keys --keyserver hkp://subkeys.pgp.net
```

Probably, you'll also need to have a list of all your keys. Use the command:

```
$ gpg --list-keys
```

If you prefer an easier way, however, *right-click* the key and select *Delete*.

7.2.2.4 PGP/GPG to encrypt and decrypt a file

The terminal command to encrypt a file is:

```
$ gpg --output secret.gpg --encrypt --recipient [your recipient] [file]
```

You can also send it as an attachment. If you can't do it, you may want to generate an ASCII-coded output, in order to send it as text, using the `--armor (-a)` parameter:

```
$ gpg --armor --encrypt --recipient [your recipient] [file]
```

The generated file will be `[file].asc`, containing the ASCII value of the text you have written. It will be like this:

```
-----BEGIN PGP MESSAGE-----
```

```
Comment: GPGTools - https://gpgtools.org
```

```
hQIOAwfq5Jrby+ZxEAf+N/ozNDVnsURxXb/lcKyPB/V4QuIGG5nQVAIZ5KO
```

```
[...]
```

```
pVhvtqu+q2yiE4khriBkpZD709uaf1kxfTaRosmRMI74duShAEQUuwjnyA1aOc'
```

```
-----END PGP MESSAGE-----
```

This way, you can send the content of an encrypted file without attaching it, but simply pasting it into a mail (however, keep in mind the file size, otherwise you may send dozens of MBs of text!). It's worth mentioning that, in this case, the `--encrypt` parameter acts as an identifier, telling the `gpg` program: "hey, you must encrypt now!". And what about decrypting? `--decrypt`, of course!

```
$ gpg --output [file] --decrypt secret.gpg
```

7.2.2.5 PGP/GPG for data signature

Besides encrypting messages, OpenPGP allows you to sign them: such signature acts as a sort of certificate, confirming the true ownership of who wrote the message. What is it for? Imagine you are sharing messages with an acquaintance on a forum: if the latter is attacked and their account compromised, you wouldn't know who is actually messaging you.

The fact they have your public key doesn't necessarily mean the user is who they claim to be: they may have gotten the key from the web, from your non encrypted messages history or from other sources. To demonstrate they are who you think they are, the other person will have to sign the message using their private key. Let's see what to do in this case.

The command to sign with your key makes use of the parameter `-s` (or `--sign`):

```
$ gpg -s [file]
```

Your file will be renamed with the `.gpg` extension. By default, the command

also compresses the file value, so if you wish to have a legible value, use:

```
$ gpg --clearsign [file]
```

It will be then saved in .asc format. If you wish to verify it, use the command:

```
$ gpg --verify [file]
```

The --clearsign can be appended with encryption values, so if you want to encrypt and sign a text document, you can use the command:

```
$ gpg -s --encrypt --recipient [your recipient] [file]
```

I know, it's quite a bunch of commands to remember, but I suggest you to make practice (perhaps using the gpg man) instead of using the GUI; you will familiarize with them shortly and you'll be more productive than using the graphical interface.

7.2.2.6 PGP/GPG for data integrity

In the data integrity chapter, we mentioned a problem that we haven't solved yet: how can you be 100% sure that a file is healthy and coming from a reliable source? Such doubt concerns the fact that a comparison between checksums – thus between results produced by hash algorithms – may be somehow altered within the hosting server.

With asymmetric encryption and, especially, the OpenPGP model, we can finally answer the above question: we will use the logic behind the public and private keys to ascertain that the source is totally trusted and the download file is perfectly healthy. Let's get back to Debian. First of all, acquire the file signature by downloading the SHA512 hash:

```
$ wget http://cdimage.debian.org/debian-cd/current-live/amd64/iso-hybrid/SHA512SUMS
```

and the related .sign containing the signature:

```
$ wget http://cdimage.debian.org/debian-cd/current-live/amd64/iso-hybrid/SHA512SUMS.sign
```

to avoid any conflict, import the key with the ID 6294BE9B (you got it from <https://www.debian.org/CD/verify>) from Debian GPG keyring:

```
$ gpg --keyserver keyring.debian.org --recv 6294BE9B
```

now you can verify the .sign (it must be named as the original file, in this case SHA512SUMS e SHA512SUMS.sign):

```
$ gpg --verify SHA512SUMS.sign
```

If everything is fine, you will receive the Valid Signature status:

```
gpg: Valid Signature from "Debian CD signing key <debian-cd@lists.debian.org>"
```

Now you have confirmed the hash is valid, so you can finally be sure that the downloaded SHA512 is a good counter-proof. If you want to test how the GPG verification works, you can edit the checksum file:

```
$ nano SHA512SUMS
```

perhaps adding a character at the end of the file. Save it and verify it again:

```
$ gpg --verify SHA512SUMS.sign
```

this time, you will get an error:

```
$ gpg: Invalid Signature from "Debian CD signing key <debian-cd@lists.debian.org>" [unknown]
```

Now you're ready to perform the checksum with the .ISO file (visit the checksum page to learn more).

7.2.2.7 PGP/GPG for email encryption

Email encryption can help you avoiding any connection monitoring (i.e., someone may read your mails passing through the network with no safety protocols in place), as well as preventing people from accessing your mailbox and reading your encrypted contents. Obviously, if you don't want to be localized, you have to stay away from mailing services allowing only the

Cleartnet access (Gmail, Yahoo, Hotmail, Libero, etc.); you should only rely to services allowing access from TOR nodes, Proxies and VPNs... Although it may sound quite obvious, we have to make a second consideration: never use the same key used for anonymity operations with your email in the *cleartnet*, or one you accessed at least once without the proper precautions. That would allow anyone with the right skills to identify you. There are different cases and tools allowing to use mail encryption. Here you will find some, right to properly kick start your search:

- Enigmail^[88]: Thunderbird and SeaMonkey extension; requires GnuGP already installed.
- Mailvelope^[89]: extension allowing you to use the OpenPGP encryption within webmails like GMail, Yahoo Mail, Outlook, etc. using Chrome and Firefox-based browsers.
- GPGMail^[90]: included in the GPG Suite, this tool allows you to encrypt from the Mail program on OSX
- APG^[91]: available for Android, allows to easily integrate GPG with files and mails
- SecureGmail^[92]: extension that allows to protect mails from the GMail services and all GApps-based systems

Among the Debian repositories, you can find Icedove^[93]: created by *Mozilla Foundation* and re-branded in *Debian style*, this mail client is based on Thunderbird. In order to install it, use the command:

```
$ su
```

```
$ apt-get install icedove
```

then install the Enigmail extension too, in order to use GPG in Icedove:

```
$ apt-get install enigmail
```

Once launched, you can activate the PGP signature and encryption by simply clicking the “*Enigmail*” button, usually on top of the mail header (Figure 27).



Figure 27: using Enigmail within the Icedove client on Debian GNOME 3

Now that you learned how PGP/GPG works, I wish to remark the importance of this tool examining a recent fact.

During the *Silk Road* takedown, it emerged that many admins, including the famous creator *Ross Ulbricht*, didn't encrypt their communications. When Ross was arrested, detectives found hundreds of files in his computer. According to *Silk Road 2* users, such files included documents with personal information about admins and moderators, saved as text files, which helped the investigators to identify his accomplices.

7.2.3 Where to store the PGP/GPG keys

After all that paranoia, it would be ridiculous if you left the encryption keys (and other important files) nicely exposed in your computer, don't you think? Here's why, if someone gets to pay you a visit, you have to be ready to hide, or destroy – in the worst case scenario – a removable media like a USB drive or a SD card, if your computer allows you use it (even better if a micro SD with SD adapter).

Using a micro SD may be the best way to hide your keys; such card is so small you can hide it between your fingers, inside your ear (why not?), in your shoes or in your underwear. Just use your imagination! Anyway, we will review how to store keys and other information in depth in the “Data Backup” chapter.

Conversely, hiding or destroying a USB drive is not that easy; besides the concealing difficulties, it is technically more difficult to destruct. SDs, instead, can be cracked with no effort, making any content illegible. To have a deeper insight of this topic, skip to the “Data Shredding” chapter.

7.3 Disk Encryption

Data encryption can be applied to an entire disk or part of it; this feature is available on any operating system.

- On Windows, it's the BitLocker technology.
- On macOS/OS X, you need to create an encrypted partition.
- Usually, on GNU/Linux the feature is called Whole Disk Encryption or Full Disk Encryption

Just like the client-level encryption, if you lose your key (passphrase) there are no chances of retrieving it; you can only format the disk and install a new OS. Since you may need to encrypt portions or entire partitions to be used across multiple Operating System, we will focus on a cross-platform software, which is System-agnostic and capable of managing different technologies.

7.3.1 TrueCrypt

TrueCrypt wrote part of the IT encryption history, being the first-born of a software lineage which introduced the average users to the full disk encryption without being IT engineers. This project has been abandoned in 2014, with the end of Windows XP support, but it's still available for who needs the related program versions. Luckily, we can use an array of forks which resolved its restrictions and issues, but I wished to dedicate a paragraph to the possibilities offered by *TrueCrypt* and its new iterations:

- Allows to encrypt entire partitions
- Allows to create two partitions: if you are forced to unlock the disk, you can access one partition using one password, while the second partition will require another password.
- Supports the hardware acceleration offered by the next generation CPUs features

- Supports three encryption algorithms: AES, Serpent and Twofish. You can also mix them.

Before we proceed, some general caveats and recommendations; they won't apply for all cases, but you must consider every possibility:

- Never defrag and index encrypted volumes, since they could leave traces in the system logs

- Never use journaled file systems, instead choose file systems without journaling (ex., FAT32, exFAT or ext2)

- Use the complete formatting, non use any fast option

- When you open files, remember they'll be stored in the RAM and in temp folders: remove them once you're done (especially when you are not using a Live OS)

- In any case, take all the necessary precautions to protect the device you're using (no Internet, updated system, protections in place, etc.)

7.3.2 Veracrypt

Available for all the major OSs and backwards compatible with Truecrypt, Veracrypt^[94] is without any doubt the go-to application. The program is quite user-friendly and easy to use.

7.3.2.1 Installing Veracrypt

Just for the other topics, we will skip the installation stage on Windows and macOS, since it's quite a no-brainer; installing it on GNU/Linux, however, is not very hard but we recommend you to strictly follow these steps. Firstly, download^[95] the tar.bz2 file for the GNU/Linux version (currently, v1.19). Open the terminal and go to the Downloads folder:

```
$ cd $HOME/Downloads
```

extract the package using the command:

```
$ tar xjf veracrypt-1.19-setup.tar.bz2
```

Now, you will have 4 files: identify the ones containing the term “gui” in the name. Then, install the program:

```
$ su
```

```
$ bash veracrypt-1.19-setup-gui-x86
```

Choose between the x86 and x64 version. If you don't know your Operating System architecture (remember, x86 means 32-bit processors, while x64 means 64-bit), you can use this command:

```
$ hostnamectl
```

If you still have doubts about which one to install, always prefer the x86 (it's compatible with x64 processors, but slower).

7.3.2.2 Using Veracrypt

Let's how we can use Veracrypt, creating a partition or a container to store all our most important files, hidden from everyone else.

- 1) In order to create the first container, choose one of the available slots, click the *Create Volume* button, then choose between *creating a container* and *encrypting a partition*. It's up to you to decide which is the most convenient situation.

2) The second part brings another dilemma: standard or hidden encryption? In the first case, if you don't use an unlocking passphrase, you won't be able to access the folder/partition; in the second one, you may use a second *passphrase*, replacing the main one, so that, if you are forced to decrypt the partition, you may decide which passphrase (and then which partition or container) to expose. However, it would still be possible to compare the visible files size with the available space in the partition, and understand this is a hoax, but it's better than nothing anyway.

3) (Optional) If you choose the container and not the partition, specify where to store it.

4) Now, choose the algorithm type. You can use the available configuration. In case of doubt, please refer to the "Encryption" chapter.

5) Choose how much space to allocate to the container or volume.

6) Now, you have to choose a password. If you chose the Hidden option, you will have to complete this step twice: the first one for the "fake password".

7) (Optional) During the creation, you will be prompted if you wish to enable the *PIM*, a multiplier introduced in the latest Veracrypt versions, allowing to specify a numeric value for your combination (it must have a minimum of 20 characters, though). The PIM multiplies the chances of finding a key: if you specify a PIM value between 1 and 485, n^{485} is the probability of finding the password, meaning that you will multiply your password security by 485. This measure could be excessive, considering that a 24-character password that was encrypted in 512-bit AES can be forced, but this operation would require the same lifespan of the universe. Just saying...

8) (Optional) You can also use a keyfile, in other words an auto-generated file containing a password. Such option may be safer than a normal password, since it would contain any kind of character, not limited to the keyboard layout charset; furthermore, it protects against keylogging attacks.

9) Now it's time to choose the partition type. In order to know which works best, I recommend you read the drawbacks under the "*Truecrypt*" chapter. Personally, I always prefer an exFAT file system, since it's quite compatible with my systems; feel free to choose the best fit for you anyway.

10) Choose if you wish to enable the Cross Platform support option. This can be useful, for example, if you're using macOS and you wish to minimize the volume risks.

11) If not already shown, you'll find a loading bar moved by your mouse. This feature allows you to generate a random encryption key, based on the erratic mouse movement. The more you move the mouse, the harder will be to crack the encryption key.

The volume is now *created*. Proceed to mount the drive or file, according to your choices:

- 1) Select an empty slot
- 2) Choose Select File or Select Device
- 3) Click *Mount*
- 4) Type the *passphrase* or load the *keyfile*

5) Enter your user account password to let the System generate a new virtual partition

If you're following this guide using Debian, you'll get an error and won't be able to proceed, because your user is not authorized to create a virtual partition. To do it, you need to add your user to the sudo users list. Run the following commands from the Terminal:

```
$ su
```

```
$ visudo
```

Add the user to the end of the file (in this case it's stefano9lli, but change it according to your own):

```
stefano9lliALL=(ALL) ALL
```

(Keep in mind that the first space is assigned by the [TAB] key of your keyboard, while the second is a normal space.) *Save using the CTRL+X combination, the Y key and pressing ENTER*. This way, you can use your user as

an admin on Veracrypt as well as in any other situation! Repeat the volume mounting steps. As you can see, a new partition will appear: here you can safely store all your files. Other users will not be able to see them, unless they know the encryption password. When you're done, you can dismount a volume by clicking the Dismount button from the program.

7.3.3 Zulucrypt, LUKS and family

In the GNU/Linux world, a new encryption tool, Zulucrypt^[96], is gaining popularity. Its strength is the support of TrueCrypt and VeraCrypt-created formats, as well as *LUKS*, a referential hard disk encryption method in the Linux world. *LUKS* is considered as a standard in the Penguin environment, therefore you need to know its existence and, possibly, how to engage it: in the Windows environment you can find an adaptation provided by the FreeOTFE tool^[97], while on *OSX* you could once find *OSXCrypt*, which is seemingly abandoned by now. Going back to the Linux environment, the *dm-crypt* module provides support to the LUKS encryption and is available in almost all GNU/Linux distros, so you shouldn't have problems using it. However, *dm-crypt* is quite hard to use for a novice user; in this case, it would be easier to use a tool called *cryptsetup*, which supports the LUKS method via the *dm-crypt* module. A good knowledge of partitions, mountpoints and general GNU/Linux commands (they may change according to the distro families) is required to use it, so we recommend to read the official *cryptsetup* manual^[98].

7.4 Steganography

The Steganography technique is used to conceal messages within containers that may appear harmless at first glance: already in ancient Greece, Herodotus writes how Demaratus the Spartan used wax-covered tablets to alert the cities nearby to a possible Persian invasion. When messengers were caught, the enemy spies would find wax tablets with messages on them, unaware that the original message was underneath.

7.4.1 Steganography with LSB method

In IT, the most common steganographical approach is the technique known as LSB (*Least Significant Bit*), based on the theory that a small portion of a large image, video or audio file can be altered to conceal other data.

Imagine you have a large 1920x1080 picture, containing 2 millions pixel. Who would ever imagine that one of them hides a secret message? You would need to zoom the picture *pixel by pixel* and know the exact location in order to identify it. Additionally, most of the *steganographic* tools use reading algorithms to “break” a pixel, choosing a non significant one that would not be prominent at the center of the image. This brings a bigger problem, if you consider that even an expert eye may be deceived.



Such kind of programs integrate in turn a set of ciphers to further encrypt the message, so that no scanning software may decrypt the content (perhaps a dictionary attack would make it). However, this method is not exempt from attacks: *steganalysis* focuses on running statistical tests to verify the presence of messages within image/video/audio files. Therefore, we must consider Steganography as vulnerable as all other defense methods. Furthermore, keep in

mind that the image must circulate as it is: resizing or optimizing it would compromise the internal data for good. If the image is partially visible (i.e., due to a failed buffer), the content would never be legible.

7.4.1.1 LSB Steganography Tools

Different tools are available in the IT Steganography world – here are some:

- SilentEye (silenteye.v1kings.io), available for Windows, Mac and Linux, is perhaps the best UI Steganography tool. *Free*.
- OpenPuff (embeddedsw.net/OpenPuff_Steganography_Home.html for Windows) is a nice tool offering Steganography in different image/video/audio/flash formats, as well as 256-bit key encryption. It also features a randomic algorithm based on the user's hardware *OpenSource*.
- Outguess (www.rbcafe.com/software/outguess/ for macOS) allows to conceal messages into JPG images. *Free*.
- iSteg (www.hanynet.com/isteg/ for macOS) is the GUI for outguess 2.0, which allows you to conceal messages into images. *OpenSource*.
- Camouflage (camouflage.unfiction.com for Windows) allows Steganography within images and Word files. It also provides messages encryption. Unfortunately, the project is abandoned. *Free*.
- Outguess Rebirth (www.outguess-rebirth.com for Windows) allows Steganography into images. It may be transported into external memories and offers encryption options. *OpenSource*.
- MP3stego (www.petitcolas.net/steganography/mp3stego/ for Windows) allows to conceal messages into mp3 audio files. However, the development has been abandoned since 2006. *OpenSource*.
- QuickStego (quickcrypto.com/free-steganography-software.html for Windows) is a simple program capable of concealing messages into images with .bmp output only. *Free*.

We also want to list StegHide, which we're going to learn and use in the coming lines.

7.4.1.2 Steghide

StegHide is a convenient tool developed for Windows and Linux. The last release dates back to 2003. Although you can find much better and updated alternatives – like SilentEye – StegHide is a good tool to operate within a test environment nevertheless. We preferred it in this document because it's easy to install in the GNU/Linux environment, compared to SilentEye, which has not been updated recently as well (especially the Debian version).

You can install it on *Debian* machines simply using this command:

```
$ sudo apt-get install steghide
```

Now, let's say we want to enter this text: "Hi everybody" within an image called klimt.jpg. First, we have to create the text.txt file with the command:

```
$ nano text.txt
```

Save the file with *CTRL+X*, confirm with the *Y* key and click *Enter*. Now, launch the steghide program as follows:

```
$ steghide embed -ef text.txt -cf klimt.jpg
```

Let's try to quickly figure out what we've just done:

- steghide, here we identified the program to invoke, namely steghide
- embed, we used this parameter to tell the program to make an embed process
- -ef, this option specifies the name and the directory of the file we want to embed
- -cf, this option specifies the name and the directory of the file that must contain the text

Running the command, we will be prompted to enter a passphrase to be used to protect our data. Let's not ignore this request and proceed. After a moment, the image will be manipulated and will contain the selected text.

The *reverse* process, or the data extrapolation, is done by this command:

```
$ steghide extract -sf klimt.jpg -xf text.txt
```

Where:

- steghide, again, here we evoke the program to use
- extract, here we define the type of job to be done, namely an extraction
- -sf, to specify the file name and directory from which we want to extract data
- -xf, to define the file name and directory that will contain the extracted content

If you compare the two pictures, it's quite impossible to see any differences at a glance. Impressive, isn't it? The program also allows to change settings like encryption type, compression and many other values. All the documentation is well explained using the command:

```
$ man steghide
```

7.4.2 Cover Generation Steganography

A second and less popular (but still efficient) approach is known as cover generation: this system is based on entering information into a long text where nobody would ever suspect the presence of a hidden message.

If you watched *The Silence of the Lambs* you should remember how Buffalo Bill managed to send messages to Hannibal Lecter by writing letters to a newspaper and positioning words in a certain way to hide the content.

One of the most authoritative sites for this service is certainly spammimic.com: this website allows you to use different – and more or less interesting – encryption algorithms.

7.4.2.1 Pure Steganography with SPAM method

This method allows you to conceal a message within a fake SPAM message. Sending it to your recipient, anyone tracking the connection would see it as their typical SPAM message. Just consider this extremely long example:

Dear Friend , We know you are interested in receiving
cutting-edge news ! If you no longer wish to receive
our publications simply reply with a Subject: of "REMOVE"
and you will immediately be removed from our club !
This mail is being sent in compliance with Senate bill
2016 , Title 3 ; Section 305 ! This is not multi-level
marketing ! Why work for somebody else when you can
become rich in 70 days . Have you ever noticed more
people than ever are surfing the web and society seems
to be moving faster and faster . Well, now is your
chance to capitalize on this ! WE will help YOU increase
customer response by 110% & increase customer response
by 180% . The best thing about our system is that it
is absolutely risk free for you ! But don't believe
us . Mrs Simpson of Alabama tried us and says "Now
I'm rich, Rich, RICH" . This offer is 100% legal !
Do not go to sleep without ordering ! Sign up a friend

and you'll get a discount of 90% . Best regards . Dear
Sir or Madam ; Especially for you - this cutting-edge
announcement ! We will comply with all removal requests
. This mail is being sent in compliance with Senate
bill 2516 , Title 9 ; Section 303 . This is a legitimate
business proposal ! Why work for somebody else when
you can become rich inside 28 weeks ! Have you ever
noticed more people than ever are surfing the web and
people love convenience . Well, now is your chance
to capitalize on this ! WE will help YOU increase customer
response by 150% and turn your business into an E-BUSINESS
. You can begin at absolutely no cost to you . But
don't believe us ! Prof Simpson of Idaho tried us and
says "I was skeptical but it worked for me" . We are
licensed to operate in all states ! You will blame
yourself forever if you don't order now . Sign up a
friend and you get half off . Thank-you for your serious
consideration of our offer . Dear Friend ; This letter
was specially selected to be sent to you . If you no
longer wish to receive our publications simply reply
with a Subject: of "REMOVE" and you will immediately

be removed from our mailing list . This mail is being

sent in compliance with Senate bill 2416 ; Title 7

, Section 302 . This is NOT unsolicited bulk mail !

Why work for somebody else when you can become rich

in 10 WEEKS ! Have you ever noticed society seems to

be moving faster and faster and most everyone has a

cellphone ! Well, now is your chance to capitalize

on this . We will help you process your orders within

seconds plus use credit cards on your website ! You

can begin at absolutely no cost to you ! But don't

believe us ! Prof Anderson who resides in Missouri

tried us and says "Now I'm rich, Rich, RICH" . This

offer is 100% legal . Do not go to sleep without ordering

! Sign up a friend and you'll get a discount of 20%

! Best regards .

Decrypting it, you would obtain the following message:

Ciao a tutti i lettori da Stefano Novelli! (Greetings to all readers from Stefano Novelli!)

You'd never say that, would you? Such method can be subject to bruteforce (especially considering that spammimic always generates the same position); however, you can also use a password^[99] that will change the positions making the attack more difficult to who wants to discover your message.

7.4.2.2 Pure Steganography with PGP method

Even in this case a misleading message will be generated; apparently, you will be sending or receiving OpenPGP-encrypted messages.

```
-----BEGIN PGP MESSAGE-----
```

```
Charset: ISO-8859-1
```

```
Version: GnuPG v1.2.5 (MingW32)
```

```
Comment: Using GnuPG with Thunderbird - http://enigmail.mozdev.org
```

```
Q2lhbyBhIHR1dHRpIGkgbGV0dG9yaSBkYSBTdGVmYW5vIE5vdmVsbGkh
```

```
-----END PGP MESSAGE-----
```

There is also a method known as fake Russian (which I personally consider as pointless, unless you use a message which is already encrypted) and the spaces method that, unlike the first one I already introduced, uses the number of spaces to specify the character to show. Obviously, you can integrate this method with the classic PGP encryption, as seen in the previous chapter, to further increase the protection of sent and received messages.

7.5 Data Backup

So far, we discussed about how to hide your files. Now it's time to understand how you can save everything about your works, private life, top-secret documents and whatnot. Consider this chapter as very important, since your digital life may depend on how successfully you'll be applying all what I'm about to tell you.

Data Backup is a key process to avoid losing everything you worked on due to a physical failure, a crash or a mistake you had not to make. This process must be part of your daily checklist, something you must do *every single day*. Don't think the digital format is something indestructible: data technology is based on extremely small magnetic frequencies that can be subject to external forces at any time; just a shake, a storm or a simple hit and everything will be destroyed. I want to share a little story about myself:

I was just over 18 when I got my first clients. One of them had a clothing e-

commerce and needed a restyling for the entire portal. The fee we agreed on was not huge, but I was young and I could've used those few hundreds of Euros. I'd worked for that layout for two months; drafts, counter-drafts, templating studies, curated and efficient JavaScript events and a CSS structure that even the CMS developers would have envied. That evening, I went to physically clean the computer, since I believed all that dust could hurt my PC. I accurately put all components on my desk, except the main Hard Disk that I left on the case surface, God knows why. Without noticing it, I bent the case just a couple of centimeters to reach a difficult point, so the HDD slid off the metal surface and fell to the ground. It jumped only 40cm, crashing to the floor.

A utterly dead Hard Disk. I immediately connected it to the SATA and I heard the mechanical disk spinning, although the stylus kept on ticking intermittently for some weird reason. Nothing to do! I tried to disassemble it, in a desperate attempt to make it run again, without mentioning the fact that I had no idea of how to align it on axis again – and I honestly haven't any clue yet. A utterly dead and buried Hard Disk.

That event was followed by a declaration of hatred against the whole world for one hour and a half, only to drift off to desperation afterwards. It took about 2-3 hours before I recovered and started the whole work from scratch. From scratch. From the start

I still wonder how those 40cm were enough to irreversibly damage a HDD so massive – at least, for us mere mortals. That night I hanged out with my friends: I took the HDD with me as a sign of mishap, to tell everyone how absurd was that accident (maybe showing them that no scratches were visible on the case and so on). Encouraged by alcohol, we decided to destroy that HDD with our hands and by any means. We used different techniques: burned with alcohol and repeatedly hit with a shovel, dragged on the road, crushed (using the car), exploded with firecrackers. We didn't manage to pry it open, not even tossing it from the car at a speed of 100km/h. If you pass by that road today, you can still see the signal with a bent iron pole.

7.5.1 How many Backups do you need?

In order to keep everything under control, you need at least two Backup disks, possibly located in different areas. The first could be also left in the PC or, maybe, used as an external HDD, while you can keep the second one in the car, at work, at a friend/family member's. If anything happens to one of the two places, the other HDD will be safe.

Perform a backup when you run a new dump, when you have new logs to work on, when you create a new bitcoin wallet. ALWAYS perform a backup for everything you deem somehow important. Don't take it as a fixation, of course, but remember that the more time you will dedicate to your digital life the highest is the loss you could suffer – may it be money, work and whatnot.

Just think: perhaps one day you'll have – or you already have – hundreds or thousands of Euros in a wallet. And suddenly... puff! All gone. Who are you going to blame? Don't be too sparing when it comes to purchasing one or more additional storages: if needed, split your works into multiple memories (so you will also have a good archive, for a faster data search).

7.5.2 Rsync

Everyone have their own file organization method, therefore it would be pointless to list the best programs for a given Operating System, telling whether File History or Backup and Restore for Windows or Time Machine for macOS are adequate for security copies or not.

However, we can consider the fittest tool for this type of operations, rsync, which is available for all UNIX-based operating systems, and for Windows through a third-party setup with cygwin. Compared to the available counterparts, this software offers many advantages: it's commonly used among system administrators, therefore it's deeply documented, and offers a very effective file copy algorithm, also through ssh protocol (for remote copy) as well as the possibility to compress files on-the-fly and in different formats.

7.5.2.1 Rsync installation

Rsync is already available in macOS v10.4 and higher and, naturally, in most of the GNU/Linux-based distros, Debian included. If for any reason you won't find it installed, you can do it typing the following command from the terminal:

```
$ sudo apt-get install rsync
```

On *Windows* you can refer to the Cygwin^[100] program instead (which allows you to install most of the tools already existing in Linux), or cwRsync^[101].

7.5.2.2 Local copy with Rsync

The great versatility of rsync is documented by the excellent list of the supported parameters. You can access it launching the usual --help parameter:

```
$ rsync --help
```

or using man:

```
$ man rsync
```

Before you familiarize with its features, you should keep in mind the copy structure. Rsync manages inputs and outputs exactly like the Linux cp tool, so it will treat the first value as the element(s) to copy, and the second one as the destination path:

```
$ rsync [filetocopy] [copydestination]
```

Now, let's practice with parameters. Imagine you want to *copy a file from folder1 to folder2*, located in your user's home:

```
$ rsync -a $HOME/rsync $HOME/rsync_backup
```

Needless to say, [username] will be replaced with the current user nick, right? In this case, we used a parameter: -a. What is it for? The -a parameter copies all files *recursively* – including the ones in the folders – keeping the original structure, permissions and other information. However, we may want to compress the folder content *on-the-fly*: why not use the -z parameter?

```
$ rsync -az $HOME/rsync $HOME/rsync_backup
```

As we saw, we used -az, therefore we matched -a and -z parameters. In this case, the operation will be recursive and files will be compressed before reaching the destination, and then will be extracted locally. This feature may be useful for great amounts of data. Using rsync hasn't any kind of restrictions: you can experiment with copying your files directly into your external storage:

```
$ rsync -az /home/[username]/folder1 /media/[username]/[partitionname]
```

7.5.2.3 Remote copy with Rsync

Going deeper through the IT world, sooner or later you may choose to rent a

Dedicated Server, a VPS or have your own remote machine.

I won't explain how to configure a Server to accept SSH connections here, I hope you already know how to do this, and if not, you can find the documentation online or rent a Server or a VPS to start experimenting with rsync in the network too. The network protocol recognition is automatic, by preceding the data destination with the login data of the machine and its host, followed by a colon. For example, if you wish to copy remote data to your local computer, use:

```
$ rsync -a [user@host]:/folder1 /home/[username]/folder2
```

Here, [user@host] gets the login data value, together with the IP address of the machine or its domain. When needed, rsync will prompt you the SSH access password. If you changed the port of your server (this is TCP 22 by default) into another one for security reasons, you must tell it to rsync. In this case, the parameter is slightly more complex, but easy applicable nevertheless:

```
$ rsync -a --rsh="ssh -p PORT" [user@host]:/folder1 /home/[username]/folder2
```

By default, rsync cannot show the copy progress status. This can be a problem, especially when you are not sure about the size of the file to copy, as well as the transfer speed. To know the copy remaining time use the --progress parameter:

```
$ rsync -a --progress [user@host]:/folder1 /home/[username]/folder2
```

If, instead, you're used to make remote backups that are already in the directories but you don't want to download them every time, you may also specify the maximum size (and, when needed, the minimum one too) of the files you're going to process. Parameters are --max-size and --min-size, like the next example:

```
$ rsync -a --max-size=10M [user@host]:/folder1 /home/[username]/folder2
```

This way, the files larger than 10 Megabytes will be ignored.

7.6 Cold Boot RAM Extraction

If you have already used *GNU/Linux* distros designed for anonymity or pentesting, you will have surely noticed tools or modes for RAM-level attacks prevention. OK, let's take a step back.

RAM means Random Access Memory, that extremely fast memory used by Operating Systems and applications to provide values to the processor, which will manipulate and distribute them across the different resources. The RAM is the fastest memory in the computer, because it doesn't sort data, which is only temporarily allocated to the computer; once you stop using your computer, the RAM memory will lose all its data. If a RAM memory gets full (unlike a HDD/SSD), the system will keep on writing and reading, overwriting the older data.

The RAM contains the temporary data, i.e. when you write a Word file, it will temporarily store all the saves which are not memorized yet. Unlike other types of ROM memory, RAM is not encrypted at all. In IT, the most common type of RAM memory is the DRAM (Dynamic Random Access Memory). Unlike the SRAM (Static R.A.M.), such memory has an architecture that allows the surrounding system to clear sectors in short time and then add new elements.

The DRAM also includes sub-levels named DDR (is this familiar to you?). If you feel at home with computer disassembling, you may know that RAM currently reached the DDR4 standard, however it's not rare dealing with DDR3 or even DDR2 types. As we said, when a computer shuts off, its RAM memory gets wiped. The question is: *how does it get wiped?*

7.6.1 How to perform CBRE

The following is a research conducted in July 2008 in San Jose, by a group of Princeton University researchers, from Electronic Frontier Foundation and Wind River Systems, who spoke at the USENIX Security symposium, showing [\[102\]](#) how it's possible to extract data from the RAM minutes after the computer has shut off, even removing the RAM from the motherboard (Figure 29).

According to the research, DRAMs are not erased immediately, thus allowing enough time to perform forensic acquisitions on the operating system

last status. Such technique has been demonstrated by successfully recovering encryption keys from some of the most famous software in the IT world (including *BitLocker*, *TrueCrypt* and *FileVault*), revealing that it didn't require any particular tool. Furthermore, the research demonstrated that user login passwords or RSA private keys from an Apache web server can also be fetched from the OSX environment.

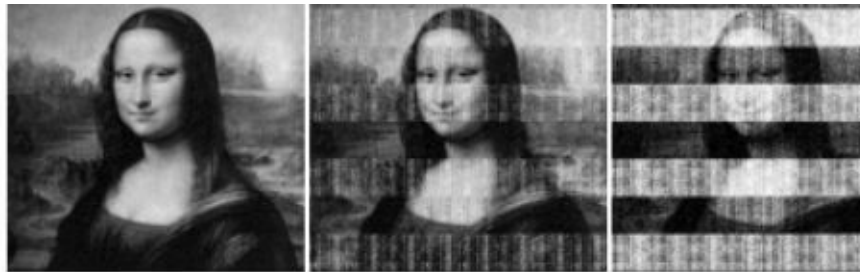


Figure 29: how the degradation of an image in RAM looks like. From left to right: 5 seconds, 30 seconds, 60 seconds and 5 minutes.

The following method will not be explained in this course, since it requires advanced reversing and RAM memories destruction skills. It will suffice to know that the “cold” term relates to the technique in use for the extraction: using a spray atomizer, the RAM temperature is brought to -50°C (Figure 30), this way data will be kept for some minutes or until static current runs through the memory.



Figure 30: demonstration of the Cold Boot RAM Extraction method

7.7 Metadata & EXIF Data

In IT, metadata are elements within files, usually not visible for the final user, that contain various data to allow the programs they interact with to function properly. Metadata may contain information about your identity and are traceable in different formats: *pictures, documents, video, etc...*

The story of *w0rmer*, codename of *Higinio O. Ochoa III* is quite famous in the IT world. Self-proclaimed as one of the Anonymous movement members who violated the USA law enforcement website, he was identified through a picture posted by his girlfriend, with this quote: “PwNd bu w0rmer & CabinCr3w <3 u BiTch’s!”. In that case, FBI found the girl using the picture metadata (later identified as EXIF Data).

With no doubt, imaging is one of the resources that made IT great in time. Nowadays, we are used to different formats (*JPG, PNG, TIFF* and so on); each of them has its own characteristics and is good for different scenarios.

The EXIF Data are metadata residing in media formats (images and some videos) that reveal additional, and quite interesting, information: you can identify the unique code of the device who took the picture (the *machine ID*), as well as brand and model, time, resolution and, if present, even GPS coordinates.

7.7.1 How to view the EXIF Data

The next generation image viewers pre-installed in the Operating Systems can show the images metadata in different formats. On Debian with GNOME 3, the official image viewer contains a sidebar showing the metadata gathered from an image by default. If not present, you can enable it on this menu: *View -> Sidebar* (or pressing *CTRL+F9*).

7.7.1.1 MAT: Metadata Anonymisation Toolkit

Certainly, one of the most popular metadata management programs in the IT world is MAT: Metadata Anonymisation Toolkit^[103]. This tool is pre-installed on different GNU/Linux distros and available in most of the repositories: you can also find the git repo^[104]s and the stable sources^[105].

You can *install it on Debian* by using the command:

```
$ sudo apt-get install mat
```

MAT can manage different formats and is available in CLI and, most commonly, in GUI versions. Such programs allows you to put one or more files into a list, then you can access their metadata fingerprint with a double-click (Figure 31).



The screenshot shows a window titled "Metadati di exiftest.JPG". It contains a table with two columns: "Name" and "Content". The table lists various metadata fields and their corresponding values.

Name	Content
GPS Img Direction	298.1968504
Scale Factor To 35 mm Equivalent	7.0
Compression	JPEG (old-style)
Subject Area	2015 1511 2217 1330
Sensing Method	One-chip color area
Circle Of Confusion	0.004 mm
Make	Apple

Figure 31: details of a test image with MAT

In this example, you can see many data related to the picture, including GPS coordinates, resolution, ISO, smartphone model and so on.

MAT also offers a convenient function for Metadata removal; such feature can be enabled by clicking the “Scour” button.

Why don't you try with some of your pictures? Try using a camera/smartphone, then try again with an online image. You can also try with different types of extensions or even files.

Please note that, when testing images from Internet (and especially from

social networks), it may happen that Metadata are not read. It may be caused by the site upload code, which could further compress the image in terms of format and resolution, in order to save space on their server and external bandwidth. Keep in mind, however, that each service may store the original files you uploaded.

Do you want to quickly erase EXIF Data from a JPG? Convert it into .PNG! This format doesn't support the EXIF Data as a standard.

7.7.1.2 Alternate software for Metadata

We only mentioned MAT because it's opensource and quite reliable for our purposes. However, there are alternate programs that work with the Metadata; the following list includes some of them with a short description of their features:

- Free Photo Viewer^[106] (*Windows*) - FPV allows to extract information for images in the JPEG and RAW formats. It also allow to fetch data like aperture, ISO value, focal length, time stamp, flash settings and so on. FVP also comes with a simple image organizer.

- IrfanView^[107] (*Windows/OSX*^[108]/*Linux*^[109]) - Available both in 32 and 64bit version, it's one of the oldest programs for who works in this field. It opens a huge number of extensions (also MP3, EPS, PSD, SWF and so on) and can be enhanced with plug-ins.

- *Photo (OSX)* - An embedded application of the Apple Operating Systems. Opening any photo, you can use the *cmd+i* shortcut or right-click -> Get Info. You can add custom metadata, like faces, description and keywords, but you cannot modify the existing ones.

- *Image Browser (Windows)* - WIB is embedded in every Microsoft Operating System. To access the image properties, *right-click -> Properties -> Summary Tab*.

- ExifPilot^[110] (*Windows/OSX/Linux*): command line tool, developed in PERL. It allows to open any kind of Metadata.

- GeoSetter^[111] (*Windows*): I think this is one of the best tools around.

Unfortunately, it's only available for Windows, but does amazing things: besides opening a vast number of digital extensions, it allows to change the geo-coordinates (including altitude), the IPTC values and much more. It's with no doubt one of the best tools to modify EXIF Data, since you can manipulate them to look convincing (instead of covering up data).

- ExifEditorApp^[112] (OSX): available for Apple OSs, this app allows to change EXIF and IPTC metadata.

- ExifDateChange^[113] (Windows): this tool is exclusively available for Microsoft OS, and comes both in free and paid versions. It's conveniently available in portable version as well.

Naturally, the list is not limited to the above; many more are available, such as Batch Purifier LITE^[114], EXIFCleaner^[115], PhotoME^[116] and so on. Just look up! Before we proceed, I must remind you that removing Metadata *is not the ultimate solution* to all your problems: the file you work on may be manipulated with Steganography, watermarks and other non-standard metadata. Furthermore, some of the programs we're going to mention allow to manage only the surface part of metadata: in example, you won't be able to change the values of an image within a PDF. You can also prevent most of the metadata contained in text documents using simple text formats (the so-called plain-text, most commonly known as .txt). Use them if you need to!

7.8 Camera sensors

This short chapter is intended to warn you to a new practice, which is globally applied by big *data mining* companies over the net. As you should know, each camera sensor release a *unique signature*, which is almost undetectable due to the minimal hardware differences characterizing it. As for a bullet that can be used to identify a weapon, a picture may allow to pinpoint the camera it was took with. It's worth mentioning that this has nothing to do with the *EXIF Data*, which work in a totally different manner. If you want to explore this topic, you can find online a document written by three researches of the New York Computer Science and Engineering Department^[117]. Unfortunately, no fast and proved methods to obfuscate such data are available today: if performed with the right tools, however, digital manipulation should ensure a good trace removal, i.e. changing color levels, saturation, contrast, sharpness, structure and so on. The study still highlighted the decreasing possibilities of

searching into over exposed photos (page 7 of the “*Sensor Noise Camera Identification: Countering Counter-Forensics*” research).

7.9 Data Shredding

One day, you may need a given file no more – whether it’s encrypted or not. That day, don’t just throw it in the bin: the file will stay there or, at least, leave some traces of its presence, possibly making it recoverable. In this chapter, we will cover all the known methods to completely destroy every evidence within your computer, especially the ROM memories storing your data.

7.9.1 How to perform Data Shredding

When the possibility to recover deleted files from a PC was discovered, dozens of free and commercial software sprang out, allowing to solve this problem. At the moment, we can break down the tools into three broad categories:

- Disk Cleaners
- File Wipers
- Physical drive destruction

7.9.1.1 Disk Cleaners

Such category includes software using different methods to *sanitize* a given Hard Disk. Essentially, they clear the drive sectors still containing *phantom data* information (a sort of in-memory reminiscence), which will be used by the operating system only when no free space is left on the Hard Disk.

However, the reliability of Disk Cleaners has been placed under discussion by many industry experts, since the techniques used are too weak and only focused on winning the “speed benchmark competitions”; furthermore, many of the software performing this task use to leave traces within the operating system proprietary logs.

BleachBit

BleachBit is an opensource program that cleans the disk space, optimizes the computer performance and ensures a better privacy for the user. Available for

Windows, macOS and GNU/Linux, Bleachbit maintains the promise providing tools designed to remove cache, cookies, history and logs from the main browser, also integrating a convenient feature to verify and rewrite the unallocated disk space (we are going to cover this topic in the “File Shredding” chapter).

Naturally, installing the program on *Debian* is a no-brainer:

```
$ su
```

```
$ apt-get install bleachbit
```

However, if for any reason it’s not available among the Operating System repositories, you can download^[118] and install the package directly from the official site. Since the program is extremely easy to use (it actually requires just two clicks) we won’t explore it further.

Other Disk Cleaning software

We can find many other tools designed for Disk Cleaning online. The following list includes those I feel to recommend:

- CCleaner^[119] for Windows, macOS and Android
- Glary Utilities^[120] for *Windows and Android*
- Clean Master^[121]

7.9.1.2 File Shredding

The *File Shredding* practice deals with this situation with a more straightforward approach, overwriting the memory position which the pre-existing file was allocated to with random bytes. The more that position will be overwritten, the more will be the chances of erasing the original file data for good. There are different opinions about the number of reiterations required for a proper File Shredding elimination: for example, the NSA recommends 3, the Department of Defense 7, while Peter Gutmann (who invented the Gutmann method) used even 35 reiterations for his most famous pattern. Everybody will make their own assessment, but 35 may indeed be a disproportionate number, even if the explanation makes sense (of course it does!); actually, however, 5-6 reiterations

may be enough for the randomization to allow an infinite number of file rebuilding hypotheses. For the sake of clarity, today, the Gutmann method is not effective anymore, because his studies were based on old patterns used in the IDE Hard Disks of the late '90s. Furthermore, we have to consider that, from 2001 on, many storage media manufacturers focused on Data Shredding, to the point to standardize their products with a technology known as ATA Secure Erase^[122]; according to a research conducted in 2011, however, only half of the worldwide manufacturers adopted this feature.

How to perform File Shredding

File Shredding is not particularly difficult; you can find many tools for any Operating System. Unfortunately, however, no one (except BleachBit) is cross-platform, therefore we must summarize one for each OS:

- CCleaner^[123] includes the Drive Wiping feature to clear the space occupied by deleted files. Only available for *Windows*.

- Once again, a huge number of File Shredding tools is also available for *Windows*: Eraser^[124], Securely File Shredder^[125], Freeraser^[126], WipeFile^[127], Secure Eraser^[128] and many more.

- On *Mac OS/OSX*, Permanent Eraser^[129] is seemingly the most reliable one

- The most recommended tool for *GNU/Linux* is shred^[130]; on *Tails* you can find Nautilus Wiper^[131]

- DBAN^[132] (to format an entire partition WITHOUT using the OS – you'll have to burn a Live copy)

Shred on Linux

If you need to shred an entire partition, you can always use *shred*, a good old command line tool. Get the list of your active partitions. You may want to use the *fdisk* tool as follows:

```
$ sudo fdisk -l
```

so you can be sure you selected the correct path to the partition you wish to

erase (let's assume */dev/sdb*). Now it's time to wipe the partition. If time is not on your side, you may want to use the *fast wipe*. This procedure is faster because a blank value is written over every sector:

```
$ sudo shred -vzn 0 /dev/sdb
```

In this case, the `-vzn` parameter will tell `shred`

- `v`, show progress
- `z`, overwrite the last shredding step (to hide it)
- `n`, define the number of iterations
- `0`, having defined 0 iterations, the value will be NULL, then 0

This will make the partition illegible, since the sectors won't contain any value. If, instead, you want to be confident on the result, you can always use the tool to perform more complex tasks. For example, by launching:

```
$ shred -vzn 3 /dev/sdb
```

You will instruct the program with the same things, but in this case three iterations will be in place, tripling the sector rewriting steps, thus performing a *safer* partition wipe. `Shred` is also a good program to delete individual files, by using the `--remove` parameter, as in the following example:

```
$ shred --remove [filename]
```

DBAN for standalone

DBAN^[133] (Darik's Boot and Nuke) is a free and opensource tool allowing file shredding over the entire hard disk. DBAN doesn't depend on any operating system in order to run, since it's a GNU/Linux-base distro itself. To use it, then, you need an external media (CD, USB etc..) and a short BIOS reconfiguration (exactly like when you run a Linux Live USB). DBAN should be used before physically destroying a drive, in order to increase the chances of making the disk illegible. Such tool offers *different possible removal algorithms*:

- Quick Erase

1 step - Security Level: Low

This method only writes a blank value (0) in every sector. It's only advisable if the partitions will be rewritten, i.e. if you have to reinstall an Operating System in it.

- RCMP TSSIT OPS-II

8 steps - Security Level: Medium

The Royal Canadian Mounted Police Technical Security Standard for Information Technology, Annex OPS-II: Media Sanitation. This module implements a data randomization process.

- DoD Short

3 steps - Security Level: Medium

The quick method used by the American Department of Defense. It is based on the steps 1,2 and 7 of the 5220.22-M model.

- DoD 5220.22-M

7 steps - Security Level: Medium

The standard method used by the American Department of Defense.

- Gutmann Wipe

35 steps - Security Level: High

The method described by Peter Gutmann in his document: "Secure Deletion of Data from Magnetic and Solid-State Memory".

- PRNG Stream

4/8 steps - Security Level: Medium/High

This method fills the device sectors using a pseudo-random numeric generator. This is probably the best method for the next generation disks, since the generation patterns vary. Such method offers a 4-step medium and a 8-step high security level.

Using DBAN

Working with DBAN doesn't require any particular skill. Once it's Live-mounted, it will appear as in Figure 32.



Figure 32: DBAN initial screen

You will find a list of the disks being used in your system, in this case it's an ATA Disk (although used in a Virtual Machine, it will be fine for testing purposes). On the bottom of the screen, you will find the keyboard shortcuts to enable the different features (Figure 33).



Figure 33: keyboard shortcuts to navigate DBAN

Always follow this caption to move across the program. Moving upwards and downwards, you can choose the partition to be formatted, then press the *M* button to select a deletion method (one of the above algorithms). For our testing, we will choose *PRNG Stream* selecting it and pressing *Space*.

The PRNG Stream provides the *Pseudo Random Number Generator*, an external tool that only generates (pseudo)random numbers. Pressing the *P* key you can choose between the two algorithms (each one will include a description). The *V* key allows you to perform a verification and choose how to run it: I recommend to leave it on *Verify Last Pass* so you can check the deletion only once the operation is done. *Verification Off* will disable the verification, while *Verify All Passes* will verify at the end of each passage (making the process much longer). The *R* button allows you to specify the number of deletion cycles. As we mentioned for the PRNG Stream method, in order to have a high security deletion, we'll proceed with 8 steps, as we will specify within the program (Figure 34).

```
Options
Entropy: Linux Kernel (urandom)
PRNG:   ISAAC (rand.c 20010626)
Method: PRNG Stream
Verify: Last Pass
Rounds: 8

Statistics
Runtime:
Remaining:
Load Averages:
Throughput:
Errors:

Rounds
> 8_
This is the number of times to run the wipe method on each device.
syslinux.cfg: nuke="dwipe --rounds 8"
```

Figure 34: 8-step definition for PRNG Stream shredding

Now that you're ready, go back to the program Home and click *Space*. Next to your partition, you will see *[wipe]* (Figure 35).

```
▶ [wipe] ATA Disk UBOX HARDDISK 1.0 8GiB (8589MB)
```

Figure 35: disk wiping confirmation through DBAN

Now you're ready to wipe your storage. Press *F10* and allow the deletion process to end.

File Shredding and SSD – Everything you need to now

If you arrived so far, I guess you need to have a constantly clean storage, always ready to withstand any forensic searching. You must know that File Shredding methods are effective for mechanical disks, while they *may* be ineffective for SSDs.

The reason is that all the variables determining the success or the failure of a File Shredding operation over a SDD are multiple: in mechanical disks, files are seemingly deleted, but still keep the occupied space to avoid any slowdown in the deletion process. When new data is created, mechanical disks rewrite the sectors flagged as “deleted”. In SSDs, when files are deleted, the SSD will choose whether to write the sector or not: such choice is taken by an internal controller, which can be controlled through a module known as TRIM (<https://it.wikipedia.org/wiki/TRIM>), that flags blank sectors and let the operating systems reuse them at once. TRIM is enabled by default in the latest versions of all OSs and it should ensure the immediate rewriting of the newly deleted sectors. SSD, then, have an internal logic that uses the TRIM to rewrite a

sector almost immediately, thus acting as a pseudo-shredder. Such consideration also suggests that 1 deleting iteration (step) is enough for solid state disks. However, the logic behind file shredding tools poorly fits into SSD architecture, therefore the only real solution in this scope is the total and complete wipe of the entire disk (not just a partition) using tools like DBAN and whatnot. Once again, a single deleting step will suffice. As an alternate solution, you may encrypt the file you wish to hide: doing this, the file is overwritten on itself. This way, the old version should become illegible, still maintaining the accessibility to the file, under a minor compromise. Naturally, you can apply this technique to the whole disk (please see the disk encryption part).

7.9.1.3 Physical Drive Destruction

This category includes all the techniques you can use to completely or partially destroy a physical storage device. We must state here that the destruction of a hard disk – both 3.5” and 2.5” – is an extremely strenuous operation. In mechanical hard disks, for example, you will take several minutes to reach the magnetic media, and such operation often requires a huge amount of effort and time. When it comes to the solid state drives (SSD), the operation may be easier, however you should exactly know where to “drill” between the metal plate and the surface where all the data storing media reside, avoiding any possible risk. As you can imagine, destroying a drive must be a quick task, something you should be able to perform in seconds. All the following methods relate to a Hard Disks and require you to know how to disassemble the magnetic media (for mechanical units) and the flash memories, looking like many microSDs soldered over the circuit (for SSDs).

Mechanical Destruction

Applies to: SD units, CDs/DVDs, Mechanical HDDs, SSDs

I cannot recommend using maces, tossing or whatever comes to your mind. Without the proper tools, the disk may stay intact. Alternatively, you can use a good nail gun and cripple the disk with holes: quite probably, nails will irreversibly damage the internal parts; plugging the drive, however, will likely cause a short circuit.

This operation can be useful for disassembling the internal parts of mechanical disks and SSDs and disposing them elsewhere.

In the case of mechanical disks, you can use a robust tool like a smith hammer to

smash the disk to the point of bending it, causing its demagnetization.

For CDs/DVDs, just use some sanding paper; after a couple of rubs, the surface mirror layer should go away like ash.

Demagnetization

Applies to: Mechanical HDDs, SSDs, USB memories, SD units, CDs/DVDs

The Degausser method – or demagnetization – is process that literally fries electronics, sending an electromagnetic pulse (Emp) to the device. You can find some professional-grade tools (Garner seems to be particularly reliable) or you can build one at home (just search *Create a Degausser* or *Degausser DIY*). We've also successfully verified this method with EPROM memories using a home-made taser made with an anti-mosquito racket (you can find many tutorials online), however we cannot recommend you to perform this task unless you have the right electronics skills (you may burn the entire machine!); such method, however, is inadvisable with mechanical disks. In the case of USB or SD storage, you can try with a microwave oven, although this is an incineration job rather than demagnetization.

Incineration

Applies to: Mechanical HDDs, SSDs, USB memories, SD units, CDs/DVDs

A word of warning before we proceed: besides the danger of setting anything ablaze, such practice may be illegal in your country, because disks contain metals that produce extremely dangerous fumes! Anyway, you need to reach 1115°C, or *Curie's Temperature*, resulting in the loss of some ferromagnetic properties of cobalt (found in some HDDs parts).

You can use:

- an industrial blast furnace, available from companies in the iron and steel industry
- an oxy-acetylene torch, for about 100 \$, reaching 3000-3100°C
- other torches, but you should verify their iron melting (1500°C) capability
- termite – you can make it at home, but it's extremely dangerous, since it reaches 2200°C

Sinking

Applies to: N/A

The mere “sinking” of a HDD in water won’t physically harm the device. Water may damage the logical board (the set of controllers and components that are soldered on the PCB), which is quite easy to replace, however.

The surface layer of mechanical HDDs magnetic disks is built with cobalt alloys covering glass, aluminum and a ceramic substrate. In this case, water must reach the aluminum in order to cause an actual oxidation damage to the disk. If the disk is not powered and not writing, then, water alone cannot damage the inner parts: perhaps, the only real way to proceed is “marinating” the disk in seawater for several days. Once dried up, the evaporated water will leave small parts of salt, metals, silicon and other agents on the electronics. This way, once the drive is powered on, damage should occur, compromising the magnetic disk. However, an experienced company in the industry will replace the whole disk electronics (only keeping memories) at ease, neutralizing such risk.

Chemical corrosion

Applies to: Mechanical HDDs, SSDs, USB memories, SD units, CDs/DVDs

Once again, I recommend you to be very careful, especially if you are unfamiliar with this process. You should perform this operation on an opened disk, namely you should remove all the covers on the memories or, in case of a mechanical disk, the storage disks. The most common solution is using *hydrochloric acid* (or muriatic acid), one of the most corrosive fluids in the world (did I already suggest you to be careful?); you can buy it at any convenience store or hardware shop in solutions varying from 30 to 37% in concentration. Nitric acid seems to be another good solution, although it’s harder to find (however you can find it in concentrations up to 65% at any hardware shop): furthermore, mixing it with hydrochloric acid in a 1:3 ratio, you can obtain the world-famous white spirit, which dissolves the most durable metals, like gold and platinum. Anyway, you should pour the acid into a heat resistant plastic container – due to the chemical reaction – and use a sufficient quantity to fully sink the entire disk (possibly allowing a tolerance of a couple of mm, just for safety) for a hour or two. Pay extra attention to your hands and clothes, and never look directly without protections, don’t breathe the exhalations and don’t close the container for any reason (it may explode!).

8. Data Recovery

Now it's time to verify if the applied methods actually make storage illegible. Please keep in mind that Forensic Search is a very complex professional field, therefore we'll only introduce the basics, as we don't mean to provide an advanced course about this topic.

8.1 *Post-Mortem Forensics*

For most of the forensic search operations, the working environment must be as aseptic as possible, without any program that may alter the Operating System nature, once launched. For example, a programmer – even a beginner – may create a simple script to encrypt/decrypt/hide/move/delete a file in the drive, hiding it from the operator and the program. You can even build a background tool to recognize and block/deceive a program after a disk scan. Now, a forensic researcher should have some safety copies in place – in order to prevent any incident – as well as the right tools to do the job with any risk of compromising the system. For this reason, *you should use a Live Operating System* containing the computer search tools; as we will go through the document, we'll use a distro containing some tools that you can also install in your Operating System. This kind of research is known as post-mortem forensics.

8.1.1 Which OS for P.M. Forensics?

First of all, we can identify two types of Operating Systems:

- *Rescue Kit OSs*
- *Forensics OSs*

The *former* are specifically developed for data recovery (together with partitioning, anti-virus and other tools), while the *latter* are best suited for navigating a system and minimizing damages.

The Rescue OSs were popular in the past, but are now deprecated: from the famous Hiren's Boot CD^[134] to Ultimate Boot CD^[135] up to FalconFour's Ultimate Boot CD^[136], they're all abandoned to date. The only one that seems still under development is SystemRescueCd^[137].

Conversely, the Forensics OSs “market” is still flourishing: besides the fact that many pentest distros include dedicated toolsets, entire operating systems exclusively designed for such practice are also available. You can use purpose-specific distros (we will list them shortly) or create your own. Most importantly, however, the OS should never impact the target disk.

Although you can mitigate such risk using Write Blocker^[138] (a tool laying between the computer and the Hard Disk that blocks any disk alteration), you should consider a distro with the RAM usage feature. Using the *RAM mode* – usually listed as a boot loader option – you can access any memory attached to the system in read-only mode, preventing any disk alteration; consider this practice as mandatory, since a Write Blocker may cost more than 500 \$ – and I guess not everyone would likely invest such amount of money. In the IT sec, CAINE^[139] is the most popular one: a 100% Italian Ubuntu-based distro, which is also used by the law enforcers, since it provides results that can be used in the court. The development is managed by Nanni Bassetti, the project founder who keeps on updating the distro together with the online community.

Tip: For the aforementioned reasons, we will use a GNU/Linux designed for Computer Forensics. However, Windows user can use another good and effective tool: Recuva (www.piriform.com/recuva), produced by Piriform (the same authors of CCleaner) and available online for free. The difference between a software and a GNU/Linux distro lays in the type of approach you are looking for: in this case, we will refer to Live Forensics instead of Post Mortem Forensics.

8.1.2 Caine OS

CAINE OS is actually a GNU/Linux distro designed to work in Live mode, loaded from a USB drive or a DVD. In this guide, we will make a limited use of it, since our only purpose is to verify the presence of files and partitions we expect to be removed. CAINE does integrate professional verification and reporting tools to provide irrefutable evidence to the court – something quite unnecessary in the scope of our course. On the next pages, we will use CAINE to test some software in the distro; however, if you need you can directly install them on Debian (on your personal distro) and do your testing from there. You’ll lose the joy of discovery, but it’s a good alternative anyway. CAINE also offers a *mount in read-only logic*: meaning that you have to choose the partition to mount BEFORE you can use it, avoiding to compromise the areas you’re going to

recover.

8.1.2.1 TestDisk or PhotoRec, which one?

TestDisk is a tool designed to recover entire partitions that have been deleted from a hard disk. Besides this outstanding feature, it also offers corrupted boot sectors recovery with FAT and NTFS file systems and the Master File Table on NTFS partitions. The tool comes with no GUI, being command-line only, although it's quite easy to use, so it shouldn't be a problem. Our purpose, however, is to verify if the file in the hard disk were deleted; we don't need to recover corrupted partitions. We only want to ensure that, once a file is deleted, no visible traces are left behind. PhotoRec is a TestDisk complementary tool that allows to recover files, documents, videos, images and more from external or internal storage devices. The special feature of PhotoRec is that it works independently from the file system, and does not directly run in write mode, ensuring the integrity of the storage under test and avoiding any dreadful sector-rewrite error in the partition. The drive must always stay in read mode: if you write even a single piece of data into the storage space, you may irreversibly compromise the data recovery. PhotoRec is available for any *operating system*, including: Dos/Win9x, Windows (32/64-bit), Linux (32/64-bit), OSX/macOS (Intel/PowerPC), *BSD; it is also available in the package format with TestDisk for free from the official site^[140]. Furthermore, you can use it over a vast array of *file systems*: exFAT/FATx, NTFS, ext2/ext3/ext4, HFS+; I also want to add btrfs that, although not officially supported, seems to work quite well. You can use it over any standard external media, as long as the operating system recognizes them and can access their content. The tool can read (almost) any format, from the classic JPEG/PNG/ZIP/PDF to the rarest LZO/XAR/PPM/RA and up to the proprietary ones like PSD/MHBD/MAX/GI and so on^[141].

8.1.2.2 PhotoRec Mini Use Guide

PhotoRec comes in two versions: *GUI* and *CLI*. Obviously, the GUI version is easier, since it can manage everything from the graphical interface. If it's not pre-installed on your distro, you should find *QPhotoRec* (PhotoRec GUI version) among the installable programs. In that case, proceed using the terminal:

```
$ sudo apt-get install qphotorec
```

Allow the installation to complete, then find the program among the installed tools; if you can't find it, open the terminal again and type:

```
$ sudo qphotorec
```

The program will appear as in Figure 36.

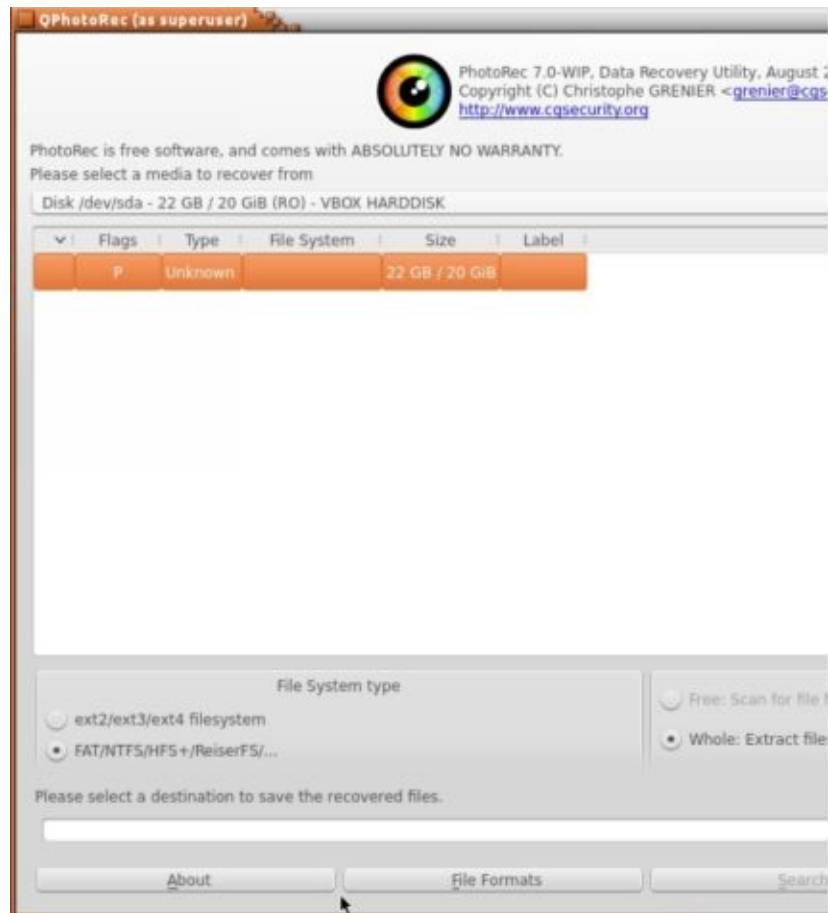


Figure 36: initial screen of QPhotorec, GUI version

If you can't see the target partition, you have to select the disk containing it. Select the destination partition, the File System type, the Free/Whole scan (Free will do in our case) and choose the path where the results will be saved using the "Browse" button. You'll just have to wait for the program to finish the drive scan! If you prefer the good old terminal, first of all ensure that the latest version is installed:

```
$ sudo apt-get install photorec
```

If you can find the program in the Operating System, proceed launching it with the command:

```
$ sudo photorec
```

As we already seen, we evoked sudo again, since we need to ensure that PhotoRec is launched in admin mode. Now you can see a screen listing all the disk discovered in the system (Figure 37).

```
PhotoRec 7.0, Data Recovery Utility, April 2015
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

PhotoRec is free software, and
comes with ABSOLUTELY NO WARRANTY.

Select a media (use Arrow keys, then press Enter):
Disk /dev/sda - 22 GB / 20 GiB (RO) - VBOX HARDDISK
>Disk /dev/sdb - 30 GB / 28 GiB (RO) - TDK LoR TF10
Disk /dev/sr0 - 3158 MB / 3012 MiB (RO) - VBOX CD-ROM
```

Figure 37: initial screen of Photorec, text version

Choose one using the Up/Down keys, select it with Enter or, in case of errors, press the Q key.

```
Disk /dev/sdb - 30 GB / 28 GiB (RO) - TDK LoR TF10

Partition          Start      End      Size in sectors
No partition       0 0 1 29553  5 32  60524736 [Whole disk]
> 1 P FAT32        0 0 3 29553  5 32  60524734 [TESTDISK]
```

Figure 38: choosing a partition of the entire disk

Now, choose the target partition (Figure 38). Selecting Whole Disk, you will recover the full disk. Choose the type of file system in use (Figure 39).

```
To recover lost files, PhotoRec need to know the fi
file were stored:
[ ext2/ext3 ] ext2/ext3/ext4 filesystem
> [ other ] FAT/NTFS/HFS+/ReiserFS/...
```

Figure 39: choosing the type of file system in use

If you selected a partition, you will be prompted if you wish to look up

across the whole partition or the blank sectors only (Figure 40).

```
2 P MS Data          201  0  1 29542  63 32  606
T]
Please choose if all space need to be analysed:
>[ Free ] Scan for file from FAT32 unallocated space
[ Whole ] Extract files from whole partition
```

Figure 40: choosing the type of scan to perform

Now you're ready to select the folder where your search will be saved (Figure 41). Remember that the keys used above also apply here (in particular, Enter to access a folder and Q to go back), with the addition of the C key to select the folder (and sub-folders) where you wish to work (if you accessed the wrong folder, go back by clicking the two dots at the beginning of the list).

```
Please select a destination to save the recovered files.
Do not choose to write the files to the same partition t
Keys: Arrow keys to select another directory
      C when the destination is correct
      Q to quit
Directory /media/sde2
drwxr-xr-x  0  0    32768 30-Jul-2016 11:42 .
drwxr-xr-x  0  0    240 30-Jul-2016 11:39 ..
```

Figure 41: choosing the path where the recovery results will be stored

If everything went as planned, the software will start digging into the desired partitions, putting everything into a dedicated folder (Figure 42).

```
Pass 1 - Reading sector      63408/60092416, 7 files found
Elapsed time 0h00m16s - Estimated time to completion 4h1
apple: 3 recovered
jpg: 2 recovered
gz: 1 recovered
tx?: 1 recovered
```

Figure 42: running the Photorec program

Now, let's see some usage examples: we formatted (with no shredding) a 32GB USB drive named "TESTDISK". Then we created a FAT partition where we placed some files (Figure 43).



Figure 43: test images for recovery

Each file has been renamed according to the action we performed:

- *deleted but not empty.jpeg* : an image that was deleted without removing the temp files or emptying the bin
- *deleted.jpeg* : an image that was deleted, removing the temp file and emptying the bin
- *normal.jpeg* : an image we didn't performed any action to
- *normal.jpeg.gpg* : an encrypted image
- *secure-shred-1.jpeg* : an image that was deleted using file shredding with DoD Short type, 1-step algorithm
- *secure-shred-7.jpeg* : an image that was deleted using file shredding with PRNG Stream type, 7-step algorithm
- *shred-1.jpeg* : an image that was deleted using file shredding with Quick Erase type, 1-step algorithm
- *shred-7.jpeg* : an image that was deleted using file shredding with DoD type, 7-step algorithm

Let's see the behavior of Photorec (Figure 44).

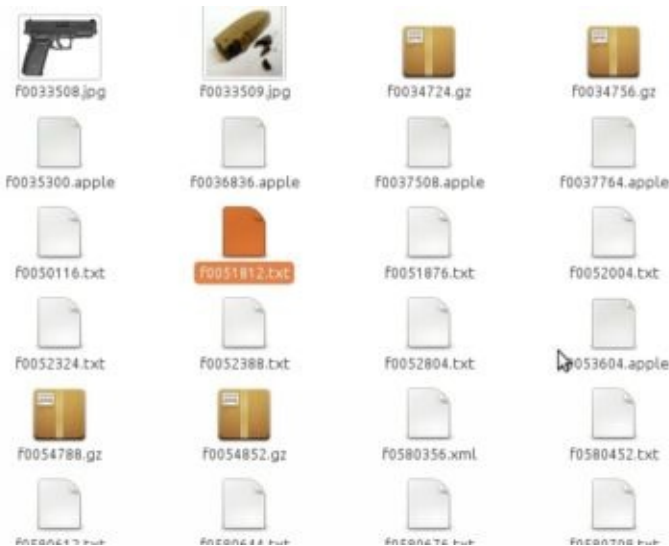


Figure 44: recovery results with Photorec

In our case, we recovered over 3GBs of files (Figure 45)! But how?!

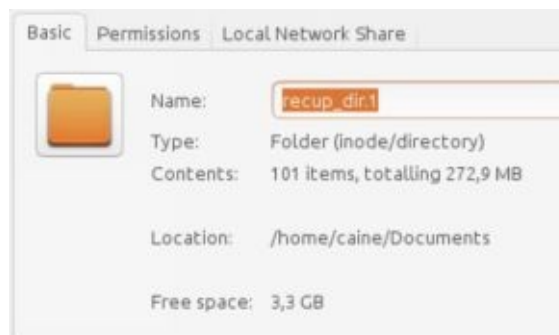


Figura 45: Folder that contains recovered files

At the beginning of this example, we mentioned that our drive was formatted in FAT through a simple format command. Before the formatting, it contained a Windows installer and, earlier, it worked as a normal USB drive, used to move files from a Mac Operating System to a Windows one. In one of the recovered dirs, we can find some .apple files, proving that the previously used operating system was OSX indeed. Many opened .txt files proved that the drive could have contained Windows files, also suggesting that can have been used as a Windows 10 installer (as mentioned above). And what about files?



Figure 46: details of the Photorec recovered files

We can still see some (Figure 46):

- *f0033380.jpg* : is the deleted.jpg file
- *f0033381.jpg* : is the deleted-but-not-empty.jpg file
- *f0033508.jpg* : is the normal.jpg file
- *f0033509.jpg* : is the shred-1.jpg file
- *t0034436.jpg* : is the preview of the secure-shred-1.jpg file
- *t0034500.jpg* : is the preview of the shred-7.jpg file

We can deduce that only the normal deletions and the Quick Erase were ineffective, while the DoD and PNRG techniques have been successful and that, after a partitioning operation, some files have been probably recovered (in this case, the Windows installer), together with some previous programs (and this would explain the amount of recovered data). However, we must consider that the Operating System used to create the driver was MacOS, which took the liberty to create some previews to our images during the data verification, thus exposing their content to the public access, although in low resolution.

9. Vulnerability

Despite all the countermeasures you may adopt to stay anonymous, unfortunately in the IT world there's always a chance to become a victim. It is known that the U.S. Government is the biggest buyer of not-yet-disclosed vulnerabilities (the so-called 0days), weak points that are constantly used to perform secret pentests. The following is a quotation from *John McAfee*, the famous anti virus CEO, who said:

There isn't too much security anymore, especially in the online world. Give me some simple information about you, and I promise I'll be able to activate your webcam and see everything you do in three days.

I want to add something that happened to me a couple of years ago:

I remember a dental technician – someone not involved with IT Security – who used to cover his webcam with a small piece of dark duct tape. I said to myself: “this guy is paranoid!”. A couple of days later, an article reported an exploit which had been used for months or even years to spy the users of that particular laptop (for the sake of clarity, it was a MacBook Pro). Who's familiar with this type of laptops knows that a green LED is lit when the webcam is on. Well, that exploit also allowed to turn the status LED off!

What can we learn from this story, then?

9.1 General Precautions

For example, covering the webcam when you don't need it is not a bad idea after all! Of course we can constantly monitor the network traffic to see if someone is connecting to our notebook/computer, but this would distract us from your tasks; furthermore, the attacker may use a backdoor installed in our computer and then arbitrarily hide their data from our network monitor.

The same can be done with the microphone: in this case, the best option (if possible) is to physically remove it from the device; alternatively you can also deactivate it from the Operating System, but, in case of attack, it can be reactivated with no effort.

Device monitoring is also feasible on a smartphone, and this can be a critical problem. Unsurprisingly, the competent authorities can perform electronic surveillance using the microphones (or getting images) on smartphones: the problem is that using a phone with no mic would be pointless, furthermore I am quite sure that not everyone is capable of disassembling it without damages. According to some research taken from Wikipedia, electronic surveillance can also be performed over a smartphone without a plugged battery^[142]. In this case, the simplest solutions would be leaving the smartphone in a microwave oven, which isolates the electro-magnetic fields, blocking any transmitting wave. Remember, don't turn the oven on!

When it comes to the emails we receive, although it may sound redundant after all we said about security, let's face the truth: you should NEVER open any attachment when your are not 100% sure of its origin.

And what about the Operating System? At the beginning of the course, we mentioned that you can stay relatively safe with any Operating System; however, you have to consider that GNU/Linux and *BSD are the only 100% trustworthy operating system. Windows and OSX/MacOS are proprietary OSs and could contain not only trojans and spyware, but also exploits the online community may not be able to fix or aware of, since the source code is handled only by the respective developers.

If you have even a small doubt about any file, you should always open it from a Virtual Machine. This way, the opened files will be virtualized in an external environment (unless the latter contains an exploit itself, capable of

“breaking the wall” of the Virtual Machine itself) and, if containing anything that may compromise your privacy and safety, they will be restricted to that environment.

If you don't trust your BIOS, flash it: some malware can penetrate the BIOS and, in this case, no Anti virus will be able to access it (remember that an anti virus only works when the Operating System is launched or, in some cases, just before the startup). Ensure the firmware in use matches the one provided by the developers and never trust custom firmwares built by strangers or people whose reliability is not acknowledged by the online community.

Speaking of Anti virus: are they truly useful? There are different perspectives: someone think they don't hurt, someone deem them as indispensable, and there's also who says they're useless, following their instinct and habits. As usual, the truth lies somewhere in between: it's all about what you do, how much you trust Anti virus programs and your choices when you decide whether to open a file or not. Certainly, Anti virus are not 100% perfect, since they use shared databases and some heuristic search algorithms to figure out what a file or a program will do once opened, but this is just statistics and they may return a false positive (a non-virus seen as a virus) or ignore a true threat. The truth is, when a devices gets infected or aimed by a governmental agency, the chances an Anti virus detects it are zero. Furthermore, most of the modern IT virus programs are obfuscated and modified at the source, in order to make the AVs operations harder – or even neutralize them. These are some of the reasons why we didn't – and won't – cover the most trusted Anti virus software.

Of course, it doesn't mean that we won't properly safeguard our Operating System: it must be constantly updated instead, using the latest versions of programs and technologies in general (do you remember the famous Heartbleed?) and you should configure it to always stay under your control. For example, the System may have a feature to automatically connect to a WiFi network: it would be quite easy to expose it and compromise the user's security.

10. Enhanced OSs

The GNU/Linux is fascinating for many reasons, including the extreme customization possibilities that allowed entire communities to build their own version and deploy it to the world. Today, we can find thousands of GNU/Linux distros for any demand: among these, the anonymous distros world seems to be one of the most flourishing.

10.1 Live OS

An Operating System you can launch from Hard Disks, as well as from USB drives, CDs/DVDs and even from SD cards, as long as you have enough digital space to allow the due operations. In years, a new usage method has been developed: Live OS, a feature that allows you to use a GNU/Linux distro without modifying your primary hard disks. We are offered such chance not only to test the distro without harming our partitions, but also to leave no trace within a computer, as it has been discovered.

Everything happening within a Live system stay in the Live system: no temp files are saved, no permanent logs are generated and the full environment comes to life and dies once the storage containing it is plugged in or removed. However, you may need some files or programs to stay available even after the system shutoff: the Persistence Mode was built for this purpose, allowing you to store any preferences, files and edits even after the computer is powered off.

10.1.1 Tails OS

Tails OS^[143] is a GNU/Linux Live distro available since 2009. It belongs to the Debian family and perfectly works with all the commands we explained in this course. Comes with all the tools you need to ensure a good anonymity and security to your Computer; you will also find a pre-configuration that routes all the connections directly to TOR, blocking the incoming ones. It's certainly an interesting distro, since it's ready-made: all the possible configurations are available in the Greeter, the pre-launch menu of the operating systems that also allows to activate the *I2P network*, toggle the *Mac Spoofing* and the root account on and off, create a persistent, encrypted space, establish TOR bridge configurations and much more. LUKS is already built-in as a standard for partition encryption.

Even the GUI is designed for anonymity: you will find the *wiping* feature directly from Nautilus (the explorer), a pre-installed *GPG* integration (even in the mail client), a pre-configured Iceweasel browser for TOR navigation and the basic tools for the most common operations in the IT world. You will also find the pre-installed *OTR* technology, allowing to encrypt the communications via Pidgin chat, the messaging program always included in most of the Penguin distros.

10.1.2 Live OS & Persistence: the risks

Live OSs are designed to be used both in computers you own and not: for example, you can run them at Internet Points, public terminals or borrowed computers. As we mentioned, the *persistence mode* allows you to integrate the Operating System with a partition which remains intact even after the computer is powered off; remember that a Live distro loses its memory (the so-called Amnesia effect) when the user launches the shutdown command for the entire machine.

If you choose to have the persistence mode in place, you should consider all the conditions from the "Data Security" as true, therefore you must apply encryption, data shredding and all the methods that help you preventing the exposition of your memory content to other people. Furthermore, I recommend you to check if the Live you're using provides the option to encrypt the persistence mode: in this case, check the presence of LUKS among the supported formats, so that you won't have any problem accessing it in the future,

even outside the Live OS. You can further explore the encryption of full disks on Wikipedia^[144].

10.1.3 Live OS & Virtual Machines: the risks

VMs are a truly excellent tool: in a nutshell, they allow to create a computer within a computer! They're often used when the Operating System is not compatible with some software (ex., when you want to use Windows applications from a Mac).

However, I recommend you to perform all your tests in a Live working environment, at least for now. The reason is related to different choices in terms of workspace sanitization: in Tails, for example, the "anti-forensic" properties would be compromised if, launching a GNU/Linux distro within a Virtual Machine, the latter writes the host computer swap with files otherwise destroyed in Live; additionally, it may happen that, putting the VM in hibernation or stand-by, the WHOLE operating system is stored in a temporary page file, thus exposing all the Tails content (by the way, VirtualBox & Co. are integrating disk encryption options on top of the software).

In anonymity scenarios, using Live GNU/Linux distros is strongly advisable. When necessary, you can set-up a USB/SD drive to contain user-reserved space for their configurations, resulting in a hybrid system capable of working as a Live, as well as storing files and whatnot like a normal installation.

10.2 Virtualized environments

When it comes to IT Security, a virtual environment can ensure a good isolation in different scenarios: just think that, if you wish to study the behavior of some malware, it's crucial to use a virtualized system, in order to safeguard the central Operating System. Just like a Live, everything happening in a Virtual Machine (usually) stays within the Virtual Machine: I wrote 'usually' because a VM may be attacked and slip out of the host computer control, but we're going off topic, probably.

Using an operating system anonymously within a VM is utterly wrong: most of the aforementioned procedures refer to obfuscation methods that require the full control of the hardware in use by the System (just think about the Mac Spoofing). As the term suggests, virtualization is all about virtualizing hardware; if we decide to perform the Mac Spoofing for a virtualized environment, we

would ONLY change the virtual Mac Address, and not the real one! Such operation can be done ONLY from the host Operating System, namely the one hosting the VM and not vice versa.

But what if the host is the Operating System providing its Virtual Machines with anonymity? Then the scenario would be extremely beneficial for the user.

10.2.1 Qubes OS

The Qubes OS^[145] project dates back to 3 September, 2012 – created by an IT researcher, Joanna Rutkowska. This particular Operating System introduces a security approach defined as *isolation*: essentially, it's taken for granted that every piece of software may be potentially harmful and that a single bug may compromise the entire IT system.

Qubes is based on Fedora Linux, but provides a para-virtualization system through Xen: its microkernel allows to create separated workspaces, here known as qubes, where the tools coexisting in the same domain can interact. To better understand this concept, look at the Figure 47 carefully.

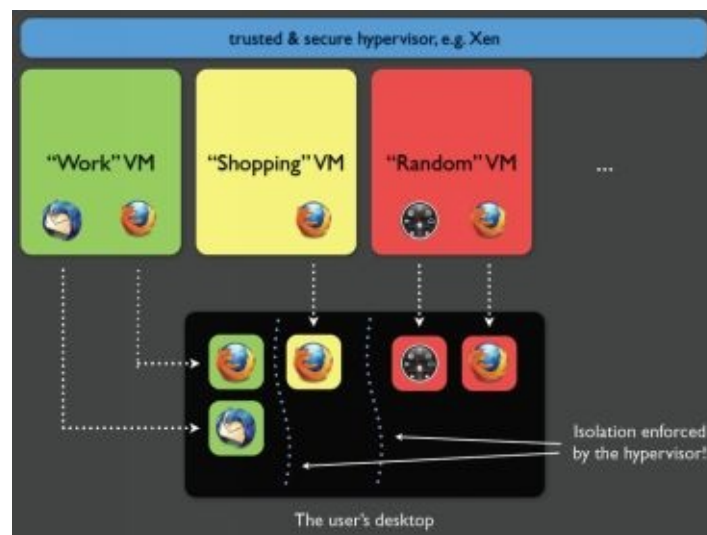


Figure 47: in this example, three virtualized environments are in place: Work, Shopping and Random.

All three environments have a Firefox active process; these, however, are treated as separate jobs, so if you are logged into Amazon in the Shopping VM, you won't be in Work and Random, thus ensuring the process isolation across

the workspaces.

Was an Operating System really needed? Couldn't they create three virtual machines instead? Sure, but the three VMs would have required three operating systems, each consuming hardware resources, requiring updates and so on.

10.2.1.1 Virtualization logic

The Qubes built-in hypervisor allows to create unlimited qubes using a single Operating System, which natively supports *Fedora*, *Debian*, *Windows* (after installing the latter^[146]) and *Whonix*; furthermore the workspaces share the same graphic environment, thus eliminating the annoying switch from an OS to another and vice versa. You may also have noticed that the three workspaces are color-coded; the same happens with the graphical level (Figure 48).

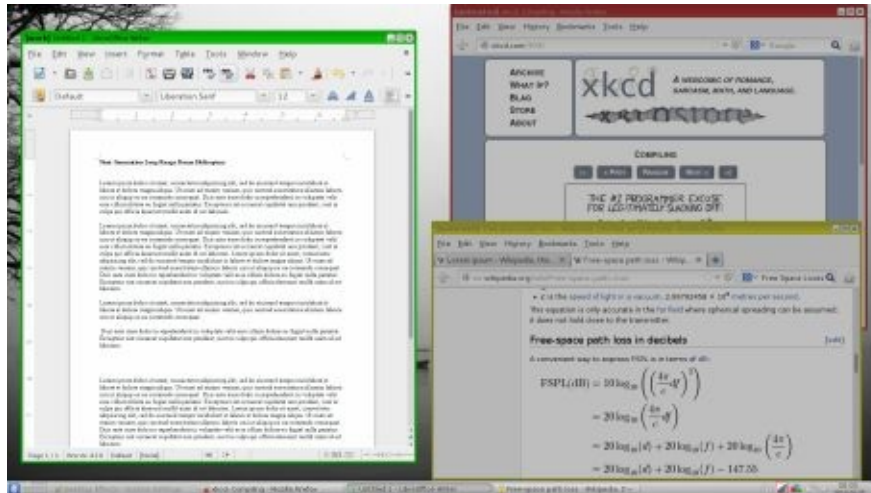


Figure 48: working logic of Qubes virtualized environments

As you can see, the three windows are represented by a different color, just like the infographics. Additionally, only one is bright, while the others are slightly shaded.

Qube OS allows to immediately recognize the workspaces breaking them down by color, as well as to show the ones you're working on in real time, highlighting the application that can inter-communicate. And when a VM is closed, all the temporary data it generated is erased.

10.2.1.2 Network and Storage Domains

As we noticed, the Network environment is the most harmful for user wishing to protect their anonymity.

Qubes OS provides a virtualization system called Network Domain: essentially, the VM concept is also applied to the network, which is virtualized in an environment controlled by a pseudo-user without root privileges and isolated from the rest of the Operating System. Consequently, it's like all the networking operations (website linking, downloads, chat and so on) are managed by another Virtual Machine, which is not able to impact the primary machine, ensuring an unprecedented security, since the users won't be exposed to network-level attacks.

The same applies to data storage, here defined as Storage Domain: workspaces must have their own disk space, obviously, in order to memorize software, data and whatnot. All the pseudo-partition share the same file system in read-only mode, thus avoiding compromises, and centralizing updates at the same time. By default, the entire file system comes already encrypted during the first installation.

10.2.1.3 Why use Qubes and not Tails OS?

As usual, let's assume that everyone will choose their preferred solution: users who prefer Tails will be probably looking for a scenario that's totally different than their usual, day-by-day tasks; Tails, in fact, allows to have a safe environment, applicable to the most common anonymous navigation operations; usability, however, is both its intrinsic limitation and main strength. Like many other Live distros, Tails OS is designed for hit-and-run operations that are not always ideal for your typical tasks.

An expert GNU/Linux user may need the security offered by Tails (as far as possible) together with the possibility of relying on a convenient environment, in order to work without having to reboot the Operation System every time. Qubes OS is the answer to such need, ensuring both isolated workspaces and the convenience of a standalone Operating System. Since it's all about individual choices, comparing features and electing the best solution would be pointless.

The bottom line is: both the Operating Systems are good for their purposes.

- Tails is designed to *stay anonymous*. When launched in a computer, it works so that no traces are left in it, changing the network adapter Mac Address

and redirecting all the traffic to TOR.

- Qubes is designed for *everyday use*. The final goal is to protect the user against any type of cyber attacks. It comes without any anonymity tools by default (except the Whonix integration), so it has to be customized by the final user.

10.2.2 Qubes OS + Tails

I guess you got disappointed when I said that Qubes is not an anonymity-designed Operating System, and you wondered: “so, why are you mentioning it?”. But, then, reading the title of this chapter you may have got excited, or (more probably) looked back to my statement: “hey, don’t use Tails on Virtual Machines!”, therefore I have to give you a reply.

It’s true that Tails should not be used in VMs, and I already explained the reasons why, the same stated by the developers: the most important is the persistence – or, better, reminiscence – of Operating System data, which remain stored in the disk. As we saw, however, Qubes uses a para-virtualization logic that completely destroys the stored data, thus facilitating the drive sanitation tasks. Consequently, the Qubes OS environment is fit for Tails virtualization, and the related steps are quite easy to follow^[147]. This way, you can leverage the power of Qubes OS together with Tails OS – as they say, killing two birds with one stone!

10.2.3 Qubes OS + Whonix

Using Tails in Qubes allowed us to understand how to virtualize a full Operating System within a Xen para-virtualized system; this, however, can be considered as a limit, becoming reality when you want to use the tools in Qubes instead of the ones virtualized in Tails.

Furthermore, only the Tails environment ensures enough security to maintain your Anonymity. Then, we need to create a further level (just like we saw with the Network and Storage Domains), allowing us to redirect traffic to a safe and anonymous communication channel. Whonix is another GNU/Linux distro based on Debian and Tor that leverages two Virtual Machines: a *Gateway* and a *Workstation*.

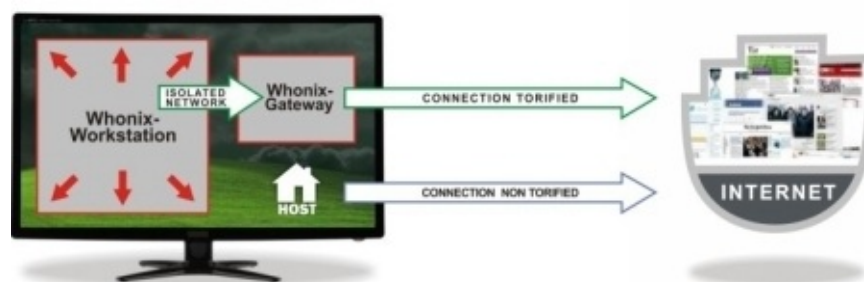


Figure 49: Whonix operation diagram

As we can see in Figure 49, the *Workstation* is an environment that allows us to work within an isolated area from the *Gateway*, a Virtual Machine already designed to connect via Tor. Nonetheless, keep in mind that Whonix has the same security limitations as we mentioned in the “Tor” chapter; additionally, unlike Tails, it is not a ready-made operating system, since you need to be familiar with the GNU/Linux environment in order to use it.

The downside of such difference is the *lack of some features* that make Tails more beneficial, like the following:

- The lack of pre-configured Mac Spoofing
- The lack of software “amnesia”, namely all the features aiming to remove any information in the computer
- The lack of metadata flushing
- The lack of a complete encryption at the mail level, due to the backwards compatibility with the SMTP protocol
- and more [\[148\]](#).

Some of these gaps can be filled through Qubes virtualization, others by applying certain techniques we already mentioned in this document. However, Whonix and Qubes are designed to be used from a fixed machine, the price to pay when you want usability rather than safety (I can assure you such balance is quite common in IT Security) [\[149\]](#).

10.2.4 Subgraph OS

We can define Subgraph OS as the latest addition to Operating Systems for privacy and anonymity. It’s still in Alpha, therefore just consider it as the draft of what it will become in the future.

Developers ensure that Subgraph OS will be a groundbreaking Operating System, and they be right to some extent: it has been designed to be a fast OS that can be also used in older computers, a safe system for users who are concerned about their privacy. The following Figure 50 is a diagram about the

Subgraph OS structure.



Figure 50: Subgraph OS working diagram

10.2.4.1 Hardened like few others

Subgraph OS is deployed by default with a ready-compiled kernel of Grsecurity, a set of patches ensuring a high level of security within the system. Grsecurity includes PaX, a component that detects many different OS attacks, like buffer overflows, using the ASLR technology to randomize memory allocations and obstruct any memory-level attack.

Subgraph OS also implies the same virtualization concept we already saw with Qubes OS: the purpose is to create isolated Sandboxes that cannot inter-communicate. In case of software exploits, the Operating System won't be impacted, making them ineffective. Such process is ensured by OZ, a sandbox framework specifically designed for Subgraph OS. If you wonder if Subgraph OS supports file system encryption, the answer is: sure it does! Furthermore, it is mandatory.

Most of the tools specifically written for Subgraph OS are high-level (probably interpreted and not compiled) so they can resist to memory attacks; additionally, the majority of the tools considered unnecessary have been removed, while the crucial ones have been integrated with security measures and, in some cases, even rewritten from scratch (like the default email client).

10.2.4.2 Network and Anonymity

Just like Qubes, we can find a networking domain: here it's called Subgraph Metaproxy and is accompanied by a Software Firewall. While the Firewall only allows the applications to connect to the Metaproxy, the latter is configured to connect each program to a single TOR relay, routing the connections across multiple channels and minimizing shared information in the network. In short, navigating the web and writing a mail will imply using two different TOR networks, and such prerogative will always be ensured by the Metaproxy. Back to the firewall, users can temporarily or permanently grant the network access to any software, removing any chance of System infection by a backdoor (unless it is already residing in a whitelisted process). App whitelisting is performed both by application name and target address; if a non-whitelisted application tries to connect, the Firewall will just kill the connection.

As you understood, Subgraph OS uses the TOR network to communicate with the external world: to be precise, it exclusively leverages the TOR network,

except for some scenarios where, for example, a direct communication to the visited portal is required (as a captive portal on a public WiFi network). Last but not least, Subgraph OS provides two custom software for communications security.

- Icedove, a Thunderbird-based client, powered by Enigmail (GPG) and TorBirdy (Anonymity via Tor)
- CoyIM, a XMPP client which was rewritten from scratch to avoid memory-level exploits, also exclusively designed for the sole TOR network

10.3 Pentest Distros

Probably, you already know the pentest-grade Linux distros: if not, Pentesting is an abbreviation for Penetration Testing, a branch of IT Security. Penetration Testing implies the assessment of the overall security of an IT structure and the surrounding environment: network, Operating System, programs and so on.

In years, Linux community have shown some interest in this ecosystem, developing distros with pre-configured applications to accelerate the testing operations, provide standardized environments and gather users with the same interests under a single spotlight. We will explore and use them in the next Hacklog volumes, when we will cover the cyber attacks; for now we will only provide a list:

- Kali Linux, based on Debian (<https://www.kali.org>)
- Backbox, based on Ubuntu (<https://backbox.org>)
- Parrot Security OS, based on Debian (<https://www.parrotsec.org>)
- DEFT, based on Debian (<http://www.deftlinux.net/it/>)
- Pentoo, based on Gentoo (<http://www.pentoo.ch>)
- NST, based on Fedora (<http://networksecuritytoolkit.org/nst/index.html>)
- BlackArch, based on Arch Linux (<https://blackarch.org>)
- Fedora Security Lab, based on Fedora

(<https://labs.fedoraproject.org/it/security/>)

- Cyborg Hawk Linux, based on Ubuntu (<http://cyborg.ztrela.com>)
- WeakerThan, based on Ubuntu (<http://www.weaknetlabs.com>)
- Samurai Web Testing Framework, based on Ubuntu (<http://samurai.inguardians.com>)
- Bugtraq (<http://bugtraq-team.com>)
- Knoppix (<http://www.knoppix.org>)

11. Online Identity

Now we have all the tools and the competencies needed to navigate in anonymity – please note that I wrote navigate, and not interact! The mere presence of TOR or any other technology in between doesn't mean you're totally safe; conversely, this sense of protection may be a double-edged sword for your real identity.

11.1 NEVER combine your identities

Regardless of the tasks you want to perform – whether in clearnet or deepweb – you must be able to *separate your activities* to avoid making connections and creating a fingerprint (is this term familiar?) of your identity. Leaving traces of your activities – email, bitcoin addresses, names, locations, etc. – allows to create a more detailed profile of the person of interest. In case someone manages to merge your two identities, they could double their information about you.

Let's get back to Ross Ulbricht, the late Silk Road admin, a portal that once allowed him – as well as many other people – to earn hundreds of thousands of dollars in the illicit market. Do you know how he got caught? When Silk Road was not famous yet, Ross was the first one asking across the clearnet if anybody knew that market – as you know, people do that to spam their websites. Together with other evidence, that episode pointed to Ross Ulbricht's identity (and consequently to other gang members).

11.2 NEVER use the same data

Even children know that a password should never contain your data (birth date, full name, location, etc.); you should instead use random alpha-numeric characters, numbers, special symbols and any other random input. You can use different programs, like [KeePassX](#) (integrated in Tails), LastPass, 1Password and others, to generate new passwords as well as store them with a master-key to unlock them all.

Due to security reasons, NEVER use a single keyring to store the passwords for your “normal” activities and those for the “alternate” ones altogether. As we mentioned above, you should never combine your identities! Besides the cases

where you would expose yourself consciously, you must also consider the traces you leave behind unknowingly:

- IP addresses
- Passwords
- Birth Date
- Billing Information
- Addresses and Locations
- Pictures and Similar Avatars
- Similar Contact Addresses
- ... anything that could point to you or even to your second/third/fourth identity and so on...

11.3 Watch Out for your Habits

If you often use some sayings, a particular dialect or Write Like This or make noticeable or repeated spelling errors, to the point that they could identify you with no doubt... do something about that!

Probably, you've already been told about these "peculiarities"; if you are particularly touchy, you could have never noticed them, but trust me, you can be identified just for the way you act or write.

Some time ago, we had a moderator who was removed for his negligence in some tasks. This person wanted to have "revenge", disrupting our communication channels with his residual powers; he did it using nicknames he often used among friends as well. Although we already obtained evidence by comparing IPs, it was enough to see his messages to identify him.

What about activity times? Are you predictable or do you work H24? Remember that – especially at the governmental level – monitoring is performed manually. Every text is thoroughly analyzed, and the writing person's character is profiled.

11.4 Disposable email

Disposable or *Temporary Email* services allow to create e-mail addresses that are not entitled to physical persons and don't require any registration.

They're partially designed to allow sign up for portals that require subscription without being included in annoying mailing lists sending SPAM materials for an indefinite period of time.

Some Temporary Emails allow to receive and send mails in complete anonymity; obviously, the user will lose the entire history of sent and received messages once the digital hourglass expires (some services allow to extend time at regular intervals).

Among the services allowing the sole mail reception, you can find:

- <http://www.throwawaymail.com>
- <https://temp-mail.org>
- <http://it.getairmail.com>
- <https://www.mailinator.com>
- <http://www.dispostable.com>
- <https://www.emailondeck.com/>
- <https://maildrop.cc>
- <https://10minutemail.com/>
- <https://www.mohmal.com/it>

while those also allowing to send mail include:

- <https://www.guerrillamail.com>
- <https://mytemp.email>
- <http://www.yopmail.com/en/>
- <https://www.crazymailing.com>

It's up to you to decide how you can use them. My only suggestion is: treat them as disposable sites, therefore they shouldn't be used for confidential information; furthermore, ensure to be properly anonymous before accessing them.

11.5 If you manage a Site/Blog/Forum

If you're mainly interested in anonymity because you wish to hide your identity, ensure to follow this guidance:

- If you're launching a new website, you may want to start with a free hosting service. There are lots of them out there, and some are designed with pre-installed CMSs (just like wordpress.com). Keep in mind that, if you wish to make a qualitative leap, probably you'll be offered some paid solutions. Anyway, you should prefer services that WON'T require billing information and also offer anonymous payment methods, just like Bitcoins (please, refer to the crypto-currencies chapter).

- It is quite improbable that the portal you're managing gets compromised; however, this is not impossible. Languages like JavaScript allow to apply stylometry techniques to analyze the number of keyboard inputs per minute and the mouse operations; such data may help someone obtaining useful information about the way you write, your skills, your frequent grammar errors, your punctuation style, your dominant hand and so on. In this case, it may be useful writing your articles or posts on a local editor first, then copying and pasting the text on the site you're managing.

- Especially in blogs, you can schedule when your articles must be published. That can be a good way to remove traces for who's trying to geo-localize you.

- Don't forget to erase all the metadata from the files you're uploading, in particular the EXIF Data from your pictures. Remember to manipulate your photos (please refer to the camera sensors chapter). If other persons are depicted, cover their faces.

11.6 Things you should NEVER do

The following list contains some instructions, or commandments, you should follow to avoid jeopardizing your work.

- NEVER navigate your personal website *while you're anonymous*
- NEVER access your social network account *while you're anonymous*
- NEVER access an account you've previously used without protections *while you're anonymous*
- NEVER access a bank/PayPal/eBay account or other sites that may contain your personal information *while you're anonymous*
- NEVER access an open WiFi network if you are not certain it's not monitored *while you're anonymous*
- NEVER underestimate the power of encryption *while you're anonymous*
- NEVER mistake anonymity with pseudo-anonymity
- NEVER use the phone verification *while you're anonymous*

12. Online Payments

Every true criminal have their own buying and selling business: skimmers, cards and documents, anonymous SIM cards, network adapters, and so on. It would be a shame if they got caught directly at home, wouldn't it? Obviously, buying online in anonymity must not be a cyber-crime world prerogative: nowadays, more and more people buy on the web, unaware that every order placed feed the biggest market analysis database ever. Buying online with no traces in clearnet became utterly impossible: orders are stored in the sellers databases, payments are managed by banks or traceable virtual payment circuits, shipments are handled by third-party companies that, due to legal requirements or, more simply, company policies, may decide to verify the content of a parcel without providing any valid reason.

12.1 Buying in the Dark Net

As we saw, Dark Net is the part of the Deep Web containing materials considered as illegal in some countries. Web criminals use the Dark Net to exchange information and products of any type; such products, obviously, cannot be traded in the "normal" market. We're not referring only to drugs, but also to weapons, stolen goods, pornography, credit cards forgery and duplication tools, prescription drugs, forged identity documents, website databases, 0day software exploits, and more. There's no point in warning you about the reliability of stores and sellers from the Dark Net: since the tools to navigate and purchase are also designed to ensure anonymity, just consider that frauds and scams happen all the times.

12.1.1 Dark Net Markets

Actually, the “dark” community in the Clearnet – reddit^[150], 4chan^[151] and many others – constantly leave feedback and links to the new selling nodes; furthermore, especially since *Silk Road* (a famous illegal marketplace) has been shut down, the new selling channels have grown almost ten fold.

12.1.1.1 Types of Dark Net Markets

In the Dark Net, you can find different types of markets, not always sharing the same transactional method. In years, we saw different types of markets we can summarize in five broad categories:

1) Centralized Markets: stores where buyers and sellers use a shared cryptocurrency wallet. They are extremely dangerous, since the managers can decide to lock an account and then cheat both parties. This category also includes the world-famous Silk Road.

2) De-Centralized Markets: a new type of market – still at the design stage – based on a transactional system that doesn’t require external navigation tools like TOR and others. Currently, the most ambitious projects are Bitmarkets^[152] and OpenBazaar^[153], which are still little used and under development

3) Forum Market: the most popular ones, since they can be created and maintained simply with a forum software. They’re totally identical to normal forums, but also contain listings announcements. Most of them also come with escrow services (selling methods where a third party arbitrate the transaction) or sell VIP verification packages, to limit frauds.

4) Multi-Signature Market: in these types of markets, the transaction takes place within a shared wallet (just like centralized markets), but here, in order to close the deal, two out of three involved parties (buyer, seller and moderator/middleman) must accept the exchange, so that they can be (almost) certain of a successful transaction.

5) Single Sale: this category includes freelance sellers offering any kind of service through their websites. They’re also the ones who, defining their own rules, perform the highest count of frauds over the Dark Net.

You must consider that some markets can be accessed by invitation only, so

being connected through the same anonymous circuit they're distributed with is not enough.

12.1.1.2 Where to find the Dark Net Markets?

Listing the currently online stores would make no sense, because they can last a couple of weeks up to some month, while this manual (hopefully) will not be updated before a couple of years! You should also consider that, after the attacks against different anonymous networks, the Dark Net Markets use to shift from a system to another; in the case of Silk Road, for example, we've seen its resurrection both on TOR and on I2P.

For this reason, I beg your pardon if I won't be able to update the list from time to time; however, you can follow one of the portals below:

- Darkwebnews (<https://darkwebnews.com/dark-web-market-list/>)

- PsychonautWiki
(https://psychonautwiki.org/wiki/Comparison_of_darknet_markets)

By the way, it's also worth mentioning Grams, a *search engine* exclusively dedicated to the Dark Net Markets (only available on TOR network at this moment; search for the correct .onion address on Internet).

12.2 *Crypto-currencies*

Concerning payments, instead, crypto-currencies are the method accepted by the online community: such currencies are acknowledged across different places (physical regions included) and, with the proper precautions, can make the user untraceable.

12.2.1 Precautions with Crypto-currencies

Obviously, it depends on the crypto-currency, but you should keep in mind that transfers are within the public domain in most of the structures of this kind, so that, if the owner of an address is identified, all their transactions and those of their contacts are exposed as well. The most stupid thing you can do is telling everyone about a payment address (called wallet) which is used for questionable or even illegal transactions.

12.2.2 Bitcoin

Bitcoin is the most popular crypto-currency in the web, and allows to make transactions anonymously, with the proper precautions, outside the control of States and Banks; the BTC technology is based on a de-centralized network, in order to avoid any possible manipulation of the network and attacks against the infrastructures storing the crypto-currency.

We will cover Bitcoin since it's the most socially accepted currency on the web: there are many others (I recommend to keep a eye on Ethereum^[154]), each with their own features, however it would be pointless mentioning dozens of crypto-currencies that may vanish at any moment.

12.2.2.1 How Bitcoins work

Bitcoins are stored in a Wallet and can be transferred just like any other online e-banking service – if you've ever used PayPal you know what I mean – but they do not depend on any bank, are not taxable and, to some extent, are also anonymous. Going back to wallets, they can be software and web-based. Actually, you can also find some hybrid types (like blockchain.info) that allow to access the wallet both from a locally installed program and a web interface. If you don't trust external wallets, you can always install the client from the official website (bitcoin.org) or use alternate ones (like electrum.org) but it will take

some time before the wallets get aligned with the Bitcoin network.

Each wallet is matched with an address, a unique alpha-numeric code identifying the wallet in the network, somewhat equivalent to a telephone number, that will send or receive Bitcoins. The address is generated during the first installation of the program or in the service subscription stage; furthermore, you can have more wallets simultaneously, and exchanging Bitcoins between them free of charge. Wallets must be secured by a password and a passphrase: these elements ensure that the only legit owner can use them, and allow to use the wallet even only temporarily within an Operating System. It is then advisable to make regular backups of your wallets and encrypt them considering what we explained in the “Encryption” chapter.

Let’s revise some general rules for Bitcoins:

- Bitcoins are digital: BTCs cannot be printed on paper (or, at least, they’re not officially acknowledged).
- Bitcoins are distributed: there are no servers managing Bitcoins.
- Bitcoins are divisible: having 1 BTC today means having hundreds of dollars. The most commonly used currency is the mBTC (it worths 0.001 BTC).
- Bitcoin is opensource: the software source code is open for modifications and available to everyone
- Bitcoin is (almost) anonymous: all transactions are public, but only addresses can be identified. If their ownership is known, privacy will be compromised.

12.2.2.2 How to obtain Bitcoins

Essentially, you can get Bitcoins in two ways:

•*Generating them*: in the IT world, this is known as *Mining*. Crypto-currency emerged as a distributed currency, therefore it is “mined” to be created (hence the use of the term ‘mining’). Downloading the mining software, you can create an actual currency you can use to purchase goods and services. Since it became quite popular, however, many experts and companies aim to generate more and more crypto-currencies, thus obstructing any form of competition.

•*Purchasing them*: this is obviously the simplest method. You can find many different crypto-currency markets, trading any type of currency (Bitcoin, Litecoin, Anoncoin, Primecoin and so on) for real money. One of the most popular is LocalBitcoins.com, which allows to contact other people of your town (or country) and purchase them using different payment methods: Debit card, Bank Transfer, PayPal, Western Union and so on. Some forums (like inforge.net) allow to purchase them from different users through the escrow method.

•*Exchanging them*: Bitcoins can also be used as a traded commodity for other currencies (including real money). Sites like BTC-E.com, bitstamp.net, coinbase.com and others offer purchasing services, including crypto-currencies. Unlike the transactions among private parties, these sites require users personal information like passports or driving licenses, actually threatening the users privacy. Furthermore, you can find many services managing their platforms with online wallets: one of them, Mt. Got, unexpectedly ceased its services in 2014, causing a loss of \$ 387 millions to their customers; its CEO was also involved in a fraudulent bankruptcy scandal.

12.2.2.3 Making Bitcoins untraceable

Bitcoin is often called “the anonymous digital coin”. This statement is wrong for two reasons:

- 1.It is not a coin, but a currency
- 2.It is not anonymous (not intrinsically)

Each Bitcoin transaction is traced: you can verify it by visiting the BlockChain.info. If you purchase Bitcoins, you will see your transaction on the site. This can be a problem for your privacy, hence you may need to hide the traces of your Bitcoins.

For the sake of clarity, if we have a website where we ask for donations showing the address next to our full name and then we purchase some marijuana from the Dark Net... well, everyone will know we smoke pot!

Mixing Service

One of the methods to “clean” BTCs is using a shared Bitcoin container known as a mixing (or tumbler) service: all users of a service put their Bitcoins

together, then they roll them out onto different transactions, changing the game and then deciding the amount to withdraw from the online wallet. The system works depositing a certain amount: a 1-3% fee is applied and it will take at least 6 transactions before returning available.

In other words, a registered user sends some Bitcoins to a mixing service: the latter will roll them out, making them anonymous. Once this is done, the user will receive them on a wallet they consider as the safest.

You can find many different interesting projects providing this service on the web: Helix by Grams, bitcoinblender.net, bitcoinmix.org, *PayShield*, bitcoin-fog.org, coinmixer.se, coinmixer.net, spacechain.io are just a few.

Their quality and reliability depend on the service trustworthiness, and especially on the way they are used. Each mixing service manage transactions their own way, as well as the percentages and everything concerning timing and graphical interfaces; anyway, the logic behind the “Bitcoin laundering” is essentially working as follows:

Be careful! When you connect to a mixer, always use the .onion links and NEVER the clearnet ones! From the site (or Internet) search for the related address.

- 1) Create a wallet in the clearnet (called wallet#1)
- 2) Send BTCs – by purchasing or transferring them – to wallet#1
- 3) Create a second wallet, this time through TOR or similar circuits (wallet#2)
- 4) Send the Bitcoins from wallet#1 to wallet#2
- 5) Send the Bitcoins from wallet#2 to the Mixer-created Bitcoin address
- 6) Create a new wallet (wallet#3) and receive your Bitcoin there
- 7) (Optional) In case of transactions, you can directly send Bitcoins from the #2 to the seller using the mixing service
- 8) From wallet#3, you can withdraw your clean BTCs and then send them to

wallet#1

9) From wallet#1, you can use the BTCs to make anonymous purchases or also in Clearnet

Treat the mixer services exactly as the VPNs, you should not trust them completely! Mixer services could log your transactions and compromise your anonymity..

CoinJoin

The Bitcoin world offers another way to clean them: CoinJoins is a compression method for Bitcoins transactions, designed to increase the parties privacy, removing unnecessary information from them. CoinJoins was created since the Bitcoin, often advertised as an anonymous payment method, is not 100% secure at all, actually; instead, we may say that its less anonymous than banks: at least, the latter won't publicly share transactions!

The CoinJoin method simply consists of connecting to a server that acts as a gathering point for people joining the same transaction: this way, it will be far more difficult to analyze all the circulating currencies; furthermore, unlike mixing services, Bitcoins cannot be stolen. Besides anonymous purchases, crypto-currencies can also be used for other purposes: the most credited methods include tax evasion and money laundering. Especially for the first reason, the Bitcoin was banned by different countries (Italy is excluded to date) and is under review by the largest world banks that, together with governments, are deciding about its fate.

It would make sense thinking that, in a not too distant future, this currency may become totally illegal, since it can be used instead of official currencies – although crypto-currencies are not real money – and this would make the entire banking system crumble down. But this is another story...

12.2.3 Beyond Bitcoin

The concept behind the crypto-currencies is still a recent reality, and, rightly, it will take years before we get a point of reference. At the moment, the latter is seemingly represented by Bitcoins, but the market tends to look elsewhere, also expecting the strong saturation of blockchains, the BTC market de-centralization and considering its limitations. There are some fascinating alternatives, then, like

Litecoin, Dogecoin, Quarkcoin, Primecoin, Peercoin, and so on.

The future of Bitcoins, however, seems to be less bright than expected: new investors are focusing on Ethereum, a new way of trading, that solves the problems and limitations of BTCs, rightfully earning the appointment as Bitcoin 2.0. For Anonymity purposes, we don't need to further explore alternative crypto-currencies, since all of them share part of the functioning logic and we are uncertain of when the Bitcoin will be replaced and by what. Only time will tell.

13. Be Free

Now you're ready to enjoy your full freedom on the web, outside of the scope of any organization and company that used you as a specimen for their experiments.

Yes, I may be exaggerating a little, but I think that it is worst believing to be free than being aware of the opposite, to some extent. Being free from the shackles of statistics, markets and analysis, from your government that doesn't want you to like certain things, from the zombies surrounding you and looking at you like a parasite only trying to be yourself.

I wanted to write this book so that you can be free and live without the constant fear that, one day, this dream may come to an end. Here's why this book is free and it will always be.

Now it's your turn: will you fight to protect this freedom? What will you do, from now on, to make things change? If you want to fight this struggle with me, please share the book with your friends, let me know your opinion and support the projects and the struggles we've been fighting for years to be free, at least on the web.

Now go and enjoy your freedom. And don't allow anyone to stop you.

Acknowledgments

Authors and Collaborators

Testi, Progettazione ed Esecuzione
Stefano Novelli

Translator
Marco Stefano Doria

Proofreader
Marco Silvestri

Audio (web series)
Mirko Marcattili

Distributed and promoted by
inforge.net - *your hacks community*

Sources & Resources

• wikipedia.it for the huge amount of information, especially about the technical parts

- deepdotweb.com and in particular the Jolly Roger's Security Guide for Beginners that I used for the stories of the featured cyber-criminals

- torproject.org for the wikis explaining the TOR network architecture

- privacytools.io for the report about mass surveillance and the summary of the key points

- *Source Sans Pro, Oxygen, Roboto Slab e Ubuntu Mono* are the fonts used for this book

Special Thanks

The success of this project has also been made possible by some of the most important IT portals, which offered their visibility across the media. Without them, Hacklog: Volume 1 would never have reached this important goal. Thank you again.

Donors

The Hacklog: Volume 1 was made possible by the monetary contributions of the hereby mentioned people, from the Hacklog Indiegogo campaign.

	Donors Diamond	
Lorenzo Pulcini	Roberto Talamonti	Francesco Buccoliero
Alex Fegatilli (Faustino50)	ddarix	Michele Colazzo
luca bizzotto	Kornel Roman	Mario Consorti
Francesco Pishedda	Gianluigi Frau	Andrea Sorrentino
Giorgio Vitale	Cristiano Alex Rado	Domenico Versace
Zanotti Andrea	Edoardo Piergentili	Luciano Barbato
Tony Fanara	heleentje64	Francesco Pvk
« MoMy »	Rossato Fabio	Federico Bevilacqua
Luca Baglivo	camap	Laempo L

	Donors Platinum	
Matteo Pernarella	Oscar Accorsi	Nicola Camodeca
Alessandro Di Franco	Gero DotNet	and.mariani
Riccardo Bassignani	Japo Jacobowski	Andrea Azzalin
Christian Paolini	Matteo Locatelli	Matteo Marangon
Simone Errico	Damiano Marchi	

	Donors Gold	
Giovanni Mangano	Pietro Ricotta	White Black
Luca Di Grazia	Pinco Pallino	Riccardo Tavano
stefano carbonaro	Davide Caputo	Tiziano Colagrossi
Salvatore Adduci	yohni makaroni	Stefano Formicola

Simone de Blasiis	Davide Zavarella
-------------------	------------------

	Donors Silver	
Michele D.	Maurizio Parton	Ciro Rutigliano
angelo.pampalone	Alberto Boto	Giuseppe Biscardi
Luigi Clemente	Matteo Chiaffitella	Luca iadicicco
Antonio Erriquez	Massimo Martini	Giovanni La Cascia
Martin Di Donna	Henry Every	

	Donors Bronze	
Daniele De Falco	Emilie Rollandin	Alessio Anzelotti
Alessandro Genova	Alessandro Genova x 2	Gennaro Grieco
Tommaso Padovano	lorenzo gregori	Antonio Silvestre

Kaiyan Chen	Davide Gabrielli	Vincenzo Di Domenico
andbri, Umbertide	micheleeee92	

	First 50 Books	
Fabio Pagnini	Laempo L	Vittorio Zamboni
Salvatore Corvaglia	Francesco Buccoliero	Francesco Ciucci
Damiano Grillo	Giuliano De Santis	Amedeo Gagliardi
alderuccio lino	Alberto Biasibetti	Alessandro Di Franco
Davide Uberti	Francesco Strippovano	Christian Perron
Giuseppe delogu	Federico Gervasoni	Lorenzo Colombo
Gianluca Giorgio	Francesco Gianchino	Giovanni Niro
Daniele Piccoli	Luigi Versitelli	Cristian Gentilezza
Federico Rocchi	Ivan Trentinaglia	Davide Scano

Alessandro Zungrone	Leonardo Aschieri	elia frigieri
redfenix45	Salvatore Scotto	Carlo Fanciulli
Fabio Vezzano	Federico Zorzi	Luca Verzani
lanfra94dani	Kirill Kuchmakra	Enrico Dametto
Francesco Bodria	Giovanni Bertozzi	Emanuele Libori
Giuseppe Capovilla	Tommaso Saglietti	Riccardo Bragadin
francesco carandini	Daniele Nuzzo	Giorgio Palombini
Roberto Perra	Gabriele Pollice	

[1]

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

[2] <http://standards-oui.ieee.org/oui/oui.txt>

[3] https://en.wikipedia.org/wiki/MAC_address#Notational_conventions

[4] <http://slagheap.net/etherspoof/>

[5]

<https://duckduckgo.com/?>

[q=how+to+create+private+dns+server&t=h_&ia=answer](https://support.opendns.com/forums/21618374)

[6] <https://support.opendns.com/forums/21618374>

[7] <https://dnsleaktest.com/how-to-fix-a-dns-leak.html>

[8] <https://help.ubuntu.com/community/AptGet/Howto>

[9] <http://proxychains.sourceforge.net>

[10] <https://github.com/rofl0r/proxychains-ng>

[11] www.proxycap.com

[12] https://en.wikipedia.org/wiki/Comparison_of_proxifiers

[13] <https://www.spamhaus.org>

[14] <https://www.barracuda.com/homepage>

[15] <https://duckduckgo.com>

[16] <https://startpage.com>

[17] <https://www.ipredator.se>

[18] www.mail-archive.com/cryptography@metzdowd.com/msg12325.html

[19] <https://nohats.ca/wordpress/blog/2014/12/29/dont-stop-using-ipsec-just-yet/>

[20] <http://www.mikrotik.com/software>

[21] [https://en.wikipedia.org/wiki/PRISM_\(surveillance_program\)](https://en.wikipedia.org/wiki/PRISM_(surveillance_program))

[22] https://en.wikipedia.org/wiki/Copyright_law_of_the_European_Union

[23] monicabarratt.net/a-discussion-about-dark-net-terminology/

- [24] <https://www.torproject.org>
- [25]
- <https://www.torproject.org/docs/debian>
- [26] www.privoxy.org
- [27] <https://www.torproject.org/projects/torbrowser.html.en>
- [28] <https://blog.torproject.org/blog/tor-browser-55a4-hardened-released>
- [29] <https://blog.torproject.org/blog/bittorrent-over-tor-isnt-good-idea>
- [30] <https://github.com/prof7bit/TorChat>
- [31] www.sourcemap.com/?page=torchat
- [32] <https://www.torproject.org/docs/tor-relay-debian.html.en>
- [33] <https://www.torproject.org/docs/pluggable-transport.html.en>
- [34] <https://github.com/Yawning/obfs4>
- [35] <https://bridges.torproject.org/bridges?transport=obfs4>
- [36] <https://trac.torproject.org/projects/tor/wiki/doc/meek#Quickstart>
- [37] <https://github.com/NullHypothesis/scramblesuit>
- [38] <https://blog.torproject.org/blog/recent-and-upcoming-developments-pluggable-transport>
- [39] cacr.uwaterloo.ca/techreports/2012/cacr2012-08.pdf
- [40] <https://github.com/blanu/Dust>
- [41] https://realworldcrypto.files.wordpress.com/2013/06/shrimpton_rwc-2014-fte-release.pdf

- [42] <https://www.torproject.org/docs/bridges.html.en#FindingMore>
- [43] <https://torcheck.xenobite.eu/index.php>
- [44] <https://blog.torproject.org/blog/did-fbi-pay-university-attack-tor-users>
- [45] <https://www.java.com/it/download/>
- [46] echelon.i2p/nachtblitz/
- [47] localhost:7657/susimail/susimail
- [48] <https://geti2p.net/it/faq#outproxy>
- [49] 127.0.0.1:7657/news - after connecting to I2P
- [50] i2pwiki.i2p/index.php?title=I2Pd - after connecting to I2P
- [51] <https://www.java.com/it/download/>
- [52] localhost:8888/plugins/ - you must use Freenet
- [53] All the shown directories are contained in Enzo's Index.
- [54] localhost:8888/chat/ - you must use Freenet
- [55] localhost:8888/plugins/ - you must use Freenet
- [56] <https://github.com/grugq/portal>
- [57] <https://openwrt.org>
- [58] <https://dd-wrt.com/site/>
- [59] https://tails.boum.org/blueprint/vpn_support/
- [60] <https://github.com/CrowdStrike/Tortilla>
- [61] www.crowdstrike.com/community-tools/

[62] [https://msdn.microsoft.com/en-us/library/windows/hardware/ff540213\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/hardware/ff540213(v=vs.85).aspx)

[63] <https://www.eff.org/it/https-everywhere>

[64] <https://www.washingtonpost.com/news/the-switch/wp/2013/12/10/nsa-uses-google-cookies-to-pinpoint-targets-for-hacking/>

[65] <https://www.ghostery.com>

[66] <https://noscript.net>

[67] <https://helpx.adobe.com/flash-player/kb/disable-local-shared-objects-flash.html>

[68] jdownloader.org

[69] <https://w3techs.com/technologies/details/cp-javascript/all/all>

[70] https://www.cvedetails.com/vulnerability-list/vendor_id-53/product_id-6761/Adobe-Flash-Player.html

[71] <https://subgraph.com/orchid/index.en.html>

[72] <https://en.wikipedia.org/wiki/WebRTC#Support>

[73] torrentfreak.com/huge-security-flaw-leaks-vpn-users-real-ip-addresses-150130/

[74] <https://diafygi.github.io/webrtc-ips/>

[75] <https://chrome.google.com/webstore/detail/webrtc-network-limiter/npeicpdbkakhahjeeohfdhnlpdkia>

[76] <https://chrome.google.com/webstore/detail/scriptsafe/oiigbmnaadbkfbmpbfijflahhl=en>

[77] <https://addons.mozilla.org/en-US/firefox/addon/happy-bonobo-disable->

[webrtc/](#)

[78] <https://panopticklick.eff.org>

[79] <https://fingerprint.pet-portal.eu/?menu=6>

[80]

<https://chrome.google.com/webstore/detail/stopfingerprinting/kfhlgmfkolojpmh>

[81] browserspy.dk

[82] cdimage.debian.org/debian-cd/current-live/amd64/iso-hybrid/

[83] implbits.com/products/hashtab/

[84] <https://www.gnupg.org>

[85] <https://gpgtools.org>

[86] <https://www.gpg4win.org>

[87] <https://www.gpg4usb.org/download.html>

[88] <https://addons.mozilla.org/it/thunderbird/addon/enigmail/>

[89] <https://www.mailvelope.com>

[90] <https://gpgtools.org/gpgmail/index.html>

[91] www.thialfihar.org/projects/apg/

[92] <https://www.streak.com/securegmail>

[93] <https://wiki.debian.org/Icedove>

[94] <https://veracrypt.codeplex.com>

[95] <https://veracrypt.codeplex.com/wikipage?title=Downloads>

[96] mhogomchungu.github.io/zuluCrypt/

[97] <https://it.wikipedia.org/wiki/FreeOTFE>

[98]

<https://gitlab.com/cryptsetup/cryptsetup/wikis/FrequentlyAskedQuestions>

[99] www.spammimic.com/encodepw.shtml

[100] cygwin.com/

[101] <https://www.itefix.net/cwrsync>

[102] citpsite.s3-website-us-east-1.amazonaws.com/oldsite-htdocs/pub/coldboot.pdf

[103] <https://mat.boum.org>

[104] <https://gitweb.torproject.org/user/jvoisin/mat.git>

[105] <https://mat.boum.org/files/>

[106] www.exifsoftware.com/free-photo-viewer/

[107] www.irfanview.com

[108] Through WineBottler

[109] Through Wine

[110] www.sno.phy.queensu.ca/~phil/exiftool/

[111] www.geosetter.de/en

[112] exifeditorapp.com

[113] <https://www.relliksoftware.com/exifdatechanger/download>

[114] www.digitalconfidence.com/downloads.html

[115] www.superutils.com/products/exifcleaner/

- [116] www.photome.de
- [117] ws2.binghamton.edu/fridrich/Research/EI7541-29.pdf
- [118] <https://www.bleachbit.org/download>
- [119] www.piriform.com/ccleaner
- [120] www.glarysoft.com/glary-utilities/
- [121] <https://www.cmcm.com/en-us/clean-master-for-pc/>
- [122] https://en.wikipedia.org/wiki/Parallel_ATA#HDD_passwords_and_security
- [123] <https://www.piriform.com/ccleaner/download>
- [124] <https://sourceforge.net/projects/eraser/>
- [125] www.freewarefiles.com/downloads_counter.php?programid=91261
- [126] www.codyssey.com/apps/utilities/freeraser.html
- [127] www.gaijin.at/en/dlwipefile.php
- [128] <https://www.ascomp.de/en/products/show/product/secureeraser/tab/description>
- [129] <https://itunes.apple.com/it/app/permanent-eraser/id500541921?mt=12>
- [130] inux.die.net/man/1/shred
- [131] https://tails.boum.org/doc/encryption_and_privacy/secure_deletion/index.en.html
- [132] www.dban.org/
- [133] www.dban.org/
- [134] www.hirensbootcd.org

[135] www.ultimatebootcd.com

[136] <https://falconfour.wordpress.com>

[137] www.system-rescue-cd.org/SystemRescueCd_Homepage

[138] https://it.wikipedia.org/wiki/Write_blocker

[139] www.caine-live.net

[140] www.cgsecurity.org/wiki/Download_TestDisk

[141] Refer to the full list at www.cgsecurity.org/wiki/File_Formats_Recovered_By_PhotoRec

[142] https://en.wikipedia.org/wiki/Cellphone_surveillance

[143] <https://tails.boum.org>

[144] https://en.wikipedia.org/wiki/Comparison_of_disk_encryption_software

[145] <https://www.qubes-os.org>

[146] <https://www.qubes-os.org/doc/windows-appvms/>

[147] <https://www.qubes-os.org/doc/tails/>

[148] <https://www.whonix.org/wiki/Warning>

[149] You can find the documentation on how to install Qubes + Whonix here: <https://www.qubes-os.org/doc/whonix/install/>

[150] <https://www.reddit.com>

[151] www.4chan.org

[152] <https://voluntary.net/bitmarkets/>

[153] <https://openbazaar.org>

[154] <https://www.ethereum.org>

[1]

qui