hackLOG

Manuale sulla Sicurezza Informatica & Hacking Etico

Volume 1 **Anonimato**

Stefano Novelli

AVVERTENZE

La violazione di un computer o rete altrui è un reato perseguibile penalmente dalla legge italiana (art. 615 ter del Codice Penale). Alcune delle procedure descritte sono da ritenersi a titolo scolastico/illustrativo/informativo e messe in pratica solo su dispositivi in nostro possesso o in ambienti di test controllati, pertanto il lettore solleva gli autori di questo documento da ogni responsabilità circa le nozioni assimilate durante il corso e le conseguenze verificabili.

NOTE SULL'OPERA

I contenuti di Hacklog: Volume 1 sono rilasciati gratuitamente per tutta la rete e disponibile in vari formati, secondo l'autoregolamentazione dell'ethical hacking e il rispetto delle realtà culturali che lo praticano.

Sei libero di poter prendere parti del documento per qualunque opera, citandone opportunamente la fonte (Hacklog di inforge.net) e possibilmente ove possibile con link ipertestuale in calce. Essendo un progetto che ha richiesto molto tempo ritengo che se il documento sia stato utile ai fini di progetti terzi venga condiviso per rispetto verso il sottoscritto, i suoi collaboratori e coloro che hanno creduto in esso.

DIRITTI D'AUTORE

Ai miei amici, ai miei cari,

a coloro che hanno reso possibile tutto ciò.

A tutti gli hackers del mondo

o aspiranti tali.

Stefano Novelli

GLOSSARIO

PREFAZIONE

Ciao e benvenuto nell'Hacklog, il corso gratuito di Sicurezza Informatica e Hacking Etico. Il mio nome è Stefano Novelli e mi occupo della stesura di questo corso: ho deciso di produrre questo documento per dare modo a chiunque di interessarsi alla Sicurezza Informatica in modo facile ma al tempo stesso con un format professionale, introducendoti a capire i modi di pensare e le tecniche utilizzate sia dai malintenzionati che dagli esperti del mondo della cybersicurezza.

Hacklog è il risultato di diversi anni di studio nel mondo dell'Hacking e dell'IT Security: sono raccolte testimonianze, tecniche e pensieri che circondano questo particolare mondo visto da diverse prospettive tramite documenti, corsi di formazione ed esperienze dirette nel campo della Sicurezza. Come abbiamo già detto poco fa l'Hacklog è un corso pensato per chi vuole conoscere e approfondire le tematiche che circondano la Sicurezza Informatica. Il manuale non mira a formare professionalmente un esperto di **IT Security** ma piuttosto a fungere da trampolino di lancio per capire questo mondo e per imparare ad effettuare da solo i tuoi test - e perché no, magari darti la giusta spinta per intraprendere questo ramo di studi. Tutto ciò per dirti che *nessun corso* online potrà mai insegnarti tutto ciò che devi sapere sulla Sicurezza Informatica: per questo esistono i Master Universitari che sono impegnativi in termini di tempo e di economia ma che sono gli unici veri metodi per intraprendere - professionalmente parlando e a parte rare accezioni - la via dell'esperto in IT Security.

Questo corso è pensato **per te** - studente o autodidatta - che vuoi conoscere cos'è l'Hacking Etico e la Sicurezza Informatica, che vuoi imparare le tecniche principali per eseguire test di sicurezza sulle tue macchine e come difenderti dai malintenzionati che brulicano in un marcio mondo oggi chiamato cybercrimine.

Se ti dicessi che puoi iniziare anche se non sai niente sull'informatica ti mentirei. Ti dirò di più: di "scuole" di Sicurezza Informatica in Italia ce ne sono davvero poche e quelle che valgono la pena di essere seguite sono molto costose e impegnative. Non è mia intenzione demoralizzarti, anzi voglio rincuorarti di una cosa: il fatto che tu sia qui è già un **ottimo inizio**! Questo significa che hai voglia di apprendere e ti assicuro che in questo settore è una risorsa estremamente importante, se non addirittura fondamentale.

Quello che ti chiederò durante il proseguimento di questo corso sarà di:

- Avere un atteggiamento positivo al corso (è importante non demoralizzarsi subito!)
- Informarsi maggiormente sugli argomenti che non ti sono stati chiari, usando strumenti come Wikipedia e simili
- Prendere appunti, se vuoi anche con carta e penna (ti aiuterà a ricordarti meglio gli argomenti di cui parleremo ed avere schemi visivi che ti rimarranno impressi nella mente)
- Confrontarsi con altre persone se non riesci a capire qualcosa (se vuoi puoi scrivere sul forum di inforge.net o qualunque altra community analoga)

Sappi però che daremo per scontati gli *argomenti fondamentali dell'Informatica*, come ad esempio la differenza tra hardware e software, che cos'è un sistema operativo, come scaricare programmi e cose così. Non ti chiedo troppo ma capirai da te che è praticamente inutile spiegare la Sicurezza Informatica a chi non sa neanche cosa vuol dire il termine informatica.

E ora iniziamo pure, ti auguro una buona lettura.

ANONIMATO

L'anonimato su Internet è un tema che negli anni si è rivelato di una certa importanza, tanto che ad oggi esistono strumenti di ogni tipo per evitare di lasciar **tracce**. Il bisogno di essere dei fantasmi in rete non è un'esclusiva dei cyber-criminali: in alcune parti del mondo (come in *Cina*, *Arabia Saudita*, *Iran* o *Nord Corea*) la censura governativa è talmente ferrea che diventa quasi obbligatorio utilizzare strumenti pensati per l'anonimato al fine di evitare il monitoraggio da parte delle agenzie di spionaggio - statali o private che siano. In alcuni di questi paesi vige ancora la *pena di morte* e la necessità di essere anonimi diventa spesso una questione di sopravvivenza.

Nel resto del mondo essere anonimi può essere utile anche in altre situazioni, ad esempio se si vogliono denunciare le condizioni di lavoro o le discutibili politiche interne di un'azienda ma anche essere **liberi** di poter navigare senza prender parte ad un sistema fortemente analitico, fornendo ai grandi colossi della rete informazioni su ciò che acquistiamo o vendiamo, su ciò che ci piace o non, alimentando così l'esperimento sociale di massa condotto dalle più grandi potenze mondiali (mappa in Figura 1).

L'anonimato è anche uno strumento fondamentale per gli **hacktivisti**, coloro che praticano *attivismo digitale*. Uno degli esempi di hacktivismo che ha scosso il mondo dell'informatica e l'opinione della gente comune che ha su di esso è indubbiamente il movimento *Anonymous*, che già dal nome, fa capire la necessità di essere non rintracciabili durante le proteste online.

Se il tuo intento è quello di mettere in **sicurezza** la tua struttura informatica in realtà vi è anche un'altra ragione, ossia facendo uso dell'anonimato come mezzo di *prevenzione* affinché tu non sia esposto alla rete e dunque potenzialmente violabile da chiunque. Se ti occupi di operare nel campo delle indagini informatiche vorrai invece conoscere quali sono gli strumenti che i cyber-

criminali usano per portare a compimento i loro attacchi, traendo in inganno chi investiga su di loro.

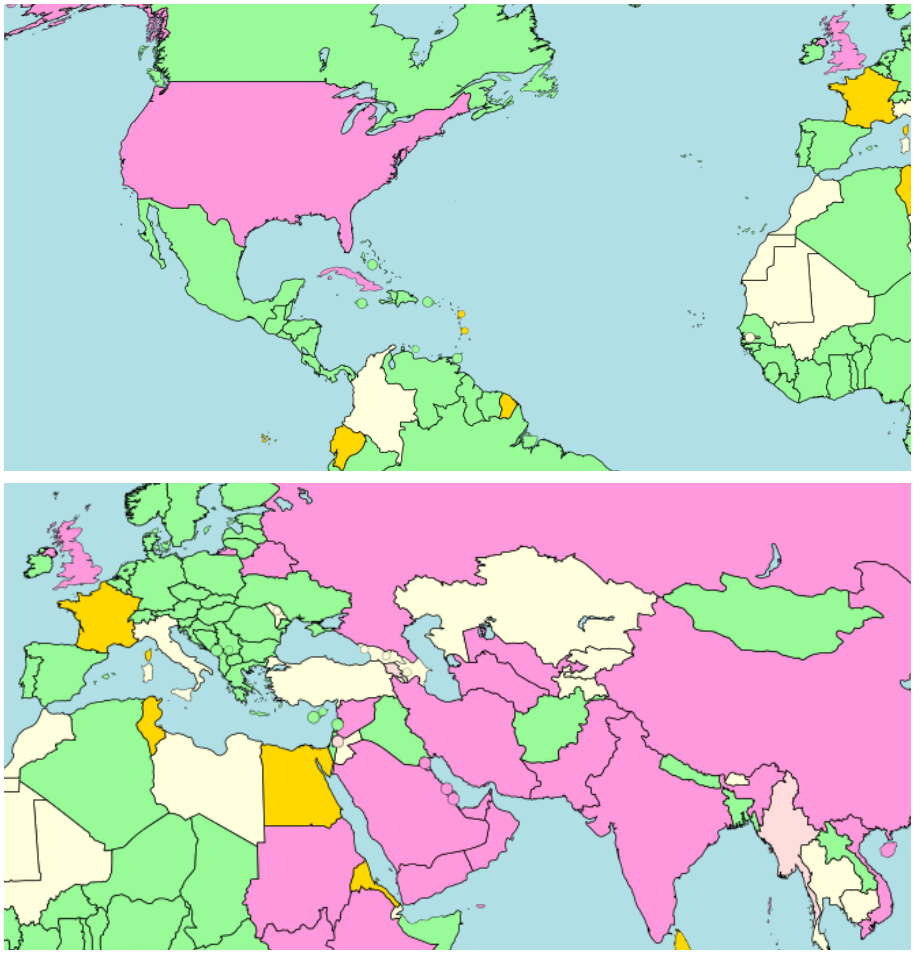


Figura 1: una mappa mondiale della censura e della sorveglianza adottati dai governi di tutto il mondo. L'Italia (in bianco) è classificata come "livello selettivo". Fonte ONI[1]

[1] https://en.wikipedia.org/wiki/Internet_censorship_and_surveillance_by_country

1. IL SISTEMA OPERATIVO

Quando usiamo un dispositivo informatico stiamo in realtà usando il *Sistema Operativo* che vi è installato: senza di esso, il computer rimarrebbe una scatola inanimata di cavi, condensatori e componenti elettronici che altrimenti non servirebbero a nulla. Il Sistema Operativo è il software che gestisce tutto all'interno di un computer: si occupa di comprendere ciò che l'utente scrive, cosa mostrare a video, avviare i programmi e così via.

Nel panorama Desktop esistono diverse famiglie di Sistemi Operativi; le tre principali sono: **Windows, macOS** (ex OSX) **e GNU/Linux**. Forse andrò un po' controcorrente nel dire che GNU/Linux non dev'essere l'unico Sistema Operativo che gli esperti di sicurezza dovrebbero utilizzare; anzi, ritengo che ognuno di essi abbia i suoi pro e contro e che sia adatto in base alle situazioni che gli si pongono davanti. Ciò che è certo, almeno per quanto riguarda l'Anonimato, è che il Sistema Operativo che più si avvicina al concetto stesso dell'essere anonimi è *GNU/Linux*.

GNU/Linux è un progetto opensource, il che vuol dire che non solo è liberamente gratuito ma anche modificabile, re-distribuibile ed esente da codice volutamente nocivo. Questa natura si sposa perfettamente con le esigenze dell'utente che vuole rimanere anonimo: si garantirà l'uso di un sistema che nasce senza distorsioni, non manipolato e più difficilmente monitorabile da parte di agenzie di spionaggio, governi, aziende del settore, malintenzionati e via dicendo. Il grande potere di GNU/Linux è l'estrema duttilità che consente a chiunque di creare la propria distribuzione: sono nate grandi community grazie a questo principio, addirittura intere aziende - come le più celebri *Red Hat*, *Novell* o *Canonical* - basano il loro fatturato sull'ecosistema del pinguino, garantendo ogni anno migliaia di posti di lavoro. E credetemi quando vi dico che esiste veramente un'infinità di distribuzioni: dalle storiche Debian o Slackware, a quelle più user-friendly come *Linux Mint* o *Ubuntu*, a quelle pensate per i videogiocatori come

Steam OS, per le produzioni *audio/video*, per stare dentro *microcomputer*, *Server*, *Firewall*, *Router* e via dicendo. Tra quest'infinità di distribuzioni troveremo anche quelle pensate per *l'Anonimato*.

1.1 Quale distribuzione scegliere?

Ho sempre pensato che non esiste la distribuzione perfetta per ogni cosa.

Ritengo che preferire una distribuzione GNU/Linux ad un'altra non debba essere solo una mera questione di software pre-installati ma anche e soprattutto alle esigenze e alle *conoscenze* che ha l'utente (tenendo conto anche della condivisione che l'utente ha con la filosofia del progetto).

Se non hai mai messo mano su una distribuzione GNU/Linux questo può essere il *momento migliore* per farlo! In alcune situazioni potrebbe essere necessario dover utilizzare comunque Windows o macOS: tratteremo anche questi Sistemi Operativi solo marginalmente.

Durante questo corso faremo uso principalmente di *Debian*, una distribuzione madre a cui si appoggiano le più popolari distribuzioni sulla rete come *Ubuntu*, *Linux Mint*, *Elementary OS*, *Kali Linux*, *Parrot Security OS*, *Backbox*, *Tails* e molte altre. Se è il tuo primo approccio ti consiglio di partire direttamente da Debian, imparerai molte più cose e ti sarà più facile - una volta presa la mano - saltare da una distribuzione all'altra.

In questo documento non spiegheremo come installare e far funzionare Debian: se hai acquistato il libro in formato cartaceo troverai un inserto denominato "*Manuale d'Installazione di Debian GNU/Linux*" (disponibile anche online gratuitamente su www.hacklog.it) dove si spiega come installare una versione funzionante di Debian e imparare a sopravvivere ai problemi più comuni. Se per qualche motivo Debian non ti piace o hai difficoltà a installare le tue periferiche, puoi provare con *Ubuntu* o *Linux Mint* che al loro interno integrano

driver proprietari e di più immediato utilizzo. Aldilà della posizione di alcuni elementi, i comandi che utilizzeremo funzioneranno anche per queste.

Se invece ritieni di essere abbastanza esperto nell'uso di una distribuzione allora non avrai problema anche ad utilizzare un'altra di queste sotto-distribuzioni o addirittura di un'altra famiglia. Verso la fine di questo libro troverai un resoconto completo di tutte le distribuzioni Linux pensate per l'anonimato (e in parte anche per il pentest), così da poter effettuare nuovamente i vari test utilizzando ambienti di lavoro specificatamente pensati per l'anonimato e non.

1.1.1 Le Virtual Machine

Se hai seguito i vecchi corsi dell'Hacklog saprai che c'è un modo più rapido e indolore per avere Linux nel proprio computer senza partizioni ovvero quello di utilizzare una Virtual Machine. La Virtual Machine è un tipo di macchina che si finge un computer completo ma che in realtà vive all'interno di un altro Sistema Operativo: questo garantisce una forte compatibilità del software e una maggiore usabilità del Sistema, tuttavia può risentirne di *prestazioni* e soprattutto espone l'utente a *seri rischi* per la sua sicurezza e privacy. Rispetto quest'ultimo punto troverai le motivazioni di tale affermazione al capitolo "Live OS" verso la fine del corso.

Essendo infine un ambiente virtualizzato il Sistema dovrà sottostare a delle regole imposte dal Sistema Operativo centrale, quindi potrebbero verificarsi problemi con software di anonimato. Per questi motivo l'uso delle Virtual Machines è da ritenersi non consigliabile per la messa in pratica della maggior parte delle tecniche qui presentate.

1.1.2 Le Live Distro

Andando avanti vedremo come può essere più sicuro utilizzare alcuni tipi di distribuzioni Linux che vengono distribuite solo ed esclusivamente Live, cioè che consentono il loro utilizzo senza l'installazione nel proprio PC. Sebbene siano estremamente utili ne parleremo solo alla fine degli argomenti tecnici poiché non consentono - esattamente come per le Virtual Machine - l'applicazione di alcune tecniche di anonimato.

1.1.3 Il terminale

Uno degli aspetti più importanti di questo corso sarà l'uso del **terminale**, un software installato di default in tutti i Sistemi Operativi. Sebbene faremo in modo che si creino meno possibili problemi, può capitare che il terminale possa comportarsi in maniera diversa in base al tipo di Sistema Operativo in uso. Questo è uno dei motivi per cui consigliamo di seguire *solo certe distribuzioni (a base Debian GNU/Linux)*, in questo modo sapremo anticipare le risposte di ogni Sistema Operativo ed eviteremo di incappare in problemi irrisolvibili.

Quando faremo uso della linea di comando useremo il programma denominato "Terminale". Il terminale si presenta all'incirca in questo modo:

```
$ 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 =
```

Esempio di output del terminale

Da questa schermata considera che dovrai scrivere solo **ping** www.inforge.net, trascurando i possibili dati che cambieranno in base a delle situazioni che non possiamo determinare. Non tener conto del simbolo del dollaro ($) iniziale, servirà solo a capire che è da lì che inizia una nuova linea.

> Tieni sotto mano questa pagina nel caso in cui ti sia perso in qualche meandro del Sistema Operativo!

Per conoscere i file e le directory presenti nel percorso in cui ci troviamo

```
$ ls
```

Per entrare in una cartella

```
$ cd {nomecartella}
```

Per tornare indietro di una cartella

```
$ cd ..
```

Per copiare un file

```
$ cp {nomefile} {nomenuovofile}
```

Per spostare o rinominare un file

```
$ mv {nomefile} {nomenuovofile}
```

Per creare una cartella

```
$ mkdir {nomecartella}
```

Per usare un editor di testo (useremo la combinazione CTRL+X per chiudere l'editor e Y/N per confermare la scelta di un'eventuale sovrascrittura):

```
$ nano {nomefile}
```

E via dicendo. L'uso del terminale prevede l'utilizzo di programmi che richiedono anche dei parametri indicati con il carattere - (meno): se vogliamo conoscere il funzionamento e i parametri permessi dal comando ls usiamo --help:

```
$ ls --help
```

Oppure richiamando il tool man:

```
$ man ls
```

Tieni inoltre presente che quando installeremo nuovi programmi su Debian useremo i comandi apt:

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

Sebbene non supportati ufficialmente da questo documento, potrebbe essere possibile installare lo stesso pacchetto nelle distribuzioni a base Red Hat (Fedora, CentOS etc...) con il comando:

```
$ yum install [nomepacchetto]
```

o anche su i sistemi a base Arch Linux con il comando:

```
$ pacman -S [nomepacchetto]
```

Questi e altri comandi dovranno essere sempre lanciati da root (amministratore). Il prefisso da utilizzare in questo caso è:

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

Se quest'ultimo non fosse presente sarà necessario effettuare prima il login da root con il comando:

```
$ su
```

2. TRACCE INFORMATICHE

Freschi di installazione Debian è arrivato il momento di conoscere quali sono le **tracce** che è possibile lasciare in rete. Quando parliamo di "tracce informatiche" ci riferiamo a tutti quei valori digitali che in qualche modo permettono di risalire alla nostra *identità*. Queste possono identificare il nostro *computer* oppure la nostra *scheda di rete*, elementi che possono incastrarci quando ad esempio ci ricolleghiamo in reti senza protezioni.

Nel peggiore dei casi se si utilizza il nostro contratto Internet ci sono buone probabilità che si risalga al *nome* e *cognome* dell'intestatario della *connessione*. Esistono diverse tecniche per risalire ad una persona anche dopo che questa ha navigato in anonimato: più avanti esamineremo in che modo è possibile che ciò accada e i relativi accorgimenti per evitare che succeda.

2.1 MAC Address

L'indirizzo **MAC** (acronimo di Media Access Control) è un codice univoco a 48bit che viene assegnato dai produttori sulle proprie schede di rete 802.x; questo codice viene scritto direttamente nella memoria *EEPROM* della scheda e viene utilizzato per la prima fase di autenticazione ad una rete locale da un dispositivo di rete – quale un router, uno switch o altro – che assegnerà poi un indirizzo IP locale.

Il MAC Address è quindi composto da 6 coppie di caratteri alfanumerici che comprendono i numeri da *0* a *9* e lettere da *A* ad F (la cosiddetta numerazione esadecimale, ossia a base di 16) ed è così rappresentato: **ab:bc:cd:de:ef:f0**. Le prime tre serie di numeri (ab:bc:cd) fanno riferimento al produttore: per

conoscere la lista dei produttori in base al prefisso è possibile consultare la lista degli standard IEEE[1].

Ora immagina di collegarti alla rete Wifi di un hotel o di una piazza pubblica: in questo caso ci sarà una struttura di rete predisposta al protocollo DHCP - un sistema che assegna automaticamente un IP locale al relativo MAC Address - consentendoti di navigare liberamente su Internet! L'importanza di non lasciar traccia di un MAC Address sta nel fatto che questa informazione viene memorizzata all'interno del dispositivo di rete che (non) sempre permette di eliminarne i log, pur essendone i proprietari; inoltre è probabile che questo MAC Address non rimanga solo ed esclusivamente all'interno del router/switch ma venga anche comunicato all'ISP (Internet Service Provider) che lo potrebbe memorizzare a sua volta nei propri database.

2.1.1 Determinare il MAC Address

Al fine di testare le prossime tecniche che ci permetteranno di modificare il MAC Address per prima cosa dobbiamo essere in grado di determinare qual è il MAC Address in nostro possesso. Per farlo si può utilizzare un tool da linea di comando presente in ogni sistema operativo (su Windows si chiamerà Prompt dei Comandi mentre per Linux e macOS sarà Terminale).

Il comando per Windows da lanciare è **ipconfig** mentre per macOS e Linux è **ifconfig;** quest'ultimo in realtà sta per essere definitivamente abbandonato dal software **iproute2** (evocabile dal comando **ip**). Tieni presente che tutti i comandi devono essere lanciati da root, quindi ricorda di usare il comando **su** per ottenere gli accessi di amministrazione. Ad ogni modo tutti i comandi permettono di

[1] http://standards-oui.ieee.org/oui/oui.txt

mostrare in che modo sono configurate tutte le schede di rete presenti nel 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 0xffffff00 broadcast
192.168.0.255
nd6 options=1<PERFORMNUD>
media: autoselect
status: active
```

Dove *{interface}* sarà il nome della nostra scheda di rete. Solitamente *eth0* sta per scheda Ethernet mentre *wlan0* sta per scheda Wifi. È possibile che questi identificatori siano diversi in base al numero di schede presenti nel nostro PC. Nel caso volessi verificarlo puoi vedere quali schede sono abilitate con il comando:

```
$ ip link show oppure ip a
```

Quello che dobbiamo riconoscere è il MAC Address che, come già detto, è composto da *6 coppie di caratteri esadecimali*, suddivisi da doppi punti. Nel nostro caso il MAC Address sarà 61:a8:5d:53:b1:b8 .

2.1.2 MAC Spoofing

Fortunatamente in (quasi) tutte le situazioni camuffare il MAC Address – nel gergo informatico effettuare il **MAC Spoofing** – risulta essere un'operazione particolarmente semplice ed indolore.

In **GNU/Linux** bisogna lanciare giusto un paio di comandi dal terminale:

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

Considera che impostando questo MAC Address il tuo computer non potrà più navigare in rete. Dovrai generare un MAC Address valido, cosa che non tratterò in questo caso in quanto abbastanza complesso[1]. Puoi riavviare il network manager usando il comando:

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

Useremo invece un tool che è presente nella maggior parte dei repository di distribuzioni GNU/Linux e permette di generare un MAC Address random. Il programma, che andrà preventivamente installato, è *macchanger.* Per installarlo su Debian lanceremo il comando:

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

Ci verrà chiesto se vogliamo cambiare il MAC Address da subito. Se selezioniamo *No*, possiamo comunque farlo manualmente con i tre comandi:

```
$ ifconfig {interface} down
$ macchanger -r {interface}
$ ifconfig {interface} up
```

In Linux il comando ifconfig permette non solo di verificare le configurazioni ma anche di comandare le schede di rete. Come abbiamo visto, con il comando

[1] https://it.wikipedia.org/wiki/Indirizzo_MAC#Formato_degli_indirizzi

ifconfig {interface} down abbiamo comandato alla nostra scheda di rete (nell'esempio riconosciuta con l'identificativo eth0) di spegnersi.

In questo modo possiamo utilizzare il comando *macchanger* per generare un valore random (grazie al parametro *-r*) e applicarlo alla scheda di rete eth0. Una volta effettuati questi passaggi siamo pronti a riattivare la scheda con il comando *ifconfig {interface} up*. Sentiti libero di sostituire il comando *ifconfig* con il nuovo *ip (iproute2)*. Se dovessi avere problemi di connettività puoi riavviare anche qui con il comando:

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

Nonostante sia un'operazione relativamente semplice esistono anche diversi script in rete in grado di automatizzare tutto il processo. Eccone alcuni:

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

Nell'universo **Windows** (Figura 2) esistono diverse opzioni, una fra tutte la modifica delle impostazioni concessa direttamente dal seguente percorso:
*Pannello di Controllo -> Sistema -> Hardware -> Gestione Periferiche -> Schede di rete -> Nome della scheda -> *click destro* -> Proprietà -> Avanzate -> Indirizzo di Rete -> Valore:*

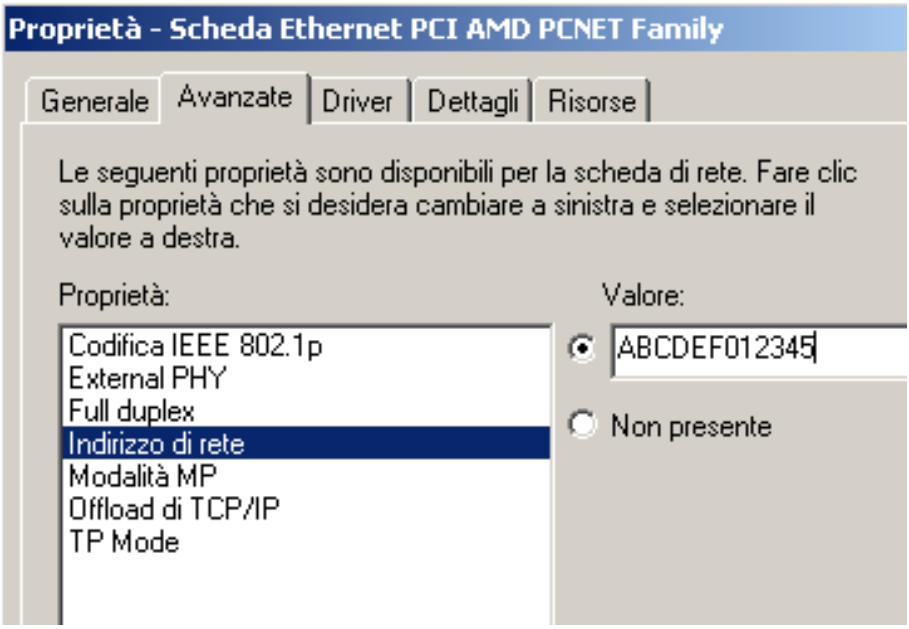


Figura 2: in questa sezione ci sarà possibile modificarne il valore su Windows OS.

> NB: è possibile che questa funzione non sia presente in tutte le schede di rete, in quanto la funzione viene resa disponibile a discrezione del produttore e dei driver presenti.

In rete sono comunque disponibili diversi tool che svolgono questa funzione, nel caso tu avessi tempo da investire potresti provarne uno dei seguenti:

- MacMACs (https://www.irongeek.com/i.php?page=security/madmacs-mac-spoofer)
- 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-spoofer (https://github.com/angusshire/mac-spoofer)

Per **macOS** è relativamente semplice modificare il MAC Address di una scheda di rete Ethernet. Basta in effetti lanciare i seguenti comandi:

```
$ sudo ifconfig en0 ether aa:bb:cc:dd:ee:ff
$ sudo ifconfig en0 lladdr 00:11:22:33:44:55
```

Le cose si complicano invece quando si parla di una scheda *Wifi*. In questo caso è necessario patchare il kernel[1] e lasciamo questa pratica a chi si sente esperto nel panorama della mela.

> Se questa risulta essere una pratica complicata ma si vuole rimanere in ambiente Mac OS valuta l'acquisto di una chiavetta USB esterna, la quale permetterà di effettuare il MAC Spoofing facilmente come con la scheda di rete Ethernet.

2.2 Hostname

Normalmente la configurazione di un hostname è un'operazione che viene effettuata in fase d'installazione, tuttavia in alcune occasioni i Sistemi Operativi Windows, MacOS e anche in certe Linux questa possibilità viene nascosta.

L'**hostname** è il nome che possiamo assegnare a un dispositivo a cui vogliamo che si venga dato un identificativo per riconoscerne il ruolo all'interno di una rete: spesso questa informazione è però lasciata al caso, quindi non è raro trovarvi la username dell'utente (nei Mac addirittura sarà qualcosa come *MacBook-Pro-di-Stefano.local*), il Sistema Operativo in uso o altre informazioni che non vorremmo vengano lette da qualche curioso all'interno della rete *LAN*.

[1] http://slagheap.net/etherspoof/

2.2.1 EFFETTUARE IL CAMBIO HOSTNAME

In qualunque distribuzione **Linux** e **MacOS** il comando per conoscere il proprio hostname è... hostname!

```
$ hostname
```

L'ultima versione di Debian ha integrato *systemd*, quindi possiamo usare anche:

```
$ hostnamectl
```

che ci fornirà maggiori informazioni riguardo il nostro Sistema Operativo, compresi gli hostname statici, l'ID macchina, la versione di GNU/Linux in uso, l'architettura e così via. Sempre con hostname possiamo decidere di *modificare temporaneamente* questo valore digitando:

```
$ su
$ hostname [nuovohostname]
```

Possiamo *verificare* il cambiamento rilanciando i comandi già citati oppure chiudendo e riaprendo la sessione del terminale.

Se la necessità è quella di *modificare permanentemente* l'hostname del computer, in ambiente Linux si userà il comando:

```
$ su
$ sysctl kernel.hostname=[nuovohostname]
```

mentre in macOS sarà:

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

Come spesso vedremo durante il corso, **Windows** ha un modo tutto suo di gestire le sue informazioni. In questo caso bisognerà far *Click Destro* sull'icona del *Computer*, quindi selezionare *Proprietà*. Nella schermata posta sotto la tag "*Nome Computer, Dominio e Impostazioni Lavoro*" si troveranno tutte le informazioni relative all'Hostname.

2.3 Domain Name System

Prima dell'invenzione dei DNS era possibile collegarsi a un computer in una rete solo conoscendone il suo *indirizzo IP*, ovvero quella serie di numeri che identifica un dispositivo informatico in una rete (di cui ne parleremo tra poco). Con l'aumentare dei dispositivi era ovviamente impossibile pensare di ricordarne la serie, così si è ben pensato nel lontano 1983 di inventare un sistema in grado di facilitare la memorizzazione, utilizzando un nome univoco (ad esempio inforge.net) anziché *l'indirizzo IP* di riferimento (come 192.124.249.10).

Nacque così la logica dei DNS e assieme ad essa i **server DNS**: questi si occupano di tradurre il nome di un *dominio* al corrispettivo *indirizzo IP*. Quando parliamo di dominio non facciamo riferimento solo al sito web ma all'intera rete: con il termine dominio infatti indichiamo un'intero network fatto di computer che condividono la stessa logica e le regole applicate da chi li amministra. Al momento ci basta sapere che, se non espressamente specificato, il nostro computer o rete Internet risponderà unicamente ai DNS proposti dall'*ISP* in nostro possesso.

Come avremo modo di vedere, una delle maggiori minacce per la privacy di un utente sulla rete sono proprio i *fornitori di linea Internet (ISP)*, ragion per cui i DNS in uso andrebbero sostituiti a prescindere dal fatto che si voglia essere anonimi al 100% o meno; considerando poi l'enorme vantaggio in termini di risposta da parte di servizi alternativi molto più efficienti e affidabili. E non solo: come avrai già visto per molti siti, per velocizzare le operazioni di takedown governativi gli organi competenti si preoccupano di censurare direttamente la risoluzione di un dominio direttamente da un DNS anziché bloccarne i server: è già successo per molti siti (come *The Pirate Bay*) e continuerà a succedere in futuro. Usando DNS non filtrati non solo ti garantirai un pizzico di anonimato in più ma accederai automaticamente a una lista completa e non filtrata di tutti i siti web disponibili realmente sulla rete Internet.

2.3.1 Scelta dei DNS

È possibile far uso di due tipi di DNS: *pubblici* e *privati*.

Usando **DNS pubblici** non solo migliori il tuo anonimato e la tua privacy ma le richieste ai DNS saranno più veloci e la tua navigazione sarà più sicura (se ti preoccupano siti pieni di malware). Solitamente i DNS pubblici fanno uso di due indirizzi IP: essi si chiamano *DNS primario* e *DNS secondario*. Considera il DNS secondario come un "paracadute" nel caso in cui il primo sia temporaneamente non disponibile oppure pieno.

In rete al momento ce ne sono diversi offerti da altrettante società: non pubblicherò gli indirizzi IP in quanto potrebbero modificare continuamente, quindi ti consiglio di seguirne i link ufficiali e scegliere quello che ritieni il più idoneo:

Nome DNS	Sito Ufficiale
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
OpenDNS	https://www.opendns.com
OpenNIC	https://www.opennicproject.org
PowerDNS	https://www.powerdns.com
Validom	http://validom.net/

Considera quelli sottolineati come i più consigliati per le scorribande digitali; evita se possibile DNS di grandi multinazionali come Google (conoscendone il modus-operandi è meglio evitarle).

In alternativa è possibile crearsi dei propri **DNS privati** in un proprio Server Dedicato o VPS; essendo un tipo di lavoro estremamente complesso e mirato al settore sistemistico mi sento di consigliarlo solo ai veterani del networking facendo uso di una delle tante guide presenti sulla rete[1].

2.3.2 Modifica dei DNS

Per utilizzare i DNS alternativi, nella maggior parte dei casi, si può ricorrere a due strade:

1. Modifica dei DNS nel router/modem (consigliato)
2. Modifica dei DNS nel Sistema Operativo

Il primo caso è possibile applicarlo direttamente nel Router o Modem in uso, facendo uso della web interface del proprio dispositivo di rete. Basterà accedervi da web browser all'indirizzo del gateway (ottenibile utilizzando i comandi visti per il Mac Spoofing tramite iproute2, ifconfig o ipconfig), digitare la password di amministrazione e inserire gli IP sotto le voci che consentono la modifica dei DNS.

Sul forum di OpenDNS[2] si trova una bella lista completa di quasi tutti i prodotti in commercio e di come modificarne i valori. Nel caso in cui stessimo lavorando su un Sistema Operativo anche qui le cose sono davvero molto semplici.

[1] https://duckduckgo.com/?q=how+to+create+private+dns+server&t=h_&ia=answer

[2] https://support.opendns.com/forums/21618374

Ad esempio, in un ambiente **Windows** (Figura 3) ci basta seguire il percorso *Start -> Pannello di Controllo -> Connessioni di Rete -> *click destro sulla rete che stai usando -> Proprietà -> Protocollo Internet (TCP/IP) -> Proprietà -> Abilita la spunta "Utilizza i seguenti indirizzi server DNS.*

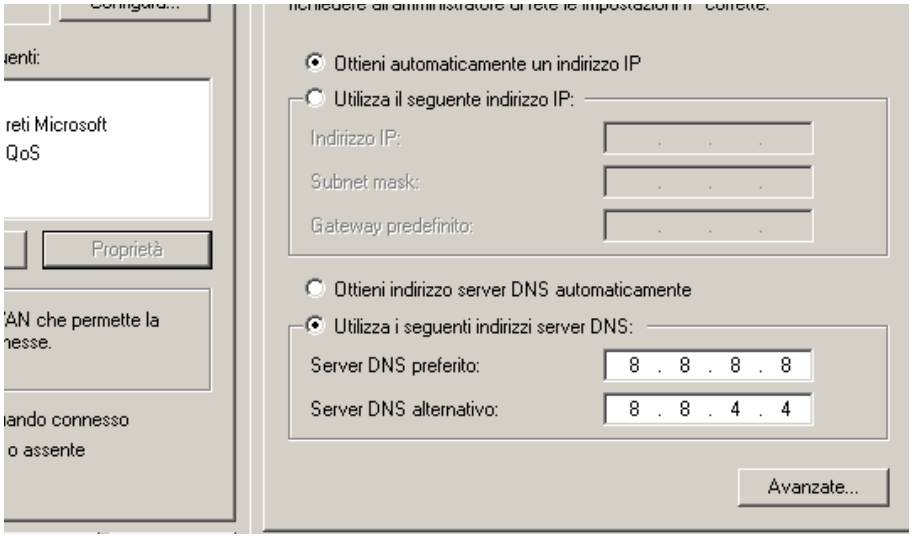


Figura 3: modifica dei DNS su Windows

Nel nostro esempio abbiamo modificato i DNS del nostro OS facendoli puntare a quelli di Google (Windows chiama i DNS primari con "preferito" e secondari con "alternativo").

Nei sistemi operativi **macOS** (Figura 4) basta seguire il percorso *Mela -> Preferenze di Sistema -> Network -> Avanzate -> Tab "DNS" -> Completa i campi come da screen cliccando sul tasto +.*

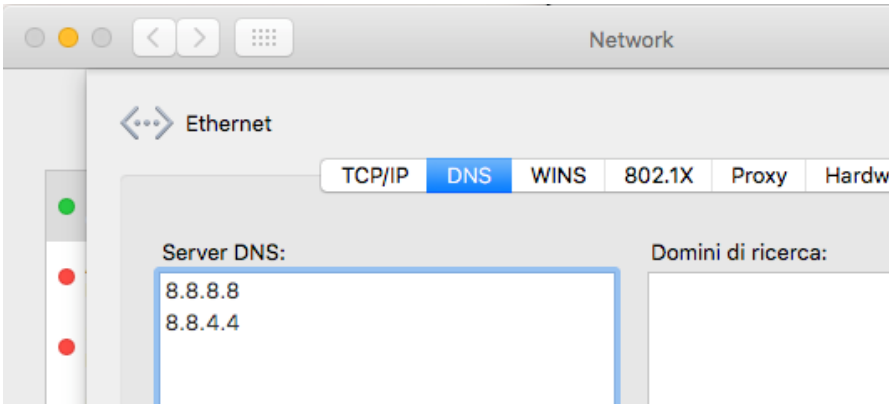


Figura 4: Modifica dei DNS nei sistemi operativi OSX/macOS

Nel mondo di **GNU/Linux** ovviamente dipende dal tipo di distribuzione e dal Desktop Manager in uso. Nel nostro caso utilizzando Debian con GNOME 3 (Figura 5), la modifica dei DNS si trova sotto il *Network Manager (tasto in alto a destra) -> Scegli la rete (eth0) -> *clicca sulla rotellina* -> IPv4 -> DNS -> aggiungi i DNS con il tasto +.*

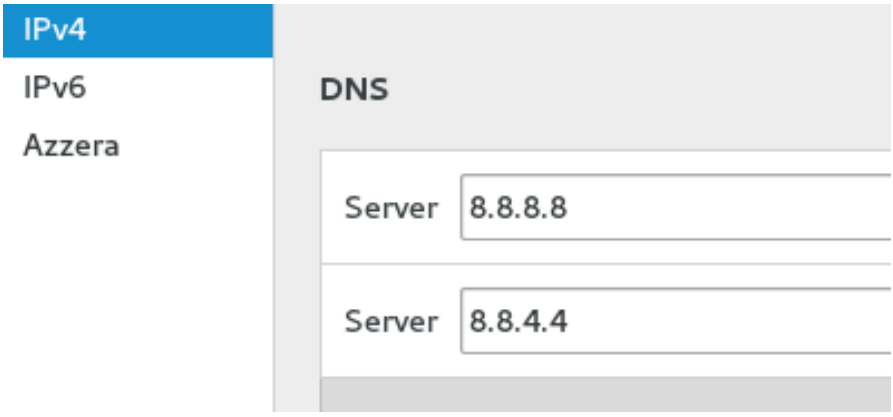


Figura 5: Modifica dei DNS nei sistemi operativi Debian e GNOME 3

La fortuna dei linux-users è la possibilità di fare quasi tutto da terminale (compreso il poter modificare i DNS). È possibile mettere mano al file "resolv.conf" ed editarlo con l'editor nano.

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

All'interno del file bisognerà scrivere (se esistono già dei valori sostituire o commentare con il #):

```
nameserver {DNS}
nameserver {DNS}
```

Ti ricordo che per salvare i file su nano usa la combinazione di tasti *CTRL+X*, quindi "*S*" o "*Y*" per confermare le modifiche e *INVIO* per effettuare la modifica definitiva. Ora riavvia il *network-manager*:

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

Puoi *verificare* i DNS in uso digitando:

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

2.3.3 Cache DNS

I sistemi operativi negli anni hanno introdotto diverse funzioni volte a migliorare le prestazioni generali. Tra queste emerge il **caching DNS**, un processo che si occupa di *memorizzare* la risoluzione di un dominio all'interno di una lista presente nel computer: il motivo di questa scelta è dovuta alla rarità con cui i domini tendono a cambiare gli indirizzi IP di destinazione, quindi risulta essere un compito inutile risolvere ogni volta l'indirizzo IP di un dominio. Questo però crea anche un problema di *privacy*: il caching dei DNS espone un'intera lista di domini che l'utente finale ha visitato, anche se questi ha preso tutte le precauzioni base per essere anonimo (compresa la navigazione in incognito).

Fortunatamente ripulire la cache dei DNS risulta essere un compito abbastanza semplice, anche perché non è raro che amministratori di sistema

debbano effettuare operazioni di manutenzione alla loro infrastruttura di rete. Una volta raggiunta questa fase è necessario dare una bella ripulita alla cache di tutti i nostri vecchi DNS in locale.

Nel mondo **Windows** è possibile lanciare il comando:

```
$ ipconfig /flushdns
```

Inoltre vorresti fare i tuoi esperimenti senza ogni volta ripulire questa dannata cache. Su Windows puoi abilitare e disabilitare temporaneamente questa funzione sempre da linea di comando:

```
$ net stop dnscache
```

In quel di **macOS** invece possiamo trovare diverse varianti in quanto alcuni tool di certe versioni non sono più presenti sulle nuove (e viceversa). Quella più funzionante sembra essere la seguente:

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

Nel mondo **GNU/Linux** possiamo prima installare nscd:

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

quindi flushare la cache:

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

Troverai maggiori approfondimenti nella rete[1].

[1] https://dnsleaktest.com/how-to-fix-a-dns-leak.html

2.4 Indirizzo IP

L'indirizzo IP è una *serie univoca di numeri* che identifica un dispositivo informatico collegato a una rete. L'indirizzo IP, così come lo conosciamo oggi, è in formato IPv4 ed è costituito da quattro serie di numeri che valgono da 0 a 255: un esempio di un indirizzo IP è 192.168.1.1 .

Nel corso dei prossimi anni la rete Internet effettuerà un passaggio graduale a un nuovo formato, *IPv6*, che consentirà a molti più dispositivi di avere un proprio codice identificativo. Fino ad allora comunque questo intero corso farà uso di esempi con *IPv4*. In molti inoltre confondono quello che è l'indirizzo IP pubblico da quello locale: un indirizzo IP viene assegnato da una rete e questa, proprio come gli IP, può essere sia locale che internet.

L'indirizzo **IP locale** viene quindi assegnato da un dispositivo di una rete interna, come ad esempio un Modem o uno Switch, e serve per identificare un dispositivo all'interno di una rete come può essere un computer in una rete locale. Nei casi più comuni, gli indirizzi IP vengono assegnati con i valori 192.168.0.x o 192.168.1.x

L'indirizzo **IP pubblico** viene invece assegnato dall'ISP, ovvero dal provider che offre il servizio a Internet: tale indirizzo serve a identificare una rete o un dispositivo informatico. Gli indirizzi pubblici, essendo assegnati dagli ISP, non possono essere modificati dall'utente finale ma possono essere solo coperti. Infine, gli indirizzi IP pubblici possono essere statici oppure dinamici, quindi possono essere sempre gli stessi o cambiare al riavvio di un modem (ciò dipende dal tipo di contratto Internet che il cliente ha stipulato).

2.4.1 Determinare l'IP in uso

Per conoscere l'IP pubblico che stiamo utilizzando possiamo usare diversi servizi online. Il metodo più semplice è far uso di un browser e quindi visitare uno dei seguenti portali:

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

Se preferisci prendere confidenza con il terminale integrato a **Linux** potremo usare il programma wget:

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

Per conoscere il funzionamento dei parametri -qO -, lancia il comando:

```
$ wget --help oppure man wget
```

2.4.2 Proxy

L'intento di un cybercriminale è quello di nascondere il suo indirizzo IP pubblico – quello che quindi lo rende riconoscibile sulla rete Internet – mentre per quello locale non si farà troppi problemi in quanto avrà già "ripulito" il suo *MAC Address* e quindi ogni informazione presente nella rete interna non lo incastrerà; come sai, l'IP locale viene assegnato da un router e questo non basta a riconoscere chi è il proprietario di un computer: l'unico elemento in grado di garantirlo è il *MAC Address*. C'è da dire che il cyber-criminale esperto quasi sicuramente non opererà *mai* da casa sua né tantomeno da qualche rete vicina: nonostante le precauzioni sa perfettamente che bisogna nascondere ogni minima traccia che lo renda

ricollegabile a qualche ipotetico reato, compresa la connessione che userà "a scrocco" durante le sue sessioni d'attacco. Ecco allora che può far affidamento ad uno degli strumenti più antichi dell'informatica: il Proxy.

I **Proxy** (tecnicamente open proxy) sono essenzialmente dei server – chiamati appunto proxy server - che possono effettuare diverse operazioni, tra:

- Fornire navigazione anonima
- Effettuare la copia di pagine web
- Effettuare un filtering a livello software, agendo come una specie di Firewall

È doveroso considerare che i proxy ad oggi sono sempre meno utilizzati per la navigazione in incognito, sostituiti da altri metodi più efficaci; rimangono comunque utili in certe occasioni - specie nella programmazione - ed è quindi importante conoscerli. Quello che fa il proxy è in sostanza porsi tra il *client* e il *server*, facendo quindi da "tramite" tra le due risorse.

2.4.2.1 TIPI DI PROXY

Come già accennato esistono diversi tipi di proxy che variano per finalità d'uso e progettazione. Sebbene sia molto interessante capire in che modo vengono utilizzati intelligentemente nelle infrastrutture *server*, nel nostro caso ci limiteremo a spiegare le differenze che riguardano la navigazione in anonimato.

Proxy HTTP/HTTPS

Come è possibile intendere da subito, i proxy HTTP/HTTPS sono in grado di filtrare le informazioni che navigano all'interno del protocollo *HTTP* e la sua forma sicura *HTTPS*. Per farla breve (almeno per ora) diciamo che *HTTP* è un protocollo di comunicazione pensato per interpretare informazioni a livello di *World Wide Web*. È in assoluto il protocollo più famoso e lo si trova in due forme:

- HTTP (senza crittografia)
- HTTPS (con crittografia SSL oppure TLS)

Tornando ai proxy HTTP è bene considerare che sono i più popolari e facili da ritrovare in quanto i server devono gestire solo questo protocollo e dunque ottimizzare meglio le macchine per fare solo questo tipo di lavoro. Rispetto ai *SOCKS* (che vedremo tra poco) sono generalmente più reattivi ma ovviamente limitati al loro protocollo. Questi tipi di proxy sono a loro volta suddivisi in sottocategorie di "qualità". Sebbene ogni agenzia che distribuisce questi proxy usa un proprio "metro di giudizio", è convenzione definirli a loro volta in 3 livelli:

- **Proxy non anonimi**: non camuffano l'indirizzo IP originario e aggiungono solitamente una sola stringa agli header (le informazioni inviate nei pacchetti) al server che lo riceve.
- **Proxy anonimi**: camuffano l'indirizzo IP ma alternano gli header al server che lo riceve.
- **Proxy èlite**: camuffano l'indirizzo IP e non alterano gli header.

Proxy SOCKS4

Il vantaggio di utilizzare un proxy con supporto al protocollo *SOCKS4* anziché *HTTP/HTTPS* è il poter reindirizzare qualunque informazione a base *TCP*. Questo significa in buona sostanza che si possono filtrare non solo i servizi del *World Wide Web* - che di natura sono basati anch'essi sul *TCP* - ma l'intera gamma di protocolli che supporta questo tipo di servizio. È possibile trovarne una variante denominata *SOCKS4a*.

Proxy SOCKS5

Sostanzialmente identico al precedente, il *SOCKS5* riesce a reindirizzare informazioni anche sul protocollo *UDP*, rendendolo di fatti il più sicuro. Il protocollo *SOCKS5* ha permesso anche ai proprietari del proxy di abilitare un

sistema di autenticazione interno e il supporto agli *IPv6*. Questo permette di utilizzare i proxy *SOCKS5* con qualunque tipo di software utilizzi la connessione ad Internet come programmi di *posta, chat, p2p* etc… Esso è la diretta evoluzione del protocollo *SOCKS4*.

Web Proxy (o CGI Proxy)

I Web Proxy sono dei veri e propri siti web che non necessitano alcuna configurazione o tool particolare all'interno del computer ma permettono di navigare direttamente in anonimato. In rete ce ne sono moltissimi, eccone alcuni che abbiamo trovato in rete per te e testato:

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

Una lista completa è disponibile su www.proxy4free.com

2.4.2.2 DOVE REPERIRE I PROXY

Una volta che abbiamo capito a cosa servono i proxy dobbiamo sapere anche dove **trovarli**!

Tramite Liste

Il ragazzo alle prime armi probabilmente userebbe *Google* digitando come chiave di ricerca "*proxy list*": quello che però non sa è che lui è l'ultima ruota di un carro

fatto di milioni di persone che a loro volta fanno un uso sconsiderato dei *proxy*. Ciò significa che nel 99,9% dei casi otterrà *proxy non buoni*, vale a dire già riconosciuti come abusati e quindi bannati, filtrati o addirittura non più attivi poiché chiusi dall'host, mentre quelle ancora funzionanti saranno lenti e instabili.

Per dovere di cronaca, i siti più attivi e popolari da cui recuperare proxy sono:

- 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

Ecco allora che nasce l'esigenza di dover trovare costantemente nuovi proxy che siano abbastanza veloci, (quasi) non bloccati da siti e servizi e che offrano un buon compromesso generale di anonimato.

Tramite Proxy Scraper

I Proxy Scraper sono dei software pensati per effettuare lo scraping - raccolta sul web - dei proxy in modo che si ottengano più velocemente e senza sforzo le ultime proxy list. Anche in questo caso consigliamo di usare un qualunque motore di ricerca; noi ne abbiamo trovati alcuni, sperando che vi possano essere utili:

- Net Ghost
- GatherProxy Scraper
- Proxy Harvester
- Holy SEO Proxy Scraper

Ti metto in guardia circa l'uso di questi programmi. Quasi tutti sono di dubbia qualità di programmazione o peggio potrebbero contenere codice malevolo per il vostro Sistema Operativo (non sempre i Proxy Scraper sono pensati per scopi nobili e chi è causa del suo mal…).La cosa migliore da fare quindi sarebbe programmarsi uno scraper da sé, facendo uso di un linguaggio di programmazione e tanta tanta pazienza.

Tramite Liste Premium

Le liste Premium sono quei siti o quelle newsletter/mailing list che contengono al loro interno liste di proxy non ancora resi pubblici. Queste liste sono quasi sempre a pagamento o riservate solo a gruppi d'èlite. In realtà sono rimasti davvero pochi i servizi pubblici che offrono proxy list a pagamento e le ultime rimaste non sono proprio così esclusive:

- Hidemyass (circa 24€/vita)
- Premium Proxy Switcher (circa 9€/mese)
- ProxySolutions (circa 18€/mese)
- SharedProxies (circa 8€/10 proxy)
- Coolproxies (circa 10€/mese)

2.4.2.3 COME USARE I PROXY

In tutto il Sistema Operativo

Arrivati a questo punto ognuno troverà più o meno difficile il giusto modo per collegarsi a un proxy, questo dipende non solo dal sistema operativo in uso ma anche dalla versione dello stesso.

Per fare un esempio: su **Windows** (Figura 6) è possibile settare a tutto il computer un proxy seguendo il percorso *Pannello di Controllo -> Opzioni Internet -> Connessioni -> Impostazioni LAN -> Server Proxy*, mentre su Windows 8 si possono seguire più strade contemporaneamente (creando così una discreta confusione). Una volta raggiunto il percorso si potrà inserire l'indirizzo proxy e la porta nei rispettivi campi.

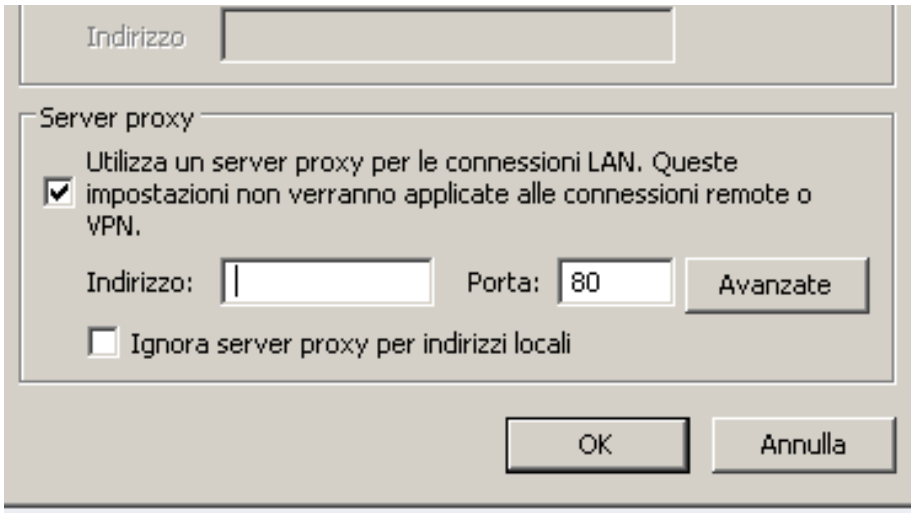


Figura 6: Utilizzo di un Proxy in ambiente Windows OS

Fortunatamente negli ambienti grafici di **GNU/Linux** (Figura 7) le cose sono anche qui semplici: in Debian con GNOME 3 è sotto la voce *Impostazioni -> Rete -> Proxy di Rete.*

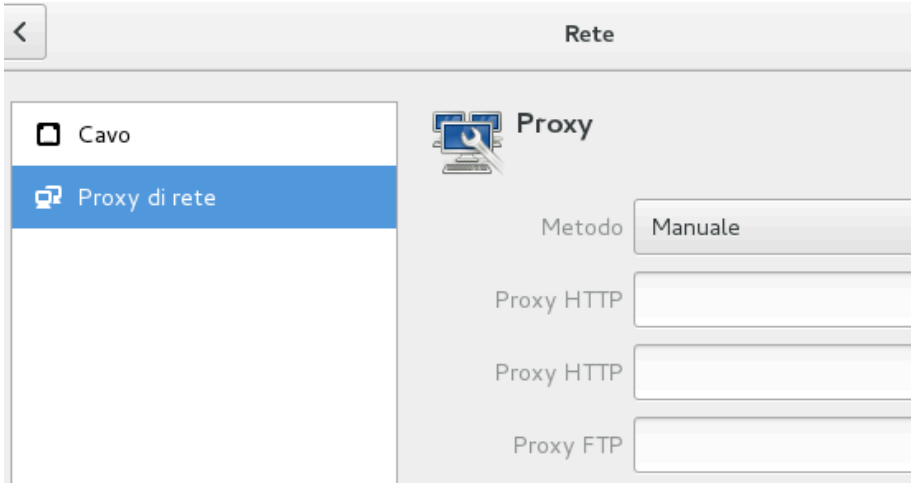


Figura 7: Utilizzo di un Proxy in ambiente Debian con GNOME 3

Chiudiamo questo piccolo capitolo parlando del sistema operativo Apple: in **macOS** (Figura 8) è possibile raggiungere la voce seguendo il percorso *Preferenze di Sistema -> Network -> Avanzate -> Proxy*. Qui è possibile specificare anche i diversi tipi di servizi che si vogliono filtrare e di assegnare a ognuno i relativi dati d'autenticazione (in caso di SOCKS5).

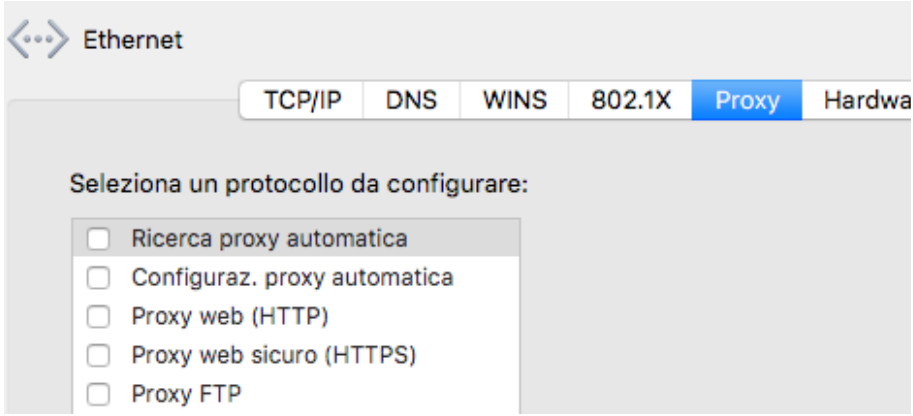


Figura 8: Utilizzo di un Proxy in ambiente OSX/macOS

Tornando alla linea di comando in Linux si può ovviamente eseguire la stessa operazione, utilizzando un *editor di testo* (*nano*) e modificando il file presente nel percorso */etc/environment*

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

Considerando ciò che abbiamo detto per la configurazione grafica, *compileremo* il file incollando le seguenti righe:

```
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"
```

Tieni presente però che alcuni programmi interni (come *APT* per le distribuzioni basate su Debian*Ubuntu) effettueranno il *bypass* di questa lettura[1].

Durante l'uso di programmi

Alcuni software - come ad esempio programmi di sharing, chat e quant'altro - danno la possibilità all'utente finale di far uso di *configurazioni proxy interne*. I motivi possono essere vari (proxy aziendale, universitario e via dicendo) e ciò permette di poter utilizzare un proxy anche per anonimizzare le connessioni in entrata ed in uscita. Per sapere se un programma fornisce la funzionalità proxy fai riferimento alla documentazione ufficiale.

[1] https://help.ubuntu.com/community/AptGet/Howto

Proxychains

Uno dei migliori software in grado di permettere l'uso mirato dei proxy è sicuramente **proxychains**[1]**,** forse in assoluto il miglior *proxifier* attualmente in circolazione. Il suo sviluppo purtroppo è stato interrotto nel 2013: fortunatamente è stato introdotto un suo fork chiamato *proxychains-NG*[2]. Il bello di proxychains è il garantire che qualunque programma - e tutte le sue dipendenze - comunichino in esterno solo ed esclusivamente tramite protocolli SOCKS4, SOCKS5 o HTTP/S.

Attenzione però: Proxychains è disponibile ufficialmente solo per i sistemi *UNIX*, quindi *GNU/Linux*, *macOS* (tramite Brew) e *BSD*. Per limitare i problemi faremo uso della versione storica (quella abbandonata) ma ancora perfettamente funzionante e presente nei repository di Debian 8.

Procediamo quindi alla sua installazione:

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

Il suo funzionamento è previsto anche da utenti normali, quindi con il comando exit ritorniamo al nostro user predefinito. Vediamo come utilizzare ora il programma:

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

Come puoi vedere il suo funzionamento è semplicissimo, basta infatti anteporre la stringa "proxychains" di fronte al comando che si vuole usare. Tuttavia lanciando il comando riceveremo un errore come il seguente:

[1] http://proxychains.sourceforge.net

[2] https://github.com/rofl0r/proxychains-ng

```
$ 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
```

Questo succede perché proxychains di default va a leggere un proxy configurato sulla porta 9050 del nostro computer in *locale*, quindi con ip 127.0.0.1. Per far in modo che legga una lista dei nostri proxy dobbiamo creare un file di configurazione. Da terminale quindi lanciamo:

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

In questo modo abbiamo prima creato la cartella che contiene il file di configurazione, quindi abbiamo avviato l'editor di testo nano sul *path* "segreto" nella cartella del nostro utente. È interessante vedere per la prima volta il richiamo di *$HOME*, una variabile in grado di farci accedere istantaneamente alla cartella assoluta del nostro utente. Nel mio caso l'utente si chiama *stefano9lli*, quindi il percorso della cartella sarà */home/stefano9lli*, che quindi si andrà a completare al resto.

Osserviamo anche che, dopo il richiamo della variabile *$HOME*, viene richiamata la cartella *.proxychains* . Nel mondo UNIX, un punto di fronte a una cartella indica che quest'ultima deve risultare nascosta durante l'uso di un file manager. Andremo quindi a creare il file *proxychains.conf* all'interno di questa cartella. Ora siamo pronti ad aggiungere alcuni valori:

```
strict_chain
proxy_dns
[ProxyList]
http proxy porta
```

Salviamo con la combinazione *CTRL+X*, il *tasto S* e *INVIO*. Ti ricordo che puoi usare una delle seguenti configurazioni per utilizzare diversi protocolli:

```
strict_chain
proxy_dns
[ProxyList]
http proxy porta
socks4 proxy porta
socks5 proxy porta
```

Rilanciamo il nostro wget con la nuova configurazione (nel nostro esempio il proxy è 177.73.177.25 e la porta è 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
```

Come vediamo stavolta il sito di *ipinfo.io* ci restituisce l'ip del proxy anziché quello in nostro possesso, a dimostrazione di come proxychains abbia funzionato. Considera che molti proxy in realtà potrebbero essere configurati per funzionare solo in determinate situazioni, quindi potrebbero rifiutarsi di rispondere a certe tipi di richieste a dominio o da programmi senza user-agent. Il modo migliore ovviamente è provare. Tutti gli usi di proxychains sono spiegati nel manuale:

```
$ man proxychains
```

Proxycap

Forse la più famosa controparte di proxychains per il mondo **Windows** è rappresentata da Proxycap[1] (Figura 9), un programma che da oltre 10 anni viene sviluppato dal team Initex. Proxycap è in grado come proxychains di dirottare tutte le comunicazioni Internet ma viene fornito anche di una GUI grafica; purtroppo è anche a pagamento. Considera inoltre molte altre *alternative*. È disponibile una pagina su Wikipedia[2] che permette di vedere una comparazione tra i vari proxifier in rete.

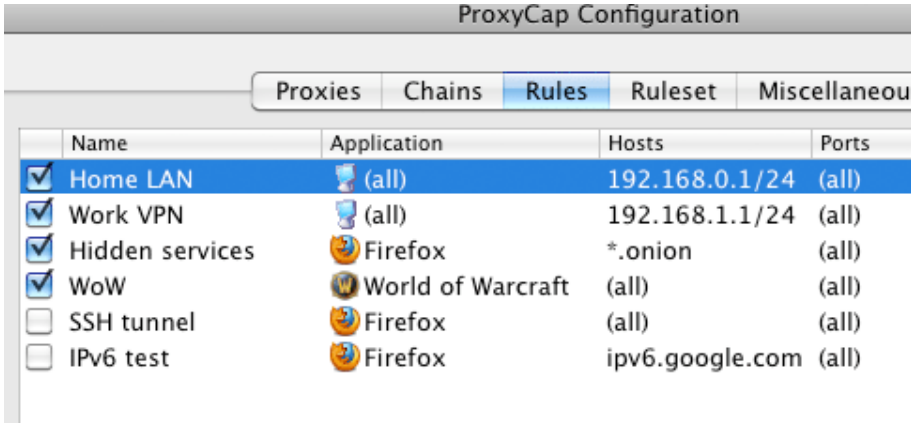


Figura 9: Schermata del funzionamento di ProxyCap

[1] www.proxycap.com

[2] https://en.wikipedia.org/wiki/Comparison_of_proxifiers

Durante la navigazione web

In questa parte tratteremo della configurazione dei proxy tramite **browser**, tuttavia vi avvisiamo che screenshots e menù potrebbero cambiare leggermente a causa delle versioni dei *Sistemi Operativi* e dei *Browser*. Nel nostro caso tratteremo solo dei principali Browser di navigazione. I browser di sistema (*Safari*, *Internet Explorer*, *Edge* etc...) fanno sempre uso della configurazione di sistema.

> Considera che ogni Browser ha anche un supporto alle estensioni e tra queste troverete sempre una GUI per velocizzare l'uso di proxy. Naviga negli store di ogni Browser e sicuramente troverai l'estensione più adatta alle tue esigenze.

Google Chrome / Chromium

Dal browser di Google accedi alle *Impostazioni* dal pulsante in alto a destra. Dalla scheda che si aprirà clicca su "*Mostra impostazioni avanzate…*" quindi clicca su "*Modifica impostazioni proxy*". In questo modo verrai riportato direttamente al prompt delle configurazioni dei Proxy del tuo Sistema Operativo.

Mozilla Firefox

Dal browser del panda rosso indirizzati verso il tasto delle *Impostazioni* in alto a destra, quindi apri la tab *Avanzate* e attiva in alto la tab *Rete*. Da qui troverai la voce "*Connessione*" e affianco il pulsante "*Impostazioni*". Una volta aperto avrai la possibilità di configurare il browser con le proprie impostazioni proxy utilizzando la voce "*Configurazione manuale dei proxy*" oppure basandosi sulle impostazioni di sistema.

Opera Browser

In Opera le cose sono davvero semplici. Aprendo il menù accediamo alla voce *Impostazioni*, quindi entriamo in *Preferenze*. Dalla tab *Avanzate* entriamo in *Rete*, quindi clicchiamo su *Server proxy*. Modifichiamo ora le impostazioni del client.

Attenzione alle Blacklist

Può capitare in molte occasioni che i proxy finiscano all'interno di **Blacklist**, database online in cui vengono memorizzati quegli indirizzi IP che sono stati usati per abusi nel web, truffe, spam e così via. Le liste vengono archiviate da servizi (i cosiddetti *Honeypot*) sia gratuiti che a pagamento per dare la possibilità a portali web, *Firewall*, *CDN* e così via di fare un rapido confronto tra l'IP visitatore e il database degli IP "maligni". I più popolari sono Spamhaus[1] e Barracuda[2] ma ce ne sono molti altri. Per verificare se il proprio IP è stato blacklistato puoi utilizzare tutti i servizi di verifica IP che lo supportano oppure utilizzando il servizio specifico di WhatIsMyIPAddress.com

2.4.2.4 QUANTO SONO SICURI I PROXY?

La domanda a questo punto che dovremmo farci è: ma i proxy garantiscono davvero l'anonimato al **100%**? La risposta nella maggior parte dei casi è *assolutamente no*. Per quanto sicuri possano sembrare, i proxy servers sono gestiti da servizi esterni che pagano il mantenimento di un server in grado di ospitare le nostre richieste al mondo di Internet.

Ecco, sono servizi esterni, ciò significa che sono gestiti da terzi che solitamente non sappiamo chi sono, cosa fanno e perché sono dei benefattori in

[1] https://www.spamhaus.org

[2] https://www.barracuda.com/homepage

questo senso. Delle volte possiamo trovarci di fronte ad associazioni che combattono la censura oppure proxy universitari per la ricerca ma possiamo trovarci anche di fronte ad aziende che lucrano sulla nostra navigazione (ad esempio per effettuare indagini di mercato) o, nel peggiore dei casi, ad *honeypot* gestiti da organizzazioni governative come *NSA* o *FBI* che monitorano il traffico.

Senza contare poi che il proxy server può memorizzare tutto - o quasi - ciò che fai (siti navigati, login, operazioni effettuate etc…) e tutte le informazioni che rilasci nella rete (indirizzo IP, browser, sistema operativo etc…) rendendolo a tutti gli effetti un'arma a doppio taglio. Ciò non significa che l'IP spoofing tramite proxy è inutile, anzi: la sua popolarità e semplicità d'uso ha permesso la creazione di moltissime librerie per ogni linguaggio di programmazione e così ha permesso di definire nuovi modi d'uso, ad esempio molti *bruteforcer* / *bot* / *stresser* - e chi più ne ha più ne metta - fanno ancora uso delle proxy list.

3. COMUNICAZIONI SICURE

Fino ad ora abbiamo visto come un indirizzo IP può essere una traccia assai pericolosa da lasciare durante la navigazione del web; qualunque server in questo mondo è in grado di loggare e memorizzare un *indirizzo IP* visitatore e di associarlo a qualunque azione egli compia. Nascondere un IP Address (nel gergo informatico, spoofare l'IP) non è sufficiente a insabbiare le attività di un internauta nella rete: basti pensare che qualunque richiesta non cifrata può essere monitorata non solo dai governi ma anche dagli *ISP* (*Internet Service Provider*, i fornitori della rete), da altri servizi e malintenzionati di qualunque tipo.

Poco fa abbiamo introdotto i protocolli *HTTPS*, il nuovo modo in cui i computer stanno iniziando a comunicare all'interno del web. La crittografia dei dati sta avendo un ruolo sempre più importante nell'ecosistema informatico e i protocolli sicuri stanno sostituendo quelli più deboli (*SSH -> TELNET, SFTP -> FTP, HTTPS -> HTTP* e via dicendo). Quindi, a meno che il programma che usiamo o il proxy a cui stiamo facendo affidamento non sono espressamente criptati, tutte le nostre operazioni all'interno di Internet sono facilmente monitorabili.

Una nota che riguarda la navigazione WWW: se privacy e anonimato sono le vostre priorità dimenticatevi per sempre di Google e affini e puntate su *motori di ricerca* che non vi monitorano come DuckDuckGo[1] oppure StartPage[2]. Perché? Prendiamo per esempio *Youtube*. *Youtube* è un servizio acquistato e gestito da Google e Google, lo sappiamo, traccia qualunque cosa. *Youtube* prende nota di qual è il tuo IP e quale video stai vedendo, quindi butta giù un profilo utente chiamato *fingerprint* e sa già cosa ti piacerebbe vedere dopo o magari acquistare mentre visiti siti web con *Google Adwords*. Un circolo vizioso.

[1] https://duckduckgo.com

[2] https://startpage.com

3.1 VPN (Virtual Private Network)

Abbiamo visto come i proxy sono strumenti utili ma a causa di una serie di problemi non riescono a garantirci il giusto compromesso tra sicurezza e velocità. Ti dirò di più: navigare con un open proxy oggi è praticamente impossibile, oltre che insicuro! Tempi di latenza incredibili e down improvvisi lo rendono inutilizzabile per operazioni più lunghe di 5 minuti! In molti ritengono che le VPN siano i *Proxy del futuro*. Sarà vero? Andiamo a scoprirlo.

Le **VPN** (acronimo di *Virtual Private Networks*) sono del "tunnel" criptati che, esattamente come i proxy, effettuano da tramite tra client e server; questo significa che tutto il traffico Internet passa attraverso questo *tunnel criptato*, impedendo a chiunque di monitorare la connessione.

Le VPN sono originariamente pensate per creare una *rete LAN* di computer collegati tramite Internet, esattamente come una rete fisica, senza però affrontare tutti i costi che comporta (localizzazione dei dispositivi, collegamenti fisici etc…) e con tutti gli accorgimenti di sicurezza del caso come Firewall, Proxy e via dicendo. Utilizzando una VPN non dovremo preoccuparci né di trovare liste funzionanti né di trovare tipi di protocolli particolari: tutto il traffico che viaggia in una VPN viene veicolata e cifrata con standard di qualità solitamente garantiti ad almeno 128 bit.

Il maggior vantaggio rispetto a un *proxy* è l'elevata reattività che spesso la VPN garantisce; l'architettura di tale infrastruttura e la *geolocalizzazione dei server* permette di ottimizzare le richieste alla rete Internet; inoltre non è necessario riconfigurare browser e tools per essere anonimi in quanto il *tunneling* - generalmente - viene effettuato su tutto il sistema.

3.1.1 Tipi di VPN

Nel mercato delle VPN possiamo definire almeno tre tipi di VPN: *Trusted VPN*, *Secure VPN* e *Hybrid VPN*.

Nel corso di questo capitolo parleremo delle **Secure VPN** in quanto le Trusted richiedono speciali contratti con gli *ISP* e non sono facilmente applicabili nelle realtà comuni: quest'ultime sono infatti pensate quasi esclusivamente per le reti aziendali dove si deve garantire che le informazioni arrivino sempre al destinatario.

Le Hybrid invece sono l'unione delle Trusted e delle Secure e poiché non parleremo delle prime, escluderemo anche le seconde. Ciò che realmente determina la qualità della sicurezza di una VPN - oltre ovviamente alle policy e alla stabilità dei servizi che vedremo a fine capitolo - sono sia i tipi di protocolli forniti, sia la sicurezza delle chiavi garantite.

Spesso non basta parlare di VPN per essere *sicuri*: ad esempio fino a pochi anni fa il noto provider di VPN iPredator[1] offriva solo connettività tramite protocollo PPTP: questo tipo di protocollo era già sotto accusa di non essere sicuro al 100% in quanto dismesso dalla Microsoft (che lo ha inventato e brevettato) e siamo ormai quasi sicuri che lo spionaggio governativo sia già in grado di *crackarlo* in breve tempo. Questo è uno dei tanti esempi che abbiamo messo in luce ma vediamo i protocolli uno ad uno e tiriamo le somme sulle loro caratteristiche e qualità.

3.1.1.1 PPTP, PER CHI CERCA LA VELOCITÀ

Il protocollo PPTP (acronimo di Point-to-Point Tunneling Protocol) è stato sviluppato da *Microsoft* per la creazione di reti VPN aziendali tramite il collegamento dial-up telefonico.

[1] https://www.ipredator.se

È un protocollo pensato esclusivamente per le VPN e generalmente fa affidamento a *MS-CHAP* per la gestione dell'autenticazione. Essendo stato per anni uno strumento molto popolare oggi è facilmente installabile (o addirittura preinstallato) in qualunque dispositivo sul mercato ed è anche molto rapido in quanto richiede poche risorse per il suo funzionamento.

Il protocollo PPTP, che può supportare solo chiavi a base da *128 bit*, ha iniziato a cedere ai colpi delle vulnerabilità tanto da costringere nel 2012 la Microsoft a dichiararlo insicuro, nonostante quest'ultima avesse rilasciato decine di patch per assicurare la situazione. Questo protocollo è oggi considerato insicuro e sicuramente già violato dall'*NSA* ma comunque utile per attività a bassa latenza come *gaming online*, *torrent*, *streaming* etc…

3.1.1.2 L2TP/IPSEC, PER CHI VUOLE SICUREZZA E REATTIVITÀ

L2TP (acronimo di *Layer 2 Tunnel Protocol*) è un protocollo di tipo VPN che di base non offre alcuna sicurezza dei dati; questo è il motivo per cui viene spesso affiancato da una suite denominata *IPsec*. L2TP/IPsec è quindi un mix di *protocollo di tunneling* e di crittografia già implementato nei Sistemi Operativi di ultima generazione, permettendo quindi una facile configurazione via client e una buona velocità generale.

Al momento non esistono vulnerabilità conosciute relativamente gravi per questo protocollo quindi posso consigliartelo se vuoi mantenere un buon livello di privacy e sicurezza, tuttavia una ricerca condotta da due esperti[1] fa intendere che l'*NSA* sta lavorando assiduamente per violarlo. Sebbene questo non sia ancora stato provato, alcune fonti[2] confermano che IPsec sia uno dei target principali dell'*NSA* e che teoricamente un attacco sarebbe possibile.

[1] www.mail-archive.com/cryptography@metzdowd.com/msg12325.html

[2] https://nohats.ca/wordpress/blog/2014/12/29/dont-stop-using-ipsec-just-yet/

Ad ogni modo, L2TP/IPsec esegue l'incapsulamento dei dati in due passaggi con chiavi di cifratura a *256 bit*, rendendolo di fatti tecnicamente più lento rispetto al PPTP ma grazie al supporto *multi-threading* implementato nei *kernel* di ultima generazione permette di cifrare e decifrare sfruttando l'architettura di calcolo dei processori *multi-core*.

L'unico piccolo difetto di questo protocollo risiede nel fatto che l'L2TP di default viaggia sulla porta *UDP 500*: quest'ultima viene spesso bloccata dai firewall business e costringe ad effettuare *port-forwarding* su *router* e *access point* più sofisticati (rendendo problematica la navigazione specie nelle reti aperte).

3.1.1.3 OPENVPN, PER CHI VUOLE IL TOP DELLA SICUREZZA

Con OpenVPN si intende un software opensource creato appositamente per creare tunnel cifrati tra due sistemi informatici e che sfrutta protocolli di crittografia a base *SSLv3/TLSv1* e la libreria *OpenSSL*. Il fatto di essere totalmente open garantisce a questo sistema la giusta trasparenza per considerarla come la soluzione più affidabile e sicura; attualmente, sono pochissimi i rischi che un ente di spionaggio governativo riesca a violarlo.

La sua natura open lo rende anche un prodotto estremamente configurabile che ci permette di utilizzarlo su qualunque porta senza effettuare il *port-forwarding* (sfruttando ad esempio anche la porta *TCP 443* per soddisfare richieste di tipo *HTTP* attraverso *SSL*) sul dispositivo di rete in uso. La libreria che utilizza (*OpenSSL*) può far uso di diversi cifrari (come *Blowfish*, *AES*, *DES* etc…) tuttavia la maggior parte dei provider di VPN fanno uso quasi esclusivo di cifrari AES o Blowfish. Quest'ultimo, a base 128 bit, è il cifrario di default presente in OpenVPN.

AES è invece un cifrario relativamente nuovo ed è attualmente utilizzato da diversi governi mondiali per proteggere i loro dati: essendo in grado di gestire

blocchi a *128-bit* può manipolare informazioni grandi fino a 1GB, a differenza di *Blowfish* che essendo a base 64-bit ne può gestire solo la metà.

Rispetto al protocollo IPsec risulta essere meno veloce e questo può risultare deleterio specie in quei dispositivi che non hanno molta potenza di calcolo: la causa di questa lentezza nasce dall'assenza di un supporto nativo al *multi-threading* che quindi non permette di sfruttare le CPU di nuova generazione in commercio.

Sebbene non sia uno standard de-facto come i precedenti PPTP e L2TP/IPsec, il mercato dei provider VPN ha accolto con gioia OpenVPN e la *community* degli sviluppatori ha rilasciato il client per tutti i maggiori Sistemi Operativi, inclusi i dispositivi mobile.

3.1.1.4 SSTP, PER GLI UTENTI WINDOWS

SSTP (acronimo di Secure Socket Tunneling Protocol) è un protocollo di *tunneling* introdotto da Microsoft e nativo per tutte le versioni di *Windows* da *Vista* in poi, mentre è disponibile ma non pre-installato nei sistemi a base *Linux* e *BSD*. Al momento non risultano esserci piani affidabili per il mondo mobile così come per i *router firmware* più blasonati (ad eccezione di Router-OS[1], attualmente l'unico Sistema Operativo per Router che lo supporta).

Come per OpenVPN fa uso della crittografia a base *SSLv3* permettendo quindi l'uso del tunnel cifrato anche dietro reti protette da firewall; il protocollo SSTP può essere utilizzato in concomitanza con l'autenticazione di *Winlogon* o *smartcard*. Attualmente è il protocollo di sicurezza utilizzato nella cloud di Microsoft denominata *Windows Azure*. A differenza di OpenVPN è tuttavia un protocollo chiuso e lo scandalo *PRISM*[2] che ha visto in collaborazione *Microsoft* e *NSA* non fa certo dormire sonni tranquilli.

[1] http://www.mikrotik.com/software

[2] https://en.wikipedia.org/wiki/PRISM_(surveillance_program)

3.1.2 Quale VPN scegliere?

Bene a questo punto tiriamo le somme: *quale tipo di VPN fa per te?* Personalmente mi sentirei di consigliarti una **OpenVPN** in quanto riesce a raccogliere tutte le caratteristiche che ricerchiamo in una VPN, vale a dire il miglior compromesso tra velocità, sicurezza e trasparenza di sviluppo. L'unico piccolo problema potrà risultare nella maggiore difficoltà di installazione e di utilizzo rispetto agli altri (in quanto non è presente una funzione built-in in quasi nessun OS) tuttavia ogni società nella maggior parte dei casi offre una documentazione sufficiente per risolvere tutti i problemi nei setup e nei giorni a venire. **L2TP/IPsec** è anch'esso molto popolare e, a meno che non tu non viva nella paranoia più completa, garantisce un'ottima velocità e una buona sicurezza generale. Sinceramente mi sento di sconsigliare **PPTP** e **SSTP**: il primo è sicuramente obsoleto e rischia di fare più danni che altro, il secondo è più indicato al mondo aziendale che non all'anonimato.

3.1.3 Come scegliere una VPN

Fare una lista delle migliori VPN online, decretando quale sia la migliore, non sarebbe una scelta saggia a causa del continuo mutarsi del mercato attuale; come però abbiamo già fatto per i proxy ci limiteremo a dare le indicazioni su quale VPN è quella giusta per te in base alle caratteristiche che cerchi, quindi butteremo giù un reseconto delle più popolari VPN in circolazione.

3.1.3.1 NON USARE VPN FREE

Magari ti sarai chiesto: *le VPN sono gratuite o a pagamento?*

La risposta è entrambe, tuttavia ci tengo a precisare che da qui in avanti parlerò solo di VPN a pagamento. Perché?

Motivo n°1: mantenere un servizio VPN ha dei costi

Alcuni dei migliori servizi VPN come HideMyAss, NordVPN o ExpressVPN offrono qualcosa come più di 1000 servers dislocati in tutto il mondo. E pensa, questi server costano! Costa mantenerli, costa sostituirli quando si rompono, costa gestirli. E a meno che tu non creda che in questo mondo sia pieno di benefattori che spendono centinaia di migliaia di dollari al mese per mantenerli non fidarti delle VPN gratuite!

Motivo n°2: i provider potrebbero vendere i tuoi dati

Ma come fa a guadagnare una VPN? Semplicemente potrebbero vendere i tuoi dati. Non sto parlando di username e passwords (anche se non è detto!) ma di veri e propri honeypot utilizzati per fare statistica e venderli ai migliori offerenti.

Motivo n°3: i provider potrebbero riutilizzare la tua banda

Una volta entrato nel circuito fai parte della rete virtuale, quindi diventi automaticamente "complice" della rete; questo significa non solo che il tuo Internet va più lento (questo era scontato) ma che tu possa finire alla "fine della coda" e risultare il responsabile di una pratica non corretta da parte di altri utenti.

Motivo n°4: i provider potrebbero riempirti di pubblicità

Questa è una pratica in voga tanto nel mondo dei free proxy quanto in quello dei free vpn. Gli adware presenti nelle Free VPN possono essere sia installati assieme al client che essere mostrati durante la navigazione manipolando il sorgente delle pagine Web che andrai a visualizzare.

Motivo n°5: Non sei tutelato

Quando acquisti un servizio sei tutelato da un documento che accetti automaticamente sia te che l'azienda venditrice: questo documento si chiama Termini e Condizioni d'Uso che, assieme alla Privacy Policy, formano il documento legale che stabilisce il rapporto tra i due. Nel caso delle VPN Free questi documenti sono spesso confusi ed essendo gratuiti l'utente pensa: vabbè, finché è gratuito chissene! In realtà come vedremo a breve i ToS e le Privacy Policy sono di fondamentale importanza per avere una VPN di qualità che ti garantisce efficienza e sicurezza nella navigazione.

3.1.3.2 POLICY DEI NO LOGS

I logs sono quei file che vengono generati per ogni attività effettuata all'interno di un sistema informatico: nel caso delle VPN, i *log* possono memorizzare informazioni come IP, dati d'accesso e altre informazioni che non vengono cifrate prima *dell'handshake* (che poi porterà al *tunneling* vero e proprio e quindi alla *cifratura* totale). Ma prima una breve storia.

Conosci il gruppo lulzsec? Si, gli stessi che hanno violato la Sony e la CIA.
Sapevi che Cody Kretsinger, in arte recursion, membro dei Lulzsec, è stato arrestato dopo che i federali sono risaliti alla sua identità chiedendo i log di accesso a un provider VPN - tale HideMyAss - che l'hacker utilizzò per violare la Sony Pictures?

Se stai scegliendo una VPN logless non limitarti agli slogan pubblicitari ma controlla le *Privacy Policy* dichiarate dai provider.

3.1.3.3 SE NON HANNO I TUOI DATI NON POSSONO INCASTRARTI

Immagina di essere il titolare di un'azienda provider di VPN e nel cuore della notte ti bussa *l'FBI* (o la *CIA*, la polizia o chi ti pare) con un mandato di perquisizione dei dati dei tuoi *server*. Te la sentiresti di fare il paladino della giustizia e difendere uno sconosciuto che dall'altro capo del mondo si è messo a giocare con i computer di qualche multinazionale? La risposta, manco a dirlo, è ovviamente no! Non esiste *nessun provider VPN* che rischierebbe anni di carcere per te. Non esiste nessun benefattore di questo genere, quindi ricordati sempre che il provider fa sempre i suoi interessi e con le giuste pressioni è disposta a venderti (come nel caso di HideMyAss).

Il punto chiave è allora capire che *un provider VPN non può rilasciare informazioni di te che non ha***,** quindi non possono essere incriminati per non aver collaborato dando informazioni che - di fatti - non hanno. Normalmente un provider VPN richiede informazioni personali per creare account e processare pagamenti, quindi chiederanno: nome, email, dati di pagamento e indirizzi di fatturazione.

I migliori provider VPN ultimamente hanno capito che è possibile consentire maggiore anonimato ai propri utenti offrendo loro pagamenti con le cryptomonete (di cui parleremo più in là): questo consente, con le dovute precauzioni, di rendere anonima la compravendita del servizio sollevando i venditori dal peso di memorizzare i dati di fatturazione.

3.1.3.4 LEGISLAZIONE INTERNAZIONALE CONSERVAZIONE DEI DATI

Ogni nazione ha al suo interno delle *leggi specifiche* che riguardano qualunque argomento; tra questi troviamo anche leggi in materia di protezione dei dati e privacy. Nella mappa qui in basso (Figura 10) troverai una cartina geografica con le nazioni colorate dal *rosso* sfumando al *verde*, dove i primi hanno una legislazione molto ferrea in materia di conservazione dei dati mentre quelli in verde sono molto flessibili (gli stati in bianco non hanno alcuna legge in merito).

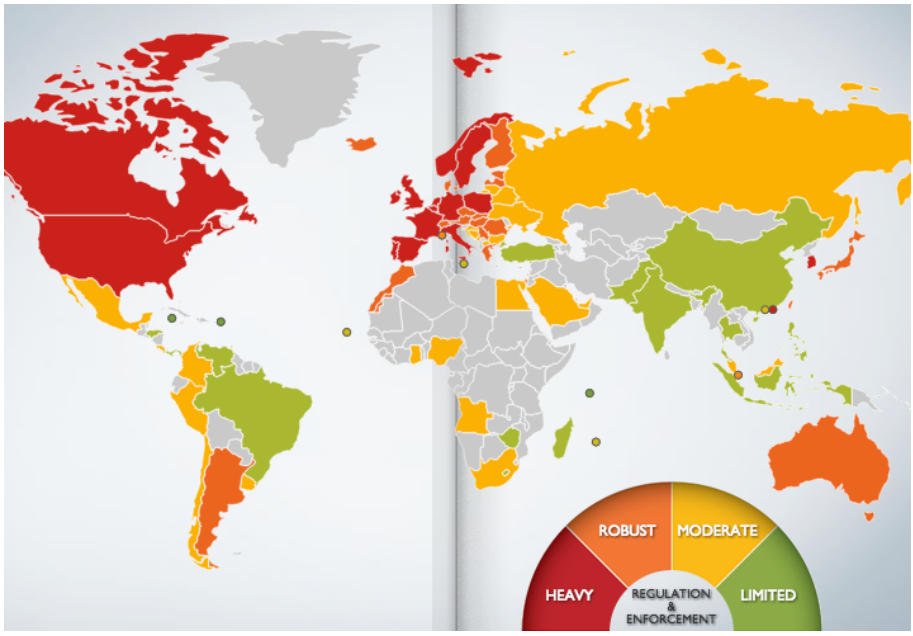


Figura 10: La seguente mappa e le relative informazioni sono disponibili online sul sito dlapiperdataprotection.com

Per fare un esempio reale, NordVPN è una società con sede a *Panama*, una nazione fortemente libertina circa le leggi sulla conservazione dati. Non a caso è anche definito un paradiso fiscale dove 120 banche segrete fanno gli interessi a ricchi impresari (tra cui molti evasori) e società offshore. In questa nazione le aziende non sono neanche tenute a presentare bilanci e i residenti a fare la

dichiarazione dei redditi, figuriamoci se un rivenditore VPN è tenuto a memorizzare i dati fiscali di un cliente!

Allo stesso modo prendiamo HideMyAss con sede nel Regno Unito: la compravendita online richiede la presentazione di documenti, pagamenti tracciabili, bilanci e soprattutto leggi in materia di abusi informatici regolamentati dal *Computer Misuse Act* che rende difatti libera la strada al Governo di ordinare perquisizioni in ogni dove.

3.1.3.5 METODI DI PAGAMENTO

Tra le caratteristiche che contraddistinguono una VPN sicura da una non sicura troviamo i metodi di pagamento supportati. Nel caso in cui ti venga in mente di affittare una VPN con servizi di pagamento come Paypal, carta di credito o bonifico bancario (intestati a tuo nome) lasci delle tracce non indifferenti. Per quanto la privacy policy di una VPN sia ferrea, le tracce dei pagamenti sono in mano alle banche (che come ben sappiamo vano a braccetto con i governi).

Una VPN che accetta solo pagamenti tracciabili - *carta di credito, bonifico bancario, vaglia* e via dicendo - non può essere definita una VPN sicura; a differenza di VPN gratuite, che l'unica cosa che possono avere sono il tuo IP e un'eventuale account registrato, le VPN a pagamento possono avere dettagli ben più pericolosi per il tuo anonimato, come appunto gli estremi di fatturazione di una carta di credito o di un conto bancario.

In questo caso si dovrebbe preferire una VPN che offre pagamenti in *cryptovalute* come *Bitcoin*, *Litecoin* etc… e prendere anche le giuste precauzioni per evitare che i *wallet* siano esposti a rischi di tracciabilità (parleremo della sicurezza sull'uso delle *cryptovalute* più avanti).

3.1.3.6 NOTIFICHE DMCA

Il *DMCA* (acronimo di Digital Millennium Copyright Act) è un insieme di leggi americane che tutelano la distribuzione illegale di materiale protetto da diritti d'autore. Pur essendo una legge d'oltremare per certi versi è simile alla legge sul diritto d'autore dell'UE[1] e potrebbe in qualche modo applicarsi anche nel nostro Stato. Non approfondiremo questo discorso in quanto è un argomento tecnicamente legale. L'unica cosa di cui possiamo essere sicuri è che l'abuso del *DMCA* potrebbe far decidere al provider della VPN di bloccare il tuo account per evitare problemi con la legge.

3.1.4 Lista delle VPN

La lista che ti presento ora raccoglie alcune delle VPN più popolari che ho ricercato nella rete: una lista più completa è disponibile sul sito vpndienste.net.
Come noterai ci sono alcune VPN sottolineate: credo che queste siano le migliori nel caso in cui si voglia evitare di essere tracciati durante la navigazione, in quanto nelle loro Privacy Policy dichiarano di non memorizzare IP durante l'uso dei loro servizi e su quello che offrono (protocolli, dati, nazione, tolleranza e tipi di pagamenti).

Nome VPN	Stato	Dati raccolti	Log IP	DMCA	Tipi	P2P	BTC
AIRVPN	Italia	Dati personali	✔	-	OpenVPN	✔	✔
BTGuard	Canada	Dati personali		-	PPTP OpenVPN	✔	✔
Boxpn	Turchia	Dati personali	✔	?	PPTP L2TP SSTP	✔	-

[1] https://it.wikipedia.org/wiki/Legge_sul_diritto_d'autore_dell'Unione_europea

Nome VPN	Stato	Dati raccolti	Log IP	DMCA	Tipi	P2P	BTC
ExpressVPN	USA	Nome Indirizzo Email Carta di Credito	✔	✔	PPTP L2TP OpenVPN	-	✔
HideMyAss	UK	Indirizzo Email Dati di Fatturazione Indirizzo IP	✔	✔	PPTP OpenVPN L2TP	✔	✔
iPredator	Svezia	Indirizzo Email	✔	?	PPTP OpenVPN	✔	✔
MULLVAD	Svezia	-	-	-	PPTP OpenVPN	✔	✔
NORDVPN	Panama	Indirizzo Email Username/ Password Dati di Fatturazione	-	-	PPTP L2TP OpenVPN	✔	✔
PRQ	Svezia	Indirizzo Email	-	-	OpenVPN	✔	✔
Private Internet Access	USA	Indirizzo Email Dati di Fatturazione Cookie temporanei	-	✔	PPTP L2TP OpenVPN	✔	✔
PureVPN					PPTP L2TP OpenVPN IKEv2 SSTP		✔
Security Kiss	UK	Indirizzo Email Nome Dati di Fatturazione	✔	?	PPTP L2TP OpenVPN	?	✔
SHADEYOU	Olanda	Username/ Password	-	✔	PPTP L2TP OpenVPN	✔	✔

Nome VPN	Stato	Dati raccolti	Log IP	DMCA	Tipi	P2P	BTC
TorGuard	USA	Dati personali	-	✔	PPTP L2TP OpenVPN SSTP	✔	-
OCTANEVPN	USA	Dati personali Indirizzo Email Dati di pagamento	-	✔	PPTP L2TP OpenVPN	✔	✔
SLICKVPN	USA	Indirizzo Email Username/ Password Dati di pagamento Google Analytics Cookies temporanei Dati Webserver	-	✔	OpenVPN	✔	✔
SECUREVPN.TO	Multiple	Dati personali	-	✔	OpenVPN	✔	✔
Steganos	Germania	Nome Indirizzo Numero di telefono	✔	?	?	?	?
VyprVPN	USA	Dati personali	✔	✔	PPTP L2TP OpenVPN	?	-
WiTopia	USA	Nome Indirizzo Email Numero di telefono Carta di credito	✔	✔	PPTP L2TP OpenVPN Cisco IPsec	-	-

Ti consiglio di fare estrema cautela ai **siti recensori di VPN**. Quest'ultimi hanno il vizietto di metter su dei portali fittizi dove sponsorizzano i loro servizi valutandoli 5 stelle e falsare risultati di ogni tipo. Mi raccomando, scegli con cura e confrontati con persone reali.

3.1.4.1. VPN MULTI HOP (A CASCATA)

Nel momento in cui un utente si connette a un servizio VPN il suo traffico Internet è protetto verso una singola VPN. Con **Multi Hop** si intende una caratteristica che definisce la pratica di connettersi ad una VPN da una VPN (e via dicendo). Le connessioni multihop offrono significativi vantaggi in termini di privacy e anonimato, garantendo non solo diversi strati di protezione delle informazioni ma anche la dislocazione della giurisdizione in cui operano le diverse VPN collegate tra loro.

Questo “hopping” potrebbe comunque causare rallentamenti non indifferenti e non credo che ci sia bisogno di spiegarne il motivo. Per il resto funzionano esattamente come le VPN a connessione diretta (*client->VPN*) con la sola differenza che tra i due si interpongono una o più VPN aggiuntive (*client -> VPN -> VPN* e via dicendo).

Al momento gli unici provider VPN (che sono riuscito a trovare) in grado di offrire questa soluzione sono:

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

3.1.5 Uso della VPN

Utilizzare una VPN in qualunque Sistema Operativo è un’operazione estremamente semplice, considerato che tutti i maggiori produttori si preoccupano di dare configurazioni già pronte da dar in pasto ai client di tunneling o meglio ancora fornirne di proprietari che si attivano in un click. Questo è valido per tutti i Sistemi Operativi ad eccezione del panorama GNU/Linux: la difficoltà di sviluppare per il mercato deframmentato del pinguino un unico strumento costringe i produttori a snobbare questo mercato, fornendo loro

solo la connettività dei protocolli. Questo problema però crea un vantaggio: la community Linux può affidarsi ad un unico client per gestire tutte le connessioni VPN, consentendo così anche a noi di avere un'unica strada da seguire. Nella configurazione di test useremo come provider NordVPN e il protocollo OpenVPN.

Da terminale scarichiamo e installiamo il client OpenVPN:

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

Rechiamoci alla cartella di installazione del programma:

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

Ogni provider fornisce una lista di VPN già pronti per essere dati in pasto al client. Scarichiamo questo file:

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

Abbiamo ora scaricato un file zippato (senza estensione). Dobbiamo estrarlo con il comando unzip:

```
$ unzip zip
```

Abbiamo ora tutti i file estratti. Mostriamoli con il comando ls:

```
$ ls -al
```

Una volta scelto il server a cui ci colleghiamo, lanciamo il comando openvpn:

```
$ openvpn [nomefile]
```

ad esempio:

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

Digitiamo Username e Password. Siamo ora collegati alla VPN e pronti a testare il tunnel di rete. Possiamo verificarlo scaricando il nostro IP in rete:

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

Per chiudere la connessione alla VPN usiamo la combinazione di tasti CTRL+C. Riverifichiamo ora il nostro IP.

3.1.6 Testare la qualità di una VPN

Finalmente hai affittato la tua VPN - o sei ancora in prova - ma non sei sicuro della scelta che hai fatto? In effetti non hai tutti i torti, soprattutto perché sai bene che ci sono delle dinamiche nel mondo Internet che sono molto complesse. Ad esempio un'errata configurazione di una VPN può consentirti di nascondere l'IP al sito finale ma la risoluzione del DNS potrebbe non essere criptata, consentendo quindi al tuo ISP di loggare le richieste ai domini e quindi rendere vana la cifratura.

> Nei Tester che presenteremo a breve vedrai alert relativi a Javascript, Apple-X, Cookies, WebRTC, Java… tutte queste vulnerabilità verranno trattate in un capitolo a parte denominato "**Risorse Locali**".

3.1.6.1 TORRENT TEST

I test che andremo ad eseguire ci permetteranno di assicurarci che la VPN funzioni correttamente anche con i protocolli P2P, in particolare Torrent. Qui purtroppo non è più sufficiente visitare la solita pagina "what is my ip address" ma è necessario sfruttare lo stesso client Torrent e una serie di mini-hacks. Vediamolo nel dettaglio. Innanzitutto ti introduco altri tre servizi web che offrono questo check:

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

Come effettuare il test VPN su Torrent

Per prima cosa armati del tuo client Torrent di fiducia quindi scarica uno speciale file .torrent o un magnet link e aprilo nel client Torrent (Figura 11).

Figura 11: Il torrent è in fase di scaricamento

A questo punto ogni servizio avrà un suo modo per fare il test: nel caso di *TorGuard* ci basterà scaricare il torrent e visualizzare la pagina dei tracker attivi; per verificare l'IP in uscita vedrai nello status del tracker l'IP assegnato dalla VPN (Figura 12).

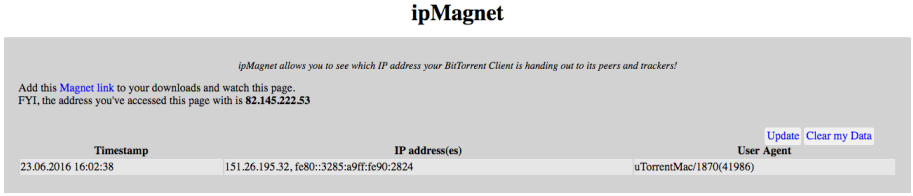


Figura 12: Dal sito ipMagnet vedrai come viene visto il tuo IP su Internet

Gli altri funzionano in maniera simile, basta solo che segui le istruzioni su ogni pagina web.

3.1.6.2 DNS LEAK TEST

In rete girano diversi servizi in grado di effettuare test per verificare se ci sono "perdite" tra te e i DNS. Verso l'inizio di questo manuale ne abbiamo già parlato, se per qualche motivo dovessi avere ancora dei dubbi ti consigliamo di tornare indietro e fare un ripassino! È possibile, in certe situazioni, che anche in una rete all'apparenza totalmente anonima il sistema operativo continui ad utilizzare i

DNS di default del proprio ISP, compromettendo totalmente l'anonimato dell'utente.

Il problema non è da prendere sottogamba: i normali servizi di recupero IP danno un falso senso di sicurezza all'utente sotto VPN, non allarmandolo che *non basta nascondere solo l'IP Address*. A questo si aggiunge anche un secondo problema: metti caso di aver modificato i tuoi DNS utilizzando i vari Google, OpenDNS, Comodo etc… penserai quindi tra te e te che il tuo ISP non è più in grado di leggere le tue richieste. Ebbene non è proprio così. Alcuni ISP sono in grado di "rileggere" la connessione al DNS sfruttando dei proxy DNS trasparenti.

3.1.6.3 Come difendersi dal DNS Leak

Se vuoi difenderti dal DNS Leak del tuo ISP devi fare in modo che il tuo sistema utilizzi i DNS della VPN *oppure* dei DNS alternativi. Prima di impazzire con il setup del tuo sistema operativo, assicurati che la tua VPN di default non abbia già la funzione di **DNS Leak Prevent**. Sebbene siano poche, le VPN che offrono questo servizi si contano sulla punta delle dita.

- 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)

Per quanto riguarda le *soluzioni software* al momento sono:

- VPN Watcher (a pagamento / disponibile per Windows, Mac, Android, iPhone, iPad / www.ugdsoft.com/products/vpnwatcher/)

- VPNCheck (a pagamento / disponibile per Windows, Linux / www.guavi.com/vpncheck_free.html)

- VPN Lifeguard (opensource / disponibile per Windows / https://sourceforge.net/projects/vpnlifeguard/)

- TunnelRat (free / disponibile per Windows / www.tunnelrat.net)

- VPNetMon (free / disponibile per Windows / vpnetmon.webs.com)

Questi software si occupano di verificare che i DNS siano sempre gli stessi indicati e, in caso qualcosa vada storto, provvederà a staccare la connessione Internet.

3.1.6.3 KILL SWITCH (PROTEZIONE DI CADUTA DELLA CONNESSIONE)

La Kill Switch (Figura 13) è un'importante - oserei dire vitale - funzione integrata all'interno di molti client VPN che permette di effettuare un *taglio alla rete* qualora il tunnel smetta di funzionare. Possiamo dire che è una specie di detonatore di rete che si attiva nel momento in cui la VPN stacca il tunneling e non è più disponibile.

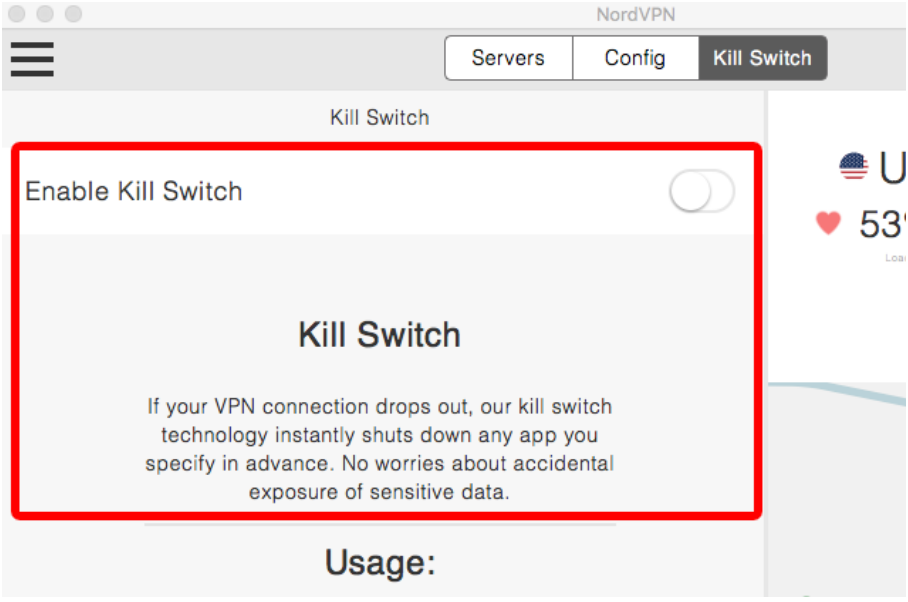


Figura 13: Funzione di Kill Switch integrata nel client di NordVPN

Senza questa funzione il tuo dispositivo, a VPN staccata, cerca di rieffettuare la connessione a Internet lasciandoti scoperto. È fortemente consigliato abilitarla soprattutto in caso si utilizzino applicazioni in *background* (ad es. scaricando da Torrent) o sia necessario allontanarsi dal dispositivo (ad es. se uno scan prende più tempo del dovuto).

Non è facile stabilire quali *provider VPN* offrono questa soluzione; ognuno di essi chiama la funzione “Kill Switch” con un nome proprietario, pertanto l’unico consiglio che posso dare è di effettuare una ricerca approfondita su ogni sistema e valutarlo con attenzione.

4. CLEARNET E DEEP WEB

Fino ad ora abbiamo parlato solo ed esclusivamente di come navigare in sicurezza e anonimato all'interno della **Clearnet**, vale a dire quella parte di Internet accessibile da qualunque dispositivo e motore di ricerca in grado di comunicare con i protocolli TCP/IP secondo gli standard più comuni. In realtà nel corso degli anni gli internauti hanno sentito l'esigenza di dover creare un nuovo tipo di rete che non fosse accessibile senza le dovute precauzioni. Questa rete oggi viene intesa come **Deep Web**.

Alcuni inconsciamente oggi ritengono che il Deep Web sia la parte "malvagia" di Internet, mentre la Clearnet (o Surface Web) sia quella legale. La verità è che con Deep Web si identifica quella parte non indicizzabile dal World Wide Web, ossia quel circuito che senza le dovute precauzioni (come l'uso di software specifici) non è accessibile. Quando invece ci si riferisce a quel mondo "distorto" fatto di vendita di armi, droghe e pedopornografia, ecco allora che il termine più corretto è Dark Net (o Dark Web per la navigazione via web). Qualora l'argomento fosse di vostro gradimento potete approfondire l'analisi sulla terminologia delle seguenti parole in un interessante articolo a riguardo[1].

Etimologia a parte, è importante non trascurare la possibilità di avere un circuito alternativo all'Internet comune. Un accesso al Deep Web può essere utile - se non fondamentale - per alcune operazioni come il tenersi in contatto con collaboratori, ottenere informazioni eliminate dalla Clearnet, acquistare exploit non ancora pubbliche e via dicendo.

Ok, perché tutta questa manfrina? Ora che conosciamo i fondamentali per la navigazione in anonimato nella Clearnet (sebbene ancora da approfondire nei prossimi capitoli) tratteremo per ogni software/network anche una piccola parte riservata al Deep Web e al come muoversi in questo particolare mondo.

[1] monicabarratt.net/a-discussion-about-dark-net-terminology/

4.1 TOR

È arrivato il momento di parlare di TOR[1]: si lo so, alcuni non ne sentono la mancanza e forse un po' hanno ragione, si parla sempre delle stesse cose! Cercherò di rendere il meno tediosa possibile questa parte, sorvolando le ovvietà e andando direttamente al sodo. Ma prima, un ripassino!

4.1.1 Cos'è la rete TOR

TOR è una rete anonima che è stata ideata per consentire la navigazione sicura proteggendo la privacy degli utenti. Questo software viene mantenuto dal *The Tor Project*, associazione che riceve fondi dal Dipartimento degli Stati Uniti d'America per lo sviluppo e la ricerca della rete TOR. Il logo della cipolla che rappresenta il progetto rende l'idea del funzionamento di questa rete: i server TOR agiscono da router riuscendo a costruire una rete privata virtuale a *strati,* proprio come una cipolla. Questa stratificazione è composta dai seguenti elementi:

- Client: gli utenti
- Middleman: i server che si occupano di effettuare il rimbalzo dei dati nella rete
- Exit router: i server finali della catena che si occupano di "uscire" in Internet
- Bridge router: simili agli exit router con l'eccezione che il loro identificatore è privato, permettendo di bypassare il blocco agli utenti TOR.

4.1.2 I TOR Projects

Per facilitare l'accesso alla rete TOR, il TOR Project ha avviato lo sviluppo di diversi progetti pensati per la navigazione in diverse occasioni. Questi sono:

[1] https://www.torproject.org

- Tor Browser (https://www.torproject.org/projects/torbrowser.html.en): un pacchetto contenente un browser (Firefox), il plugin HTTPS Everywhere (per forzare le connessioni SSL), il plugin NoScript (per bloccare Javascript) e ovviamente il client di Tor. È disponibile in versione con installer e portabile per tutti i Sistemi Operativi.

- Orbot (https://guardianproject.info/apps/orbot/): un client che permette di collegarsi alla rete TOR e proteggere il traffico di tutte le applicazioni di un dispositivo Android.

- Tails (https://tails.boum.org): una distribuzione GNU/Linux pensata per la navigazione anonima, consentendo di veicolare la connessione sulla rete TOR, fornita di strumenti di crittografia e tool che consentono di non lasciare tracce.

- Arm (https://www.atagar.com/arm/): tool da linea di comando che permette di monitorare e configurare la rete TOR.

- Atlas (https://atlas.torproject.org): uno strumento via web che consente di verificare lo stato dei relay della rete TOR.

- Pluggable Transports (https://www.torproject.org/docs/pluggable-transports.html.en): in quest'area troviamo quei software di terzi che vengono supportati per l'anonimato.

- Stem (https://stem.torproject.org): una libreria Python che consente di interagire con TOR.

- OONI (https://ooni.torproject.org): un software che permette di rilevare la manipolazione di traffico e il monitoraggio della nostra connessione da parte dei governi.

Riguardo Tor Browser è necessario sapere che nella sua versione preistorica era fornito della versione **Bundle** (chi ricorda Vidalia e Privoxy?) e della versione Browser.

4.1.3 Installazione di TOR

Grazie alla sua popolarità TOR è presente in quasi tutti i repository esistenti. Basterebbe infatti lanciare il comando:

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

Tuttavia in Debian raramente utilizzeremo l'ultima versione stabile e inoltre gli stessi sviluppatori del Tor Project sconsigliano di usare quelli presenti in Ubuntu e derivati, in quanto risultano essere poco aggiornati e quindi inaffidabili. La cosa migliore da fare a questo punto è quella di inserire i repository ufficiali di TOR direttamente nella nostra distribuzione Debian; per prima cosa allora apriamo con l'editor nano il file /etc/apt/sources.list

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

Usando Debian 8 Jessie, come consigliato dal sito ufficiale[1], aggiungiamo le seguenti righe a fine file:

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

salviamo con CTRL+X, premiamo "S" e clicchiamo Invio. Torneremo così al terminale. Ora per evitare di avere problemi con la certificazione dei file sarà necessario importare le chiavi GPG.

[1] https://www.torproject.org/docs/debian

```
$ gpg --keyserver keys.gnupg.net --recv
A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89
$ gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89
| apt-key add -
```

Diamo una bella aggiornata ai nostri repository, quindi installiamo il pacchetto TOR:

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

Ecco fatto, ora siamo pronti a usare TOR, che figurerà come un proxy in locale in ascolto sulla porta 9050 tramite SOCKS e sulla porta 9150 per Tor Browser (di cui parleremo a breve). Possiamo verificare lo status del servizio digitando:

```
$ service tor status
```

per fermarlo invece useremo:

```
$ service tor stop
```

per avviarlo:

```
$ service tor start
```

e per riavviarlo:

```
$ service tor restart
```

Uno dei metodi che uso per verificare il funzionamento di TOR è quello di utilizzare proxychains (che abbiamo visto nel capitolo dei Proxy) configurandolo affinché si colleghi al proxy in locale di TOR. Prima di tutto però assicuriamoci di vedere dove effettivamente si trova TOR e su quale porta ascolta:

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

Il comando netstat ci permette di ottenere tutta la lista dei processi attivi che fanno uso delle risorse di rete mentre con grep filtreremo i risultati solo per i processi che indichiamo. L'operatore | (pipe) serve a concatenare i due programmi. Il risultato di questa espressione darà 127.0.0.1:9050, dove 127.0.0.1 è l'ip in locale (quindi del nostro PC) e 9050 la porta in uso. Prima di modificare la configurazione di proxychains ritorniamo utenti normali:

```
$ exit
```

quindi riapriamo il file proxychains.conf

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

e modifichiamolo in questo modo

```
dynamic_chain
proxy_dns
[ProxyList]
socks4 127.0.0.1 9050
```

salviamo con *CTRL+X*, *tasto S* e *INVIO*. Notiamo come abbiamo modificato il valore strict_chain in dynamic_chain, questo perché durante l'uso di TOR è possibile scontrarsi contro relay non operanti. La funzione dynamic_chain ci permette di essere più elastici nell'uso dei proxy, mentre strict_chain sarà talmente fiscale da bloccare ogni possibile modifica della struttura del proxy.

Verifichiamo ora il nostro IP attuale:

```
$ wget http://ipinfo.io/ip -qO -
82.51.116.171
```

e confrontiamolo con quello in uscita tramite 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
```

Ovviamente potrai configurare tutto il sistema affinché il traffico dell'intero Sistema Operativo passi attraverso il network-manager oppure modificando il file di configurazione /etc/environment come visto nel capitolo dedicato ai Proxy.

È importante considerare che in caso si volesse utilizzare TOR per la navigazione web potrebbe essere necessario far uso di Privoxy, un servizio di web proxy in grado di mutare le richieste HTTP, disabilitare pubblicità e molto altro. Questo è già integrato in TOR browser, quindi se hai necessità di navigare con TOR ti consigliamo di proseguire. In alternativa, visita la pagina ufficiale del sito[1] e procedi alle FAQ dedicate.

4.1.4 Gli usi di TOR

Una volta che TOR è attivo all'interno del nostro Sistema Operativo possiamo utilizzarlo in diversi modi. Vediamo quali sono i servizi e gli usi più comuni.

4.1.4.1 TOR COME BROWSER

Il Tor Browser Bundle è forse il progetto più famoso del TOR Project. È un browser basato su *Firefox ESR* preconfigurato per collegarsi al proxyserver *SOCKS* interno di TOR all'indirizzo *127.0.0.1:9150*. Assieme al browser vengono forniti:

[1] www.privoxy.org

- *TorLauncher* che si occupa di avviare il collegamento alla rete TOR in modalità fantasma;

- *TorButton* che permette di controllare le identità e impostazioni del client TOR;

- *NoScript* che previene l'esecuzione del codice Javascript (per maggiori informazioni salta al capitolo sulle Risorse Locali);

- *HTTPS Everywhere* che forza le connessioni web a utilizzare il protocollo HTTPS (anche qui approfondire al capitolo Risorse Locali).

Il client è disponibile nelle versioni *Windows*, *OSX* e *Linux* all'indirizzo web ufficiale Tor Browser[1]; è possibile scaricarne tre versioni:

- *Stable*, la versione stabile

- *Experimental*, la versione nightly più aggiornata (ma meno testata)

- *Hardened*, la versione alpha del progetto disponibile solo per Linux x64[2]

Installazione di TOR Browser

Mentre per *Windows* e *MacOS* troveremo i binari da eseguire con un doppio click, su *GNU/Linux* ci divertiremo un po' con il terminale al fine di prenderci il manico. Procuriamoci la versione desiderata (italiano andrà benissimo) e l'architettura disponibile, scaricandola dal sito ufficiale. Se per qualche motivo ci sono dubbi su quale scaricare, preferiamo sempre la *32-bit*.

Una volta scaricato il file apriamo il terminale e rechiamoci nella cartella dei file scaricati:

```
$ cd $HOME/Scaricati
```

[1] https://www.torproject.org/projects/torbrowser.html.en

[2] https://blog.torproject.org/blog/tor-browser-55a4-hardened-released

Nel nostro caso il file si chiama “tor-browser-linux32-6.5a3_it.tar.xz”. Questo lo sappiamo perché abbiamo ottenuto la lista dei file presenti lanciando il comando:

```
$ ls
```

Procediamo ora all’estrazione del file compresso:

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

Un consiglio: potrebbe risultare tedioso ogni volta scrivere a mano il nome di un file o di una cartella. Usando un terminale a base UNIX è disponibile l’autocompletamento di un file: per farlo si digita una parte del nome (es tor-) e quindi si completa con il tasto [TAB]. Ad esempio:

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

In questo modo il terminale auto-completerà il nome del file. Verrà estratta la cartella che contiene l’eseguibile: essa si chiamerà tor-browser_it/. Entriamoci con il comando:

```
$ cd tor-browser_it
```

Per lanciare l’eseguibile è presente uno script chiamato start-tor-browser.desktop. Avviamolo con il comando:

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

Ulteriori informazioni su TOR Browser

Il TOR Browser Bundle può essere utilizzato sia in *clearnet* che in *deepweb*. La comodità che risiede in questo software, oltre alla portabilità che consente di utilizzarlo anche da supporto esterno come chiavette USB o SD, è l'avere preinstallato il core TOR e il TorButton (Figura 14) che permette di manovrare le connessioni senza una GUI esterna (come succedeva con la vecchia versione). L'intera rete TOR viene quindi gestita dal TorButton cliccando sulla cipolla verde vicino alla barra degli indirizzi del browser.

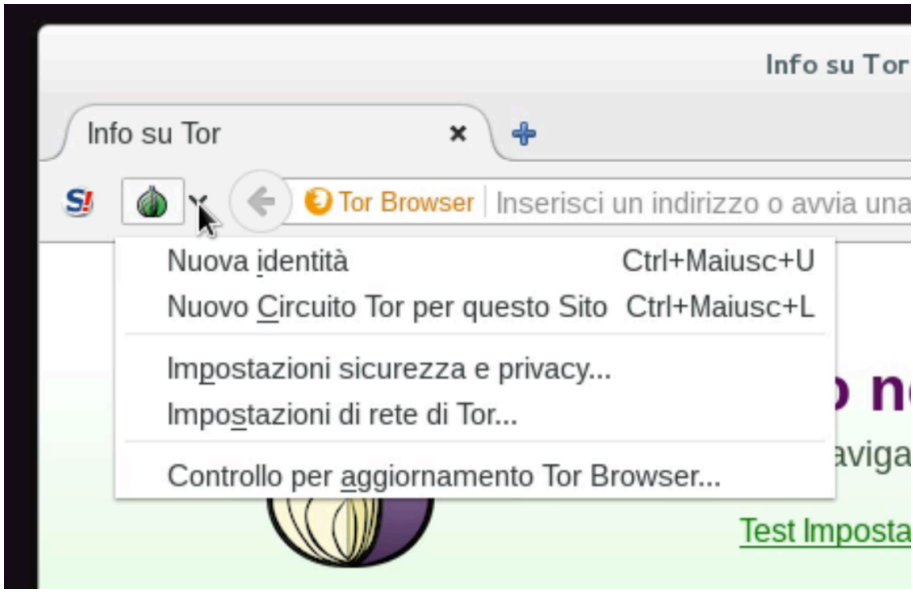


Figura 14: Pulsante TOR Button su Firefox

Nella voce *Impostazioni Sicurezza e Privacy* possiamo decidere di operare su quattro caratteristiche già presenti nelle preferenze di Firefox e grazie ai Livelli di Sicurezza utilizzare quattro profili utente che ne determinano il livello di "paranoia" (Figura 15).

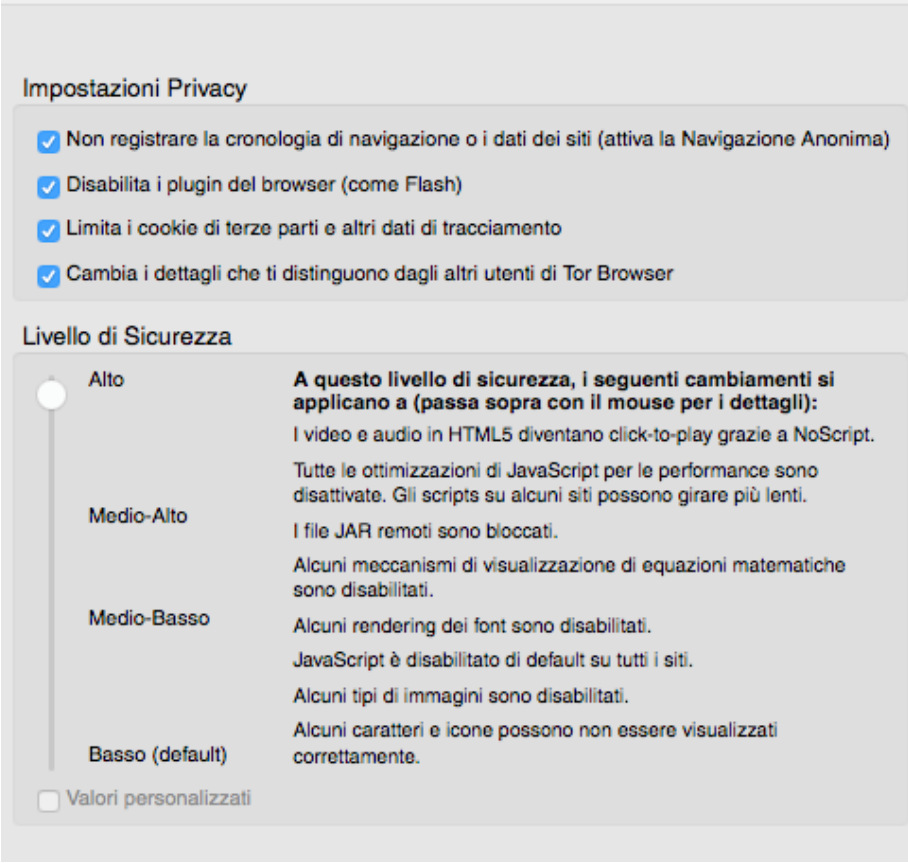


Figura 15: Impostazioni avanzate di TOR Browser

4.1.4.2 TOR COME P2P

Il *TOR Project* sconsiglia la condivisione P2P di qualunque genere[1], riferendosi a Torrent in particolare poiché quello più famoso. I motivi per cui Tor non andrebbe usato per la condivisione P2P sono essenzialmente due:

1) La rete Tor non ha la capacità di supportare applicazioni ad alto volume di consumi di banda. Se tutti i "Tor-nauti" condividessero file utilizzando la tecnologia P2P, la rete Tor si saturerebbe.

2) La rete Torrent potrebbe "tradirti". Torrent, e molti altri network P2P, hanno bisogno di comunicare il tuo indirizzo IP a un database pubblico per

[1] https://blog.torproject.org/blog/bittorrent-over-tor-isnt-good-idea

metterti in contatto con i tracker e quindi collegarti ai peer. In questo senso, il client Torrent potrebbe inviare il tuo indirizzo IP direttamente al tracker così uscire dal network Tor per la fase di download/upload, effettuandone una connessione diretta.

In realtà con le giuste precauzioni è possibile usare Torrent ma in ogni caso non è consigliato. Per condividere in anonimato nei circuiti P2P si consiglia l'uso delle VPN o di I2P (di cui parleremo in seguito).

4.1.4.3 TOR COME CHAT

Servizi come Gmail, Hotmail, Skype, Facebook Messenger così come i vecchi Yahoo! Messenger e MSN e qualunque altra forma di comunicazione su Internet può essere monitorata e memorizzata per tempi davvero lunghi, anche per più di 5 anni. Più in avanti parleremo anche di come cifrare i messaggi all'interno della rete ma fino ad allora introduciamo solo il software TorChat.

TorChat[1] è un programma di instant messenger decentralizzato e anonimo che fa uso della rete Tor per comunicare su Internet tramite il meta-protocollo .onion . Questo permette lo scambio di messaggi e dati multimediali cifrati end-to-end. TorChat ha il pregio di essere disponibile nativamente per *Windows, Linux* e gli smartphone di nuova generazione. Esiste anche una versione non ufficiale per i sistemi OSX[2], da usare a tuo rischio e pericolo.

Installazione di TorChat

Se abbiamo integrato i repository del TOR Project per installare TOR allora avremo anche disponibile l'installazione torchat. Prima di tutto però eseguiamo un aggiornamento del sistema:

[1] https://github.com/prof7bit/TorChat

[2] www.sourcemac.com/?page=torchat

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

Notiamo come, per la prima volta, evochiamo il simbolo di concatenazione &&. Questo ci permette di eseguire due comandi distinti che non devono comunicare tra di loro - a differenza di quanto visto con il simbolo | (pipe). I due comandi apt-get update e apt-get upgrade servono rispettivamente ad aggiornare i repository e i software presenti nel nostro sistema. Raggiunta questa fase installiamo tranquillamente torchat:

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

Ad installazione finita possiamo avviarlo direttamente da terminale digitando:

```
$ exit
$ torchat
```

Come funziona TorChat

In TorChat ogni utente ha un ID univoco alfanumerico composto da 16 caratteri. Questo ID viene creato casualmente da Tor quando il client viene avviato per la prima volta e si presenta fondamentalmente come un indirizzo .onion . A questo punto si otterrà un codice come ad esempio *murd3rc0d310r34l.onion*, quindi il tuo ID sarà *murd3rc0d310r34l*. Questo potrà essere comunicato ad altri utenti che vorranno messaggiare con te.

Riguardo la sicurezza di TorChat

In molti casi si è discusso, e se ne sta ancora parlando, circa la reale sicurezza che *TorChat* offre ai suoi utenti. Il dubbio nasce dal funzionamento stesso del tool: esso crea un servizio all'interno del computer ospite e si occupa di trasferire semplicemente dei dati (un po' come avviene con netcat) sottoponendo il

computer agli stessi attacchi di deanonimizzazione già utilizzati in qualunque altro network anonimo.

Il secondo problema può essere quello riguardante il trasferimento dei dati: non c'è un controllo manuale sull'accettazione del trasferimento di un file e tutta la parte temporanea viene scritta nel path */tmp*: teoricamente un attacker potrebbe trasferire dati random al tmp del Sistema Operativo, che essendo montato in RAM, ne causerebbe il crash. Nei casi più gravi si potrebbe addirittura ipotizzare un exploiting della macchina a seguito di un overflow o di altri tipi di attacchi teoricamente accettabili.

L'ultimo elemento che lascia in dubbio riguarda l'impossibilità di prevenire ad altre persone che sappiano se e quando un ID di *TorChat* è online oppure no e nel caso si voglia tagliare i rapporti con qualcuno bisogna creare un nuovo *TorChat* ID. In definitiva *TorChat* è uno strumento utile che deve essere utilizzato solo ed esclusivamente con persone di cui ci fidiamo e di limitarne comunque l'utilizzo solo se strettamente necessario.

4.1.4.4 TOR COME PROXY SOFTWARE

Come per i Proxy, e a differenza dei tunnel VPN, è necessario configurare i propri tool affinché operino nella rete TOR. Una volta che TOR è attivo abbiamo disponibile un proxy SOCKS a tutti gli effetti all'interno del nostro computer.

A questo punto ci è possibile eseguire i nostri software proxati con Proxychains o Proxycap (vedi capitolo riguardante i Proxy Server) collegandoci all'indirizzo 127.0.0.1 (o localhost) tramite la porta 9050. Abbiamo affrontato questa situazione quando abbiamo installato e testato TOR (e non TOR Browser), quindi torna a qualche capitolo precedente per sapere come fare.

4.1.5 I TOR Relay

Nell'universo di TOR, i Relay si occupano di regalare banda agli utenti della rete che così possono usufruirne gratuitamente. Il torporject[1] raccomanda agli utenti TOR di attivare la funzione Relay nel caso in cui abbiamo più di 250kb/s sia in upload che in download.

Nello schema che mostra la lista degli elementi TOR, i Relay fanno parte della categoria Middleman ed Exit Node: chiunque può decidere di eseguire un Relay nella propria rete e specificare se vuol essere un Middleman, un Exit Node o entrambi. Ai fini di questa guida il setup di un relay non è fondamentale, tuttavia se vuoi contribuire allo sviluppo della rete TOR puoi farlo creando un tuo relay personale.

4.1.6 I TOR Bridges

I bridge di TOR - chiamati *bridge relays* - sono dei nodi della rete TOR che permettono di bypassare il filtraggio da parte di ISP e siti web circa l'uso della rete TOR. Per far in modo che il sistema funzioni efficacemente, non è disponibile una lista completa dei bridge relays altrimenti ISP ed eventuali honeypot di siti web li riconoscerebbero subito e li bloccherebbero.

È possibile tuttavia impartire al client di TOR Browser l'ordine di utilizzare i bridge utilizzando l'opzione "*Il mio fornitore di servizi Internet (ISP) blocca le connessioni alla rete TOR*" , voce attivabile nelle *Impostazioni di rete di TOR* (nel caso in cui usiate il TOR Browser cliccate sull'icona della cipolla verde in alto a sinistra).

[1] https://www.torproject.org/docs/tor-relay-debian.html.en

4.1.6.1 USO AVANZATO DEI BRIDGES

Nel caso in cui volessimo impostare manualmente i propri bridge, ad esempio si vuole usare Tor Expert Bundle, distribuzioni Linux TOR-based come Tails oppure con TOR Browser tramite la configurazione avanzata, per prima cosa è necessario raggiungere la pagina Bridge di Torporject (https://bridges.torproject.org/bridges), saltare allo step2, compilare (l'impossibile) captcha in alto e ottenere così un valore di questo tipo (le *** sono state aggiunge):

```
92.***.0.174:9001
65B2F8E594190A3***********59B0E32FC45720
194.***.208.26:27049
```

Possiamo avviare TOR Browser e dargli in pasto i nuovi bridges ottenuti (solo Figura 16).

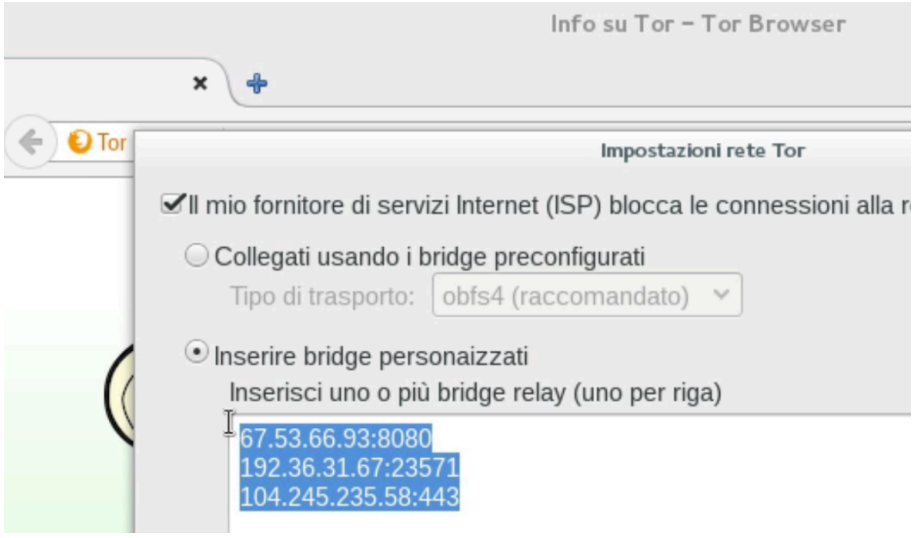


Figura 16: Inserimento dei bridge su TOR

4.1.7 I Pluggable Transports

Considera però che anche i bridges possono essere blacklistati in quanto il loro accesso può essere effettuato da chiunque, censori compresi. Per aggirare questo controllo gli sviluppatori di TOR hanno introdotto una nuova funzionalità chiamata *pluggable transports*. I P.T. hanno il compito di trasformare il flusso del traffico TOR in traffico "pulito" tra il client e il bridge che altrimenti potrebbe essere intercettato dall'ISP con una tecnica chiamata *Deep Packet Inspection* (DPI) che consiste nel classificare i flussi di traffico IP e, una volta confrontato il pattern, bloccati a monte.

Al momento la tecnologia P.T. è in fase di sviluppo attivo e necessita di operatori e sviluppatori per integrarla al meglio nel TOR Project. Per maggiori informazioni visita la pagina del sito ufficiale[1]. I P.T. più comuni al momento sono definiti *bridge offuscati*: per definizione, questi bridge si occupano di offuscare il traffico difficilmente interpretabile per gli ISP. La loro tecnologia fa uso di algoritmi che mischiano i pacchetti in entrata e in uscita; tali algoritmi vengono identificati dai protocolli. I protocolli di questo tipo sono tre: *obfs2*, *obfs3* e *obfs4*.

Obfs2 (la versione due, chiamata anche "Twobfuscator") è il più semplice dei due: la sua tecnologia si occupa di prendere i dati in entrata e uscita nel traffico e riordinarli casualmente. Com'è emerso negli ultimi studi, questo protocollo può essere crackato intercettando l'handshake iniziale (un po' come accade con la sicurezza WEP delle reti WiFi) rivelando così le informazioni contenute. È una versione deprecata non più in sviluppo né supportata dallo sviluppo TOR.

Obfs3 (il "Threebfuscator") è molto simile al protocollo precedente ma utilizza il metodo Diffie Hellman per lo scambio delle chiavi in fase di handshake (argomento che verrà spiegato in "Crittografia").

[1] https://www.torproject.org/docs/pluggable-transports.html.en

Obfs4 è la quarta versione di questo protocollo, sebbene come dice lo sviluppatore stesso "è più vicino a ScrambleSuite che a obfs2/obfs3". L'ultima versione di questo protocollo sembra essere quella più sicura e attualmente presente di default su Tor Browser. Informazioni aggiuntive di questo protocollo sono presenti nel Github ufficiale[1]. È possibile ottenere una lista di Obfs4 nella pagina ufficiale di Tor Project[2].

4.1.7.1 PROTOCOLLI MEEK E SCRAMBLESUIT

TOR è in grado di comunicare con molti altri protocolli oltre quelli della famiglia Obfs* (Figura 17).

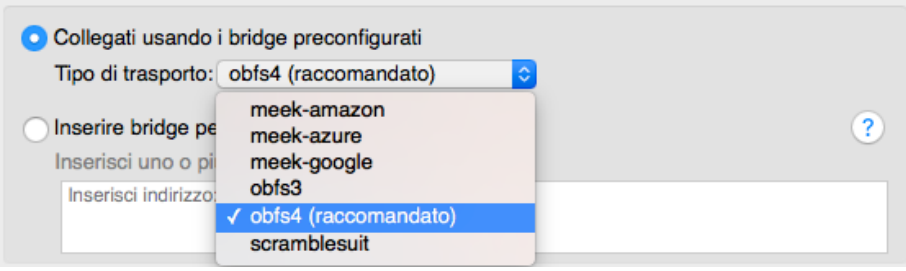


Figura 17: Selezione dei bridge su TOR

Meek-*

I protocolli della famiglia **meek-*** sono stati ideati nel 2014 per permettere di effettuare un tunneling in un circuito HTTPS. Viene utilizzata anche una tecnica chiamata "*domain fronting*" che si occupa di nascondere all'ISP il fatto che si sta comunicando con un TOR bridge. Come puoi vedere affianco alla sigla meek- troverete il nome di un noto servizio web: nel caso si scelga Amazon ad esempio l'ISP crederà che stiamo comunicando con il noto sito ecommerce (o meglio con la cloud AWS), Azure con il cloud Microsoft e Google… beh, con i servizi Google. Come spiega anche il TOR Project i protocolli a base meek-* sono più lenti dei vari

[1] https://github.com/Yawning/obfs4

[2] https://bridges.torproject.org/bridges?transport=obfs4

obfs-* e andrebbero utilizzati solo nel caso in cui l'ISP provveda a bloccare quest'ultimi. Se la situazione lo richiede è possibile seguire la guida ufficiale di TOR Project che spiega come configurare il client per usare questo meek[1]; in caso di dubbi puoi tranquillamente saltare questo tipo di protocollo (o al massimo eseguire dei test). Al momento sembrano essere l'unica valida alternativa in caso di censorship avanzata, come avvenuto in Cina a fine 2015, tuttavia è ancora acerba e nel corso degli anni la situazione potrebbe evolversi.

ScrambleSuit

Il progetto ScrambleSuit - citando la pagina ufficiale di Github[2] - nasce per risolvere due problemi:

- Proteggere l'utente dagli attacchi di monitoraggio richiedendo un "segreto" condiviso tra client e server sfruttando una comunicazione out-of-band tramite BridgeDB (il servizio di bridge listing di TOR).
- Proteggere dagli attacchi di analisi alternando il flusso dati. ScrambleSuit è in grado di alterare il tempo e la lunghezza dei pacchetti che vengono comunicati.

ScrambleSuit è stato pensato per essere un protocollo di trasporto indipendente i protocolli SOCKS, quindi i vari HTTP, SMTP, SSH e via dicendo. Questo dovrebbe far intendere meglio come funziona anche il protocollo Obfs4; quest'ultimo viene dichiarato però più stabile e veloce[3], pertanto si consiglia l'utilizzo di questo protocollo solo se Obfs4 non è disponibile. Allo stato attuale ScrambleSuit non è più in via di sviluppo.

---

[1] https://trac.torproject.org/projects/tor/wiki/doc/meek#Quickstart

[2] https://github.com/NullHypothesis/scramblesuit

[3] https://blog.torproject.org/blog/recent-and-upcoming-developments-pluggable-transports

Tra i protocolli abbandonati troviamo anche SkypeMorph[1], Dust[2] e FTE[3]. Tutta la documentazione completa sui relay e i P.T. è disponibile nella pagina ufficiale di Tor Project[4].

4.1.8 Testare la sicurezza di TOR

In questa parte del documento ci preoccuperemo di effettuare dei test per controllare la sicurezza di TOR Browser.

> Come per le VPN, nei Tester via Browser vedrai alert relativi a Javascript, Apple-X, Cookies, WebRTC, Java… tutte queste vulnerabilità verranno trattate in un capitolo a parte denominato "**Risorse Locali**".

4.1.8.1 TEST TOR VIA BROWSER

Il sito a cui faremo riferimento per i nostri test sarà TorCheck[5], prodotto da xenobite. I risultati dei test sono visibili in Figura 18 e 19.

[1] cacr.uwaterloo.ca/techreports/2012/cacr2012-08.pdf

[2] https://github.com/blanu/Dust

[3] https://realworldcrypto.files.wordpress.com/2013/06/shrimpton_rwc-2014-fte-release.pdf

[4] https://www.torproject.org/docs/bridges.html.en#FindingMore

[5] https://torcheck.xenobite.eu/index.php

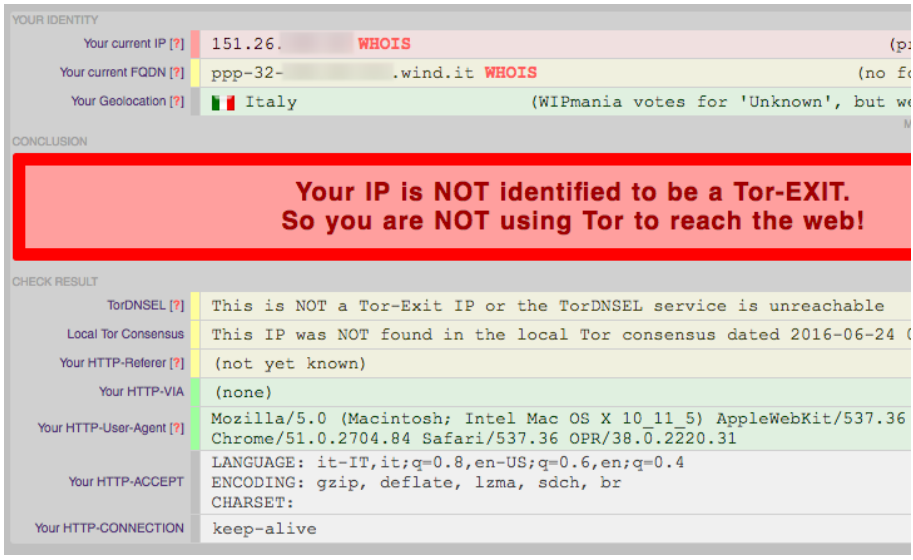


Figura 18: TorCheck senza l'uso di TOR Browser

Figura 19: TorCheck con l'uso di TOR Browser

Da questa pagina riceviamo un report che ci indica una serie di voci e valori. Com'è possibile vedere sulla screen in alto, i campi contrassegnati con lo sfondo in *verde chiaro* indicano una buona protezione, mentre i campi contrassegnati con lo sfondo in *rosso* chiaro indicano qualcosa da risolvere (nb: nel nostro

esempio riceviamo la voce “Your real IP” come rossa, forse a causa di un bug). Vediamo in cosa consistono queste voci:

- ***Your real IP***: il tuo indirizzo IP reale. Se questo viene mostrato significa che la tua sicurezza può essere compromessa.
- ***Your current IP***: qui viene indicato l’indirizzo IP che viene mostrato al sito che stai visualizzando. Se tutto va secondo i nostri piani, ricevi un indirizzo IP estraneo al tuo (sarà quello dell’exit node)
- ***Your current FQDN***: con FQDN si fa riferimento al nome del dominio che specifica i livelli del DNS. Questo identificatore ci avvisa che il nostro indirizzo IP viene ancora loggato dall’ISP durante la risoluzione dei domini.
- ***Your Geolocation***: qui viene indicata la posizione geografica risalendo all’indirizzo IP. Tale posizione è approssimativa e fa riferimento alla centrale dell’ISP e non all’indirizzo fisico di chi usa la connessione.
- ***TorDNSEL***: qui si verifica se l’indirizzo IP che “esce” fa parte della lista degli Exit Node. Questa voce è importante in quanto ci permette di sapere se la nostra connessione in uscita non sia stata manipolata oppure se l’exit node viene riconosciuto come proveniente da TOR.
- ***Local Tor Consensus***: qui non si sono documentazioni in merito.
- ***Your HTTP-Referer***: qui si può verificare se lasciamo tracce di tipo Referer. Il valore Referer permette a un sito web di vedere da dove il client arriva (es: da una ricerca, da un sito, da una mail etc…).
- ***Your HTTP-Via***: qui viene mostrato il valore che informa il server che tipo di richiesta viene effettuata tramite il proxy di Tor (*es: Via: 1.0 fred, 1.1 inforge.net (Apache/1.2)*)
- ***Your HTTP-User-Agent***: qui viene mostrato il lookup del nostro browser e sistema operativo. L’HTTP-User-Agent può essere manipolato e vedremo come fare nel prossimo capitolo riguardante le Risorse Locali.

- ***Your HTTP-ACCEPT***: qui vengono mostrati i valori che il tuo browser accetta, ad esempio possono esserci informazioni sulla lingua, sui cookie, sulla cache e via dicendo.
- ***Your HTTP-CONNECTION***: qui viene segnalato il valore della Connectionn del browser. Solitamente troverai il valore keep-alive

4.1.9 TOR e il Deep Web

La rete TOR è forse il più famoso strumento per accedere al Deep Web, o meglio per accedere al Deep Web di TOR. Senza di esso il nostro browser non sarebbe in grado di risolvere i domini con estensione .onion, vale a dire quei siti che vengono hostati - ovvero serviti in rete - da server e computer collegati a TOR.

4.1.9.1 DOVE TROVARE I SITI .ONION

Questa è una bella domanda. Quando cerchi qualcosa dove vai? Ovviamente su Google! Come ti ho già detto però Google (ma anche Bing, Yahoo e compagnia bella) sono la peste nera di chi vuole mantenere l'anonimato. Quindi cosa fare?

Il primo passo che un aspirante *deepnauta* dovrebbe intraprendere è fornirsi della **The Hidden Wiki**, una pagina in puro stile Wikipedia che raccoglie alcuni dei principali siti .onion in circolazione. Per trovare la Hidden Wiki basta googl… ehm, cercare sul motore di ricerca la parola chiave "The Hidden Wiki" e finire in qualche sito - meglio se con una certa autorevolezza - così da ottenere un indirizzo .onion tipo questo: http://zqktlwi4fecvo6ri.onion (al momento è quello attivo ma potrebbe andare down) o addirittura siti web in clearnet.

A proposito della Hidden Wiki: ce ne sono davvero tante, le più rinomate al momento sono quelle di "ion" che però non è molto aggiornata. In alternativa alla official abbiamo la "Mirror Version" che risulta essere quella più completa. Una terza scelta invece può essere la HackBlock's Hidden Wiki, aggiornabile dalla community (ma attenzione a cosa visitate). Ad ogni modo ne vengono create a

decine ogni giorno (e altrettante ne vengono chiuse) quindi è necessario lavorarci un po' su con tanta pazienza e un buon motore di ricerca.

4.1.10 La rete TOR è davvero sicura?

Negli anni il TOR project si è ritagliato una fetta importante nel mondo di Internet, riconosciuto ormai come il network anonimo per eccellenza.
Alcune testate giornalistiche lo hanno immolato tanto da definirlo lo "strumento di navigazione anonima perfetto", distorcendo un po' quella realtà che accomuna tutti i software, ovvero che "la perfezione non esiste".

TOR è tutt'altro che perfetto: è pur sempre un software, un programma costruito da esseri umani in grado di sbagliare. Ed è stato anche violato: l'FBI - con l'aiuto di ricercatori esterni finanziati dal Dipartimento della Difesa - ha compromesso tra il 30/01/2014 e il 04/07/2014 il circuito, riuscendo a monitorare centinaia di migliaia di connessioni (e facendo chiudere il famoso Silk Road). Un anno prima, sempre l'FBI riuscì ad arrestare un pedofilo irlandese sfruttando un bug presente in Firefox 17, la stessa versione usata da Tor Browser.
L'ultima attacco risale al 2015 quando un'università ha ricevuto 1 milione di dollari per sabotare la rete TOR[1]. Di questo ne parleremo tra poco.

Ma allora, *TOR si o TOR no?* Dunque, TOR è uno strumento e se usato bene può dare grandissimi vantaggi in termini di anonimato ma affinché se ne sfruttino le reali potenzialità è necessario saperci mettere mano.

[1] https://blog.torproject.org/blog/did-fbi-pay-university-attack-tor-users

4.1.10.1 TOR E IL PROTOCOLLO HTTP

Come abbiamo già detto Tor Browser viene fornito di default di HTTPS Everywhere, un addon pensato per Firefox che forza le connessioni HTTPS (HTTP + protocollo SSL/TLS) verso i siti web.

Perché si dovrebbe forzare la connessione HTTPS? Semplicemente perché TOR è solo un router di traffico e non un programma in grado di criptare i dati in rete. Il compito della rete Tor è solo quello di assicurare l'anonimato della sorgente della richiesta, cifrando le connessioni interne, ma non effettua il processo di crittografia all'esterno del circuito Tor. Questa operazione viene quindi effettuata dal protocollo HTTPS, sempre se il sito ospite supporti tale protocollo (in caso contrario potresti aver difficoltà a navigare nel sito stesso). Gli attacchi di intercettazione di dati all'interno di una rete non cifrata vengono chiamati "eavesdropping".

4.1.10.2 TOR E GLI EXIT-NODE COMPROMESSI

Uno dei rischi maggiori che si possono correre navigando nella rete TOR è quello di imbattersi in un exit node compromesso: l'exit node è l'ultimo nodo che da TOR arriva alla rete Internet. Senza le dovute precauzioni, il traffico che viaggia in entrata e in uscita da un exit node potrebbe essere non cifrato, il che significa che il proprietario di un exit node (come un'agenzia di spionaggio) riuscirebbe a monitorare il traffico di rete.

Il solo collegamento alla rete TOR non permette però di risalire al mittente della richiesta, anche perché la natura stessa di TOR impedisce ciò (ricordiamoci che TOR è costruita su più collegamenti tra computer che serve appunto a non rendere rintracciabile la sorgente della richiesta): si può tuttavia risalire alle informazioni in chiaro che si pubblicano in rete come informazioni personali, email, password e via dicendo. Gli exit-node possono anche reindirizzare gli

utenti verso siti web fasulli per rubare i dati personali degli utenti; questo è anche uno dei principali motivi per cui bisogna preferire sempre connessioni HTTPS (in caso di siti web fasulli riceverai una notifica di certificato non corretto).

4.1.10.3 TOR BROWSER, I PROBLEMI DEL "PRECOTTO"

Il Tor Browser Bundle è un progetto sviluppato dal The Tor Project in collaborazione con la EFF: esso è il primo - e spesso unico - passo per chi vuole interagire subito con la rete Tor. Il problema dell'utilizzare il Tor Browser Bundle è causato dalla natura del bundle stesso: essendo un pacchetto All-in-One che lo rende uno Starter Pack da l'illusione di protezione all'utente che quindi si dimenticherà ben presto delle prossime pagine, pensando: "Hey, ma chi me lo fa fare di configurarmi tutto per filo e per segno? Tanto esiste il Bundle". Sapete come sono stati arrestati quelli di Freedom Hosting (il servizio web che offriva hosting a molti siti della darknet) ? Sfruttando vulnerabilità all'interno del Tor Browser Bundle. A buon intenditor…

4.1.10.4 TOR, GOOGLE & CO.

Negli anni Google è riuscita a creare una rete tra i propri servizi in grado di anticipare le domande e i desideri di un utente. Considera che i servizi Google sono presenti un po' dappertutto: Browser, Sistema Operativo (Android e Chrome OS), Account, Addon, Prodotti e via dicendo Ripeto, non è una cosa impossibile essere anonimi al 100% passando per Google ma è comunque altamente sconsigliato: meglio utilizzare motori di ricerca che non loggano IP e dati di ricerca come DuckDuckGo oppure StartPage

4.1.10.5 TOR NON È A PROVA DI IDIOTI

Perdonami questa parte ma andava fatta, in un modo o nell'altro… Come si può pretendere di essere anonimi se, mentre stiamo acquistando un nuovo exploit

sulla Dark-Net, siamo collegati al nostro account Facebook? No non è una follia, succede anche spesso: c'è chi ad esempio sotto Tor fa la verifica a due fattori del proprio account (lasciando il proprio numero di cellulare!), chi accede alla propria posta, chi si registra con i propri dati personali e via dicendo. Quella che andrò a raccontare ora è la storia di un TOR abusing ai danni dell'università di Harvard.

Il 18 Dicembre 2013 è stato arrestato un ragazzo di 20 anni che risponde al nome di Eldo Kim. Il suo reato è stato quello di procurare un allarme bomba all'Università di Harvard, a Cambridge, per evitare di presentare alcuni esami finali.

Per fare ciò Eldo ha utilizzato un software di anonimizzazione chiamato TOR e un servizio di posta spazzatura Guerrilla Mail, che permette di creare e inviare temporaneamente email senza i dati dell'utente.

Il software TOR ha svolto egregiamente il suo lavoro, nascondendo il suo operato sia all'ISP che al servizio di posta, ma non alla sua Università. Eh già, il caro Eldo ha commesso l'errore di effettuare tutto tramite il WiFi dell'Università che, per evitare abusi, consente l'accesso solo tramite username e password assegnati a ogni matricola. Tramite un controllo incrociato di dati d'accesso al Wifi e ai protocolli e server in uso si è scoperta l'identità del ragazzo che poi ha ammesso di aver fatto ciò di cui è stato accusato. In quel caso è stata l'ingenuità di non rendersi conto - o addirittura di non ricordare - che erano necessari user e pass per accedere al network: come ogni Hotspot, ai dati vengono associati un indirizzo IP locale che a sua volta memorizza nei logs le attività. Il ragazzo è stato condannato a cinque di prigione e a una multa di 250.000$.

Immagino che questa storia basti a rendere chiara l'idea del messaggio che voglio dare, e non solo per quanto riguarda la "stupidità" ma soprattuto le conseguenze che potrebbero essere intese come sproporzionate al reato commesso. Immagina le conseguenze nell'acquistare illegalmente qualcosa nella Dark Net o pubblicare

un messaggio scomodo in una nazione dittatoriale dove vige la pena di morte. Ricordati che TOR non è magia: è un programma che connette tanti utenti alla stessa rete. Che tu ne conosco la struttura di programmazione o sappia semplicemente come funziona è pur sempre un tool scritto da esseri umani (che possono sbagliare) e da solo non garantisce l'anonimato al 100%. Usa la testa.

4.2 I2P

Nel panorama Internet spesso si sente di parlare di **I2P**, il network alternativo a Tor. I2P (di default) non permette di navigare nella clearnet, la parte "pulita" di Internet, ma è un progetto espressamente pensato per la navigazione nella propria darknet, quindi accostarlo a TOR non è proprio esatto. Ma andiamo per ordine.

A differenza di TOR che richiede gli Onion Router per sopravvivere, **I2P** (acronimo di *Invisible Internet Project*) è una rete decentralizzata basata completamente sulla tecnologia peer-to-peer. Allo stato attuale è ancora un progetto in beta ma continuamente aggiornato, con rilasci di ogni 6/8 settimane; inoltre gli sviluppatori ritengono che la presenza di bug è talmente remota tanto da considerarla una rete stabile.

I2P è disponibile preinstallato in molte distribuzioni GNU/Linux tra cui iPrediaOS, Liberté Linux, Whonix o il più popolare e famoso Tails. È interessante vedere come, a differenza di TOR, I2P non costringe l'utente ad utilizzare il protocollo HTTPS. Il motivo è causato dal fatto che I2P cripta già da sé la connessione prima ancora che arrivi all'HTTP

4.2.1 Utilizzo di I2P

I2P è scritto in linguaggio Java, quindi è necessario aver installato il Java Runtime Environment[1] disponibile per i Sistemi Operativi più popolari. Una volta installato è necessario avviare il software che si occuperà di eseguire tutto il processo per collegarti alla rete peer-to-peer. Nel 99% dei casi vedrai aperto solo un terminale o comunque non succederà niente.

4.2.1.1 INSTALLARE I2P

Come per TOR Browser, anche su **Windows** e **macOS** l'installazione di I2P è facilmente attuabile grazie ai binari precompilati. Su **GNU/Linux** (nel nostro caso Debian 8 "Jessie") sarà necessario invece aggiungere i repository ufficiali dal sito i2p2.de . Per farlo modifichiamo il file sources.list:

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

incolliamo quindi nel file i sources che ci vengono forniti (se stiamo usando una versione diversa di Debian incolleremo la parte con il codename appropriato, come in questo caso "Jessie"):

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

Siamo ora pronti per aggiornare i repository.

```
$ apt-get update
```

Qualcosa tuttavia andrà storto. Questo perché, come hai visto dai repository in alto, abbiamo usato il protocollo https, non presente di default in apt. Prima di

[1] https://www.java.com/it/download/

aggiornare il nostro parco software dobbiamo quindi installare il pacchetto apt-transport-https (come consigliato dal terminale stesso):

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

Ora possiamo rilanciare l'aggiornamento:

```
$ apt-get update
```

Anche qui un altro problema: ci vogliono i certificati! Scarichiamoli con il comando:

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

Ci verrà chiesto di voler essere sicuri della scelta in due occasioni, digitiamo per entrambi il *tasto S* e confermiamo con *Invio*. Infine, installiamo i2p:

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

Come per TOR è consigliato non utilizzare I2P da root. Eseguiamo allora il logout da root:

```
$ exit
```

quindi avviamo il servizio con il comando:

```
$ i2prouter start
```

4.2.1.2 IL PRIMO AVVIO DI I2P

In realtà il servizio di I2P è già perfettamente funzionante; per assicurartene visita la **Console Router I2P** (Figura 20) all'indirizzo http://127.0.0.1:7657. Se si apre una schermata come quella seguente vuol dire che il servizio I2P è già funzionante, o perlomeno, il daemon è riuscito a creare un web server in locale che ci permette di usare il Console Router I2P.

Questa console ci permette di configurare e di tener sotto d'occhio lo stato del network. Il corretto funzionamento di I2P prevede che la connessione a Internet sia "relativamente" libera, vale a dire che non devono esserci regole in entrata e in uscita troppo restrittive. Nel caso di un firewall generico presente nei router questo non dovrebbe essere un problema; discorso diverso invece per utenti "NATtati" (gli utenti Fastweb ad esempio sanno di cosa sto parlando). RTFM a parte, I2P ha bisogno di un po' di tempo - giusto un paio di minuti - affinché il pairing con la rete p2p sia abbastanza stabile.

Figura 20: Console di gestione del Network I2P

4.2.1.3 CONFIGURAZIONE DEL BROWSER CON I2P

Una volta avviato il servizio di I2P è possibile configurare il nostro browser preferito affinché riesca a connettersi ai servizi. Per farlo puntiamo il nostro browser ai seguenti indirizzi:

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

Se non sai come modificare i proxy del tuo browser fai riferimento al capitolo precedente che riguarda appunto i Proxy Server. Comunque giusto per non sbagliare in Figura 21 si veda come viene configurato *Firefox.* Dovrebbe essere alla portata di tutti, credo.

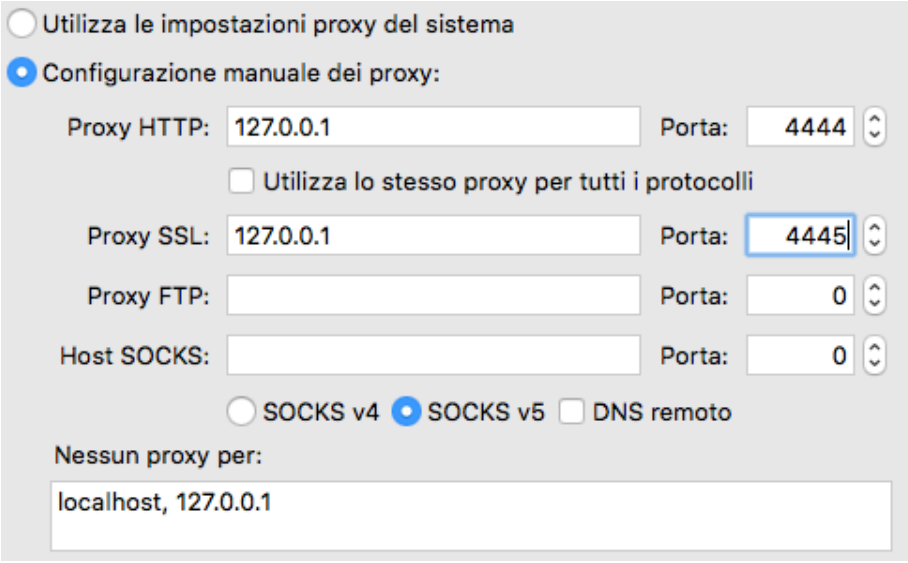


Figura 21: Configurazione di Firefox con I2P

4.2.1.4 RISORSE UTILI DI I2P

A questo punto la domanda che tutti si fanno è: e adesso che faccio?

Come abbiamo già detto I2P non permette quasi mai di connettersi alla rete esterna ma è un circuito "chiuso" o limitato solo ad alcuni servizi.

Eepsites

Gli eepsites sono dei siti particolari con estensione .i2p . Sono accessibili solo ed esclusivamente usando il circuito I2P (esattamente come i nodi .onion della rete Tor) e fanno parte del mondo legato alla darknet. Di questo parleremo più in là nel capitolo che riguarda il "Deep Web". Da notare come ogni utente presente nel network I2P può avere da subito il proprio sito .i2p gratuitamente e con poca fatica. Per farlo basta seguire l'indirizzo http://127.0.0.1:7658 dove viene spiegato come modificare il sito e richiedere il dominio ufficiale .i2p .

Irc2p

Irc2p è il nome del tunnel che viene creato nel momento in cui si starta la prima volta I2P. È possibile configurare qualunque *client IRC* e collegarsi al seguente

indirizzo, quindi usarlo come un vero e proprio server. Puoi connetterti al server IRC di I2P collegandoti all'indirizzo 127.0.01:6668.

Blogging

All'interno del circuito I2P ci sono due eppsites che offrono servizi di *microblogging:* http://id3nt.i2p e http://jisko.i2p . Ultimamente però non sono molto reattivi e tendono a caricarsi dopo diversi minuti.

Torrent

Poteva mancare il client ufficiale per il download via *torrent*? Ma certo che no!
Il nome del progetto è I2PSnark ed è raggiungibile da subito all'indirizzo http://localhost:7657/i2psnark/ . Considera che il torrenting via I2P può essere sicuro quanto maledettamente lento: il problema è causato non solo dal processo di cifratura ma anche dal fatto che non ci sono molti peer in grado di supportare l'I2P. Per questo motivo oltre al client Torrent è necessario affiancare un tracker dedicato. Di seguito ne troverai qualcuno interessante:

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

eDonkey

Forse è il progetto I2P più famoso, nelle sue ultime versioni è stato distribuito assieme al core di I2P rendendolo di fatti un tool stand-alone. L'indirizzo eepsite di riferimento è http://echelon.i2p/imule . Esiste anche un progetto parallelo chiamato Nachblitz[1] che funge da client alternativo a iMule (ma perfettamente compatibile) e basato sulle librerie .NET Framework, disponibile per Windows e Linux con emulazione tramite Wine.

[1] echelon.i2p/nachtblitz/

Mail

Susimail[1] è un particolare servizio I2P e funge da interfaccia mail di due servizi di posta (POP3 e SMTP) che possono essere anche utilizzati con un client di post in locale. Gli account email vengono gestiti da un router ufficiale (hq.postman.i2p) che offre un servizio di mailing sia per la darknet che per la clearnet. Questo servizio ci permette di avere un indirizzo di posta all'interno della rete I2P - l'indirizzo sarà username@mail.i2p - e di inviare e ricevere anche su Internet - l'indirizzo sarà username@i2pmail.org

> NB: il servizio potrebbe essere controllato dai gestori del servizio che comunque offrono il servizio in maniera gratuita e consigliano di cancellare mail non più necessarie lasciando libero spazio anche agli altri utenti. In ogni caso esiste una politica di report abuse e il vostro account potrebbe venire eliminato, per cui ti consiglio di non abusare del servizio.

Scopri di più

Questa è solo una lista approssimativa di tutti i servizi presenti nel network I2P (e ci mancherebbe!). I progetti più importanti sono presenti all'indirizzo http://echelon.i2p/ e racchiudono la storyline di sviluppo di ogni singolo progetto.

4.2.1.5 NAVIGAZIONE ANONIMA IN CLEARNET

Ok, prima che tu pensi di essere pazzo ti rassicuro: avevo detto che I2P è un network assolutamente chiuso a se stesso… ebbene, non è proprio così. Nel corso degli anni l'affezionata community di I2P ha pensato a un modo di usare la propria tecnologia per anonimizzare la navigazione nella clearnet; il metodo più utilizzato è quello di usare gli outproxy, ovvero dei router - come gli exit node di TOR - in grado anche di effettuare il collegamento all'Internet "normale".

[1] localhost:7657/susimail/susimail

Se vuoi provare esiste un modo per navigare su Internet con I2P, basta seguire le FAQ del sito ufficiale[1]. Tuttavia il loro funzionamento non è garantito (allo stato attuale esiste un solo outproxy tra l'altro molto instabile) perciò se intendi navigare in Clearnet forse I2P non è il network che fa per te.

4.2.1.6 DOVE TROVARE I SITI I2P

A differenza della rete TOR non abbiamo una vera e proprio Hidden Wiki ma sono presenti diversi motori di ricerca sparsi qua e la sia nella Dark Net che nella Clearnet. Il consiglio che posso darti è di utilizzare la clearnet per cercare le "i2p lists" nei motori di ricerca, quindi da lì scavare negli eepsites attivi (tenendo conto che molti link potrebbero morire da un momento all'altro). Inoltre è possibile avere una lista degli eepsites attivi - che sono stati registrati - nella rete I2P consultando la lista disponibile a http://identiguy.i2p

4.2.1.7 LE DIFFICOLTÀ DI I2P

Da poco più di un anno la community di I2P sembra essersi un po' allontanata dal progetto. Anche uno dei capi fondatori di I2P ha lasciato i due progetti Syndie e I2P e con sé anche molti altri sviluppatori e volontari. Il problema maggiore è stato non solo perdere un programmatore ma anche il dominio ufficiale i2p.net che ha fatto presagire una brutta fine per questo progetto, tuttavia il nuovo team ufficiale su geti2p.net ci tiene a dire che I2P è ancora attivo e questo è avvalorato al releasing delle nuove versioni di I2P[2]. Il progetto sembra ora in leggera ripresa ma molti degli eepsites - in particolare i progetti proposti in home - sono stati abbandonati; anche il canale ufficiale di IRC sembra essere un cimitero, non che si possa pretendere troppo ma è comunque una considerazione da fare nel caso in cui si decida di dedicarsi per tanto tempo a questo network. Un altro fattore

[1] https://geti2p.net/it/faq#outproxy

[2] 127.0.0.1:7657/news - previo collegamento a I2P

rilevante può essere la base sui cui nasce I2P; il linguaggio con cui è scritto “Java” sembra essere una succhiasangue per la CPU. Per avere un software prestante e fresco il gruppo è al lavoro su un recoding del kernel di I2P in C++ con un progetto che si chiama I2PD[1].

4.3 Freenet

La rete Freenet, così come I2P, è una tecnologia basata sul peer-to-peer che usa le risorse dei suoi utenti - banda, spazio e potenza di calcolo - per creare uno strumento di comunicazione alternativo con standard di sicurezza elevati. È stata ideata 15 anni fa e ancora oggi ha molto da raccontare, basti pensare che duckduckgo.com (noto motore di ricerca famoso per essere un’alternativa anonima a Google) ha donato a Maggio 2016 ben 25,000$ al progetto. Freenet non è un sistema di proxing pensato per la navigazione clearnet, esattamente come I2P, dunque la sua spiegazione si limiterà alla navigazione del suo Deep Web.

Il progetto Freenet nasce come uno strumento libero ed esente da qualunque tipo di censura; il primo prototipo fu progettato da Ian Clarke nel 1999 e fu pubblicato per la prima volta nel marzo del 2000. Tra i tre - cui contiamo i già citati TOR e I2P - è il progetto più anziano. Il sistema è basato interamente sulla tecnologia peer-to-peer, dunque nessuno può avere il controllo di ciò che viene pubblicato né si può, in via teorica, risalire all’autore di un contenuto.

Il network P2P è stato ideato per mettere in connessione gli utenti con i propri amici (nel caso in cui volesse avere un tipo di network più sicuro) oppure geolocalizzato con gli utenti della rete (così da avere un network più veloce e stabile). Gli utenti (chiamati nodi) sono collegati tra di loro ma nessuno sa se il

[1] i2pwiki.i2p/index.php?title=I2Pd - previo collegamento a I2P

destinatario del messaggio che sta inviando è il mittente oppure un nodo intermedio.

4.3.1 Installazione di Freenet

Freenet è stato concepito tramite il linguaggio Java, quindi se lo si vuole utilizzare è necessario procurarsi la Java Environment Runtime[1]. Da **GNU/Linux** (Debian 8) il comando per installarlo è semplicemente:

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

Se state seguendo il corso dall'inizio probabilmente lo avrete già installato poiché I2P viene già distribuito con l'installatore del JRE. Possiamo ora seguire la guida ufficiale di Freenet da cui prendiamo spunto sui passaggi consigliati da fare. Scarichiamo allora lo script auto-installante, quindi lo rinominiamo.

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

Procediamo ora al suo avvio con il comando. Nel caso in cui tu sia utente root, effettua il logout ad utente normale con il comando exit:

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

Seguirà ora un processo d'installazione guidato; procedi finché non riceverai un messaggio dell'avvenuto setup, quindi attendi il caricamento del tuo browser preferito.

[1] https://www.java.com/it/download/

4.3.2 Configurazione di Freenet

Il primo avvio di Freenet prevede una breve configurazione del sistema. Qui possiamo scegliere due tipi di pre-configurazioni esistenti:

* *Basso livello di sicurezza*: a questo livello di sicurezza verremo collegati ad utenti casuali presenti in Freenet. Quest'opzione ha il vantaggio di essere la più veloce tuttavia altri utenti potrebbero essere in grado di monitorare il traffico dati e ricondurle alla tua persona.

* *Alto livello di sicurezza*: con questa configurazione è possibile connettersi solo grazie alla presenza di amici già presenti in Freenet. Quest'opzione, sebbene più sicura, necessita di molti utenti già presenti in questo circuito.

A queste due opzioni se ne aggiunge un'altra che prevede di effettuare una configurazione personalizzata per il proprio tipo di attività. Queste opzioni possono essere modificate in seguito tramite il pannello di configurazione di Freenet. Dopo aver risposto alle varie domande del wizard verremo riportati alla dashboard iniziale da cui è possibile tenere sotto controllo la rete e altre funzioni.

4.3.3 Utilizzo di Freenet

A differenza di TOR o di I2P il sistema si installa come proxy interno tramite indirizzo http://localhost:8888 e non è necessario riconfigurare il browser di navigazione in alcun modo. Questo consente quindi di mantenere il client sempre attivo - nel caso in cui si voglia anche contribuire a rendere la rete viva - e di accedervi solo quando necessario senza l'uso di browser particolari. Raggiunta la dashboard iniziale all'indirizzo http://localhost:8888 (o http://127.0.0.1:8888 come preferisci) verrai riportato verso una lista iniziale di link a cui si può accedere, esattamente come I2P.

È possibile che aprendo un link questo non si carichi subito ma mostri una screen (Figura 22) che indica un timeout. Questo valore ci indica il tempo rimanente affinché la pagina possa essere risolta nel nostro browser:

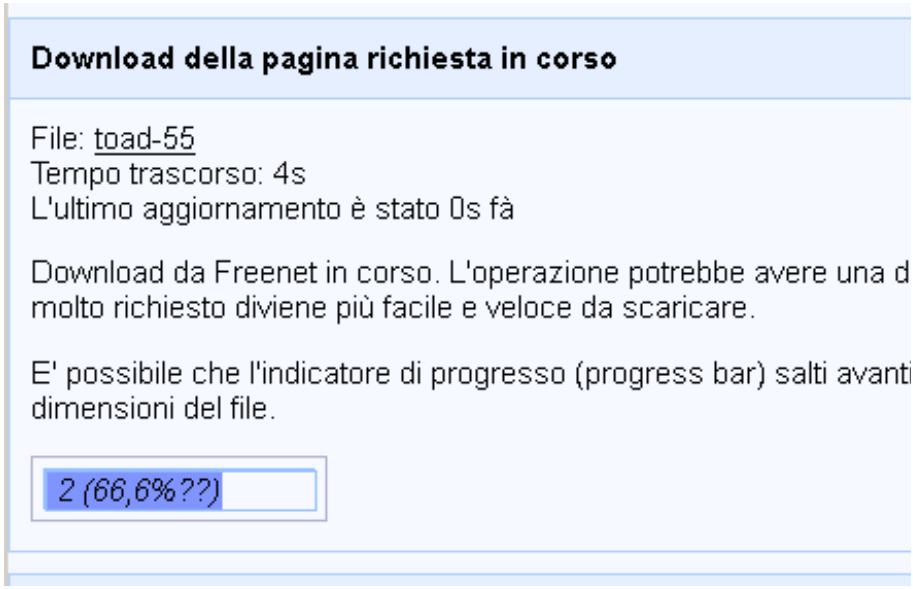


Figura 22: Caricamento di una pagina su Freenet

4.3.4 Risorse utili di Freenet

L'ecosistema di Freenet vive grazie alla sua community che costantemente genera nuovo materiale. Molto di questo è di tipo propagandistico, politico o finalizzato alla denuncia di abusi da parte di istituzioni e governi, tuttavia non è da escludere la presenza di di markets, pornografia e materiale visivamente shockante.

Freesites

I freesites sono l'essenza stessa di Freenet. Essi vengono creati dagli utenti quindi vengono caricati direttamente dalla dashboard del client di Freenet.

Il procedimento per crearne uno è spiegato nella pagina http://localhost:8888/insertsite/ ; esistono anche strumenti come Sharesite[1] e FlogHelper[2] per facilitare la creazione di questi, mentre la lista di quelli presenti è divisa in tre fasce[3]:

- **Enzo's Index**, che contiene tutti i siti organizzati per lingua, categoria etc…
- **The Filtered Index**, che contiene siti ad eccezione di quelli considerabili disturbanti
- **Nerdageddon**, che contiene perlopiù documenti opensource e siti informativi

A questi ufficiali mi sento libero di indicarne altri comunque molto forniti:

- **Linkageddon**: è organizzato come Nerdageddon ma non filtra alcun tipo di sito
- **The Ultimate FreeNet Index**: altra index stracolma di freesites, possibilità di scegliere la categoria di appartenenza
- **TPI: The Public Index**: directory di siti web gestita autonomamente dalla community. Le istruzioni per aggiungere un freesites sono presenti in fondo alla pagina.
- **AFKindex**: directory aggiornata tramite il crawling di Freenet. Sono filtrati i siti pornografici ed erotici.

> Una volta caricati, i freesites rimangono all'interno del network e condiviso dai peer finchè vengono regolarmente visualizzati. Se vengono ignorati per troppo tempo si cancellano automaticamente.

[1] localhost:8888/plugins/ - previo utilizzo di Freenet

[2] localhost:8888/plugins/ - previo utilizzo di Freenet

[3] Tutte le directory mostrate sono presenti all'interno di Enzo's Index

Social Networking

Freenet è dotato internamente di una suite di programmi pensati per consentire la comunicazione fra più utenti, cosa che non può essere possibile con i siti statici dei freesites che invece permettono solo una comunicazione *una a molti*. Per conoscere questi strumenti visita la pagina dedicata alla *Comunicazione*[1].

Mail

Nel network Freenet lo strumento ufficiale per comunicare con gli altri utenti è appunto Freemail. Questo strumento viene preinstallato all'interno del pacchetto di Freenet ma non è abilitato di default. Per attivarlo è necessario dirigersi alla pagina dei plugins[2] selezionare Freemail e cliccare sul bottone "Carica".

Questo strumento permette di avere una comunicazione solo con gli utenti di Freenet, o meglio del *Web of Trust*. Il WoT è un plugin aggiuntivo (che deve essere attivato esattamente come Freemail) e che permette di avere un'identità riconosciuta all'interno della rete stessa.

Una volta abilitato è necessario configurare un primo alias prima di poter utilizzare Freemail, quindi ci sarà possibile creare anche più alias contemporaneamente. Ciò significa in parole povere che Freemail può essere utilizzato solo dagli utenti registrati in Web of Trust e che non può essere utilizzato all'esterno della rete Freenet.

[1] localhost:8888/chat/ - previo utilizzo di Freenet

[2] localhost:8888/plugins/ - previo utilizzo di Freenet

4.3.5 La sicurezza in Freenet

Partiamo subito dal presupposto che Freenet è un network molto sicuro purché venga utilizzato nel modo giusto. Considerate che la vostra privacy sarà direttamente proporzionale al numero di "amici" a cui vi sarete collegati in Freenet: nel caso in cui non ne abbiate nessuno verrete collegati a degli sconosciuti. Per questo Freenet può essere configurato a diversi livelli di sicurezza: è possibile modificarlo dal menù in dashboard sotto la voce *Configurazione -> Livello di Sicurezza*. Più il livello sarà alto, maggiore sarà la sicurezza in rete (a discapito della velocità).

Freenet prevede inoltre una seconda opzione che riguarda le situazioni in cui il tuo computer venga confiscato: sempre dalla pagina Livello di sicurezza è possibile impostare uno dei *quattro livelli di cifratura*, dal più blando (non viene crittato nulla) a quello più paranoico (ogni volta che si riavvia tutto ciò che riguarda Freenet si elimina). Personalmente mi sentirei di consigliare il livello Medio-Alto nei sistemi Linux Live (tratteremo poi questi sistemi più avanti) mentre il livello Paranoia solo in Virtual Machine o Computer in cui si opera costantemente e stabilmente.

5. COMBO NETWORK

Eccoci qui, pronti ad affrontare uno dei grandi dibattiti sull'anonimato: TOR e VPN, come usarli insieme? O meglio ancora, *TOR tramite VPN* o *VPN tramite TOR* ?

> Per convenzione parleremo di TOR come network "anonimizzante". Trascureremo gli altri in quanto hanno di natura uno scopo finale che non comprende la Clearnet.

Prima di rispondere a queste domande è opportuno ricordarci alcuni punti chiave: le VPN, a prescindere dall'uso di TOR, andrebbero sempre utilizzate non tanto per anonimizzarsi in clearnet quanto per **proteggersi durante l'uso di hotspot non sicuri** (come aeroporti, hotel, bar etc…) o se si ha timore che il proprio router casalingo possa essere monitorato. Tutti i network, soprattutto quelli pubblici, possono essere esposti al monitoraggio del traffico. Questo dovrebbe bastare come motivo per avere sempre una VPN accesa, a prescindere da quanto ci si senta sicuri o meno dell'hotspot a cui ci si collega.

I nostri dati di navigazione, o perlomeno quelli che vogliamo restino sicuri nelle nostre mani (email, password, carte di credito e tutto il resto) dovranno passare per una VPN sicura così da evitare attacchi nella rete locale. Ricordati anche quello che abbiamo detto a proposito dei no-logs sulle VPN.

È vero che con le giuste precauzioni è possibile fare lo stesso con TOR ma è anche vero che usando la sola rete TOR c'è una buona possibilità di diventare dei sospetti. Di cosa, vi chiederete? Che tu stia compiendo azioni illegali o meno l'NSA/FBI/GCC e tutti gli organi di spionaggio governativo (e non) vogliono sapere se stai usando TOR o anche solo Linux. Una di queste blacklist viene chiamata in rete la NSA watch list[1] ma non è da escludere che anche altri si occupino di questo. Come fanno a sapere se stai usando TOR? Basta chiedere all'ISP. Quindi è

[1] https://duckduckgo.com/?q=nsa+watch+list&t=ha&ia=web

necessario non solo nascondere le nostre attività ai siti e servizi online ma anche non lasciar tracce visibili nell'ISP.

5.1 TOR tramite VPN

In questa configurazione parleremo del seguente schema:

```
Computer -> VPN -> TOR -> Internet
```

Partendo dal presupposto che il tuo dispositivo si connetta a Internet, utilizzando una VPN cifreremo a monte tutto ciò che entra ed esce dalla rete, quindi dentro questo tunneling andremo a connetterci al routering di TOR, nascondendo all'ISP l'accesso a quest'ultima rete. Prima ho citato i *no-logs* e per questo c'è un motivo: usando una VPN noi nascondiamo le nostre attività all'ISP ma se la VPN non è seria riguardo le privacy policy potrebbero essere loro a memorizzare le nostre attività, rendendo vani tutti i nostri accorgimenti.

Connettendoci a TOR dopo aver effettuato il tunneling con una VPN, il provider VPN può sapere che tu stai utilizzando la rete TOR, esattamente come è in grado di farlo anche un ISP. Quest'ultimo potrà sapere che ti stai collegando a una VPN ma non alla rete TOR. Il provider VPN può quindi memorizzare anche tutte le attività che non vengono cifrate. Ci sono vari motivi che potrebbero decidere di farci scegliere una configurazione TOR su VPN, eccone alcuni:

Pro

- Nascondi la tua attività di TOR al tuo ISP, limitando l'effetto *NSA watch list*
- Potrai accedere al deep web (indirizzi .onion e simili)
- L'entry node di TOR non vedrà il tuo IP ma quello della VPN

Contro

- Dirai al tuo VPN provider che usi TOR

- Se l'exit node è compromesso si espone il tuo VPN provider (ma non il tuo reale IP), in questo caso affidati a una VPN veramente sicura e nologs

- Alcuni exit node di TOR rifiutano i collegamenti da VPN.

Prima di procedere

Alcune VPN offrono servizi di "TOR over VPN" semplicemente effettuando una configurazione in OpenVPN. Se il tuo provider VPN te lo consente probabilmente avrà una sezione dedicata nel suo sito che spiega come fare.

5.1.1 Come effettuare TOR tramite VPN

Una soluzione a livello di Sistema Operativo è quella di utilizzare la Workstation di Whonix collegandola alla VPN scelta, mentre il Gateway sarà già collegato alla rete TOR. Parleremo di Whonix molto più in là, quindi se vuoi puoi farci un salto e tornare quando vuoi.

Sempre in campo della virtualizzazione, è possibile effettuare questa procedura con un computer Host interamente collegato alla rete TOR (in questo caso qualunque Sistema Operativo andrà bene) e nella macchina virtualizzata si effettuerà il collegamento alla rete VPN.

Un'interessante alternativa consiste nel far uso di un router hardware come P.O.R.T.A.L.[1] che, in maniera simile al collegamento che abbiamo visto con la VPN, permette di collegare un router direttamente alla rete TOR. Lascio ai più intrepidi il compito di cimentarsi in questo mondo.

[1] https://github.com/grugq/portal

5.2 VPN tramite TOR

In questa configurazione parleremo del seguente schema:

```
Computer -> TOR -> VPN -> Internet
```

L'utente si collega alla rete TOR creando la sua rete protetta. Da lì si collegherà alla VPN nascondendo l'IP dell'exit-node di TOR, facendolo risultare solo un utente VPN. Effettuando un tunneling a una rete VPN dopo essersi collegati a TOR significa far sapere al tuo ISP che tu stai utilizzando la rete TOR. Quest'ultima crea un network a sé, non si interessa di effettuare il tunneling diretto alla VPN che invece verrà effettuata a parte, dicendo così all'ISP che non solo ci stiamo collegando a TOR ma anche ad una VPN.

Pro

- Nascondi la tua identità al provider VPN
- Puoi navigare i siti web che bloccano gli exit-node di TOR
- Se la tua VPN cade rimani comunque protetto da TOR, in ogni caso preferisci sempre una VPN con funzione di Kill Switch

Contro

- Dirai al tuo ISP che stai usando sia VPN che TOR
- Non potrai accedere al deep web (indirizzi .onion e simili)
- Accendi i riflettori dell'NSA watch list su di te

5.2.1 Come effettuare VPN tramite TOR

È possibile effettuare questo tipo di connessione in maniera semi-permanente utilizzando un router che supporti OpenWRT o dd-wrt : questi sono dei firmware che supportano il collegamento alle VPN. Troverai maggiori informazioni e la lista dei router compatibili con questi firmware nei rispettivi siti ufficiali[1][2]. Una volta collegato al router potrai navigare tranquillamente utilizzando il tuo client Tor preferito.

Anche in questo caso ci sono strade alternative: si potrebbero voler utilizzare distribuzioni pre-configurate (o configurarle da se) che consentono l'uso unicamente della rete TOR per effettuare connessioni esterne: la procedura corretta per effettuarlo ad esempio in Tails è descritta nella pagina ufficiale[3] .

[1] https://openwrt.org

[2] https://dd-wrt.com/site/

[3] https://tails.boum.org/blueprint/vpn_support/

5.3 TOR su TOR

Nel capitolo riguardante la rete TOR abbiamo detto che esiste una remota possibilità che l'exit node, ovvero l'ultimo "strato" della rete che va in clearnet, potrebbe monitorare le nostre azioni. Sebbene possa risultare in parte paranoico correrò il rischio di presentare un metodo che permette di effettuare un tunneling TOR all'interno di un altra rete TOR.

Questa operazione, sebbene non risolve a monte il problema del monitoring dell'exit node, permette di mischiare le carte all'interno del routing diminuendo drasticamente le possibilità di risalire all'origine della richiesta partendo dal routing stesso, sebbene questa realtà risulti alquanto remota.

5.3.1 Tortilla

Tortilla è un programma che si occupa di reindirizzare tutte le richieste TCP e DNS all'interno del nodo TOR. Il tool viene distribuito con binari preconfigurati indipendenti a quelli ufficiali; questo permette di non aver conflitti di alcun genere. Usato assieme al Tor Browser o a una versione stand alone di TOR è possibile effettuare il doppio-tunneling esattamente come abbiamo visto tra VPN e TOR. L'unico "limite" è il fatto di esistere *solo su Windows*.

Questo comunque non sarà un problema in quanto potremo usare Windows come computer Host e utilizzare una Virtual Machine per le nostre operazioni (ne parleremo tra qualche capitolo). Tortilla è disponibile in versione opensource e distribuito nei canali ufficiali di Github[1] e precompilato nel sito ufficiale[2].

Il suo utilizzo è molto semplice: per prima cosa devo procurarti il TOR Expert Bundle dal sito ufficiale, quindi installalo sul tuo Sistema Operativo (ancora

[1] https://github.com/CrowdStrike/Tortilla

[2] www.crowdstrike.com/community-tools/

meglio se su chiavetta USB). Aprendo l'eseguibile tor.exe si aprirà il prompt dei comandi; quando dai log uscirà la voce *Bootstrapped 100% Done* significa che la connessione al circuito TOR è completata, quindi saremo pronti a lanciare il client tortilla.exe.Ricordati di eseguire entrambi i programmi con privilegi di amministratore; per alcune versioni di Windows potrebbe essere anche necessario abilitare i certificati non autorizzati, fai riferimento alla guida ufficiale di Microsoft[1].

Tutti i metodi d'applicazione di un circuito VPN su TOR sono validi anche per questo tipo di configurazione.

5.3.2 TOR su TOR è utile?

Personalmente ritengo che il collegare due reti TOR a scalata non garantisca alcun beneficio in termini di privacy che può già garantire la scelta di una buona VPN. L'uso di questa configurazione è da ritenersi puramente sperimentale e non in linea con gli standard qualitativi d'anonimato che si possono avere con una combo network di altri tipi, pertanto se si voglia mettere in pratica lo si consiglia solo per ambienti di test finalizzati allo studio.

[1] https://msdn.microsoft.com/en-us/library/windows/hardware/ff540213(v=vs.85).aspx

6. RISORSE LOCALI

In questa parte del documento tratteremo delle Risorse Locali, ovvero quell'insieme di *software* e **oggetti virtuali** che sono presenti all'interno del computer e possono in qualche modo compromettere l'anonimato di un *browser web*, di un *client* e via dicendo. Navigando in rete con il browser siamo in grado di lasciare migliaia di informazioni a nostra insaputa; quello che andremo a fare è un resoconto generale di ogni singola risorsa in grado di esporre l'utente finale allo smascheramento digitale.

6.1 Navigazione in Incognito

La modalità in **incognito** è una funzione speciale integrata in qualunque *browser* di ultima generazione che consente di disabilitare temporaneamente tutte quelle informazioni che possono compromettere l'anonimato dell'utente come *cookies*, *cronologia*, *file temporanei*, *sessioni* e *password salvate*.

6.1.1 Come passare alla modalità in Incognito

Ogni browser permette di attivare la *modalità in Incognito* eseguendo la corretta combinazione di tasti. Per comodità ti indicherò quali sono le *scorciatoie* da tastiera per i più popolari browser online:

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

Tipo di Browser	Shortcut (CTRL per Win/Linux, CMD per macOS/OSX)
Safari	CMD + SHIFT + N
Internet Explorer	CTRL + SHIFT + P
Microsoft Edge	CTRL + SHIFT + P

6.1.2 Cosa fa (e non fa) la modalità in Incognito

La modalità in Incognito può essere davvero molto comoda per effettuare diverse operazioni senza ogni voltar impazzire a configurare menù e ripulire risorse. Per intenderci, la modalità in Incognito lavora su queste risorse:

- **Cookie**: non usa cookie creati in precedenza e quelli creati in modalità in incognito vengono cancellati alla chiusura della modalità stessa
- **Cronologia**: i siti web visitati non vengono salvati nello storico di navigazione
- **Cache**: non vengono usati file memorizzati sul PC né vengono salvati file per velocizzare il caricamento delle pagine
- **Estensioni**: non vengono caricate le estensioni/addon installate nel browser di default (a meno che non le si attivi in automatico)

La comodità risiede nel non dover ogni volta cancellare cronologia, svuotare cache o cookies e disabilitare estensioni che potrebbero risultare pericolose.

Bisogna intendere la **modalità in Incognito** come ad una funzione che consente di nascondere alcune attività in locale, ricordandoti che *non è in grado di proteggere i tuoi dati all'esterno*.

6.2 HTTPS

Del protocollo HTTPS ne abbiamo già parlato quindi faremo giusto un ripassino.
Con HTTP si intende un protocollo creato appositamente per permettere la comunicazione nel *Word Wide Web* di informazioni tra server e client; con **HTTPS** si intende un protocollo *HTTP* che utilizza una connessione cifrata tramite *TLS* o il più vecchio *SSL*. Connettersi a un sito web tramite HTTPS significa mettere al sicuro i dati che vengono fatte transitare nella rete, allontanando possibili azioni di spionaggio tramite *attacchi man-in-the-middle*. Al momento quindi dobbiamo solo sapere che l'HTTPS consente di avere più sicurezza e che non utilizzarlo è un fattore di rischio.

6.2.1 Controllo sui protocolli HTTPS

Nel panorama del web il software per eccellenza che è sinonimo di prevenzione in questo campo è HTTPS Everywhere[1], sviluppato in collaborazione dal *The Tor Project* e la *Electronic Frontier Foundation*. Il tool è disponibile per i più popolari browser web e viene distribuito anche all'interno del Tor Browser.

6.3 Cookies

In informatica un **cookie** è un file di testo memorizzato all'interno di un computer che viene utilizzato da un browser web per tener traccia di informazioni come login, pagine visitate, preferenze dell'utente come grafica o lingua e via dicendo. I cookies vengono creati da un server tramite protocollo HTTP e possono essere letti o scritti solo dal rispettivo dominio. Un cookie è *composto* da:

- **Nome**: ovvero un identificatore che lo riconosce. Questo valore è obbligatorio.

[1] https://www.eff.org/it/https-everywhere

- **Valore**: ovvero il contenuto che è presente nel cookie. Questo valore è obbligatorio (ma può essere vuoto e in questo caso il suo valore sarà vuoto).

- **Scadenza**: ovvero la durata che il cookie avrà all'interno del browser. Questo valore è opzionale.

- **Sicurezza**: ovvero se il cookie deve essere trasmesso o meno solo con il protocollo HTTPS.

- **HttpOnly**: ovvero se il cookie può essere trasmesso solo tramite protocollo HTTP o se può essere manipolato anche dai linguaggi client-side come il Javascript

6.3.1 Impatto dei Cookie sulla sicurezza

I cookies vengono classificati attraverso livelli in cui possono lavorare e in base alle loro finalità tecniche, tuttavia ai fini di questo capitolo non è importante conoscerli. Quello che adesso devi sapere è che un cookie, tra le tante cose che può fare, ha anche la facoltà di memorizzare **dati statistici** e questi possono essere creati anche da *terzi siti* (i cosiddetti cookie di terza parte). Inoltre, i cookies possono lasciar traccia dei siti che abbiamo visitato in quanto praticamente tutti i siti web lasciano almeno un cookie all'interno del browser (che esso sia di preferenza, di login o di qualunque altra cosa).

Un articolo pubblicato nel 2013 dal Washington Post[1] spiega come l'NSA monitora segretamente gli utenti di Internet tramite i cookies di terze parti, come ad esempio quelli che si memorizzano quando si visita un sito che utilizza Google Analytics. Se il concetto non è ancora chiaro pensa a questo: hai presente quando alcuni annunci sembrano seguirti o addirittura replicarsi su altri siti? È tutta colpa dei cookie (tecnicamente chiamati *cookie di profilazione*) che si occupano di

[1] https://www.washingtonpost.com/news/the-switch/wp/2013/12/10/nsa-uses-google-cookies-to-pinpoint-targets-for-hacking/

memorizzare i tuoi interessi e di mostrarti campagne pubblicitarie adeguate alla tua persona.

6.3.2 Controllo sui cookie

Come già spiegato i cookies vengono utilizzati per garantire il funzionamento di certe situazioni nei siti web. Bloccarli completamente tramite le funzioni via browser *è sconsigliato* in quanto causerebbe il malfunzionamento del sito o, nei peggiori dei casi, il blocco da parte del portale. *Quindi cosa fare?*

Si può decidere di utilizzare estensioni denominate *cookie manager* in grado di bloccare per dominio i cookies oppure semplicemente di lasciarli attivi e utilizzare la modalità in Incognito del browser (vedi il capitolo precedente), l'importante è ricordarsi che i cookie di terze parti possono condividere informazioni sui siti web visitati. Questi cookies possono provenire da servizi come analytics, advertisement e CDN esterne che potrebbero tracciarne la navigazione.

Tra le tante estensioni in grado di bloccare i cookie, uno dei migliori è sicuramente Ghostery[1] che permette di bloccare a monte tutti gli *script* in grado di generare *cookie* di terze parti (si trova in nei relativi store dei browsers). Se questo non fosse disponibile per il tuo browser, puoi sempre usare NoScript[2] (vedi la parte relativa al Javascript).

[1] https://www.ghostery.com

[2] https://noscript.net

6.4 Cookies “speciali”

In aggiunta alla lista dei tipi di cookie regolari presenti in rete stanno di volta in volta nascendo anche altri tipi di cookies che possiamo definire proprietari. *Adobe* ad esempio ha creato i “**Local Stored Objects**” (chiamati anche “Flash Cookies”) che sono integrati nel Flash Player; *Mozilla* dalla sua ha inserito nelle ultime versioni di *Firefox* il **DOM storage** che permette un più rapido rendering degli elementi web.

6.4.1 Impatto dei Cookies “speciali” sulla Sicurezza

Vedesi “Impatto dei Cookies sulla Sicurezza”.

6.4.2 Flash Cookies, come bloccarli

Se proprio non puoi fare a meno di Flash - e più in là spiegheremo il perché andrebbe disattivato - è possibile fare in modo di disattivare i Local Shared Objects. Per farlo è necessario dedicare un valore “0” allo spazio che i LSO possono utilizzare. Per maggiori informazioni si può fare riferimento alla guida ufficiale di Adobe[1].

In ogni caso ripeto: è meglio lasciar perdere Flash. Se devi scaricare a tutti i costi un video puoi farlo con qualche addon per il tuo browser o un download manager come JDownloader[2] e simili.

6.4.3 DOM Storage, come bloccarlo

Disattivare il DOM Storage su Firefox è relativamente semplice. Basta digitare sull’indirizzo del browser “about:config”, cercare “*storage*” filtrando i risultati; a

[1] https://helpx.adobe.com/flash-player/kb/disable-local-shared-objects-flash.html

[2] jdownloader.org

questo punto click-destro sulla voce "*dom.storage.enabled*" e effettuando un doppio-click il valore dovrebbe diventare "*false*". È possibile bloccare il DOM Storage di Firefox anche utilizzando FireGloves (ne parleremo nel capitolo riguardante il Fingerprinting del Browser).

6.5 Javascript

Nel mondo del web il linguaggio **Javascript** è un'autorità. Esso è un linguaggio di scripting che viene utilizzato principalmente per intercettare eventi del client, vale a dire quelle azioni che l'utente compie (come il passaggio di un mouse sopra un pulsante, una notifica live, uno scroll e via dicendo), effettuando operazioni che il solo linguaggio HTML non è in grado di eseguire.

È bene ricordare che *Javascript NON è Java*: sono due linguaggi di programmazione completamente diversi, vengono utilizzati e funzionano in maniera del tutto dissimile tra di loro. Senza di esso oggi non avremmo siti web dinamici con notifiche in live e tante funzioni per velocizzare il web o renderlo semplicemente più appetibile. Considera inoltre che ad oggi - stando a una ricerca di W3Techs[1] - il 93.5% dei siti web fa uso di Javascript. Una cosa enorme insomma.

6.5.1 Impatto del Javascript sulla Sicurezza

Nonostante ciò il Javascript è in grado di interagire con le attività dell'utente ed è in grado ad esempio di raccogliere ciò che quest'ultimo scrive in una pagina web, fungendo da vero e proprio *keylogger*. Molte aziende di *analytics/ advertisement* ad esempio usano il Javascript per analizzare le *keywords* dei siti web e vendere ai loro clienti le pagine più visitate o semplicemente interessanti.

[1] https://w3techs.com/technologies/details/cp-javascript/all/all

Con il Javascript è possibile valutare (in parte) se l'utente fa uso di TOR e VPN, permette di visualizzare la lista dei plugin del browser, i font installati, la tua Time Zone (e da lì risalire alla tua nazionalità), il tuo user-agent (anche se viene *spoofato* tramite un controllo incrociato di pseudo-classi con il CSS), lo storico di pagine, alcuni programmi installati (come *OpenOffice, Adobe Reader, Microsoft Silverlight* e altri) e molte altre informazioni.

Dulcis in fundo il Javascript può essere utilizzato come "controller" a seguito di un attacco definito *XSS* (*Cross Site Scripting*) che consente a un malintenzionato di prendere possesso di una pagina web e di automatizzare operazioni lato client (ad es. copiando i cookie e inviarli ad un'altra pagina) o di reindirizzare la pagina a un fake login e impossessarsi dei dati d'accesso.

6.5.2 Controllo sul Javascript

Per ogni browser troveremo la scelta migliore. Come di consueto, ti indicherò solo le estensioni per i browser più popolari:

- **Mozilla Firefox**: per il browser del panda rosso l'estensione più autorevole è sicuramente NoScript[1]. Questa estensione è in grado non solo di bloccare il Javascript ma anche Flash, Java e ogni altra applicazione esterna. NoScript è in grado anche di intercettare e bloccare attacchi di tipo XSS e Clickjacking.
- **Google Chrome**: la controparte Google purtroppo non può fare affidamento all'ottima suite di NoScript, tuttavia è presente una validissima alternativa - a tratti anche più completa - che si fa chiamare uMatrix.
- **Opera Web Browser**: anche qui troviamo l'eccellente uMatrix[2]
- **Safari**: sul browser di macOS/OSX è possibile disabilitare il Javascript direttamente dalle *Preferenze -> Sicurezza -> Abilita Javascript*

[1] https://noscript.net

[2] https://addons.opera.com/en/extensions/details/umatrix/?display=en

- **Microsoft Edge**: è possibile disabilitare il Javascript modificando le Group Policy tramite il percorso *Configurazione Utente -> Template Amministrativi -> Componenti Windows -> Microsoft Edge.*

6.6 Flash

Flash è una tecnologia sviluppata da Macromedia e acquistata da Adobe che ha contribuito nell'ultimo decennio a rendere accessibile il formato media interattivo alla portata di tutti gli internauti. Prima di continuare bisogna dire una cosa: **Flash** è morto. O meglio, sta morendo. Le statistiche dicono che nel 2018 meno dell'1% dei siti web faranno ancora uso del Flash, il browser *Chrome* dal 2017 lo disabiliterà completamente così come *Firefox* un anno dopo. La stessa *Adobe* ha dichiarato ormai l'abbandono della tecnologia a favore dell'*HTML5*, il nuovo standard del web. Certo è che se *Flash* fa la fine di *Windows XP* allora aspettiamoci installazioni di questa applicazione almeno per altri 10 anni!

6.6.1 Impatto del Flash sulla Sicurezza

Il Flash Player è stato oggetto di critiche da molti ricercatori che lo ritengono pericoloso per l'utente, instabile e poco performante. Dall'ultima versione rilasciata a Gennaio sono centinaia (se non migliaia) le vulnerabilità non ancora risolte[1], questo a dimostrazione di come sia una mina vagante se installata all'interno di qualunque computer.

[1] https://www.cvedetails.com/vulnerability-list/vendor_id-53/product_id-6761/Adobe-Flash-Player.html

6.6.2 Controllo sul Flash

L'unico consiglio che posso darti è: *disinstallalo completamente*. Nel caso in cui ne avessi bisogno fai riferimento alla voce "Controllo sul Javascript", tutte le estensioni lì spiegate consentono di bloccare anche il *Flash Player*.

6.7 Java

Il **Java** è un linguaggio di programmazione molto popolare tra i programmatori della rete (sebbene ultimamente abbia perso il suo fascino e nuovamente decollato grazie ad *Android*) ed è stato per anni il fautore di *web applications* di tutto rispetto. Ultimamente però la tecnologia *HTML5* e i browser in generale stanno acquisendo sempre più popolarità relegando il Java Web a quello che è un linguaggio di nicchia. Per carità, è ancora oggi un valido strumento soprattutto se utilizzato per sfruttare appieno l'hardware in commercio, tuttavia è bene ricordare che buona parte dei browser moderni si sta allontanando da questo linguaggio. Il risultato è che presto o tardi Java potrebbe diventare obsoleto nel settore web.

6.7.1 Impatto di Java sulla Sicurezza

Le vecchie versioni di Java sono state oggetto di discussione in quanto non era possibile configurarne un proxy *SOCKS4/5* all'esterno costringendo quindi l'utente a disabilitarlo totalmente. Nelle nuove versioni è stato risolto con una nuova funzione sperando però che il team di sviluppo documenti meglio questa nuova possibilità. Ciò nonostante, si consiglia di disabilitare completamente il client Java in quanto un browser non correttamente configurato potrebbe causare un *DNS leak* (già spiegato ampiamente nel capitolo riguardante le VPN).

6.7.2 Controllo di Java

Il client Java può essere disabilitato facendo uso degli stessi strumenti già descritti nel paragrafo "*Controllo di Javascript*". Qualora tuttavia sia comunque necessario possiamo consigliare l'uso di Orchid[1], un browser sperimentale basato su Tor Browser che offre il pieno supporto alle librerie Java, anche nei dispositivi Android.

6.8 ActiveX

Per ActiveX si intende un'estensione creata da Microsoft per… estendere le funzionalità del browser Internet Explorer. Nonostante sia poco popolare in Europa (a differenza di applicazioni di origine orientale come le IP Camere) esso permette un controllo completo della macchina in cui gira, consentendo operazioni in grado di compromettere l'intero sistema dell'utente.

6.8.1 Impatto di ActiveX sulla Sicurezza

Come si può intendere, ActiveX è uno strumento estremamente pericoloso se messo in mano a dei criminali. Fortunatamente, è anche poco popolare e soprattutto in disuso in quasi tutti i servizi pubblici. Considera però che, a prescindere da anonimato o meno, una ActiveX è in grado di insediarsi nel dispositivo ospite e infettarlo con malware e trojan di ogni genere, rendendo nulla qualunque operazione di anonimizzazione.

6.8.2 Controllo di ActiveX

Non conoscendo la natura di ogni singola applicazione in ActiveX se ne sconsiglia altamente l'esecuzione da fonti non attendibili. Qualora venga eseguita

[1] https://subgraph.com/orchid/index.en.html

un'applicazione che ne esegue una connessione Internet a parte, assicurarsi che l'intero sistema Windows sia configurato al collegamento esterno tramite *Proxy/VPN/Tor*. Qualora possibile, verificare anche la provenienza dei certificati (le firme digitali) di ogni singolo applicativo e la loro integrità. In caso di dubbi, non consentire mai l'esecuzione lato client (opzione possibile solo su *Windows XP SP2* e successive).

6.9 WebRTC

WebRTC è una nuova tecnologia nata nel 2011 che consente con un browser di effettuare videochat tramite i linguaggi *HTML5* e *Javascript*. Tale tecnologia è preinstallata nei browser e OS di ultima generazione[1] e attualmente è possibile utilizzarla nei servizi come *Firefox Hello, Google Hangouts, Skype (in versione web), Facebook Messenger* e via dicendo.

6.9.1 Impatto di WebRTC sulla Sicurezza

Essendo una tecnologia relativamente nuova (ha appena 5 anni!) abbiamo poche *case-history*. A dir la verità ne esiste una sola. Da una ricerca condotta da TorrentFreak[2] si è scoperto che un sito in remoto può sfruttare il protocollo **WebRTC** per rivelare il vero indirizzo IP di un utente, anche se questo è collegato a una rete VPN o TOR. E non solo l'indirizzo pubblico ma anche quello locale!

Sei in paranoia? Ci mancherebbe altro, eppure (si spera) questa vulnerabilità venga sfruttata solo da pochi portali. Ad ogni modo prova a collegarti a una *VPN* e visita l'indirizzo di test[3]. Se viene mostrato il tuo reale indirizzo IP (sia esso locale

[1] https://en.wikipedia.org/wiki/WebRTC#Support

[2] torrentfreak.com/huge-security-flaw-leaks-vpn-users-real-ip-addresses-150130/

[3] https://diafygi.github.io/webrtc-ips/

che remoto) - nonostante la VPN o altri sistemi che camuffano il tuo IP - allora sei vulnerabile. Questa precisa vulnerabilità può essere approfondita nella pagina github dei ricercatori[1], corredato di proof-of-concept e spiegazione tecnica dell'attacco.

6.9.2 Controllo su WebRTC

Ragazzi, siamo sinceri, il WebRTC non è che faccia poi così impazzire! Personalmente consiglierei a tutti di disabilitarlo direttamente da Browser utilizzando estensioni come

- WebRTC Network Limiter[2] per *Chrome*, ScriptSafe[3] per *Opera* e anche *Chrome*
- Disable WebRTC Addon[4] per *Firefox*

Per *Firefox* è possibile anche disattivare la funzione direttamente da browser; per farlo è sufficiente digitare "about:config" nella barra degli indirizzi, ricercare la stringa "*media.peerconnection.enabled*" e con doppio click per impostare il suo valore su *false.*

[1] https://github.com/diafygi/webrtc-ips

[2] https://chrome.google.com/webstore/detail/webrtc-network-limiter/npeicpdbkakmehahjeeohfdhnlpdklia

[3] https://chrome.google.com/webstore/detail/scriptsafe/oiigbmnaadbkfbmpbfijlflahbdbdgdf?hl=en

[4] https://addons.mozilla.org/en-US/firefox/addon/happy-bonobo-disable-webrtc/

6.10 Fingerprinting del Browser

Tutte le tecnologie di cui abbiamo parlato fin'ora sono state analizzate affinché si dimostrasse come ognuna di esse possa diventare un problema per la sicurezza del navigatore. Quello che ancora non abbiamo spiegato è che l'insieme di queste tecnologie formano il cosiddetto fingerprinting del browser.

Con il termine **fingerprinting** (impronta digitale) definiamo quel valore unico che il browser assume dal momento in cui tutte le sue informazioni sommate portano ad un solo unico risultato. Per essere più chiari, immagina di essere in grado di smontare letteralmente il tuo *browser*. Ogni *pezzo* fa parte di un puzzle, e se questo puzzle ha un ordine unico nella sua struttura, ecco che automaticamente assume un'identità unica; se questa identità viene associata alla tua persona, non c'è *proxy/VPN/Tor* che regga. Ma cosa sono questi *pezzi*?

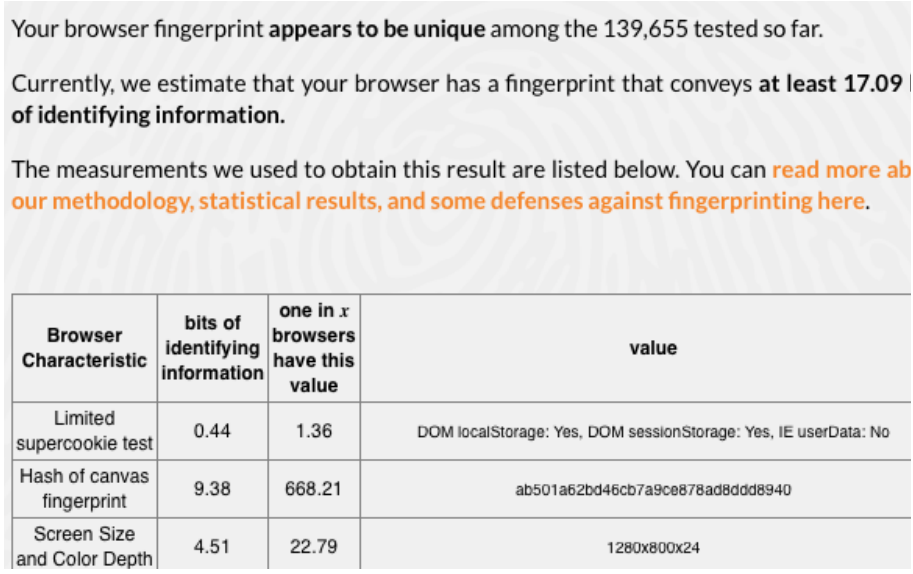
6.10.1 Definire il Fingerprinting del Browser

Innanzitutto chiariamo che un *fingerprinting* è un'operazione estremamente laboriosa e che viene effettuata solo da dei software pensati appositamente per questa operazione. Quando navighiamo in rete il nostro browser lascia "aperto" un canale che permette a qualunque sito di conoscerne delle informazioni. Queste informazioni sono:

- Risoluzione, profondità dei colori
- Plugin attivi e le versioni di ognuno di essi
- Ora corrente e Timezone
- Fingerprint del WebGL
- Lista dei font presenti nel Sistema Operativo
- Lingua corrente

- Sistema Operativo e versione
- User Agent, ovvero il browser e la tecnologia su cui si basa, quindi la sua versione
- Supporti esterni come Touchpad
- Utilizzo di AdBlock
- … e tutto quello di cui abbiamo già parlato

Resterai affascinato da sapere quante informazioni rilasciamo sui siti web che visitiamo. Se vuoi, puoi eseguire un test sul sito Panopticlick[1] sviluppato dalla EFF. Con Opera su OSX 10.11.5 appena formattato, il risultato dimostra che su oltre 139.000 test il browser è unico nel suo genere (Figura 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 of identifying information.**

The measurements we used to obtain this result are listed below. You can read more ab 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

Figura 23: Risultati di un browser Opera convenzionale su Panopticlick

[1] https://panopticlick.eff.org

6.10.2 Difendersi dal Fingerprinting del Browser

Se hai preso alla lettera ogni singolo consiglio degli argomenti precedenti con molta probabilità il tuo browser è abbastanza sicuro, tuttavia è possibile fare ancora di più. Il trucco sta nel mischiare le carte, operazione possibile manipolando le risorse sopracitate. Ogni browser permette di effettuare operazioni di "insabbiamento" come modificare la lista font, disabilitare plugin etc… tuttavia non basterebbe un libro intero per parlare solo di questo! Possiamo però utilizzare alcune estensioni in grado di venirci incontro, come ad esempio:

- FireGloves[1], disponibile per *Mozilla Firefox*
- StopFingerprinting[2], disponibile per *Google Chrome*

6.11 Download di File

In questa categoria rientrano tutti quei file che vengono **scaricati** ma che, una volta aperti, possono rivelare le informazioni dei vostri dati online. Nel caso in cui sia necessario aprire file di qualunque tipo è bene far uso di strumenti come *Virtual Machine* su computer host scollegati da Internet. I file scaricati da Internet possono contenere codice eseguibile in grado di comunicare esternamente al network in anonimato: ad esempio con le giuste conoscenze è possibile inserire codice di script arbitrario in file *Word* o *PDF*, oltre ovviamente ai classici eseguibili disponibili per il vostro sistema operativo (.exe, .dmg, .sh e via dicendo).

[1] https://fingerprint.pet-portal.eu/?menu=6

[2] https://chrome.google.com/webstore/detail/stopfingerprinting/kfhlgmfkolojpnmhgggilmillpcokmnb

6.12 Test di Sicurezza del Browser

La sicurezza del Browser è un tema molto complesso e in continua evoluzione che richiede diverse conoscenze in molti ambiti. Al momento lo strumento più completo ed affidabile per effettuare un test del proprio browser - e della sua sicurezza - viene offerto da BrowserSPY[1] che permette di verificare l'esistenza, o meglio l'esposizione, di qualunque tecnologia presente nel browser.

L'uso di questo strumento è particolarmente semplice: per ogni voce a sinistra dello schermo si aprirà una scheda riassuntiva che riguarda la tecnologia e una lista di valori che vengono esposti in rete. Assicuriamoci che tutte le voci che potrebbero in qualche modo minare al nostro anonimato siano opportunamente nascoste, approfondendo magari anche quelle che non sono state trattate in questo documento.

[1] browserspy.dk

7. SICUREZZA DEI DATI

Se nonostante tutte le precauzioni dovesse succedere a qualcuno di essere accusato di qualche reato - cosa che sinceramente non auguro a nessuno - tutti gli apparecchi informatici che per una ragione o un'altra possono ricondurre a un tipo di reato potrebbero venir sequestrati.

L'informatica forense è quella branca dell'informatica che studia i metodi per ritrovare informazioni di qualunque tipo all'interno di un dispositivo informatico. Tale campo ha riscosso molto successo negli ultimi anni; ci basti pensare a quanti casi sono stati risolti grazie a una telefonata di troppo, a una foto scattata da uno smartphone o ai file recuperati all'interno di un computer di qualche criminale. La sua evoluzione ha subito inoltre profondi cambiamenti: fino a pochi anni fa venivano sequestrati non solo computer ma anche tastiere, monitor e tappetini per il mouse, senza alcuna valida ragione!

Oggi vengono utilizzati laboratori e personale altamente qualificato e i risultati sono spesso eccellenti. Le pratiche studiate nella ricerca forense possono essere utilizzate non solo dalle forze dell'ordine - che in base alla legge potranno o meno eseguire determinate azioni - ma anche da qualunque persona abbia le abilità necessarie per metterle in pratica. Come vedremo alcune di queste abilità sono facilmente apprendibili e che, tranne in rare eccezioni, non richiede strumenti particolari. In questa parte del documento parleremo dunque di tutti quei metodi per la verifica dell'informazione, contrastare la *ricerca forense*, eliminando le tracce delle nostre azioni dal dispositivo che è stato utilizzato.

7.1 Integrità dei Dati

Per quanto una connessione sia considerata sicura non è detto che questa da sola sia in grado di garantire che i dati veicolati all'interno di una rete rimangano integri. Per integrità dei dati definiamo lo stato di originalità di tutte quelle informazioni in grado di essere inviate e ricevute: se ad esempio scarichiamo un programma da uno sviluppatore e vogliamo essere assolutamente certi che quello che ci è arrivato è lo stesso che lo sviluppatore ha distribuito, ecco allora che dovremo effettuare la verifica sull'integrità del programma.

Immagina di voler scaricare la .ISO dell'ultima versione di Ubuntu, nota distribuzione GNU/Linux per uso casalingo: se questa venisse manipolata nel server in cui è presente (ad esempio un malintenzionato riuscisse a violare i server di Ubuntu e modificare le immagini con una backdoor) o magari il download si interrompe prima del dovuto e tu inconsciamente la utilizzassi per l'uso quotidiano, potresti incappare in diversi problemi d'utilizzo. Analizziamo una case-history molto attuale sull'argomento:

Il 20 Febbraio 2016 il noto portale di Linux Mint, attualmente la distribuzione Linux più popolare, è stato violato e con esso l'ultima versione rilasciata (Cinnamon 17.3). L'attacco all'immagine disco ha permesso al criminale, conosciuto come *Peace*, di avere il totale controllo di tutti gli utenti che avessero scaricato e installato la .ISO in quelle 24 ore tramite un trojan IRC chiamato come Tsunami.

Se tutti gli utenti avessero eseguito la verifica dell'integrità dei dati (in questo caso della .ISO) probabilmente nessuno sarebbe stato infettato.

7.1.1 Checksum & Hash

Quando in informatica si parla di Checksum ci si riferisce a quella sequenza di bit che è il risultato di un calcolo effettuato sul contenuto di un'informazione. Questo calcolo viene generato da un hash, una funzione matematica in grado di restituire un valore alfanumerico (appunto, il checksum) in maniera **non invertibile**: detto più semplicemente, dando in pasto a un hash una qualunque informazione, questo produce un checksum (il risultato). Ciò permette a chiunque di generare un checksum partendo da un'informazione ma non il contrario.

Aggiungiamo che un buon hash, per essere definito tale, deve essere **resistente alle collisioni**: deve cioè produrre dei checksum univoci e che quindi non possono risultare validi per due tipi di informazioni diverse. Grazie alle sue peculiarità gli hash sono molto utilizzati nell'ambito informatico, soprattutto nel campo della memorizzazione delle password: quando noi digitiamo una password all'interno di un portale, questa - per convenzione di sicurezza - viene convertita nel suo checksum attraverso un hash specifico, così da poter confrontare ciò che l'utente ha scritto con il checksum della password nel database, senza correre il rischio di doverla memorizzare . Ad onor di cronaca, le password vengono prima "saltate" ma questa è un'altra storia.

7.1.1.1 TIPI DI HASH

Nell'informatica è possibile incontrare comunemente tre tipi di hash:

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

Ognuno di essi ha le proprie peculiarità con relativi vantaggi e svantaggi: ai fini di questo corso ci limiteremo a dire che quelli ritenuti più sicuri ad oggi sono lo SHA-256 e SHA-512.

7.1.1.2 CALCOLO DI UN CHECKSUM

Nel mondo di **macOS, Linux** e **BSD** spesso è possibile trovare un tool da linea di comando molto comodo che risulta essere **shasum**. Il seguente tool va utilizzato in questo modo:

```
$ shasum [nomefile]
```

Come per la maggior parte dei programmi nel mondo UNIX questi permettono di utilizzare diversi parametri per essere comandati al meglio. Nel caso volessimo generare il checksum con SHA a 512 bit dovremo trovare il parametro corretto nella documentazione cosha n il comando:

```
$ shasum -h
```

oppure con il comando man:

```
$ man shasum
```

Da qui scopriremo che è il parametro -a a gestire il tipo di "profondità" dell'algoritmo. Useremo allora il comando:

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

per generare l'hash a 512 bit.

Il risultato che vedremo sarà il checksum generato; sentiamoci liberi di effettuare le nostre prove con i nostri file. Ecco l'esempio di output di un file random:

```
c568ac4df6aef33d887b0326c46d340196fe722f34d696bf7ab7
ac9bd2cad933bdc9aa581612d678bead2f3550438c9b7280cd99
c2c7e469c76d9ab9d889a983  stefano9lli.txt
```

Ipotizziamo di voler verificare l'ultima versione di Debian (attualmente la 8.6.0 in versione standard) scaricata nel nostro computer e verificarne l'integrità. Prima di tutto ne genereremo il checksum in locale:

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

adesso lo confronteremo con quello fornitoci dagli sviluppatori stessi nel mirror ufficiale[1] . Sceglieremo in questo caso *SHA512SUM*, quindi troveremo nel documento la porzione interessata:

```
e9506a3746e351203757599a8ce01ba4a84260a633177ee719fa
6754b70151f82d03a2843c4aa58e17aa10c35e61369077ea3207
b956183259be8444c465e4eb  debian-live-8.6.0-amd64-
standard.iso
```

Se i due checksum risultano identici vorrà dire che ciò che abbiamo scaricato è quello che gli sviluppatori avevano previsto che avessimo.

Anche gli utenti **Windows** potranno effettuare questo tipo di operazione, utilizzando un software integrato. Il comando si chiama certUtil:

```
$ certUtil -hashfile [percorsoFile] [algoritmo]
```

quindi nel caso di un file random nel nostro Desktop tramite algoritmo SHA-512 sarà:

```
$ certUtil -hashfile C:
```

Una nota di merito va a Hashtab[2] (Figura 24), programma freemium che installa la funzione di checksum direttamente nell'Explorer di Windows, integrandosi nella voce del menù "Proprietà" al click destro di un file.

[1] cdimage.debian.org/debian-cd/current-live/amd64/iso-hybrid/

[2] implbits.com/products/hashtab/

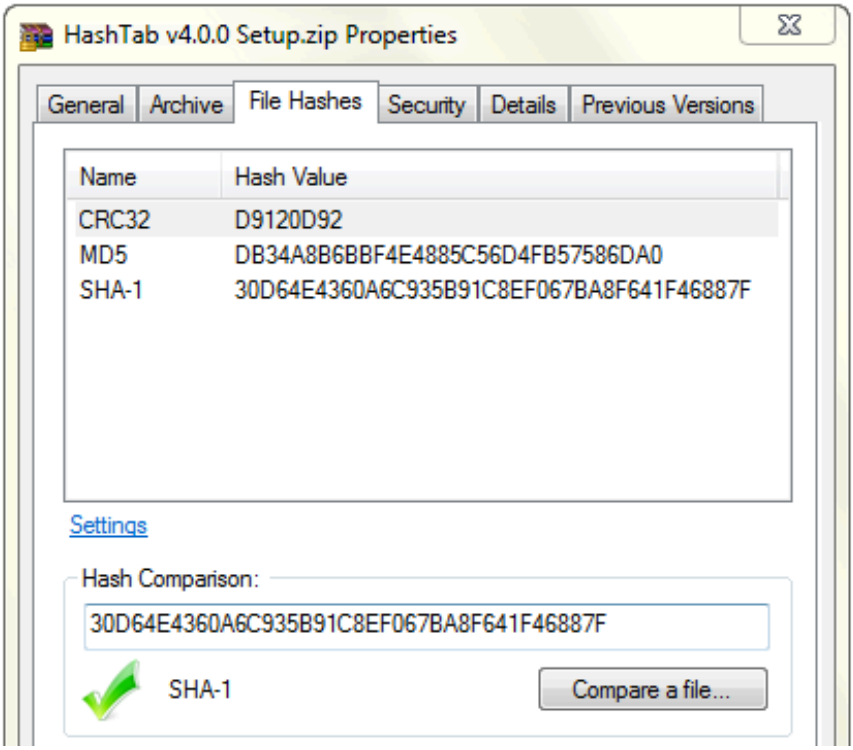


Figura 24: Screenshot del programma HashTab per Windows

7.1.1.3 CHECKSUM NELL'USO COMUNE

Durante l'inizio di questo paragrafo abbiamo spiegato come l'uso dei checksum permettesse di garantire l'integrità di un'informazione: questa affermazione è però vera nel momento in cui gli unici a generare il checksum siamo noi stessi o la fonte da cui proviene il checksum per la contro-verifica non può essere manipolata. Il checksum, così come l'abbiamo visto, funge da fingerprint di un file ma non garantisce la provenienza del checksum che useremo per il confronto. Se ovviamente qualcuno riesce a violare un sito web e i file che ospita non si farà troppi problemi anche a cambiare il checksum presente sulla pagina, non trovi?

L'uso delle hash senza alcun tipo di firma digitale può quindi risultare utile solo a fini di verifica dei dati personali, che quindi potrebbero risultare mutati solo da attacchi esterni. In questo caso il proprietario prenderà i giusti

accorgimenti per custodire il checksum originale e confrontarlo al momento del bisogno, ma non potrà utilizzarlo come timbro per garantire la verifica dell'integrità di un file. Per il momento quindi non abbiamo ancora uno strumento in grado di verificare se ciò che vogliamo scaricare dalla rete sia esattamente quello che volevamo; riprenderemo l'argomento nel capitolo "PGP/GPG per l'integrità dei dati".

7.2 Crittografia dei Dati

Arrivati a questo punto dovremmo essere abbastanza informati su tutto quello che c'è da sapere nella navigazione e nell'uso degli strumenti di anonimato su Internet. Quello che però ora ci manca è una sana e giusta preparazione dell'ambiente di lavoro e una minima conoscenza degli strumenti adatti per evitare di lasciar tracce in grado di raccontare ciò che è stato fatto con il computer. Ecco, immagina di essere un utente di *Silk Road 3.0* (o a che versione sarà ora) o in una qualunque altra community dove l'anche solo esserti registrato potrebbe farti passare dei guai seri… non vorresti certo che qualcuno sia in grado di identificarti, vero? Pensa che l'NSA è riuscita a catturare dozzine di spacciatori e clienti risalendo a username e password di Silk Road.

È curioso sapere che dopo tutte le precauzioni prese, il computer formattato, *TOR* fresco di installazione, *Bitcoin wallet* nuovo fiammante e via dicendo alla fine l'hanno beccato perché nella sua password c'era il suo numero di cellulare. Davvero, non sto scherzando. Dato che abbiamo già parlato di protocolli sicuri sappiamo quanto sia importante la cifratura dei messaggi. Questa operazione può essere applicata non solo alla connessione e ai dati che non possiamo vedere ma anche ai messaggi che scambiamo con altri utenti (amici, famigliari, venditori etc…).

7.2.1 PGP, Pretty Good Privacy

Quando si parla di crittografia delle informazioni non si può certo evitare PGP (acronimo di *Pretty Good Privacy*) uno strumento in grado di cifrare, decifrare e firmare testi, email, file e directory per incrementare la sicurezza dei tuoi documenti. Il suo funzionamento è il seguente: l'utente che vuole cifrare il messaggio creerà due chiavi: una **chiave pubblica** e una **chiave privata**. La chiave *pubblica* è quella che permette a chiunque di inviarti un messaggio criptato, la chiave *privata* è invece l'unica chiave in grado di sbloccare - e quindi leggere - il messaggio creato con quella chiave pubblica.

Questa è in buona sostanza la crittografia su cui si basa quasi tutta la comunicazione informatica: il sistema pubblica/privata è anche conosciuta come *Crittografia Asimmetrica (o Diffie ed Hellman),* mentre l'uso di una sola chiave (che PGP comunque usa) si definisce *Crittografia Simmetrica*. Se in PGP perdi la chiave privata considera l'informazione protetta come perduta per sempre.

7.2.2 GPG, GNU Privacy Guard

GNU Privacy Guard (da ora in poi GPG) è una suite di tools disponibile per *Windows, macOS, Linux e BSD*. Nasce come alternativa libera a PGP da cui ne eredita lo standard di crittografia *OpenPGP*. Consideriamo quindi GPG come un'alternativa libera a PGP, software che ha creato lo standard di cui GPG si serve per funzionare. GPG[1] viene fornito, oltre che in versione *CLI*, anche sotto diverse vesti:

- GPGTools[2], suite di tools per macOS
- GPG4Win[3], client per Windows

[1] https://www.gnupg.org

[2] https://gpgtools.org

[3] https://www.gpg4win.org

- gpg4usb[1], una versione pensata per girare esclusivamente su USB (Windows e Linux)

- … e molti altri!

> GPG è presente di default in molte distribuzioni GNU/Linux. Se preferisci la modalità grafica puoi usare seahorse (la stessa usata da Tails). Da qui in poi faremo ampio uso del terminale in quanto l'uso dell'interfaccia grafica risulta essere particolarmente intuitiva. Tutte le operazioni sui file si potranno eseguire con il tasto destro, usando poi le voci presenti in base alle situazioni. In caso di dubbi, consigliamo di imparare prima il procedimento a riga di comando, quindi di provare con la modalità grafica.

7.2.2.1 COMPRENDERE CHIAVE PUBBLICA/PRIVATA

Prima abbiamo spiegato la differenza tra chiave privata e quella pubblica, quindi non mi ripeterò, ciò che è stato già detto dovrebbe bastare per comprendere come funziona.
Riassumendo:

- La **chiave privata** (*private key*) deve rimanere segreta, è tua e non va condivisa con nessuno.

- La **chiave pubblica** (public key) può essere condivisa con il resto del mondo.

Tra chiave privata e chiave pubblica esiste una relazione che spiegherò semplicemente così: *una chiave pubblica può essere decriptata solo dalla relativa chiave privata*.

Per semplificare ancora di più la cosa faremo un esempio: *Andrea e Beatrice* sono due amici che vogliono scambiarsi messaggi. Tuttavia i due non si fidano dei canali di comunicazione e decidono di usare PGP per scriversi. Per fare in modo che entrambi possano cifrarsi e decifrarsi a vicenda dovrebbero avere una

[1] https://www.gpg4usb.org/download.html

password in comune, tuttavia per potersela comunicare dovrebbero usare il sistema di comunicazione che reputano non sicuro.

Per sopperire a questa difficoltà PGP usa un tipo di crittografia chiamata "asimmetrica", dove i messaggi vengono scambiati facendo uso di chiavi pubbliche e private. *Andrea* ha una sua chiave pubblica e una privata, così come *Beatrice.* Quando *Andrea* vorrà scrivere un messaggio a *Beatrice* userà la chiave pubblica di quest'ultima. Se *Beatrice* vorrà decifrare quel messaggio dovrà usare la sua chiave privata. Essendo *Beatrice* l'unica proprietaria di quella chiave privata solo lei potrà decodificare quel messaggio. Semplice, no?

7.2.2.2 CREARE LA PROPRIA CHIAVE PGP

In questa parte della guida impareremo a creare le nostre chiavi pubbliche e private, così da permettere ad altri internauti di inviarci messaggi cifrati che solo noi potremo leggere. Assumendo che tu stia utilizzando Debian è possibile avviare la GUI di GPG (Figura 25) lanciando il programma "*seahorse*" da Terminale, o più comunemente "*Password e Chiavi*" dalla lista delle applicazioni.

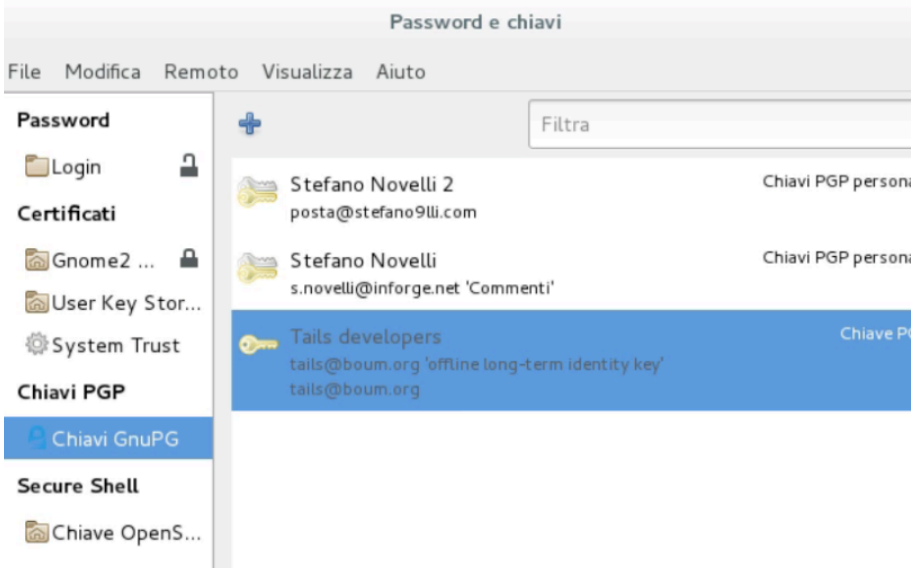


Figura 25: Schermata iniziale della GUI "seahorse" su Debian GNOME 3

Clicca ora su *File -> Nuovo* (oppure usa la shortcut CTRL+N) e clicca su *Chiave PGP*, quindi *Continua*. Definisci il tuo *Nome Completo* e *indirizzo Email*. Aprendo le impostazioni avanzate potrai modificare il *tipo di chiave* (quella consigliata è RSA) e la *forza di cifratura* (fino a un massimo di 4096, al momento la chiave più forte che si può usare).

Puoi anche scegliere di assegnare una *scadenza* e un *commento* aggiuntivo. Puoi ora cliccare su *Crea*. Ora devi assegnare una password alla tua chiave. A questo punto troverai la tua chiave appena creata sotto la voce “Chiavi GnuPG”; se così non fosse, dovrai attendere che il programma generi abbastanza entropia per la tua chiave (ti spiegherò a breve di cosa si tratta).

Se sei un amante del terminale è possibile ovviamente farlo anche da lì. In questo caso il comando da lanciare sarà:

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

Ti verrà chiesto che tipo di chiave vuoi scegliere, quindi assegnerai la lunghezza della chiave e un’eventuale scadenza. Esattamente come in versione GUI, andrai ora ad indicare il Nome ed Email; per finire digiterai la passphrase.

Ci verrà chiesto ora di “muoverci” con mouse e tastiera: questa operazione serve a raccogliere entropia da poter associare alla forza della chiave. Oltre a battere tasti a caso, posso consigliarti qualcosa che ti tiene occupato, come ad esempio una partita a Forza 4!

7.2.2.3 IMPORT, EXPORT E REVOCA DI UNA CHIAVE PGP/GPG

Per poter inviare messaggi cifrati ad altre persone è necessario prima di tutto **importare** la chiave pubblica del destinatario. Per farlo da GUI non sarà un problema, un pulsante dedicato: in Debian lo troviamo sotto la voce *File -> Importa* oppure usando la shortcut *CTRL+I*; se invece vogliamo farlo da linea di comando ci basterà digitare:

```
$ gpg --import [nomefile]
```

Se invece vogliamo **esportare** una chiave possiamo seguire la voce *File -> Export* oppure sempre da linea di comando lanciare (sostituirai [ID] con la User ID (il suo formato è tipo AB1234567) :

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

Questo tuttavia produrrà un output illeggibile; possiamo allora formattarlo in ASCII con i parametri:

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

quindi possiamo anche salvare l'output in un file, come da esempio:

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

È possibile anche **revocare** una chiave privata, magari nel caso in cui sia andata persa una chiave privata o peggio sia stata violata. Questa voce - e anche le prossime - è disponibile sotto la tab "Dettagli" di ogni chiave (Figura 26).

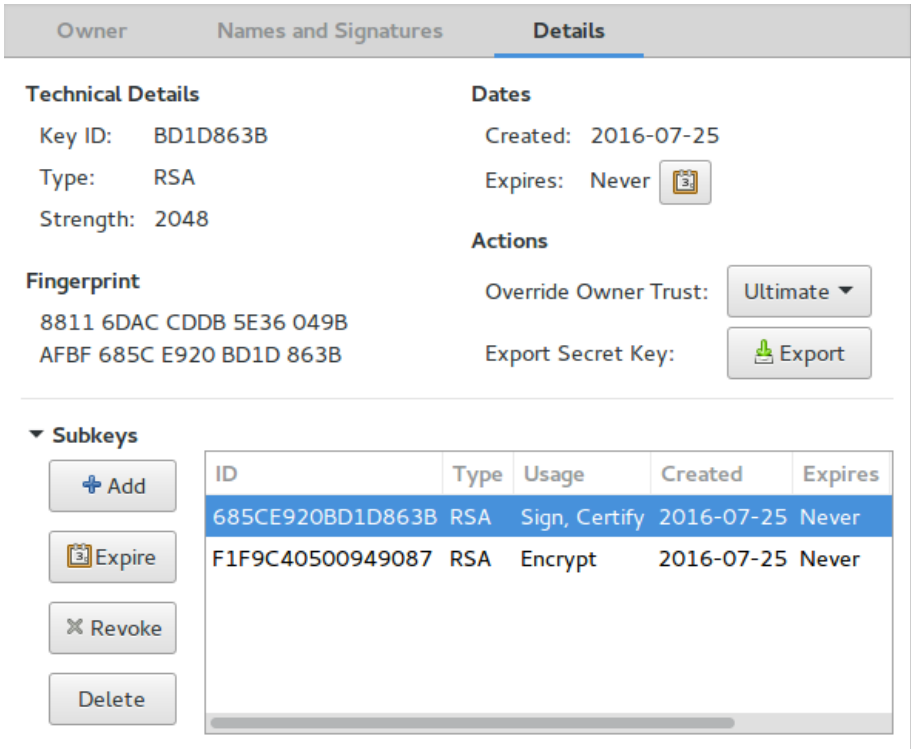


Figura 26: Dettagli di una chiave GPG su seahorse

che tradotto in linea di comando sarà:

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

per generare una revoca della chiave, compiliamo i campi come richiesto, quindi importiamo il certificato di revoca:

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

Se per qualche motivo abbiamo sincronizzato la nostra chiave con i keyserver di PGP dovremo richiederne la resincronizzazione in questo modo:

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

Infine aggiorniamo il nostro portachiavi in questo modo:

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

È probabile che avremo anche bisogno di avere una lista di tutte le chiavi a nostra disposizione. Potremo usare il comando:

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

Se comunque preferisci la strada semplice puoi far click destro sulla chiave e cliccare su Elimina.

7.2.2.4 PGP/GPG PER CIFRARE E DECIFRARE UN FILE

Su terminale il comando per cifrare un file sarà:

```
$ gpg --output secret.gpg --encrypt --recipient
```

Questo potrà essere inviato come allegato. Se la situazione non ce lo permette potremmo voler generare un output con codifica ASCII per poterlo inviare tramite testo, usando quindi il parametro --armor (-a):

```
$ gpg --armor --encrypt --recipient [destinatario]
```

Il file generato sarà [file].asc, contentente il valore ASCII del testo che abbiamo scritto. Si presenterà in questo modo:

```
-----BEGIN PGP MESSAGE-----
Comment: GPGTools - https://gpgtools.org
hQIOAwfq5Jrby+ZxEAf+N/ozNDVnsURxXb/lcKyPB/
V4QuIGG5nQVAIZ5KO8W4/+
[...]
pVhvtqu+q2yiE4khriBkpZD709uaf1kxfTaRosmRMl74duShAEQU
uwjnyA1aOcT0
```

Così facendo possiamo inviare il contenuto di un file cifrato senza allegarlo ma semplicemente incollandolo in una mail (tieni però sempre conto delle dimensioni del file o rischi di inviare decine di MB di testo!).

Da notare che in questo caso il parametro --encrypt ricopre il ruolo di identificatore che dice al programma gpg “ehi, adesso devi cifrare!”. E per **decifrare**? Ma ovviamente --decrypt!

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

7.2.2.5 PGP/GPG PER LA FIRMA DEI DATI

OpenPGP offre la possibilità non solo di cifrare i messaggi ma anche di firmarli: la firma funge come una specie di certificato che attesta la vera proprietà di chi ha scritto il messaggio. A cosa serve? Immagina di avere una corrispondenza con un nostro conoscente all’interno di un forum: se quest’ultimo venisse attaccato e l’account del nostro destinatario venisse compromesso, non sapremmo se è effettivamente lui a volerci scrivere qualcosa.

Il fatto che lui abbia la nostra chiave pubblica non significa che è effettivamente lui ad essere chi dice di essere: potrebbe aver preso la chiave pubblica dal web, dallo storico dei nostri messaggi non cifrati o da altre fonti. Per dimostrare di essere effettivamente chi crediamo egli sia dovrà firmare il suo messaggio con la sua chiave privata. Vediamo come comportarci in questo caso.

Il comando per **firmare** con la propria chiave consiste nell’uso del parametro -s (o --sign):

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

Il nostro file verrà rinominato con estensione .gpg . Di default il comando comprime anche il valore del file, quindi per avere un valore leggibile utilizzeremo:

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

Quindi verrà salvato in formato .asc . Se volessimo **verificarlo** potremo usare il comando:

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

Il parametro --clearsign si può aggiungere assieme ai valori di cifratura, quindi nel caso in cui volessimo cifrare e firmare un documento di testo potremmo usare il comando:

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

Lo so, sono tanti comandi da ricordare, ma personalmente consiglio di farci pratica (magari usando il man di gpg) anziché l'uso della GUI; in breve tempo riuscirai a prenderci la mano ed essere più produttivo di quanto tu non possa fare con l'interfaccia grafica.

7.2.2.6 PGP/GPG PER L'INTEGRITÀ DEI DATI

Nel capitolo riguardante l'integrità dei dati abbiamo esposto un problema a cui ancora non avevamo una risposta: come faccio ad essere sicuro al 100% che un file sia integro e che questo provenga da una fonte attendibile? Il dubbio era riferito al fatto che il confronto dei checksum, quindi dei risultati prodotti dagli algoritmi hash, potessero in qualche modo essere manipolati all'interno del server che li ospita.

Grazie alla crittografia asimmetrica e in particolare al modello OpenPGP possiamo finalmente rispondere a questa domanda: utilizzeremo cioè la logica che risiede alla base della chiave pubblica e chiave privata per essere certi che la provenienza sia assolutamente attendibile e che il file scaricato sia perfettamente

intatto. Torniamo alla nostra Debian. Innanzitutto ci procureremo la firma del file[1] scaricando l'hash SHA512:

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

e il relativo .sign che contiene la firma:

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

per non avere noie di conflitti importiamo dal portachiavi GPG di Debian la chiave con id 6294BE9B (che abbiamo preso da https://www.debian.org/CD/verify):

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

adesso possiamo verificare il .sign (che dovrà avere lo stesso nome del file originale, quindi nel nostro caso SHA512SUMS e SHA512SUMS.sign):

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

Se tutto va per il verso giusto riceviamo lo status di Firma valida:

```
gpg: Firma valida da "Debian CD signing key <debian-cd@lists.debian.org>"
```

Abbiamo ora confermato la validità dell'hash, quindi possiamo finalmente essere sicuri che lo SHA512 che abbiamo scaricato possa essere usato come controprova. Se vogliamo testare il funzionamento della verifica di GPG possiamo modificare il file contenente il checksum:

```
$ nano SHA512SUMS
```

aggiungendo magari un carattere a fine file. Salviamolo e riverifichiamo:

[1] cdimage.debian.org/debian-cd/current-live/amd64/iso-hybrid/

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

stavolta riceveremo un errore:

```
$ gpg: Firma non corretta "Debian CD signing key
<debian-cd@lists.debian.org>" [unknown]
```

Siamo ora pronti a rieffettuare il checksum con il file .ISO (per maggiori info visita la pagina relativa il checksum).

7.2.2.7 PGP/GPG PER CRITTOGRAFIA DI EMAIL

La crittografia email può essere utile non solo per evitare che qualcuno monitori la tua connessione (potrebbe ad esempio leggere le tue mail se queste viaggiano in chiaro in rete senza protocolli di sicurezza) ma anche per evitare che qualcuno acceda alla tua posta e riesca a leggerne i contenuti cifrati. Certo, se il tuo intento è quello di non essere localizzato è inutile che ti dica di star lontano da servizi di mailing che consentono l'accesso solo da *Clearnet* (Gmail, Yahoo, Hotmail, Libero etc…) ma di affidarti solo a servizi che consentono l'accesso da nodi TOR, Proxy e VPN...

Una seconda considerazione, forse abbastanza ovvia ma da mettere in luce, è quella di non utilizzare la stessa chiave che si usa per operazioni di anonimato sulla propria mail in *clearnet*, o comunque a cui hai avuto accesso almeno una volta senza le giuste precauzioni. Questo permetterebbe a chiunque ne abbia le abilità di risalire alla tua persona. Ci sono diverse situazioni e altrettanti strumenti che ci consentono di farne uso. Di seguito ne troverete qualcuno, giusto per poter iniziare al meglio la tua ricerca:

- Enigmail[1]: estensione per Thunderbird e SeaMonkey, necessita di GnuGP già installato.

[1] https://addons.mozilla.org/it/thunderbird/addon/enigmail/

- Mailvelope[1]: estensione che consente di utilizzare la crittografia OpenPGP all'interno delle webmail come GMail, Yahoo Mail, Outlook etc… tramite i browser Chrome e Firefox based.

- GPGMail[2]: presente nella GPG Suite, è un tool che permette di crittografare all'interno del programma Mail di OSX

- APG[3]: disponibile per Android, permette di integrare GPG facilmente nei file e nelle mail

- SecureGmail[4]: estensione che permette di mettere al sicuro la mail dei servizi GMail e di tutti i sistemi basati sulle GApps

Tra i repository Debian troveremo Icedove: creato da *Mozilla Foundation* e ribrandizzato in salsa *Debian,* è un client di posta Thuderbird-based. Per installarlo ci basta lanciare il comando:

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

quindi installiamo anche Enigmail, l'estensione che ci permette di utilizzare GPG in Icedove:

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

Una volta avviato Icedove sarà possibile attivare la firma e la cifratura PGP cliccando semplicemente sul pulsante "*Enigmail*" posto solitamente sopra l'intestazione della mail (Figura 27).

[1] https://www.mailvelope.com

[2] https://gpgtools.org/gpgmail/index.html

[3] www.thialfihar.org/projects/apg/

[4] https://www.streak.com/securegmail

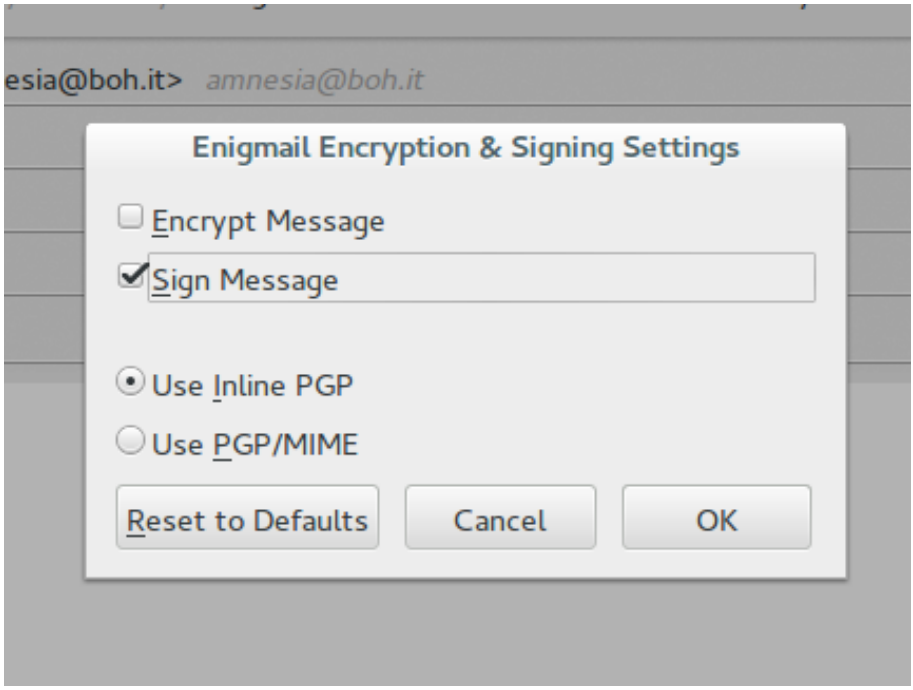


Figura 27: Uso di Enigmail all'interno del client Icedove su Debian GNOME 3

Ora che abbiamo appreso il funzionamento di PGP/GPG voglio nuovamente ricordarti l'importanza di questo strumento prendendo in esame un evento recentemente accaduto.

Nel takedown di *Silk Road* è stato scoperto che molti degli admin, tra cui il famoso creatore *Ross Ulbricht*, non usavano la crittografia nelle loro comunicazione. Quando Ross è stato arrestato gli investigatori hanno trovato nel suo computer centinaia di files. Tra questi - secondo gli utenti di Silk Road 2 - vi erano documenti contenenti informazioni personali su amministratori e moderatori salvati su file di testo, da cui ovviamente gli inquirenti hanno ottenuto il modo di risalire ai suoi complici.

7.2.3 Dove conservare le chiavi PGP/GPG

Sarebbe assurdo se dopo tutta questa paranoia lasciassimo le chiavi di decrittazione (e qualche altro file importante) in bella vista nel nostro computer, non trovi? Ecco perché, in caso qualcuno venga a farti visita, devi essere pronto a nascondere - o nel peggiore dei casi demolire - un supporto rimovibile come una chiavetta USB, o se il tuo computer te lo consente, una scheda SD (meglio ancora se una micro SD con adattatore SD).

Utilizzare una micro SD può essere il modo migliore per nascondere le chiavi; la memoria micro SD è così piccola da poter essere nascosta tra le dita, dentro un orecchio (perché no?), nelle scarpe, nelle mutande. Lascio a te l'immaginazione della cosa! Ad ogni modo, approfondiremo la memorizzazione delle chiavi e di altre informazioni nel capitolo "Backup dei Dati".

Nascondere o distruggere una chiavetta USB invece non può essere poi così facile, oltre ad essere meno semplice da occultare è tecnicamente più difficile da distruggere. La SD invece può essere spezzata con poche difficoltà, rendendo illeggibile qualunque contenuto. Per approfondire l'argomento salta al capitolo del "Data Shredding".

7.3 Crittografia del disco

La crittografia dei dati può essere applicata su un intero disco o su una parte di esso; questa funzione è presente in ogni sistema operativo:

- Su Windows la tecnologia è **BitLocker**
- Su macOS / OS X bisogna creare una partizione **encrypted**.
- Su GNU/Linux è solitamente chiamata **Whole Disk Encryption** o **Full Disk Encryption**

Come per la crittografia a livello client, se perdi la tua chiave (passphrase) non c'è possibilità di recupero; puoi solo formattare il disco e reinstallare un nuovo OS. Poiché potrebbe essere necessario cifrare porzioni o intere partizioni da poter usare anche tra più Sistemi Operativi, ci concentreremo sull'uso di un software multipiattaforma, indipendente dall'architettura del Sistema e in grado di gestire diverse tecnologie.

7.3.1 TrueCrypt

TrueCrypt ha scritto in parte la storia della crittografia informatica, essendo il capostipite di una generazione di software che ha introdotto all'utente medio la possibilità di cifrare interi dischi senza essere un tecnico informatico.

Il progetto è stato abbandonato nel 2014 in concomitanza con la fine definitiva al supporto di Windows XP ma è ancora disponibile per chi avesse bisogno di relative versioni del programma. Fortunatamente per noi esistono una serie di fork che ne hanno risolto i limiti e i problemi, ma ho pensato che fosse giusto prima dedicare una voce riguardo una le possibilità che *TrueCrypt* - e quindi le sue reincarnazioni - offrono:

- Permette di cifrare partizioni intere
- Permette di creare due partizioni: nel caso in cui tu sia costretto a sbloccare il disco, usando una password si può accedere a una partizione, usandone un'altra si accederà a una seconda partizione
- Supporta l'accelerazione hardware offerta dalle caratteristiche delle CPU di ultima generazione
- Supporta tre algoritmi di cifratura: AES, Serpent e Twofish. È possibile anche intrecciarle tra di loro.

Prima di procedere alcune raccomandazioni e controindicazioni generali; non valgono per tutte le situazioni ma diamo per buone tutte le possibilità:

- Mai deframmentare e indicizzare i volumi criptati, potrebbero lasciare tracce nei log di sistema

- Non usare filesystem journaled, preferisci filesystem che non lo prevedono (es: FAT32, exFAT o ext2)

- Usa una formattazione completa, non usare eventuali voci come "formattazione veloce"

- Quando apri i file ricorda che verranno immagazzinati in RAM e cartelle temp, procedi alla loro rimozione appena hai finito (soprattutto se non stai usando OS Live)

- In ogni caso, adotta tutte le precauzioni necessarie per garantire la sicurezza del dispositivo che stai utilizzando (internet scollegato, sistema aggiornato, protezioni etc…)

7.3.2 Veracrypt

A prendere le redini di questo programma troviamo senza dubbio Veracrypt[1], disponibile per tutti i maggiori OS e retrocompatibile con Truecrypt. Il programma si presenta in maniera molto user-friendly e pertanto facile nell'uso.

7.3.2.1 INSTALLARE VERACRYPT

Come per gli altri argomenti, salteremo l'installazione su Windows e macOS in quanto alla portata di tutti; l'installazione di Veracrypt su GNU/Linux non è neanche difficilissima ma si consiglia di seguire alla lettera i passaggi seguenti. Per prima cosa scarichiamo[2] il tar.bz2 della versione GNU/Linux (attualmente la versione 1.19). Apriamo quindi il terminale e dirigiamoci nella cartella "Scaricati":

```
$ cd $HOME/Scaricati
```

[1] https://veracrypt.codeplex.com

[2] https://veracrypt.codeplex.com/wikipage?title=Downloads

ed estraiamo il pacchetto con il comando

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

A questo punto troveremo 4 file: identifichiamo quelli che hanno il nome che contiene la parola “gui”. Procediamo quindi ad effettuare l’installazione del programma:

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

Scegliendo la versione x86 o x64. Se non conosciamo l’architettura del nostro Sistema Operativo (ricordo che x86 sta per i processori a 32bit mentre x64 sta per quelli a 64bit) possiamo lanciare il comando:

```
$ hostnamectl
```

Se hai comunque dei dubbi su quale installare preferisci sempre la x86 (che è compatibile anche con processori x64 ma anche più lenta).

7.3.2.2 UTILIZZARE VERACRYPT

Vediamo ora in che modo possiamo utilizzare Veracrypt, creando una partizione o un contenitore dove immagazzinare tutti i nostri file più importanti e all’oscuro di tutti.

1. Per creare il nostro primo contenitore scegli uno slot tra quelli presenti, clicca sul tasto *Create Volume*, quindi decidi se *creare un contenitore* oppure *cifrare una partizione*. Starà a te decidere quale situazione sia la più comoda.
2. La seconda facciata ci mostrerà un nuovo bivio: cifratura standard o hidden? Nel primo caso se non avremo una passphrase di sblocco non potremo accedere alla cartella/partizione, nel secondo caso si potrà usare una seconda *passphrase* in sostituzione a quella principale, così nel caso tu sia costretto a decifrare la partizione potrai decidere quale passphrase (e quindi

quale partizione o contenitore) mostrare. Sarà comunque possibile confrontare le dimensioni dei file mostrati con l'effettiva capienza disponibile nella partizione per capire che questo sia un imbroglio ma comunque è meglio di niente.

3. (Opzione) Se si è scelto il contenitore e non la partizione specificare dove memorizzarla

4. Scegli ora il tipo di algoritmo. Puoi utilizzare la configurazione già presente. Se hai dubbi fai riferimento al capitolo "Crittografia".

5. Decidi quanto spazio vuoi dedicare al contenitore o al volume.

6. Ora è il momento di scegliere una password. Se hai scelto l'opzione Hidden dovrai compilare due volte questo passaggio, il primo sarà quello riguardante la "password fasulla".

7. (Opzionale) In fase di creazione ci viene chiesto se vogliamo abilitare il *PIM*: questo è un moltiplicatore introdotto nelle ultime versioni di Veracrypt che consente di specificare un valore numerico alla propria combinazione (che però dovrà essere di almeno 20 caratteri). Il PIM va a moltiplicarsi alle possibilità di trovare una chiave, in pratica se noi specifichiamo un valore PIM compreso tra 1 e 485 vorrà dire che ci sono n^485 probabilità di trovare la password, il che tradotto significa che moltiplicheremo di 485 volte la sicurezza della nostra password. In ogni caso potrebbe essere un per di più, considerato che una password di 24 caratteri cifrata in AES a 512 bit si riuscirebbe a forzare solo aspettando lo stesso tempo di vita dell'universo. Così, tanto per dire…

8. (Opzionale) È possibile anche utilizzare una keyfile, vale a dire un file che viene generato automaticamente e che contiene una password. Tale opzione potrebbe risultare più sicura di una password normale in quanto conterrebbe caratteri di qualunque genere e non solo limitati al charset del layout della tastiera, in più difenderebbe da attacchi di keylogging.

9. A questo punto sarà necessario scegliere il tipo di partizione. Per sapere qual è il migliore ti consiglio di leggere le controindicazioni sotto il capitolo "*Truecrypt*". Personalmente preferisco sempre un filesystem di tipo exFAT in quanto molto compatibile con i miei sistemi, ad ogni modo sentiti libero di scegliere quello che più fa al caso tuo.
10. Decidiamo se vogliamo abilitare l'opzione di supporto Cross Platform. Questo può essere utile se ad esempio usi macOS e vuoi limitare al massimo i rischi del volume.
11. Se non ci è stata già mostrata troveremo una barra di caricamento che si muoverà con il movimento del mouse. Questa funzione permette di generare una chiave di cifratura casuale in base ai movimenti randomici del mouse. Maggiori sono i movimenti, più difficile sarà crackare la chiave di cifratura.

Il volume sarà ora *creato*.

Procederemo ora ad effettuare il **montaggio del drive o del file**, in base a ciò che abbiamo scelto:

1. Selezioniamo uno slot vuoto
2. Selezioniamo *Select File* o *Select Device*
3. Clicchiamo su *Mount*
4. Digitiamo la *passphrase* o carichiamo il *keyfile*
5. Inseriamo la password del nostro account utente per permettere al Sistema di generare una nuova partizione virtuale

Se stai seguendo la guida con Debian riceverai un errore e non potrai proseguire; questo è causato dal fatto che il tuo utente non ha i poteri per creare una partizione virtuale. Per fare ciò è necessario aggiungere il tuo utente alla lista degli utenti sudo, quindi da Terminale lanceremo i comandi:

```
$ su
$ visudo
```

E alla fine del file aggiungiamo l'utente (in questo caso è stefano9lli ma se il tuo nome utente è diverso ovviamente cambialo con quello in tuo possesso):

```
stefano9lli      ALL=(ALL) ALL
```

(Tieni presente che il primo spazio è assegnato con il tasto [TAB] della tastiera, mentre il secondo è uno spazio normale). Salva il file con la combinazione *CTRL+X*, il *tasto S* e quindi *INVIO*.

In questo modo potremo non solo utilizzare il nostro utente come amministratore su Veracrypt ma in qualunque altra situazione! Rieffettuiamo ora i passaggi descritti per montare il volume.

Come potrai notare comparirà una nuova partizione: qui potrai inserire tutti i tuoi file in sicurezza senza che altri utenti - senza conoscere la password di cifratura - potranno vedere.

Quando hai finito puoi **smontare un volume** cliccando sul tasto *Dismount* del programma.

7.3.3 Zulucrypt, LUKS e famiglia

Nel mondo GNU/Linux sta spopolando un nuovo tool di cifratura chiamato Zulucrypt[1]. Il suo punto di forza è quello di supportare non solo i formati creati da TrueCrypt e VeraCrypt ma anche *LUKS*, un metodo di cifratura di dischi rigidi di riferimento nel mondo Linux.

LUKS è visto come uno standard nell'ambiente del pinguino, quindi è giusto sapere che esiste ed eventualmente anche come interagirvi: nell'ambiente

[1] mhogomchungu.github.io/zuluCrypt/

Windows esiste un adattamento offerto dal tool FreeOTFE[1] mentre su *OSX* un tempo esisteva *OSXCrypt* ma sembra ormai abbandonato.

Tornando all'ambiente Linux è il modulo *dm-crypt* che si occupa di offrire il supporto alla cifratura con LUKS; tale modulo è presente in quasi tutte le distribuzioni GNU/Linux e non dovrebbero esserci problemi ad utilizzarlo. Tuttavia dm-crypt risulta essere particolarmente ostico da gestire per un utente alle prime armi, mentre risulta più semplice utilizzare un tool chiamato *cryptosetup* che offre il supporto al metodo LUKS tramite modulo dm-crypt. Il suo utilizzo richiede una certa conoscenza di partizioni, mountpoint e comandi generali di GNU/Linux (che potrebbero cambiare da famiglie di distribuzione), pertanto si consiglia di leggere il manuale ufficiale di cryptosetup[2].

7.4 Steganografia

La Steganografia è una tecnica utilizzata per nascondere messaggi all'interno di contenitori che a una prima occhiata possono sembrare innocui: già nell'Antica Grecia Erodoto racconta di come Demarato di Sparta, per avvisare le città vicine di una possibile invasione persiana, utilizzava delle tavolette ricoperte da cera. Nel caso in cui i messaggeri fossero stati scovati, le spie nemiche avrebbero trovato delle tavolette di cera su cui erano scritti i messaggi, non sospettando che invece al di sotto di esse ci fosse il messaggio originario.

[1] https://it.wikipedia.org/wiki/FreeOTFE

[2] https://gitlab.com/cryptsetup/cryptsetup/wikis/FrequentlyAskedQuestions

7.4.1 Steganografia con metodo LSB

Nell'informatica la steganografia più comune risiede all'interno di una tecnica identificata come **LSB** (acronimo di *Least Significant Bit*, ovvero bit meno significativo) che si basa sulla teoria che un'immagine, un video o un'audio di grandi dimensioni può essere alterata di una piccolissima porzione per contenere informazioni da nascondere.

Immaginiamo di prendere una grande immagine da 1920x1080: questa immagine contiene 2 milioni di pixel, chi penserebbe che tra uno di questi viene nascosto un messaggio segreto? Bisognerebbe ingrandire l'immagine *pixel per pixel* e conoscerne la posizione per riuscire a riconoscerlo. A questo si aggiunge che molti dei tools *steganografici* usano algoritmi di lettura per "rompere" un pixel, scegliendone uno poco significativo che non brillerebbe al centro dell'immagine. Questo comporterebbe un problema ancora più grande se consideriamo che un occhio attento potrebbe essere ingannato. Puoi farti un'idea di un'immagine complessa in Figura 28.

Figura 28: Quest'immagine contiene un messaggio segreto: riesci a vederlo?

I programmi di questo tipo integrano a loro volta una serie di cifrari volti a criptare il messaggio, così che qualunque software di scanning non riesca a decifrarne il contenuto (magari tramite attacco a dizionario).

Tuttavia questo metodo non è esente da attacchi: la *steganalisi* è la materia che si occupa di effettuare test statistici per verificare se sono presenti messaggi all'interno di immagini/video/audio. Trattiamo quindi la steganografia come un metodo di difesa vulnerabile esattamente come tutti gli altri. Considera inoltre che l'immagine deve navigare così com'è: un ridimensionamento o un'ottimizzazione dell'immagine comprometterebbe per sempre l'informazione interna. In caso l'immagine risulti parzialmente visibile (per via ad esempio di un buffer errato) il contenuto non potrà mai esserne letto.

7.4.1.1 TOOL PER LA STEGANOGRAFIA LSB

Il mondo della steganografia informatica offre diversi tools, vediamone alcuni:

- SilentEye (silenteye.v1kings.io) è forse il miglior tool ad interfaccia grafica per la steganografia, disponibile per Windows, Mac e Linux. *Free*.
- OpenPuff (embeddedsw.net/OpenPuff_Steganography_Home.html per Windows) è un gran bel tool che offre, oltre la steganografia in diversi formati immagini/video/audio/flash anche cifratura a chiavi 256 bit. Al suo interno troviamo anche un algoritmo randomico che si basa sull'hardware dell'utente. *Opensource*.
- Outguess (www.rbcafe.com/softwares/outguess/ per macOS) permette di nascondere messaggi in immagini JPG. *Free*.
- iSteg (www.hanynet.com/isteg/ per macOS) è la GUI di outguess 2.0 che permette di nascondere messaggi in immagini. *Opensource*.
- Camouflage (camouflage.unfiction.com per Windows) permette la steganografia all'intero di immagini e anche in file Word. Permette la cifratura dei messaggi. Purtroppo il progetto risulta abbandonato. *Free*.

- Outguess Rebirth (www.outguess-rebirth.com per Windows) permette steganografia in immagini. Può essere trasportato in memorie esterne e offre opzioni di cifratura. *Opensource*.
- MP3stego (www.petitcolas.net/steganography/mp3stego/ per Windows) permette di nascondere messaggi all'interno di file audio mp3. Il suo sviluppo risulta però abbandonato al 2006. *Opensource*.
- QuickStego (quickcrypto.com/free-steganography-software.html per Windows) è un semplice programma in grado di nascondere messaggi dentro un'immagine con output solo in .bmp. *Free*.

Alla lista aggiungiamo StegHide, che impareremo a scoprire e ad utilizzare nelle prossime righe.

7.4.1.2 STEGHIDE

StegHide è un pratico tool sviluppato per Windows e Linux il cui ultimo rilascio risale al 2003. Sebbene esistano alternative decisamente migliori e più aggiornate - come SilentEye - StegHide rimane un buon tool per operare in un ambiente di test. È stato preferito in questo documento in quanto più semplice da installare in ambiente GNU/Linux rispetto alla controparte SilentEye, anch'essa comunque poco aggiornata ultimamente (specie nella sua versione Debian).

L'**installazione** su macchine Debian si esegue tramite il semplice comando:

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

Ipotizziamo ora di voler inserire il testo "Ciao a tutti" all'interno di un'immagine, chiamata klimt.jpg . Per prima cosa creiamo il file testo.txt con il comando:

```
$ nano testo.txt
```

Salviamo il file con *CTRL+X*, confermiamo con il *tasto S* e clicchiamo *INVIO*. Lanciamo ora il programma steghide in questo modo:

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

Cerchiamo di capire brevemente cosa abbiamo fatto:

- **steghide**, in questa parte abbiamo identificato il programma da evocare, appunto steghide
- **embed**, con questo parametro abbiamo indicato al programma di effettuare un processo di inserimento
- **-ef**, questa opzione serve a specificare il nome e la directory del file che vogliamo inserire
- **-cf**, questa opzione serve a specificare il nome e la directory del file che vogliamo contenga il testo

Lanciando il comando ci verrà chiesto di inserire una passphrase da utilizzare per mettere al sicuro i nostri dati. Non tiriamoci indietro a questa richiesta e proseguiamo. Dopo un breve istante, l'immagine verrà manipolata e conterrà al interno il testo selezionato.

Il processo *inverso*, quindi di estrapolazione delle informazioni, si eseguirà con il comando:

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

Dove:

- **steghide**, anche qui richiamiamo il programma da usare
- **extract**, qui definiamo il tipo di lavoro da fare, ovvero estrarre
- **-sf**, qui specifichiamo nome e directory del file da cui estrarre i dati
- **-xf**, qui specifichiamo nome e directory del file che conterrà i contenuti estratti

Confrontando le due immagini a prima vista è impressionante come sia visivamente impossibile non notare la differenza, non trovi? Il programma permette anche di modificare impostazioni come il tipo di cifratura, la

compressione e molti altri valori. Tutta la documentazione è ben spiegata nel comando:

```
$ man steghide
```

7.4.2 Steganografia a Generazione di Copertura

Un secondo approccio meno popolare ma comunque efficiente è detto **a generazione di copertura**: questo sistema si basa sull'inserire delle informazioni all'interno di un testo lungo in cui non si sospetterebbe mai un messaggio.
Se avete mai visto *Il Silenzio degli Innocenti* vi ricorderete di come Buffalo Bill riusciva ad inviare messaggi a Hannibal Lecter scrivendo lettere a un quotidiano e utilizzando certe posizioni delle parole per nasconderne il contenuto.

Per questo servizio uno dei più autorevoli è sicuramente spammimic.com: questo sito web permette di utilizzare alcuni algoritmi di cifratura, alcuni molto interessanti e altri meno.

7.4.2.1 STEGANOGRAFIA PURA CON METODO SPAM

Tramite questo metodo è possibile nascondere un messaggio all'interno di un falso messaggio di spam. Inviandolo al nostro destinatario, chiunque tracciasse la connessione penserebbe al solito messaggio di spam. Considera questo lunghissimo esempio:

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 ligitimate

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 .

Se noi ora lo decifrassimo otterremmo il seguente messaggio:

```
Ciao a tutti i lettori da Stefano Novelli!
```

L'avresti mai detto? Questo metodo può essere soggetto a bruteforce (soprattutto dopo che spammimic genera sempre la stessa posizione) tuttavia si può anche utilizzare una password[1] che si occuperà di modificarne le posizioni rendendo più difficile l'attacco a chi vuole scoprire cosa diciamo.

7.4.2.2 STEGANOGRAFIA PURA CON METODO PGP

Anche in questo caso verrà generato un messaggio fuorviante, sembrerà infatti che stiamo inviando o ricevendo messaggi cifrati in OpenPGP:

```
-----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-----
```

Esiste poi il metodo fake russian (che personalmente ritengo inutile, a meno che non si utilizzi un messaggio già cifrato) e il metodo degli spazi, che a differenza del primo già presento, fa uso del numero di spazi per decidere quale carattere mostrare. Ovviamente si può integrare a questo metodo la classica cifratura in PGP, già vista nel capitolo precedente per aumentare ancora di più la protezione dei messaggi che vengono inviati e ricevuti.

[1] www.spammimic.com/encodepw.shtml

7.5 Backup dei Dati

Fino ad ora abbiamo parlato di come nascondere i nostri files. Adesso è arrivato il momento di capire in che modo salvare tutto ciò che riguarda i nostri lavori, vita privata, documenti top-secret e via dicendo. Considera questo capitolo molto importante in quanto la tua vita digitale potrebbe dipendere dal successo o dal fallimento di come applicherai tutto quello che sto per dirti.

Il Backup dei dati è un processo fondamentale per evitare che quello su cui hai lavorato vada in fumo per un guasto fisico, un crash o un errore che non dovevi compiere. È un processo che deve diventare parte della tua checklist quotidiana, un qualcosa che devi fare *ogni singolo giorno*. Non vedere il formato digitale come una cosa indistruttibile: la tecnologia dei dati si basa su piccolissime frequenze magnetiche che possono essere sollecitate da agenti esterni in ogni momento; basta una scossa, un temporale o un semplice colpo per mandare tutto in frantumi. Un piccolo aneddoto che riguarda il sottoscritto:

Avevo poco più di 18 anni quando iniziai ad avere i miei primi clienti. Uno di questi aveva un e-commerce di abiti che aveva bisogno di un restyling dell'intero portale. La quota pattuita non era stratosferica ma ero giovane e quelle poche centinaia di euro m'avrebbero fatto molto comodo. Lavorai a quel layout per due mesi; bozze, controbozze, studi del templating, eventi Javascript curati ed efficienti e una struttura CSS che avrebbe fatto invidia anche agli sviluppatori del CMS. La sera stessa andai ad effettuare una pulizia fisica del computer, pensai che tutta quella polvere non faceva bene al mio PC. Appoggiai tutti i componenti accuratamente sulla scrivania, ad eccezione dell'Hard Disk principale che, chissà per quale motivo, avevo lasciato sul piano del case. Senza rendermene conto piegai il case giusto un paio di cm per arrivare in un punto ostico, l'HDD scivolò dalla superficie metallica e cadde. Fece un salto di appena 40 cm e finì sul pavimento.

Hard Disk completamente morto. Lo ricollegai subito al SATA e sentivo che il disco

meccanico girava ma per qualche strana ragione il pennino continuava a ticchettare a intermittenza. Niente da fare, provai a smontarlo per cercare un modo disperato di farlo ripartire, senza contare che non avevo idea - e ad essere sincero neanche oggi ne ho - di come rimetterlo in asse. Hard Disk morto e sepolto.

A quell'evento seguirono scenate di odio al mondo intero per un'ora e mezza per poi arrivare alla fase della disperazione. Ci vollero circa 2-3 ore prima che mi riprendessi e ricominciassi da zero tutto il lavoro. Da zero. Da capo

Ancora oggi mi chiedo come siano potuti bastare quei 40 cm per danneggiare irreparabilmente - per noi comuni mortali - un HDD così massiccio. La sera stessa uscii con dei miei amici: portai con me l'HDD come segno di disgrazia per dimostrare a tutti l'assurdità della cosa (non so, far vedere loro che non c'era neanche un graffio sulla scocca e cose così). Tra i fiumi dell'alcool decidemmo di distruggere a mano quell'HDD con ogni mezzo. Le tecniche furono diverse: dato fuoco con l'alcool e colpito ripetutamente con una vanga, trascinato sull'asfalto, schiacciato (sempre con l'auto), circondato da petardi e fatto detonare. Neanche lanciarlo da un'auto in corsa a 100km/h e centrare il palo di uno stop riuscì ad aprirlo. Ancora oggi, se si passa per quella strada, c'è il segnale con una piega sul tubo di ferro.

7.5.1 Quanti backup servono?

Per avere sempre tutto sotto controllo sarebbero necessari almeno due dischi di Backup, dislocati possibilmente in due aree diverse. Il primo potrebbe essere lasciato anche all'interno del PC o magari come HDD esterno, il secondo dentro l'auto, a lavoro, a casa di un amico/familiare. Se succedesse qualcosa - qualunque cosa - in uno dei due posti l'altro HDD sarebbe comunque salvo.

Effettua un backup quando fai un nuovo dump, effettua un backup quando hai nuovi log su cui lavorare, effettua un backup quando crei un nuovo wallet bitcoin. Effettua SEMPRE un backup per qualunque cosa reputi anche solo

minimamente importante. Non prenderla come una fissa, certo, ma ricorda che maggior tempo dedicherai alla tua vita digitale maggiori saranno le perdite che potresti avere, sia esso tempo, soldi, lavoro o quant'altro.

Pensa se magari un giorno - o forse è già quel giorno - avrai centinaia se non migliaia di euro su un wallet. Di colpo… puff! Tutto svanito. A chi darai la colpa? Non lesinare sull'acquisto di una o più memorie aggiuntive: se necessario, dividi i tuoi lavori per più memorie (così da avere anche una buona archiviazione che ti permetterebbe una più veloce ricerca dei tuoi dati).

7.5.2 Rsync

Ritengo che ognuno abbia il proprio metodo per organizzarsi i file: inutile quindi fare una lista dei migliori programmi per questo o quel Sistema Operativo, se Cronologia File o Backup e Ripristino per Windows oppure Time Machine per macOS sono adeguati per delle copie di sicurezza.

Ciò che possiamo ritenere il tool più adatto per questo tipo di operazioni si chiama **rsync**, disponibile per tutti i Sistemi Operativi a base UNIX e anche per Windows tramite installazione terza con cygwin. Questo software ha diversi vantaggi rispetto alle controparti disponibili: è ormai usato per consuetudine tra gli amministratori di sistema, dunque molto documentato, offre un algoritmo davvero efficace per la copia di file anche tramite protocollo ssh (per la copia in remoto) e la possibilità di comprimere al volo in diversi formati.

7.5.2.1 INSTALLAZIONE DI RSYNC

È possibile trovare Rsync già in **macOS** a partire dalla versione 10.4, oltre ovviamente in quasi tutte le distribuzioni a base **GNU/Linux**, Debian compresa. Se per qualche ragione non dovesse essere presente si potrà procedere alla sua installazione digitando da terminale:

```
$ sudo apt-get install rsync
```

Per **Windows** invece si potrà far riferimento al programma Cygwin[1] che permette di installare anche la maggior parte dei tool già esistenti in Linux) oppure cwRsync[2].

7.5.2.2 COPIA IL LOCALE CON RSYNC

La grande versatilità di rsync è documentata nell'eccellente lista dei parametri che è in grado di supportare. Possiamo averne una lista lanciando il solito parametro --help:

```
$ rsync --help
```

oppure il man:

```
$ man rsync
```

Prima di familiarizzare con le sue funzionalità è bene tener conto della struttura di copia. Rsync gestisce gli input e output esattamente come il tool cp in Linux, quindi tratterà il primo valore come l'elemento (o gli elementi) da copiare, mentre il secondo come il percorso di destinazione:

```
$ rsync [filedacopiare] [destinazionedicopia]
```

[1] cygwin.com/

[2] https://www.itefix.net/cwrsync

Facciamo ora pratica con i parametri.

Ipotizziamo di voler **copiare un file presente dalla cartella1 alla cartella2** presente nella home del nostro utente:

```
$ rsync -a $HOME/rsync $HOME/rsync_backup
```

Inutile ricordare che [nomeutente] andrà sostituito con il nick dell'utente attuale, giusto? In questo caso abbiamo usato un parametro: **-a**. A cosa serve?
Il parametro -a si occupa di copiare **ricorsivamente** tutti i file - quindi anche quelli dentro le cartelle - mantenendo la struttura originale, i permessi e altre informazioni.

Vorremmo però decidere di **comprimere on-the-fly il contenuto della cartella**: perché non usare il parametro -z ?

```
$ rsync -az $HOME/rsync $HOME/rsync_backup
```

Come abbiamo visto abbiamo usato -az, quindi abbiamo abbinato i parametri -a e -z. In questo caso non solo l'operazione sarà ricorsiva ma i file verranno compressi prima di raggiungere la destinazione, quindi verranno estratti in locale. Questa funzione potrà ritornare utile per grandi quantitativi di dati.

L'uso di rsync non ha limiti di alcun genere: puoi provare a sperimentare copiando i tuoi file direttamente nella tua memoria esterna:

```
$ rsync -az /home/[nomeutente]/cartella1 /media/
[nomeutente]/[nomepartizione]
```

7.5.2.3 COPIA IL REMOTO CON RSYNC

Avanzando nel mondo dell'informatica presto o tardi potrai decidere di affittare un Server Dedicato, una VPS o avere una macchina di tua proprietà in remoto.
Non starò qui a dirti come configurare un Server per accettare connessioni di tipo

SSH, spero tu lo sappia già fare: se così non fosse, puoi documentarti online o affittare un Server o una VPS per iniziare a sperimentare rsync anche nella rete.

Il riconoscimento del protocollo di rete viene fatto in automatico anteponendo alla destinazione i dati di login della macchina e il suo host, seguito dai due punti. Se ad esempio vogliamo **copiare dei dati da remoto al nostro computer locale** useremo:

```
$ rsync -a [utente@host]:/cartella1 /home/[nomeutente]/
cartella2
```

Qui [utente@host] prende il valore dei dati di login assieme all'indirizzo IP della macchina o il suo dominio. All'occorrenza rsync si occuperà di chiederci la password d'accesso tramite SSH.

Se per motivi di sicurezza abbiamo **modificato la porta** del nostro server (che di default è una TCP 22) a un'altra dobbiamo poterlo dire a rsync. In questo caso il parametro è leggermente più arzigogolato ma comunque facilmente applicabile:

```
$ rsync -a --rsh="ssh -p PORT" [utente@host]:/
cartella1 /home/[nomeutente]/cartella2
```

Rsync di default non è in grado di mostrare lo stato d'avanzamento della copia. Questo può essere un problema, soprattutto se non conosciamo a menadito la dimensione dei file da copiare e la velocità di trasferimento. Per **conoscere il tempo rimanente di copia** usiamo il parametro --progress:

```
$ rsync -a --progress [utente@host]:/cartella1 /home/
[nomeutente]/cartella2
```

Se invece siamo soliti effettuare backup in remoto a loro volta presenti nelle directory ma non è nostra intenzione scaricarli ogni volta possiamo decidere di specificare la **dimensione massima** (e all'occorrenza anche **minima**) dei file che

stiamo per processare. I parametri sono --max-size e --min-size come nell'esempio prossimo:

```
$ rsync -a --max-size=10M [utente@host]:/cartella1 /
home/[nomeutente]/cartella2
```

In questo modo i file più grandi di 10 Megabyte verranno ignorati.

7.6 Cold Boot RAM Extraction

Se hai già utilizzato distribuzioni *GNU/Linux* pensate per l'anonimato o il pentesting avrai sicuramente notato la presenza di tools o di modalità pensate per la prevenzione di attacchi a livello RAM. Ok, un passo indietro.

RAM sta per Random Access Memory, vale a dire quella memoria estremamente veloce utilizzata dai sistemi operativi e dalle applicazioni per fornire dei valori al processore, il quale si occuperà di manipolarli e rigirarli alle varie risorse. La RAM è la memoria più veloce presente in un computer, questo perché non si preoccupa di ordinare i dati al suo interno ma solo di allocarli temporaneamente nel computer; al termine dell'utilizzo del computer, la memoria RAM perderà tutti i suoi dati. Se una memoria RAM si riempie (a differenza di un HDD/SSD) il sistema continuerà a scrivere e leggere sovrascrivendo i vecchi dati.

Nella RAM è possibile trovare dei dati temporanei, ad esempio lavorando in un file Word i salvataggi non ancora memorizzati vengono temporaneamente salvati lì. A differenza di altri tipi di memoria ROM, la memoria RAM non viene cifrata in alcun modo. Nell'informatica la tipologia più comune di memoria RAM è la **DRAM** (acronimo di Dynamic Random Access Memory). Questa memoria, a differenza della SRAM (che sta per Static R.A.M.), ha un'architettura tale da permettere al

sistema che lo circonda di ripulire i settori della memoria in breve tempo e quindi di aggiungere nuovi elementi.

All'interno della DRAM ci sono dei sottolivelli chiamati DDR (vi dice niente?). Se bazzicate un po' nell'assemblaggio dei computer saprete che attualmente le memorie RAM hanno raggiunto la DDR4, tuttavia non è raro imbattersi in versioni DDR3 o addirittura DDR2. Come abbiamo detto, quando un computer si spegne la sua memoria RAM viene cancellata. La domanda è: *in che modo viene cancellata?*

7.6.1 Come si effettua il CBRE

Quella che segue è una ricerca condotta nel Luglio 2008 a San Jose da un gruppo di ricercatori dell'Università di Prenceton, della Electronic Frontier Foundation e della Wind River Systems che hanno presentato al simposio dell'USENIX Security una dimostrazione[1] di come sia possibile effettuare un'estrazione dei dati in memoria RAM anche a distanza di qualche minuto dallo spegnimento di un computer, addirittura rimuovendo la memoria RAM dalla motherboard (Figura 29).

Sempre in base alla ricerca, le DRAM non si cancellano subito, dando così il tempo necessario a chiunque di effettuare acquisizioni di natura forense sull'ultimo stato del sistema operativo. Tale tecnica è stata dimostrata effettuando con successo un recupero delle chiavi di cifratura di alcuni dei software più famosi nel panorama informatico (tra cui *BitLocker*, *TrueCrypt* e *FileVault*) rivelando come non sia stato necessario l'uso di particolari strumenti. Nella ricerca emerge anche di come in ambiente OSX sia stato possibile recuperare le login password dell'utente o di chiavi private RSA di un web server Apache.

[1] citpsite.s3-website-us-east-1.amazonaws.com/oldsite-htdocs/pub/coldboot.pdf

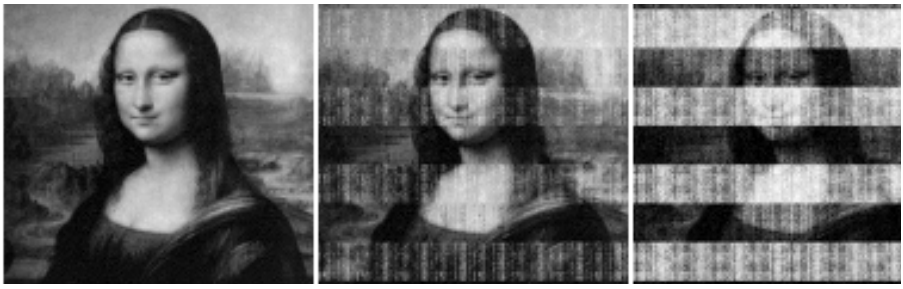


Figura 29: Ecco come si presenta la degradazione di un'immagine in una memoria RAM. Nell'ordine da sinistra verso destra: 5 secondi, 30 secondi, 60 secondi e 5 minuti.

Il seguente metodo non verrà spiegato nel presente corso in quanto richiede, oltre a conoscenze di reversing avanzate, anche la possibile distruzione delle memorie RAM. Vi basti sapere che il termine "a freddo" deriva dalla tecnica che si usa per l'estrazione: rovesciando un nebulizzatore di spray si porterà a -50°C la temperatura della RAM (Figura 30), in questo modo i dati contenuti in essa persisteranno per diversi minuti fino a che rimane della corrente statica all'interno della memoria.

Figura 30: Dimostrazione del metodo Cold Boot RAM Extraction

7.7 Metadata & EXIF Data

Nel campo dell'informatica i metadati sono elementi presenti all'interno dei file, normalmente non visibili agli occhi dell'utente finale, che contengono informazioni di varia natura per consentire ai programmi con i quali va ad interagire di funzionare in maniera ottimale. I metadati possono dire molto sulla

tua identità e sono rintracciabili in diversi formati: *fotografici, documenti, video etc…*

Nel mondo dell'informatica è famosa la storia di *w0rmer*, nome in codice di *Higinio O. Ochoa III*, proclamatosi come uno dei membri del movimento Anonymous che hanno violato il sito web delle forze dell'ordine degli USA; è stato riconosciuto da una foto pubblicata dalla sua ragazza che citava "PwNd bu w0rmer & CabinCr3w <3 u BiTch's!". L'FBI in quella situazione riuscì a risalire all'identità della ragazza tramite i metadati della foto (identificati più avanti come EXIF Data).

Una delle risorse che ha reso grande l'informatica è indubbiamente quella delle immagini. Oggi siamo abituati a diversi formati (*JPG, PNG, TIFF* e via dicendo) ognuno dei quali ha le sue peculiarità ed è quindi adatto a diverse situazioni.

Gli **EXIF Data** sono dei metadati presenti nei formati media (immagini e anche alcuni video) che rivelano informazioni aggiuntive davvero interessanti: è possibile risalire al codice univoco del dispositivo che ha scattato l'immagine (*l'ID machine*), marca e modello, orario, risoluzione e se presente persino le coordinate GPS.

7.7.1 Come visualizzare gli EXIF Data

I visualizzatori immagine di ultima generazione preinstallati nei Sistemi Operativi sono in grado di mostrare a video i metadati delle immagini in vari formati. Nel caso di Debian con installato GNOME 3, il visualizzatore immagini ufficiale conterrà una barra laterale che mostrerà di default i Metadati raccolti da un'immagine. Se questa non fosse presente è possibile attivarla dal menù *Visualizza -> Barra Laterale* (o premendo la combinazione *CTRL+F9*).

7.7.1.1 MAT: METADATA ANONYMISATION TOOLKIT

Nel panorama informatico uno dei programmi più popolari per la gestione dei metadati è sicuramente MAT: Metadata Anonymisation Toolkit[1]. Questo tool è disponibile preinstallato in diverse distribuzioni GNU/Linux e disponibile nella maggior parte dei repository: sono inoltre disponibili i repo git[2] e i source stabili[3].

È possibile *installarlo su Debian* con il comando:

```
$ sudo apt-get install mat
```

MAT è in grado di gestire diversi formati ed è disponibile nelle versioni CLI e più comunemente in versione GUI. Tale programma permette di inserire in una lista uno o più file, quindi doppio-cliccandoli si potrà accedere al fingerprint dei metadati (Figura 31).

Figura 31: Dettagli di un'immagine di test con MAT

[1] https://mat.boum.org

[2] https://gitweb.torproject.org/user/jvoisin/mat.git

[3] https://mat.boum.org/files/

In questo esempio è possibile vedere molte informazioni che riguardano la foto scattata, comprese le coordinate GPS, la risoluzione, la ISO, il modello dello smartphone e così via.

MAT offre anche una comoda funzione per **rimuovere i Metadati**; tale funzione è attivabile cliccando sul pulsante "Scour".

Perché non provi con qualche tua foto? Prova utilizzando una tua fotocamera/smartphone, quindi riprovaci con un'immagine online. Puoi provare anche con altri tipi di estensioni o addirittura tipi di file diversi.

NB: è possibile che, testando immagini provenienti da Internet (e soprattutto Social Network) capiti che i Metadati non vengano letti. Questo può essere causato dal codice di upload del sito che potrebbe ricomprimere l'immagine per formato e risoluzione così da risparmiare spazio sui propri server e banda in esterno. Tuttavia considera sempre che qualunque di questi servizi **potrebbe memorizzare i file originali da te caricati**.

> **Tip:** Vuoi cancellare velocemente gli EXIF Data da una JPG? Converti la tua immagine in formato .PNG! Questo formato non ha il supporto standard agli EXIF Data.

7.7.1.2 SOFTWARE ALTERNATIVI PER I METADATA

Abbiamo parlato solo di MAT, questo perché è opensource ed è abbastanza affidabile per i nostri scopi. Tuttavia sono presenti anche programmi alternativi in grado di lavorare sui Metadati, di seguito ne listeremo alcuni con una breve descrizione delle loro funzioni:

- Free Photo Viewer[1] (*Windows*) - FPV è un programma che permette di estrarre informazioni per immagini in formato JPEG e RAW. Consente di

[1] www.exifsoftware.com/free-photo-viewer/

estrarre anche informazioni come apertura, valore ISO, lunghezza focale, data e ora, impostazioni sul flash e via dicendo. Al suo interno troviamo un semplice organizzatore di immagini.

- IrfanView[1] (*Windows/OSX[2]/Linux[3]*) - Disponibile sia a 32 che a 64 bit è uno dei programmi più antichi che fa questo mestiere. Apre un'infinità di estensioni (anche MP3, EPS, PSD, SWF e via dicendo) ed è espandibile mediante l'uso di plugins.

- *Photo* (*OSX*) - In Italiano Foto, è un'applicazione integrata nei Sistemi Operativi Apple. All'apertura di una qualunque foto basta usare la shortcut *cmd+i* oppure click destro -> Ottieni Info. È possibile aggiungere metadati custom come volti, descrizione e keywords ma non modificare quelli attuali.

- *Visualizzatore Immagini* (*Windows*) - VIW è integrata in tutti i Sistemi Operativi firmati Microsoft. Per accedere alle proprietà immagine fare *click destro -> Proprietà -> Tab Riepilogo*.

- ExifPilot[4] (*Windows/OSX/Linux*): tool da linea di comando sviluppato in PERL. Permette l'apertura di qualunque tipo di Metadata.

- GeoSetter[5] (*Windows*): personalmente lo ritengo uno dei migliori tools in circolazione. Purtroppo è solo per Windows ma può fare cose fantastiche: oltre ad aprire un'infinità di estensioni digitali permette di modificare le coordinate geografiche (altitudine compresa), valori IPTC e molto altro. È sicuramente uno dei migliori tools per modificare gli EXIF Data in quanto permette di manipolarli in modo da sembrare convincenti (anziché insabbiarli).

---

[1] www.irfanview.com

[2] Tramite WineBottler

[3] Tramite Wine

[4] www.sno.phy.queensu.ca/~phil/exiftool/

[5] www.geosetter.de/en

- ExifEditorApp[1] (*OSX*): app disponibile per gli OS Apple, permette di modificare i metadata EXIF e IPTC.
- ExifDateChange[2] (*Windows*): tool esclusivamente per l'OS Microsoft, è disponibile in versione sia free che a pagamento. Comodo in quanto disponibile in versione portable.

Ovviamente la lista non finisce qui, ce ne sono molti altri come Batch Purifier LITE[3], EXIFCleaner[4], PhotoME[5] e molti altri. Basta cercare! :)

Prima di procedere è doveroso ricordare che rimuovere i Metadati *non è la soluzione finale* a tutti i nostri problemi: i file su cui operiamo potrebbero essere manipolati tramite steganografia, watermarks e altri tipi di metadati non standard. Inoltre alcuni dei programmi di cui parleremo permettono di gestire solo la parte superficiale dei metadati: ad esempio non sarà possibile modificarne i valori di un'immagine all'interno di un PDF. È possibile inoltre **prevenire** buona parte dei metadati presenti nei documenti di testo utilizzando semplici formati testo (i cosiddetti plain-text o più comunemente conosciuti come .txt). Se ne hai bisogno usali!

[1] exifeditorapp.com

[2] https://www.relliksoftware.com/exifdatechanger/download

[3] www.digitalconfidence.com/downloads.html

[4] www.superutils.com/products/exifcleaner/

[5] www.photome.de

7.8 Sensori delle Fotocamere

Questo breve capitolo vuole metterti in guardia su una nuova pratica che grandi aziende di *data mining* stanno applicando a livello globale sulla rete. Dovresti sapere che ogni sensore fotografico rilascia una *firma unica* quasi impercettibile a causa delle minime differenze hardware di cui sono composte.

Come per un proiettile che permette di risalire all'arma che ha sparato, una fotografia può permettere di risalire alla fotocamera che l'ha scattata. Da notare come questo non ha nulla a che vedere con gli *EXIF Data* che compiono un lavoro totalmente diverso. Qualora si volesse approfondire l'argomento è disponibile in rete un documento redatto da tre ricercatori presso il Dipartimento di Ingegneria Elettronica e Computer di New York[1].

Purtroppo al momento non esiste un metodo veloce e riconosciuto in grado di offuscare questo tipo di informazioni: la manipolazione digitale - fatta con i giusti strumenti, dovrebbe garantire comunque una buona eliminazione delle tracce, ad esempio modificandone i livelli di colore, la saturazione, il contrasto, la nitidezza, la struttura e via dicendo. Dallo studio è comunque emerso un calo delle possibilità di ricerca nelle fotografia sovra-esposte (pag. 7 della ricerca "*Sensor Noise Camera Identification: Countering Counter-Forensics*").

[1] ws2.binghamton.edu/fridrich/Research/EI7541-29.pdf

7.9 Data Shredding

Arriverà il giorno in cui non avremo più bisogno di un file, criptato o decriptato che sia. Quando arriverà quel giorno non lasciate che venga semplicemente cestinato: il file in realtà rimarrà lì, o quantomeno lascerà alcune tracce della sua presenza, e si, potrebbe essere possibile recuperarlo. In questo capitolo ci occuperemo di tutti i metodi conosciuti per distruggere completamente ogni prova presente all'interno del nostro computer e in particolare alle memorie ROM che immagazzinano i nostri dati.

7.9.1 Come effettuare il Data Shredding

Quando è stata scoperta la possibilità di poter recuperare file cancellati sul proprio PC sono spuntati come funghi decine di software - commerciali e gratuiti - che hanno promesso di risolvere questo problema.

Al momento possiamo suddividere i tools in tre grandi categorie:

- *Disk Cleaner*
- *File Wipers*
- *Distruzione fisica del drive*

7.9.1.1 DISK CLEANER

In questa categoria rientrano quei software che utilizzano vari metodi per effettuare una sanificazione dell'Hard Disk. In sostanza si occupano di liberare quei settori del drive che contengono ancora informazioni su dati fantasma (una specie di reminiscenza in memoria) e che verranno utilizzati dal sistema operativo solo quando non sarà disponibile più spazio all'interno dell'Hard Disk.

L'affidabilità dei Disk Cleaner è messa tuttavia in discussione da molti esperti del settore in quanto utilizzerebbero tecniche talvolta troppo blande col fine di vincere le "gare dei benchmark in velocità", inoltre molti dei software che

compiono questo lavoro sono soliti lasciare tracce del loro passaggio all'interno dei log proprietari o del sistema operativo stesso.

BleachBit

BleachBit è un programma opensource che si occupa di ripulire spazio disco, ottimizzare le prestazioni del computer e garantire una migliore privacy dell'utente. Disponibile per *Windows*, *macOS e GNU/Linux*, Bleachbit riesce a garantire ciò che promette fornendo strumenti volti a rimuovere cache, cookie, cronologia e log dei principali browser, integrando inoltre una piacevole funzione volta alla verifica e alla riscrittura dello spazio su disco non allocato, argomento che verrà trattato a breve nel capitolo "File Shredding".

Installare il programma su Debian è ovviamente un gioco da ragazzi:

```
$ su
$ apt-get install bleachbit
```

Se questo tuttavia non dovesse per qualche ragione essere disponibile tra i repository del Sistema Operativo si potrà scaricare e installare direttamente il pacchetto appropriato dal sito ufficiale[1]. Considerata l'estrema semplicità d'uso di questo programma (che per la cronaca richiede due click di numero) non lo approfondiremo ulteriormente.

Altri software di Disk Cleaning

In rete possiamo trovare molti altri tool pensati per il Disk Cleaning. Di seguito una lista di quelli che mi sento di consigliarti:

- CCleaner[2] per *Windows, macOS e Android*

[1] https://www.bleachbit.org/download

[2] www.piriform.com/ccleaner

- Glary Utilities[1] per *Windows e Android*
- Clean Master[2]

7.9.1.2 FILE SHREDDING

Il *File Shredding* è una pratica che fa fronte a questa situazione in modo più diretto: il suo funzionamento si basa sul riscrivere nella posizione in memoria precedentemente allocata dal file pre-esistente dei byte random. Più volte si riscriverà su quella posizione, maggiori saranno le probabilità che le informazioni del file originale scompaiano per sempre.

Ci sono molti punti di vista circa il numero di volte da seguire per una corretta eliminazione tramite File Shredding: l'NSA ad esempio ne raccomanda 3, il Dipartimento della Difesa 7 e Peter Gutmann (inventore del metodo Gutmann[3]) nel suo schema più famoso ne dimostra ben 35. Ognuno fa le sue valutazioni del caso, certo è che 35 potrebbe essere un numero sicuramente esagerato, nonostante la spiegazione fili (e ci mancherebbe!) ma in realtà potrebbero bastare 5-6 passaggi affinché la randomizzazione consenta un numero di ipotesi infinite per la ricostruzione di un file. Giusto per essere chiari, il metodo Gutmann non risulta essere più efficace ai giorni nostri in quanto i suoi studi si basano su vecchi pattern utilizzati negli Hard Disk IDE di fine anni '90.

Consideriamo inoltre che dal 2001 molti produttori di supporti di memoria si sono interessati al Data Shredding tanto da standardizzare i loro prodotti con una tecnologia definita ATA Secure Erase[4], tuttavia secondo una ricerca emersa nel 2011 sono solo la metà dei produttori mondiali che hanno adottato questa caratteristica.

---

[1] www.glarysoft.com/glary-utilities/

[2] https://www.cmcm.com/en-us/clean-master-for-pc/

[3] https://it.wikipedia.org/wiki/Metodo_Gutmann

[4] https://en.wikipedia.org/wiki/Parallel_ATA#HDD_passwords_and_security

Come effettuare il File Shredding

Effettuare il File Shredding non risulta essere una pratica particolarmente difficile, è possibile trovare davvero moltissimi tools per qualunque Sistema Operativo. Purtroppo però (a parte BleachBit[1]), nessuno è multi-piattaforma, quindi saremo costretti a riassumerne uno per ogni OS:

- CCleaner[2] ha una funzione di Drive Wiping per liberare lo spazio occupato dai file eliminati. Disponibile solo per *Windows*.
- Sempre su *Windows* sono disponibili un'infinità di tools dedicati al File Shredding: Eraser[3], Securely File Shredder[4], Freeraser[5], WipeFile[6], Secure Eraser[7] e molti altri.
- Su *Mac OS / OSX* il più affidabile sembra essere Permanent Eraser[8]
- Su *GNU/Linux* il tool più consigliato è shred[9]*;* su *Tails* è presente Nautilus Wiper[10]
- DBAN[11] (per la formattazione intera di una partizione SENZA l'uso di OS - si dovrà masterizzare una Live)

---

[1] https://www.bleachbit.org/

[2] https://www.piriform.com/ccleaner/download

[3] https://sourceforge.net/projects/eraser/

[4] www.freewarefiles.com/downloads_counter.php?programid=91261

[5] www.codyssey.com/apps/utilities/freeraser.html

[6] www.gaijin.at/en/dlwipefile.php

[7] https://www.ascomp.de/en/products/show/product/secureeraser/tab/description

[8] https://itunes.apple.com/it/app/permanent-eraser/id500541921?mt=12

[9] linux.die.net/man/1/shred

[10] https://tails.boum.org/doc/encryption_and_privacy/secure_deletion/index.en.html

[11] www.dban.org/

Shred su Linux

Nel caso si voglia shreddare un'intera partizione si può sempre utilizzare *shred*, un tool dalla cara e vecchia linea di comando. Otteniamo la lista delle nostre partizioni attive. Potremmo voler utilizzare il tool fdisk in questo modo:

```
$ sudo fdisk -l
```

così siamo sicuri che il percorso della partizione che vogliamo eliminare sia quella corretta (ipotizziamo */dev/sdb*).

È il momento di ripulire la partizione, se il tempo non è un nostro alleato potremmo voler utilizzare una *cancellazione veloce*. Questa procedura risulta essere più veloce in quanto su ogni settore scrive un valore vuoto:

```
$ sudo shred -vzn 0 /dev/sdb
```

Nel nostro caso il parametro -vzn dirà a shred:

- **v**, mostrami i progressi
- **z**, sovrascrivi l'ultimo passaggio dello shredding (per nasconderlo)
- **n**, definisci il numero di iterazioni
- **0**, avendo definito 0 iterazioni il valore sarà NULL quindi 0

Questo farà si che la partizione risulterà illegibile, in quanto i settori non conterranno più alcun valore. Se invece vogliamo essere più sicuri dei risultati possiamo sempre manipolare il tool affinché compia azioni più sofisticate. Lanciando ad esempio:

```
$ shred -vzn 3 /dev/sdb
```

Diremo al programma le stesse cose, ad eccezione che le iterazioni in questo caso saranno 3 e quindi i passaggi di riscrittura dei settori sarà triplicata, effettuando così un'eliminazione di partizione *più sicura*.

Shred è un programma ottimo anche per **cancellare singoli files**, tramite il parametro --remove come l'esempio che segue:

```
$ shred --remove [nomefile]
```

DBAN per lo standalone

DBAN[1] (acronimo di Darik's Boot and Nuke) è un tool gratuito e opensource che permette di effettuare file shredding su tutto l'hard drive. Il suo funzionamento non dipende da alcun sistema operativo in quanto lo stesso DBAN è una distribuzione basata su GNU/Linux. Per usarlo è quindi necessario un supporto esterno (CD, USB etc...) e una breve riconfigurazione del BIOS (esattamente come quando lanciamo una Live USB di Linux).

DBAN andrebbe utilizzato prima di distruggere fisicamente un drive, così da aumentare le possibilità di illeggibilità del disco. Questo strumento offre *diversi possibili algoritmi di eliminazione*:

- **Quick Erase**

  1 passaggio - Livello Sicurezza: Basso

  Questo metodo non fa altro che scrivere in ogni settore un valore vuoto (0). È consigliabile solo se la partizione andrà riscritta, ad esempio nel caso si debba reinstallarvi in esso un Sistema Operativo.

- **RCMP TSSIT OPS-II**

  8 passaggi - Livello Sicurezza: Medio

  Il Royal Canadian Mounted Police Technical Security Standard for Information Technology, Appendice OPS-II: Media Sanitazion. Questo modulo implementa un processo di randomizzazione dati.

- **DoD Short**

  3 passaggi - Livello Sicurezza: Medio

[1] www.dban.org/

Il metodo veloce utilizzato dal Dipartimento Americano della Difesa. Si basa sui passaggi 1,2 e 7 del modello 5220.22-M.

- **DoD 5220.22-M**

  7 passaggi - Livello Sicurezza: Medio

  Il metodo standard utilizzato dal Dipartimento Americano della Difesa.

- **Gutmann Wipe**

  35 passaggi - Livello Sicurezza: Alto

  Il metodo descritto da Peter Gutmann nel suo documento "Secure Deletion of Data from Magnetic and Solid-State Memory".

- **PRNG Stream**

  4/8 *passaggi - Livello Sicurezza: Medio/Alto*

  Questo metodo riempie i settori del dispositivo utilizzando un generatore numerico pseudorandom. Probabilmente è il miglior metodo nei dischi di ultima generazione poiché gli schemi di generazione variano. Questo metodo offre un livello di sicurezza medio di 4 passaggi e un livello di sicurezza alto con 8 passaggi.

Utilizzo di DBAN

L'uso di DBAN non richiede particolari abilità. Una volta montato in Live si presenterà come in Figura 32.

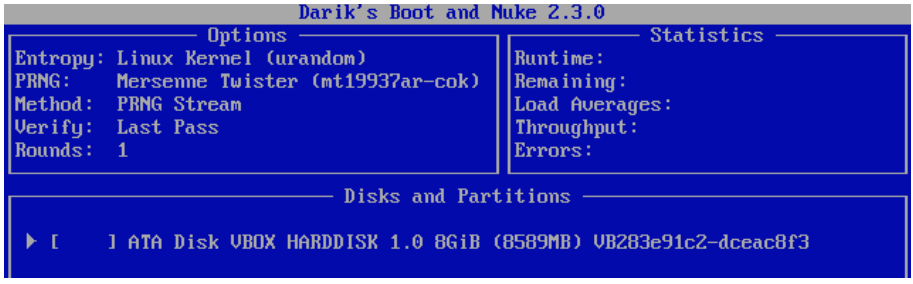


Figura 32: Schermata iniziale di DBAN

Troveremo una lista di dischi in uso nel nostro sistema, nel nostro caso è un ATA Disk (virtualizzato in Virtual Machine ma ai fini di un test andrà benissimo). In fondo alla schermata troveremo le scorciatoie da tastiera per abilitare le varie funzioni (Figura 33).

Figura 33: Scorciatoie da tastiera per navigare su DBAN

Seguiremo sempre questa legenda per muoverci nel programma. Muovendoci su e giù sceglieremo la partizione da formattare, quindi premeremo il *tasto M* per selezionare un metodo di eliminazione (uno degli algoritmi precedentemente presentati). Per una nostra prova sceglieremo *PRNG Stream* selezionandolo e premendo *Spazio*.

Il metodo PRNG Stream offre la possibilità di utilizzare lo *Pseudo Random Number Generator*, un tool esterno che si occupa esclusivamente di generare numeri (pseudo)random. Premendo il *tasto P* possiamo decidere quali dei due algoritmi usare (troverete descrizione sotto ognuno di essi). Con il *tasto V* possiamo decidere se effettuare verifica e come effettuarla: personalmente consiglio di lasciare su *Verify Last Pass* così da effettuare la verifica dell'avvenuta eliminazione solo a fine operazione. *Verification Off* disabiliterà la verifica mentre *Verify All Passes* effettuerà una verifica alla fine di ogni passaggio (rendendo molto più lunga l'operazione). Con il *tasto R* possiamo definire il numero di cicli di eliminazione. Come abbiamo detto con il metodo PRNG Stream per avere un'eliminazione ad alta sicurezza procederemo con 8 passaggi, quindi definiamolo nel programma (Figura 34).

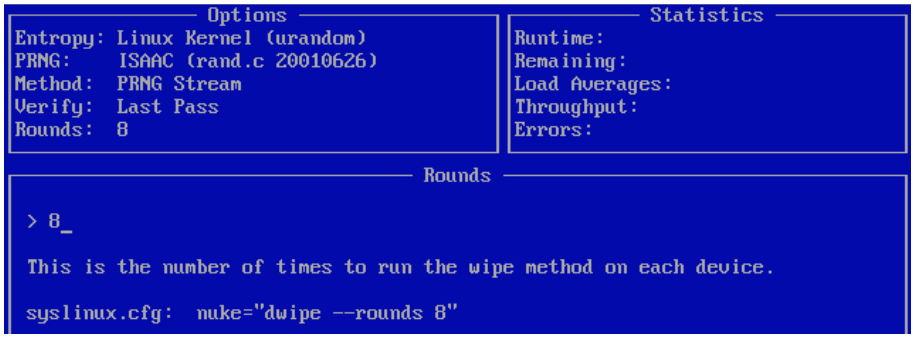


Figura 34: Definizione di 8 passaggi per lo shredding con metodo PRNG Stream

Ora dovremmo essere pronti, torniamo nella Home del programma e clicchiamo su *Spazio*. Affianco alla nostra partizione vedremo comparire *[wipe]* (Figura 35).

Figura 35: Conferma di wiping su disco tramite DBAN

Ora siamo pronti a wippare la nostra memoria. Premiamo *F10* e attendiamo la fine del processo di eliminazione.

File Shredding e SSD, cosa c'è da sapere

Se sei arrivato fino a questo punto immagino che la tua necessità sia quella di avere una memoria sempre bonificata e pronta per i riflettori dei ricercatori forensi di tutto il mondo. Quello che devi sapere è che i metodi di File Shredding sono efficaci per i dischi meccanici mentre *potrebbero* non esserlo per le SSD.

Dico potrebbero perché le variabili che determinano il successo o il fallimento del File Shredding su una SSD sono diverse: nei **dischi meccanici** i file vengono eliminati alla vista ma mantengono lo spazio occupato per non rallentare il processo di eliminazione. Quando si creano nuovi dati, il disco meccanico riscrive nei settori flaggati come "eliminati".

Nelle **SSD** quando i file vengono eliminati sarà la SSD stessa a scegliere se riscrivere o meno il settore: questa decisione spetta a un controller interno che è possibile comandare tramite un modulo chiamato TRIM (https://it.wikipedia.org/wiki/TRIM), il quale si occupa di contrassegnare i settori vuoti e di farli riutilizzare sin da subito al sistema operativo. Il TRIM viene abilitato di default nelle ultime versioni di tutti gli OS e dovrebbe garantire una riscrittura immediata dei settori appena cancellati.

Le SSD quindi hanno una logica interna che provvede, tramite il TRIM, a riscrivere quasi immediatamente un settore fungendo così da pseudo-shredder. Questa considerazione ci porta anche ad intuire che basta 1 iterazione (passaggio) di eliminazione nei dischi a stato solido. Tuttavia la logica dei tool di file shredding si sposano male con l'architettura delle SSD, quindi l'unica vera soluzione in questo senso è la pulizia totale e completa dell'intero disco (non la partizione) tramite tool come DBAN e simili. Come già accennato, basterà un solo step di eliminazione.

Un'altra soluzione potrebbe essere la cifratura del file che si vuole nascondere: cifrandolo, il file si sovrascrive su se stesso. Questo dovrebbe rendere già illeggibile la vecchia versione, mantenendo comunque l'accessibilità al file con un piccolo compromesso. Ovviamente questo può essere applicato a tutto il disco (vedesi la parte relativa alla crittografia e cifratura dei dischi).

7.9.1.3 DISTRUZIONE FISICA DEL DRIVE

In questa categoria rientrano tutte le tecniche che si possono effettuare per distruggere completamente o parzialmente un dispositivo di memorizzazione fisica. C'è da premettere che la distruzione di un disco rigido, sia di taglio da 3,5" che da 2,5", risulta essere un'operazione estremamente faticosa. Negli hard disk meccanici ad esempio ci vorranno diversi minuti per riuscire ad arrivare ai supporti magnetici, operazione spesso faticosa in termini di tempo e di sforzo.

Per le unità a stato solido (**SSD**) l'operazione potrebbe risultare più semplice, tuttavia bisognerebbe sapere esattamente dove "trapanare" tra la placca metallica e la superficie in cui alloggiano i supporti che memorizzano adeguatamente le informazioni, senza possibilità di rischi. Come è facile ipotizzare, la distruzione di un drive deve essere un'operazione estremamente veloce, tanto da poter essere attuata nell'arco di pochi secondi.
Tutti i metodi seguenti che riguardano un Hard Disk prevedono che tu sappia smontare i supporti magnetici (per quanto riguarda le unità meccaniche) e le flash memory, che assomigliano a tante microSD saldate sul circuito (per quanto riguarda le unità a stato solido ovvero le SSD).

Distruzione Meccanica

Valido per: Unità SD, CD/DVD, HDD Meccanici, SSD
Personalmente non mi sento di consigliare mazze, lanci del peso o qualunque altra cosa vi venga in mente. I rischi di lanciare intatto il disco sono elevati senza gli strumenti adeguati. In alternativa si può provare con una buona sparachiodi e riempire il disco di fori: è molto probabile che i chiodi danneggino irreparabilmente i componenti interni e comunque inserendo la memoria i rischi di causare un corto sono elevati.

Sui dischi meccanici e SSD è un'operazione che in parte può essere utile per rimuovere i componenti interni e pensare al loro smaltimento in altra sede.
Nei dischi meccanici, utilizzando un martello molto resistente - come quello da fabbro - e colpendo rovinosamente il disco tanto da deformarne la forma, causeremo una reazione di smagnetizzazione.

Sui CD/DVD basta della semplice carta abrasiva, con un paio di passate lo strato superficiale a specchio dovrebbe andarsene come cenere.

Smagnetizzazione

Valido per: HDD Meccanici, SSD, memorie USB, unità SD, CD/DVD

Il metodo Degausser - o demagnetizzazione - è il processo che consiste letteralmente di friggere l'elettronica inviando un impulso elettromagnetico (Emp) al dispositivo. Ne esistono di professionali (tra le tante una che sembra essere particolarmente affidabile è Garner[1]) oppure costruirne uno in casa (basta cercare *Create a Degausser* o *Degausser DIY*). Abbiamo anche potuto verificare con successo questo metodo delle memorie EPROM tramite un teaser costruito in casa con una racchetta anti-zanzare (anche qui troverete diversi tutorial in rete), tuttavia sconsigliamo altamente di eseguire questa operazione a meno di non avere basi di elettronica (potreste bruciare l'intera macchina!); questo metodo non è però consigliabile con i dischi meccanici. Se la memoria è una USB o una SD si può provare con il forno a microonde, anche se è più una pratica di incenerimento che di smagnetizzazione.

Incenerimento

Valido per: HDD Meccanici, SSD, memorie USB, unità SD, CD/DVD

Prima di procedere voglio mettervi in guardia: aldilà della pericolosità di infiammare qualunque oggetto, nel suolo italiano questa pratica è illegale in quanto i metalli contenuti nei dischi producono fumi altamente nocivi! Ad ogni modo è necessario raggiungere quota 1115°C, vale a dire la *Temperatura di Curie* che nel cobalto (che compone alcune parti degli HDD) causano la perdita di alcune proprietà ferromagnetiche.

Si possono utilizzare:

- un altoforno industriale, si possono trovare nelle aziende del settore siderurgico
- cannello ossiacetilenico, spesa di 100€, si raggiungono i 3000°C-3100°C

[1] www.garnerproducts.com

- altri cannelli di vario tipo, è necessario informarsi se riescono a fondere il ferro (a 1500°C)
- termite, si può comporre in casa ma altamente pericolosa, si raggiungono i 2200°C

Affogamento

Valido per: N/D

Il solo "affogare" un HDD nell'acqua non costituisce un danneggiamento fisico al dispositivo. L'acqua al massimo potrà danneggiare la scheda logica (l'insieme di controller e componenti saldati sulla PCB) che però non è difficile da sostituire.

Negli HDD meccanici lo strato superficiale del disco magnetico è costruito da leghe di cobalto che coprono vetro, alluminio e un substrato ceramico. In questo caso è necessario che l'acqua entri in contatto con l'alluminio per creare un vero danno al disco, ossidandolo. Se quindi il disco non è alimentato e il disco non sta scrivendo è impossibile che la sola acqua danneggi l'interno: forse l'unica vera strada è quella di lasciar "marinare" il disco per diversi giorni in acqua di mare. Una volta asciutto, l'acqua evaporata porterà piccole parti di sale, metalli, silicio e altre sostanze all'interno dell'elettronica. Così facendo, all'accensione del drive, questo dovrebbe danneggiarsi e danneggiare il disco magnetico. Ad ogni modo una società esperta del settore si occuperà comunque di cambiare tutta l'elettronica del disco (mantenendo solo le memorie) eliminando questi rischi.

Corrosione Chimica

Valido per: HDD Meccanici, SSD, memorie USB, unità SD, CD/DVD

Anche in questo caso ti raccomando la massima prudenza, soprattutto se non hai idea di ciò che stai facendo. Questa operazione andrebbe fatta a disco aperto, vale a dire che andrebbero tolte tutte le protezioni che coprono le memorie, o nel caso di un disco meccanico, i dischi di memoria. La soluzione più facilmente

reperibile è l'*acido cloridrico* (o acido muriatico), uno dei liquidi più corrosivi esistenti (vi ho già detto di stare attenti?) e si può acquistare in qualunque discout o ferramenta in soluzioni che variano dal 30 al 37% di concentrazione. Anche l'acido nitrico sembra essere un'ottima soluzione, sebbene più ostico da reperire (in qualunque ferramenta comunque dovrebbe esserci in concentrazioni fino al 65%): inoltre miscelandolo con un rapporto 1:3 all'acido cloridrico si ottiene la famosa acqua regia, un reagente in grado di dissolvere metalli molto resistenti come oro e platino. Ad ogni modo, l'acido andrebbe versato in un recipiente plasticoso abbastanza resistente anche al calore - per via della reazione chimica - in quantità in grado di contenere a immersione l'intero disco (magari lasciando un paio di mm di tolleranza per sicurezza) per un paio d'ore. Fai attenzione a mani e vestiti e non guardare mai direttamente senza protezioni, non respirare le esalazioni e non chiudere il contenitore per nessun motivo (potrebbe esplodere!).

8. RECUPERO DEI DATI

È arrivato il momento di testare se i metodi applicati rendono effettivamente illeggibile la memoria. Voglio ricordare che la Ricerca Forense è un settore molto complesso e trattato in maniera professionale, l'argomento sarà quindi trattato in maniera superficiale e non intende sostituirsi a ciò che ci si può aspettare da un corso avanzato sull'argomento.

8.1 Post-Mortem Forensics

In quasi tutte le operazioni di ricerca forense è necessario che l'ambiente su cui bisogna lavorare sia il più asettico possibile, senza programmi che una volta attivi in background possano modificare la natura del Sistema Operativo. Ad esempio un programmatore - anche alle prime armi - potrebbe crearsi un semplice script che cifra/decifra/nasconde/sposta/elimina file all'interno del disco, nascondendolo agli occhi di operatore e programma. Si potrebbe addirittura creare un tool in background che, al riconoscimento di un software che effettua una ricerca nel disco, lo blocca o addirittura lo inganna.

A questo punto è importante che il ricercatore forense abbia, oltre alle dovute copie di sicurezza così da evitare incidenti di percorso, anche gli strumenti adatti per effettuare il lavoro senza rischiare l'inquinamento del sistema. Ecco perché *è preferibile utilizzare un Sistema Operativo Live* contenente i tool per procedere alla ricerca nel computer; nel prosieguo del documento utilizzeremo una distribuzione contenente dei tools che si potranno comunque installare all'interno del proprio Sistema Operativo. Questo tipo di ricerca è definita **post-mortem forensics**.

8.1.1 Quale OS per la P.M. Forensics

Possiamo innanzitutto identificare due tipi di Sistemi Operativi:

- *Rescue Kit OS*
- *Forensics OS*

I *primi* sono pensati esclusivamente per il recupero di dati (con l'aggiunta di tools di partizionamento, antivirus e cose così) mentre i *secondi* sono più indicati per lavorare proprio nella navigazione di un sistema limitando al minimo i danni.

I **Rescue OS** un tempo andavano di moda ma oggi sono stati quasi tutti abbandonati: dal celebre Hiren's Boot CD[1] a Ultimate Boot CD[2] per poi passare a FalconFour's Ultimate Boot CD[3], ormai sono stati tutti lasciati al loro destino. L'unico che sembra ancora essere in via di sviluppo è SystemRescueCd[4].

Il "mercato" dei **Forensics OS** è invece più florido: oltre al fatto che molte distro di pentest integrano toolset dedicati, esistono anche interi sistemi operativi pensati esclusivamente per questa pratica. Si possono utilizzare sia le distribuzioni pensate per questo lavoro (vedremo tra poco quali sono) oppure crearsene una da sé, ad ogni modo l'importante è che l'OS non tocchi minimamente il disco su cui deve lavorare.

Sebbene questo rischio può essere limitato utilizzando un Write Blocker[5] (strumento che si interpone tra computer e Hard Disk e blocca ogni possibile alterazione del disco), è bene considerare le distribuzioni con la funzione di RAM usage. Utilizzando la *RAM mode* - solitamente indicata come opzione selezionabile in fase di boot loader - si può accedere a qualunque memoria

[1] www.hirensbootcd.org

[2] www.ultimatebootcd.com

[3] https://falconfour.wordpress.com

[4] www.system-rescue-cd.org/SystemRescueCd_Homepage

[5] https://it.wikipedia.org/wiki/Write_blocker

collegata al sistema in modalità di sola lettura, senza rischiare così di alterare i contenuti dei dischi, il ché è quasi un obbligo considerando che un Write Blocker può costare dai 500€ in su - e non tutti immagino vogliano investire una tale somma.

La più celebre in Italia è CAINE[1], una distribuzione tutta italiana basata su Ubuntu e utilizzata anche dalle forze dell'ordine, poiché fornisce risultati validi anche in un tribunale . Il suo sviluppo è diretto da Nanni Bassetti, fondatore del progetto che assieme alla community online continuano ad aggiornare la distribuzione.

> **Tip**: Per i motivi che abbiamo già spiegato utilizzeremo una distribuzione GNU/Linux pensata per il Computer Forensics. Gli utenti Windows troveranno però un altro ottimo tool capace di dare ottimi risultati: si chiama Recuva (www.piriform.com/recuva), è prodotto dalla Piriform (gli stessi di CCleaner) ed è disponibile gratuitamente online. La differenza tra un software e una distribuzione GNU/Linux sta nel tipo di approccio che si vuole avere: in questo caso parleremo di Live Forensics anziché di Post Mortem Forensics.

8.1.2 Caine OS

CAINE OS è a tutti gli effetti una distribuzione GNU/Linux pensata per funzionare in modalità Live, caricata su una USB o un DVD. In questa guida la utilizzeremo in maniera molto limitata in quanto il nostro unico scopo sarà quello di verificare l'esistenza di file e partizioni che credevamo fossero cancellate. CAINE infatti integra al suo interno anche strumenti volti alla verifica e al reporting professionale per una dimostrazione inconfutabile in un tribunale, cose che ai fini di questa parte del corso non sono necessarie.

[1] www.caine-live.net

Nelle prossime pagine useremo CAINE per testare alcuni software presenti nella distribuzione, tuttavia se ne avrete bisogno potrete installarli direttamente su Debian (o nella vostra distribuzione personale) e testarli direttamente da lì. Perderà un po' il fascino della scoperta ma è comunque un'opzione alternativa. CAINE offre anche una logica di *mount in read-only*: questo significa che non solo dovremo decidere quali partizioni montare PRIMA di poterle utilizzare ma eviteremo anche di inquinare le zone che andremo a recuperare.

8.1.2.1 TESTDISK O PHOTOREC, QUALE USARE?

TestDisk è un tool pensato per recuperare intere partizioni eliminate da un disco fisso. Oltre a questa divina possibilità offre anche il recupero di settori di boot danneggiati con filesystem FAT e NTFS e ripristinare la Master File Table delle partizioni NTFS. Il tool si presenta senza una GUI ma solo in linea di comando, questo non dovrebbe comunque spaventarci in quanto il suo utilizzo è relativamente semplice e comodo. Tuttavia lo scopo del nostro percorso è quello solo di verificare se i file presenti nell'hard disk sono stati eliminati, non ci interessa recuperare partizioni danneggiate ma essere sicuri che, una volta che un file sia stato eliminato, non lasci delle tracce visibili.

PhotoRec è un tool abbinato a TestDisk che ci permette di recuperare file, documenti, video, immagini e altro da memorie esterne o interne. La peculiarità di PhotoRec risiede nel fatto che funziona indipendentemente dal filesystem e non opera direttamente in modalità write: ciò garantisce l'integrità della memoria che si vuole testare, evitando di compiere il malsano errore di riscrivere settori nella partizione. È importante che l'unità rimanga sempre in modalità di lettura: nel caso in cui si scriva anche solo un dato nella memoria è possibile che il recupero dei dati non sia più possibile.

PhotoRec è disponibile per qualunque *sistema operativo*, tra cui: Dos/Win9x, Windows (32/64 bit), Linux (32/64 bit), OSX/macOS (Intel/PowerPC), *BSD; inoltre è disponibile in formato pacchetto con TestDisk gratuitamente sul sito ufficiale[1].

Il suo utilizzo è possibile inoltre su un'enorme varietà di *filesystem*: exFAT/FATx, NTFS, ext2/ext3/ext4, HFS+; aggiungerei a questo punto btrfs che, seppur non ufficialmente supportato, sembri funzionare abbastanza bene.

Il suo utilizzo può essere effettuato su qualunque supporto esterno standard, purché il sistema operativo lo riconosca e sia in grado di leggerne i contenuti.
È in gradi di leggere (quasi) qualunque tipo di formato, da classici JPEG/PNG/ZIP/PDF ai più rari LZO/XAR/PPM/RA fino ad arrivare a quelli proprietari come PSD/MHBD/MAX/GI e così via[2].

8.1.2.2 BREVE GUIDA ALL'USO DI PHOTOREC

PhotoRec viene distribuito in due versioni: *GUI* e *CLI*. La versione GUI è ovviamente più semplice in quanto gestisce tutto da interfaccia grafica.

Se non è preinstallato sulla tua distribuzione dovresti trovare *QPhotoRec* (la versione di PhotoRec con la GUI) tra i programmi installabili. Eventualmente possiamo **installarlo** da noi tramite il terminale:

```
$ sudo apt-get install qphotorec
```

Aspettiamo che tutto venga installato, quindi localizziamo il programma tra i tools installati; se non lo troviamo possiamo riaprire ancora il nostro terminale e digitare:

```
$ sudo qphotorec
```

Il programma si presenta come in Figura 36.

[1] www.cgsecurity.org/wiki/Download_TestDisk

[2] Lista completa su www.cgsecurity.org/wiki/File_Formats_Recovered_By_PhotoRec

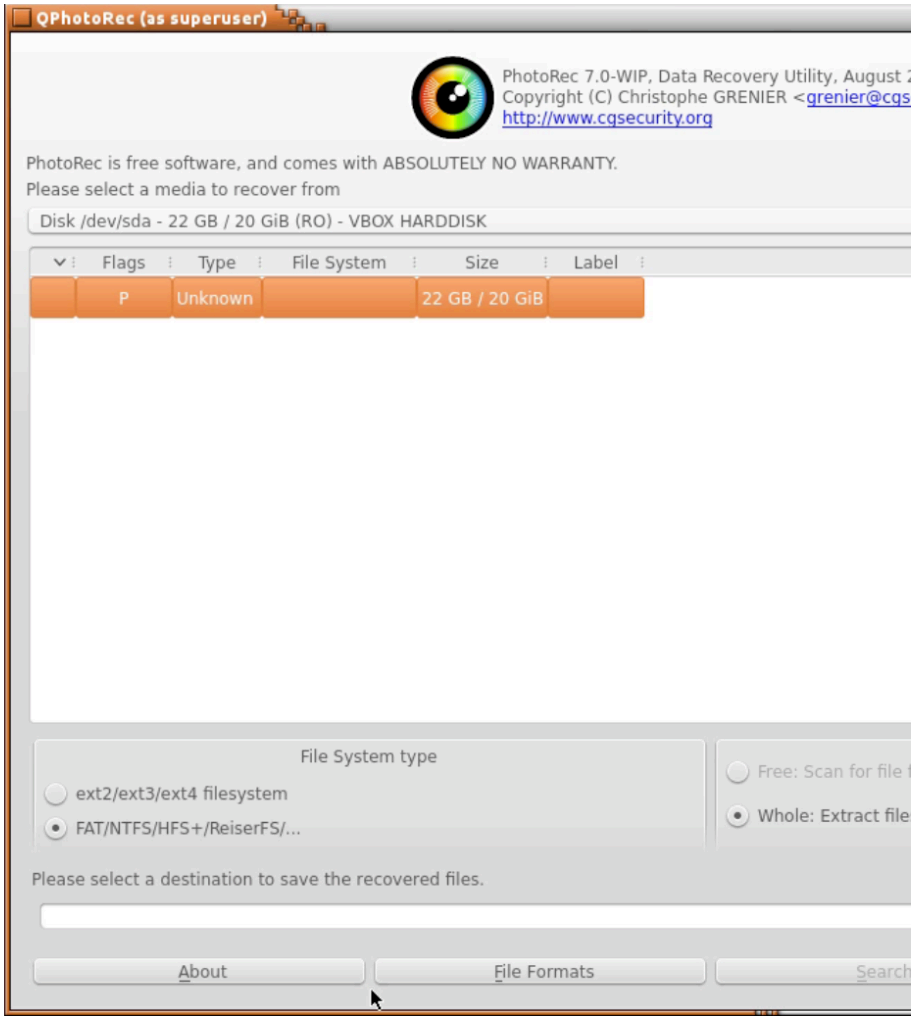


Figura 36: Schermata iniziale di QPhotorec, versione GUI

Se non riesci a vedere la partizione in cui vuoi lavorare dovrai selezionare il disco che contiene la partizione. Seleziona la *partizione* in cui vuoi lavorare, il tipo di *File System*, il tipo di scansione *Free / Whole* (nel nostro caso andrà benissimo Free) e scegli una destinazione in cui salvare i risultati col pulsante "*Browse*".

Ora non dobbiamo far altro che aspettare che il programma finisca di scansionare il drive!

Se preferisci utilizzare il *caro vecchio* **terminale** assicurati innanzitutto che sia installata l'ultima versione:

```
$ sudo apt-get install photorec
```

Se il programma è presente nel Sistema Operativo possiamo allora procedere al lancio con il comando:

```
$ sudo photorec
```

Come abbiamo visto abbiamo richiamato anche qui il *sudo* in quanto dobbiamo assicurarci che PhotoRec sia lanciato in modalità amministratore. Siamo ora di fronte a una schermata che ci lista tutti i dischi riconosciuti nel sistema (Figura 37).

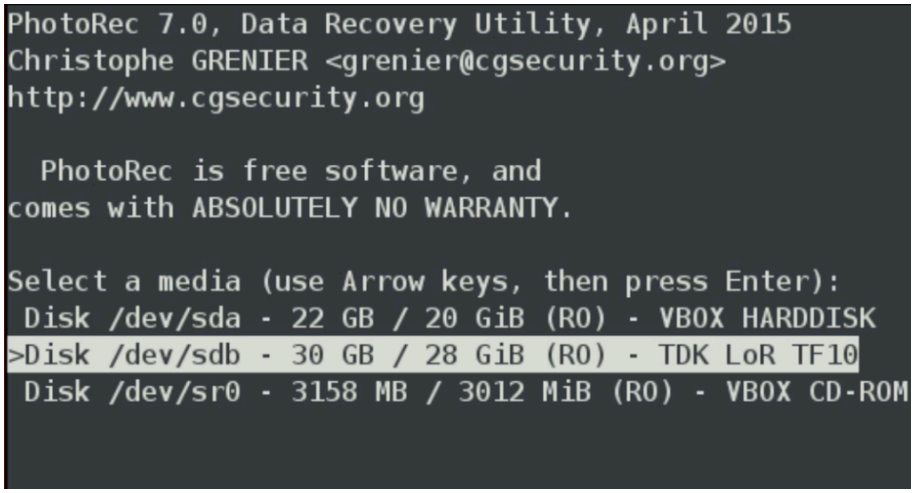


Figura 37: Schermata iniziale di PhotoRec in versione testuale

Possiamo sceglierne uno con i *tasti Su/Giù*, selezionarlo con *Invio* o in caso di errore premere il *tasto Q*.

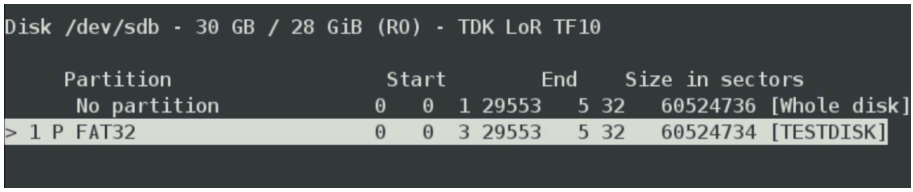


Figura 38: Scelta della partizione o del disco intero

È arrivato il momento di scegliere la partizione su cui vogliamo lavorare (Figura 38). Selezionando Whole Disk effettueremo un recupero su tutto il disco. Scegliamo che *tipo di filesystem* viene utilizzato (Figura 39).

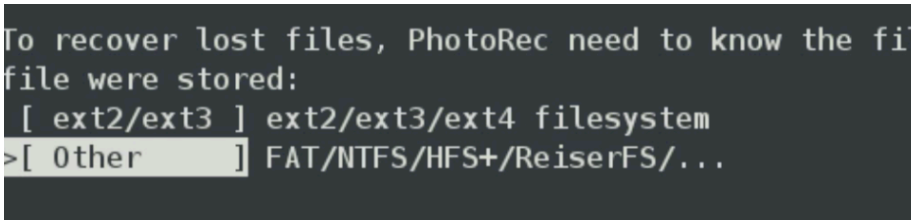


Figura 39: Scelta del tipo di filesystem in uso

Se abbiamo selezionato una partizione ci verrà chiesto se vogliamo fare una ricerca su tutta la partizione o solo dei settori vuoti (Figura 40).

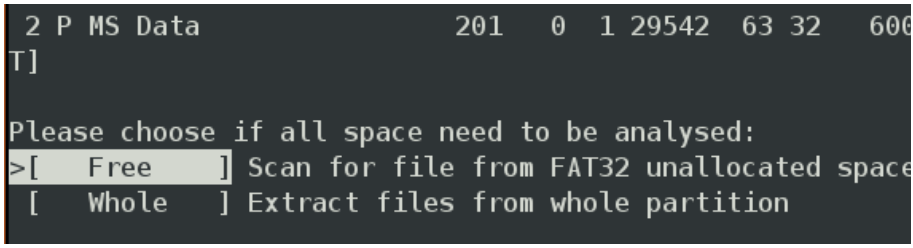


Figura 40: Scelta del tipo di scansione da effettuare

Siamo pronti a scegliere la cartella in cui salveremo la nostra ricerca (Figura 41). Ricorda che i tasti utilizzati poc'anzi valgono anche qui (in particolare Invio per entrare in una cartella e *Q* per tornare indietro) con l'aggiunta del *tasto C* per selezionare la cartella (e le sottocartelle) in cui vorremo lavorare (se sei entrato in una cartella sbagliata puoi tornare indietro cliccando sui doppi punti a inizio lista).

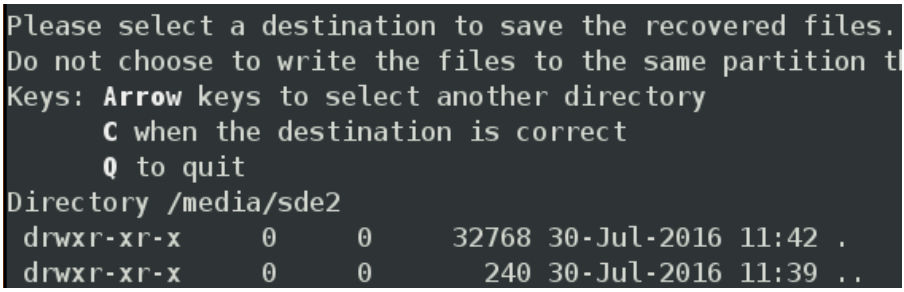


Figura 41: Scelta del percorso in cui salvare i risultati di recupero

Se è stato fatto tutto secondo i piani il software inizierà a scavare nella partizione desiderata e quindi a mettere tutto dentro una cartella dedicata (Figura 42).

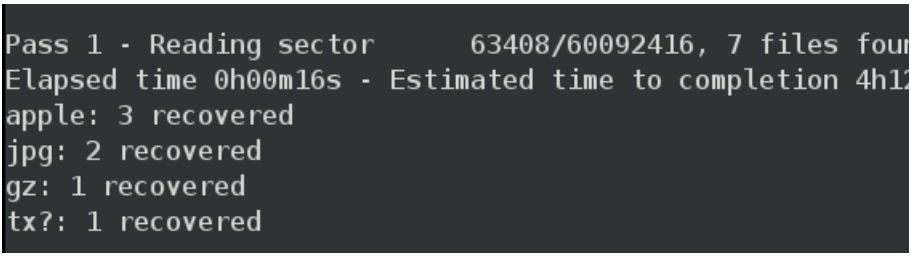


Figura 42: Esecuzione del programma Photorec

Vediamo di fare qualche esempio di utilizzo: abbiamo formattato (senza alcun tipo di shredding) una chiavetta USB da 32 GB con nome “TESTDISK”. È stata creata quindi una partizione FAT in cui abbiamo aggiunto alcuni file (Figura 43).

Figura 43: Immagini di test per il recupero

Ogni file è stato rinominato in base alle azioni che abbiamo eseguito:

- *deleted but not empty.jpeg* : un’immagine che è stata eliminata ma non sono stati eliminati file temporanei né svuotato il cestino
- *deleted.jpeg* : un’immagine che è stata eliminata e file temp e cestino sono stati svuotati
- *normal.jpeg* : un’immagine a cui non è stata applicata nessuna azione
- *normal.jpeg.gpg* : un’immagine cifrata
- *secure-shred-1.jpeg* : un’immagine che è stata eliminata tramite file shredding con algoritmo tipo DoD Short a 1 passaggio
- *secure-shred-7.jpeg* : un’immagine che è stata eliminata tramite file shredding con algoritmo tipo PRNG Stream a 7 passaggi
- *shred-1.jpeg* : un’immagine che è stata eliminata tramite file shredding con algoritmo tipo Quick Erase a 1 passaggio
- *shred-7.jpeg* : un’immagine che è stata eliminata tramite file shredding con algoritmo tipo DoD a 7 passaggi

Vediamo come si comporta il nostro PhotoRec (Figura 44).

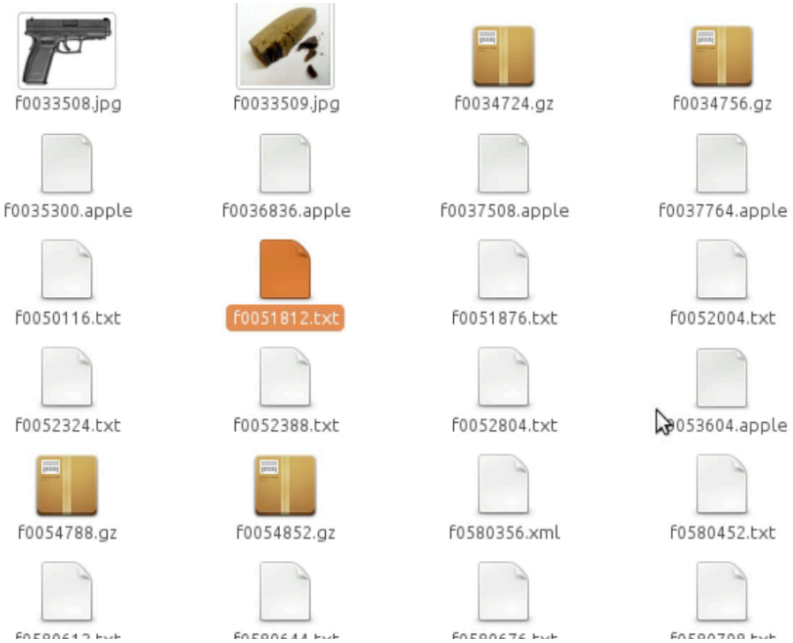


Figura 44: Risultati dopo il recupero con Photorec

Nel nostro caso sono stati recuperati oltre 3 GB di files (Figura 45)! Ma com'è possibile?!

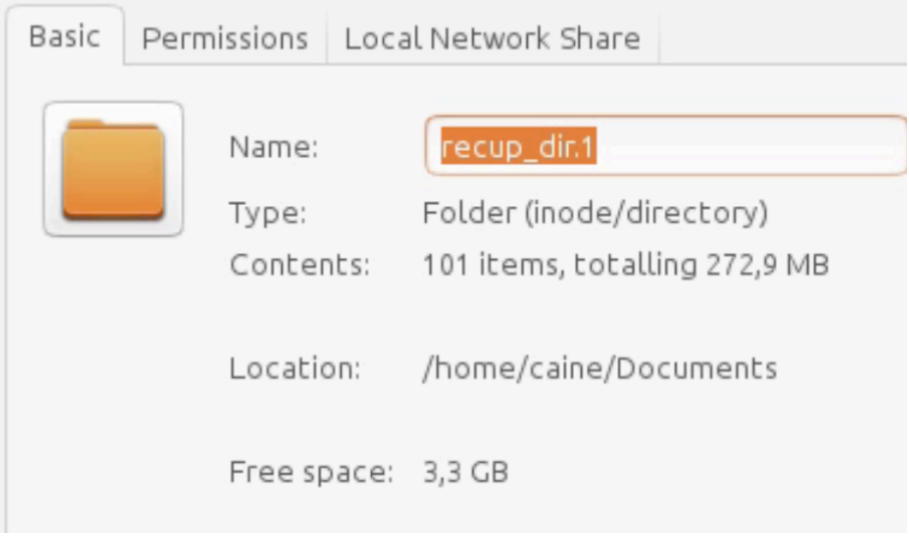


Figura 45: Cartella che contiene i file recuperati con Photorec

All'inizio di questo esempio abbiamo spiegato che la nostra chiavetta è stata formattata in FAT tramite un semplice comando di formattazione. Prima della formattazione conteneva un'installer di Windows e prima ancora fungeva da normale chiavetta USB per spostare files da un Sistema Operativo a base Mac e a base Windows. In una delle dir recuperate troviamo dei files contenenti l'estensione .apple, sintomo che il sistema operativo precedentemente usato era appunto OSX. Molti dei file .txt aperti hanno dimostrato che la chiavetta potesse contenere file pensati per Windows, addirittura che la chiavetta stessa fosse stata utilizzata come installer di Windows 10 (già affermato in precedenza). E i files?

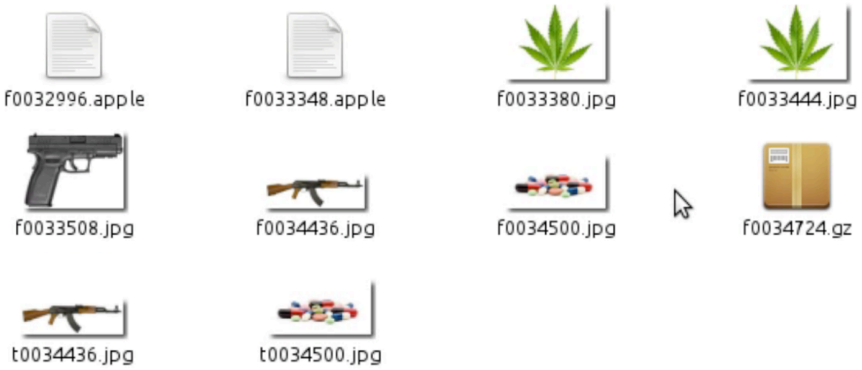


Figura 46: Dettaglio dei file recuperati con Photorec

Ce ne risultano alcuni ancora visibili (Figura 46):

- *f0033380.jpg* : risulta essere il file deleted.jpg
- *f0033381.jpg* : risulta essere il file deleted-but-not-empty.jpg
- *f0033508.jpg* : risulta essere il file normal.jpg
- *f0033509.jpg* : risulta essere il file shred-1.jpg
- *t0034436.jpg* : risulta essere l'anteprima del file secure-shred-1.jpg
- *t0034500.jpg* : risulta essere l'anteprima del file shred-7.jpg

Possiamo quindi dedurre che solo le eliminazioni normali e il Quick Erase sono risultati inutili mentre tecniche di DoD e PNRG risultano efficaci, che anche a

seguito di un partizionamento alcuni file sono stati recuperati (in questo caso l'installer Windows) e con molta probabilità anche programmi precedenti (che spiegherebbe così tanti dati recuperati). Tuttavia dobbiamo considerare che il Sistema Operativo con cui abbiamo creato la chiavetta era un MacOS che, durante la verifica dei dati, si è preso la libertà di creare anteprime delle immagini in nostro possesso, esponendo così il loro contenuto - seppur in bassa risoluzione - all'accesso pubblico.

9. VULNERABILITÀ

Per quanto tu possa aver preso tutte le contromisure necessarie per essere anonimo, purtroppo nell'informatica esiste sempre la possibilità che tu sia una vittima. Non dovrebbe essere un segreto che il governo degli Stati Uniti è il più grande acquirente di vulnerabilità ancora non scoperte (le cosiddette 0day), vulnerabilità che usa costantemente per effettuare operazioni di pentest sconosciute a noi comuni mortali. Quella che segue è una dichiarazione fatta da *John McAffee*, CEO del celebre antivirus, che afferma:

> Non c'è più questa grande sicurezza, soprattutto nel mondo online. Se mi date alcune semplici informazioni su di voi, vi prometto che in tre giorni sono in grado di abilitare la vostra webcam e vedere tutto ciò che fate.

A cui ora collegherò un evento capitatomi qualche anno fa:

> Ricordo di un odontotecnico - quindi una persona che non ha nulla a che fare con la Sicurezza Informatica - era solito usare un pezzettino di nastro isolante scuro per coprire la webcam. Pensai tra me e me: "ma questo è proprio paranoico!". Pochi giorno dopo uscì un articolo in cui si parlava un'exploit che per mesi - o forse anni - era stata utilizzata per spiare gli utenti che utilizzavano proprio quel portatile (per intenderci, era un MacBook Pro). Chi ha già usato questo tipo di portatili sa bene che all'attivazione della webcam si illumina un led verde: ebbene, questo exploit permetteva anche di spegnere il led di stato!

Quindi cosa possiamo imparare da questa storia?

9.1 Precauzioni Generali

Ad esempio che coprire la **webcam**, quando non viene utilizzata, non è tutto sommato una cattiva idea! Certo, potremmo monitorare costantemente il traffico di rete e vedere se qualcuno effettivamente si collega al nostro notebook/computer, ma questo toglierebbe tempo utile alle nostre attività; inoltre l'attacker potrebbe aver installato una backdoor nel nostro computer e quindi nasconderne arbitrariamente le informazioni dal monitor di rete.

Lo stesso si potrebbe fare con il **microfono**: in questo caso la soluzione migliore (se possibile) è quella di rimuoverlo fisicamente dal dispositivo; in alternativa si potrebbe anche disattivare dal Sistema Operativo, ma se per qualche malaugurato evento il computer venisse violato ci vorrebbe poco ad attivarlo.

Il monitoraggio di un dispositivo è fattibile anche su **smartphone**, e questo può essere un grave problema. Non è una novità che gli organi competenti sono in grado di effettuare intercettazioni ambientali utilizzando i microfoni (o catturare immagini) dagli smartphone: il problema è che sarebbe praticamente inutile andare in giro con un cellulare senza microfono, inoltre sono abbastanza sicuro che non tutti siano in grado di smontarlo pezzo pezzo senza danneggiarlo.
Secondo alcune ricerche tratte da Wikipedia, il monitoraggio ambientale può essere effettuato su uno smartphone anche se quest'ultimo è privo di batteria[1]. La soluzione più semplice in questo caso è di lasciare lo smartphone all'interno di un forno a microonde, che permette di isolare i campi elettromagnetici e non permettere quindi l'uso di qualunque onda trasmittente. Occhio a non accendere il forno!

Veniamo poi alle **email** che riceviamo: sembrerà ridicolo dirlo ora, dopo tutto quello che abbiamo detto riguardo la sicurezza, eppure consideriamo sempre

[1] https://en.wikipedia.org/wiki/Cellphone_surveillance

questa verità: non aprite MAI nessun allegato che vi viene inviato, a meno che non siate sicuri al 100% della fonte.

E per quanto riguarda il **Sistema Operativo**? All'inizio di questo corso abbiamo detto che è possibile essere relativamente sicuri con qualunque Sistema Operativo; tuttavia bisogna considerare che GNU/Linux e *BSD sono gli unici sistemi operativi a cui si può fare veramente affidamento. Windows e OSX / MacOS sono OS proprietari e potrebbero contenere non solo trojan e malware di spionaggio ma anche exploit che la community online non potrebbe fixare o anche solo saperne dell'esistenza, in quanto il codice sorgente è in mano ai rispettivi sviluppatori.

Se nasce anche solo un minimo dubbio riguardo a qualunque file, aprilo sempre prima all'interno di una **Virtual Machine**. In questo modo i file aperti verranno virtualizzati in un ambiente esterno (a meno che lo stesso non contenga un exploit capace di rompere il "muro" della Virtual Machine) e in caso contengano qualunque cosa possa compromettere la tua privacy e sicurezza saranno limitati in quell'ambiente.

Se non ti fidi del tuo **BIOS**, flashalo: alcuni malware sono in grado di insidiarsi all'interno del BIOS e in questo caso nessun Antivirus è in grado di accedervi (ricordati che l'antivirus funziona solo quando il Sistema Operativo è avviato o in alcuni casi poco prima del suo avvio). Assicurati che il firmware utilizzato corrisponda a quello degli sviluppatori e non fidarti mai di firmware custom sviluppati da sconosciuti o non riconosciuti come affidabilità dalla comunità online.

A proposito degli **Antivirus**: sono davvero utili? Ci sono diverse scuole di pensiero, c'è chi pensa che male non fanno, c'è chi li ritiene indispensabili e c'è chi invece li ritiene inutili e dovrebbe seguire il proprio istinto e le proprie abitudini. Come per molte cose, la verità sta in mezzo: tutto si basa sul tipo di attività che fai, su quanto ti fidi degli Antivirus e sulle scelte che prenderai

aprendo o non aprendo un file. Di sicuro gli Antivirus non sono perfetti al 100%, fanno uso di database condivisi e alcuni di algoritmi di ricerca euristica per cercare di interpretare cosa farà un file o un programma una volta aperto, ma è solo statistica e potrebbe dare un falso positivo (un virus non virus) o non accorgersene. Quello che è sicuro è che se il dispositivo è infetto o viene preso di mira da un'agenzia governativa, le probabilità che un Antivirus se ne accorga sono pari a zero. Inoltre, la maggior parte dei virus informatici vengono oggi offuscati e modificati alla sorgente per rendere più difficile - e certe volte nullo - il lavoro degli AV. Questi sono alcuni dei motivi per cui non abbiamo trattato - e non lo faremo - su quali Antivirus fare affidamento.

Questo non significa però che il Sistema Operativo non deve essere adeguatamente protetto: è importante anzi che sia **aggiornato** costantemente, che faccia uso delle ultime versioni dei programmi e delle tecnologie in generale (ricordate il famoso Heartbleed?) e configurato in modo che sia sempre sotto il nostro controllo. È possibile ad esempio che il Sistema abbia una funzione per collegarsi automaticamente ad una rete Wifi: basterebbe poco per esporre il sistema e compromettere la sicurezza dell'utente.

10. SISTEMI OPERATIVI AVANZATI

Il mondo GNU/Linux è affascinante per vari motivi: tra questi troviamo l'estrema personalizzazione che ha permesso a intere community di costruirsi la propria versione e distribuirla al mondo intero. Oggi troviamo migliaia di distribuzioni GNU/Linux per ogni esigenza: tra queste, il mondo delle distro anonime sembra essere uno tra i più fiorenti.

10.1 Live OS

Un Sistema Operativo può essere non solo avviato da un Hard Disk ma anche da chiavette USB, CD/DVD e persino schede SD, purché ci sia sufficiente spazio digitale per consentirgli le dovute operazioni. Negli anni è stata quindi sviluppata una nuova metodologia di utilizzo chiamata **Live OS**, una funzione che permette di utilizzare una distribuzione GNU/Linux senza modificare i propri Hard Disk principali. Tale possibilità ci viene offerta non solo per poter testare la distribuzione senza far danni alle nostre partizioni ma si è scoperto che è anche un ottimo modo per non lasciar tracce all'interno di un computer.

Tutto quello che succede all'interno di un sistema Live rimane nel sistema Live: non vengono salvati file temporanei, non vengono generati log permanenti e tutto l'ambiente nasce e muore dal momento in cui la memoria che lo contiene viene inserita o rimossa. Tuttavia potrebbe essere necessario che file o programmi siano comunque reperibili anche dopo l'arresto del sistema: per questo motivo è stata ideata la **Persistence Mode**, una modalità che consente di immagazzinare preferenze, file e modifiche di qualunque genere anche dopo lo spegnimento del computer.

10.1.1 Tails OS

Tails OS[1] è una distribuzione Live GNU/Linux nata nel 2009 e appartenente alla famiglia Debian, quindi perfettamente utilizzabile con tutti i comandi spiegati in questo corso. Viene fornita di tutti gli strumenti utili per garantire un buon anonimato e sicurezza del tuo Computer; al suo interno è possibile trovare una pre-configurazione che veicola tutte le connessioni direttamente su TOR, bloccando quelle in entrata. È sicuramente una distribuzione interessante per il fatto di essere già pronta all'uso: tutte le configurazioni possibili sono presenti nel **Greeter**, il menù di pre-lancio del sistema operativo che consente di attivare anche il network *I2P*, attivare e disattivare il *Mac Spoofing*, attivare e disattivare l'account root, creare spazio persistente cifrato, effettuare configurazioni a bridge TOR e molto altro. Per la crittografia della partizione è già integrato LUKS[2] come standard.

La GUI è anch'essa espressamente pensata per l'anonimato: sarà possibile trovare il *wiping* direttamente da Nautilus (l'explorer), l'integrazione di *GPG* già preinstallato (anche sul client di posta), un browser Iceweasel configurato per navigare su TOR e strumenti base per le operazioni più comuni nell'ambito informatico. Sempre pre-installato troverai la tecnologia *OTR* che permette di cifrare le comunicazioni via chat in Pidgin, il programma di messaggistica onnipresente in quasi tutte le distribuzioni del pinguino.

10.1.2 Live OS e Persistence: i rischi

Le Live OS sono pensate per essere utilizzate sia in computer di vostra proprietà che non: si possono avviare ad esempio in Internet Point, terminali pubblici o computer presi in prestito da qualcuno. La *persistence mode* come abbiamo detto permette di affiancare al Sistema Operativo una partizione in grado di rimanere

[1] https://tails.boum.org

[2] https://en.wikipedia.org/wiki/Linux_Unified_Key_Setup

integra anche dopo lo spegnimento del computer; ricordiamoci che una Live perde la propria memoria (chiamato effetto Amnesia) quando l'utente decide di dare il comando shutdown a tutta la macchina.

Se si decide di avere una persistence mode bisogna considerare vere tutte le condizioni presenti nel capitolo " Sicurezza dei Dati", quindi applicare la crittografia, il data shredding e i metodi utili per evitare l'esposizione del contenuto della tua memoria ad occhi esterni. Ti consiglio inoltre di verificare se la distribuzione Live che stai usando fornisce l'opzione di cifrare la persistence mode: in questo caso controlla la presenza di LUKS tra i formati supportati, in modo da non avere problemi futuri per accedervi anche al di fuori di una situazione in Live OS. Puoi approfondire la cifratura di interi dischi su Wikipedia[1].

10.1.3 Live OS e Virtual Machine: i rischi

Le VM sono strumenti davvero eccezionali: in poche parole permettono di creare un computer all'interno di un computer! Vengono spesso utilizzate quando il nostro Sistema Operativo non è compatibile con alcuni software (ad esempio si vorrebbe poter utilizzare applicazioni Windows mentre si usa un Mac).

Tuttavia, almeno per il momento, ti consiglio di effettuare tutte le prove in ambiente di lavoro Live. Il motivo di questo rigetto possiamo attribuirlo a diverse scelte legate alla sanificazione dell'area di lavoro: in Tails ad esempio le proprietà "anti-forense" andrebbero a farsi benedire dal momento che lanciando una distribuzione GNU/Linux all'interno di una macchina virtuale quest'ultima andrebbe a scrivere all'interno della swap del computer host file che altrimenti in Live andrebbero distrutti; è possibile inoltre che, ibernando o sospendendo la macchina virtuale, TUTTO il sistema operativo venga memorizzato all'interno di un pagefile temporaneo che lo contiene, esponendo tutto il contenuto di Tails in

[1] https://en.wikipedia.org/wiki/Comparison_of_disk_encryption_software

chiaro (a tal proposito, VirtualBox e co. stanno integrando opzioni di cifratura dei dischi a monte del software).

In situazioni di anonimato è fortemente consigliato l'uso di distribuzioni Live GNU/Linux. Se necessario è possibile configurare una chiavetta USB/SD affinché contengano spazio riservato all'utente e alle sue configurazioni, così da avere un sistema ibrido in grado di funzionare come una Live ma di poter memorizzare file e quant'altro come una normale installazione.

10.2 Ambienti Virtualizzati

Quando si parla di Sicurezza Informatica un ambiente virtuale può garantire una certa "tenuta stagna" in diverse situazioni: basti pensare che, se si vuole studiare il comportamento di un malware, è fondamentale l'utilizzo di un sistema virtualizzato affinché non si comprometta il Sistema Operativo centrale. Esattamente come una Live, tutto quello che succede in una Virtual Machine (di solito) rimane in una Virtual Machine: dico di solito poiché è possibile che la macchina virtualizzata venga attaccata così da prendere il controllo del computer host, ma forse stiamo uscendo dal discorso.

Utilizzare un sistema operativo in modo anonimo all'interno di una Virtual Machine è completamente errato: molte delle procedure che abbiamo anche descritto fanno riferimento a metodi di offuscamento che richiedono al Sistema di avere il totale controllo dell'hardware in uso (basti pensare al Mac Spoofing). La virtualizzazione, come dice la parola stessa, consiste nel virtualizzare l'hardware; se decidessimo di effettuare il Mac Spoofing da un ambiente virtualizzato, questo modificherebbe SOLO il Mac Address virtuale, non quello reale! Questa operazione può essere effettuata SOLO dal Sistema Operativo host, vale a dire quello che ospita la Virtual Machine, non viceversa.

Ma se fosse l'host ad essere il Sistema Operativo che offre anonimato alle sue Virtual Machine? Ecco che allora la situazione cambia notevolmente a favore dell'utente.

10.2.1 Qubes OS

Il progetto Qubes OS[1] nasce il 3 Settembre 2012 ad opera di una ricercatrice informatica, Joanna Rutkowska. Questo particolare Sistema Operativo introduce un approccio di sicurezza definito ad *isolamento*: in pratica si da per scontato che ogni software può essere potenzialmente dannoso e che basta un bug per compromettere l'intero sistema informatico.

Qubes è basato su Fedora Linux ma al suo interno fornisce un sistema di paravirtualizzazione grazie a Xen: il suo microkernel permette di creare ambienti di lavoro divisi così da permettere l'interazione tra i vari tools che convivono nello stesso dominio, qui chiamati **qubes**. Per spiegare meglio il concetto guarda con attenzione la Figura 47.

[1] https://www.qubes-os.org

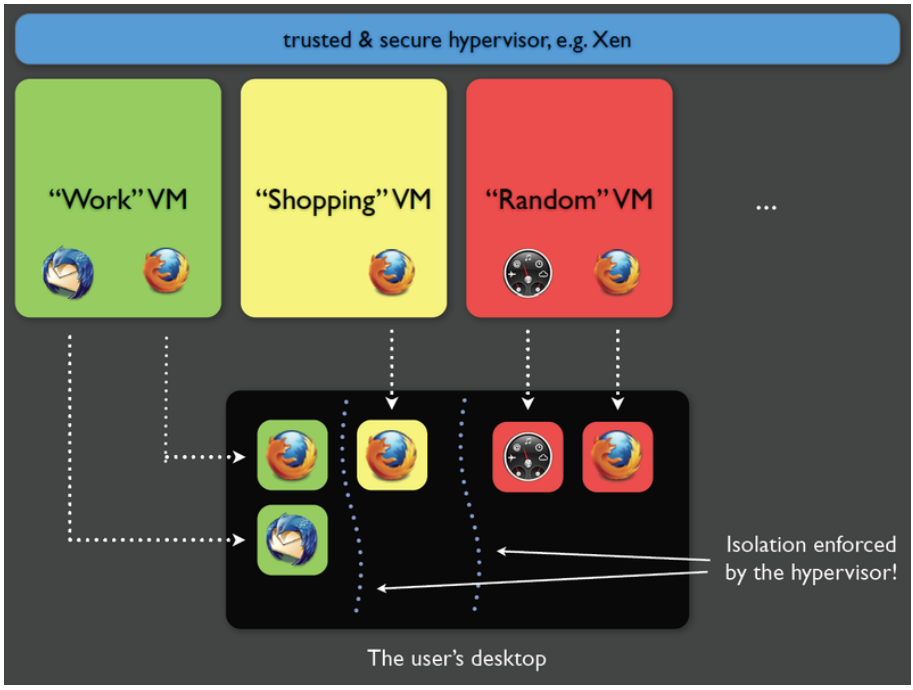


Figura 47: Nell'esempio qui mostrato, sono presenti tre ambienti virtualizzati: Work, Shopping e Random.

In tutti e tre gli ambienti è presente un processo attivo di Firefox: questo però viene trattato come processo a se stante, quindi nel caso fossimo loggati su Amazon nella VM Shopping non lo saremo in Work e in Random, garantendo così l'isolamento dei processi tra i vari ambienti di lavoro.

Che bisogno c'era di un Sistema Operativo? Non si potevano creare tre virtual machine? Certo ma le tre virtual machines avrebbero richiesto tre sistemi operativi, ognuno dei quali avrebbe utilizzato risorse hardware, sarebbero dovuti essere aggiornati e via dicendo.

10.2.1.1 LOGICA DI VIRTUALIZZAZIONE

L'hypervisor integrato in Qubes permette di creare infinite qubes utilizzando un unico Sistema Operativo che può supportare nativamente *Fedora, Debian, Windows (previa installazione di quest'ultimo*[1]*) e Whonix;* inoltre, gli ambienti di lavoro condividono lo stesso ambiente grafico, eliminando lo stressante switch tra i vari Sistemi Operativi. Avrai inoltre notato che i tre ambienti di lavoro sono divisi per colore; la stessa cosa si replica anche a livello grafico (Figura 48).

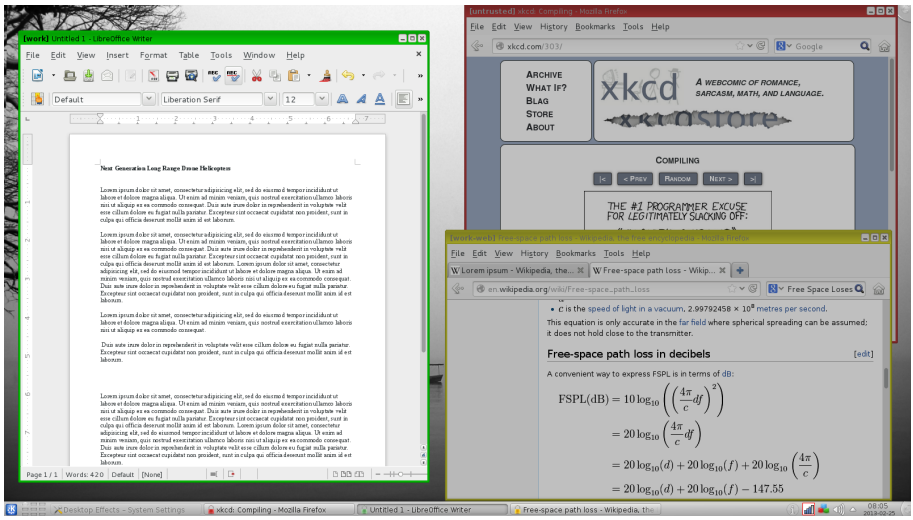


Figura 48: Logica di funzionamento di ambienti virtualizzati in Qubes

Come puoi notare le tre finestre sono rappresentate da un colore diverso, esattamente come l'infografica. In più solo una è luminosa, mentre le altre sono leggermente oscurate.

Qubes OS permette non solo di riconoscere al volo gli ambienti di lavoro suddividendoli per colore ma anche di mostrare in real time quali sono quelli in cui si sta lavorando, evidenziando le applicazioni che possono comunicare tra di loro. E quando una Virtual Machine viene chiusa, tutti i dati temporanei che essa ha generato vengono distrutti.

[1] https://www.qubes-os.org/doc/windows-appvms/

10.2.1.2 DOMINIO NETWORK E DOMINIO STORAGE

Come abbiamo avuto modo di vedere, il Network è l'ambiente più pericoloso per l'utente che vuole proteggere il proprio anonimato.

Qubes OS offre al suo interno un sistema di virtualizzazione chiamato **Network Domain**: in buona sostanza il concetto di VM viene applicato anche alla rete che viene virtualizzata in un ambiente controllato da uno pseudo-user senza privilegi di root e isolato dal resto del Sistema Operativo. Praticamente è come se tutte le operazioni di networking (collegamenti a siti web, downloads, chat e via dicendo) venissero gestite da un'altra Virtual Machine che non può assolutamente interferire con la macchina principale: questo garantisce una sicurezza senza precedenti in quanto l'utente non rischia attacchi a livello di rete.

Lo stesso concetto viene applicato anche alla memorizzazione dati, qui definita **Storage Domain**: gli ambienti di lavoro devono ovviamente avere il loro spazio su disco per memorizzare software, dati e via dicendo. Tutte le pseudo-partizioni condividono lo stesso filesystem in modalità read-mode only, così da evitare compromissioni riuscendo però ad effettuare una centralizzazione degli aggiornamenti. Di default, l'intero filesystem viene creato già criptato alla prima installazione.

10.2.1.3 PERCHÉ USARE QUBES E NON TAILS OS?

Partiamo sempre dal concetto che ogni soluzione è soggettiva: chi preferisce Tails probabilmente cerca una situazione totalmente estranea al suo utilizzo quotidiano; Tails infatti permette di avere un ambiente protetto e adeguato alle operazioni di navigazione in anonimato più comuni, tuttavia il suo limite intrinseco è anche il suo punto di forza: l'usabilità. Tails OS, così come molte altre distribuzioni Live, è pensata per operazioni mordi e fuggi che non sempre si sposano adeguatamente con il tipo di attività che si fa.

Un utente con una certa esperienza nel campo di GNU/Linux vorrebbe avere la sicurezza che Tails offre (ai limiti del possibile) ma di poter avere un ambiente comodo su cui lavorare senza dover ogni volta riavviare il Sistema Operativo. Qubes OS riesce a fare questo, garantisce ambienti di lavoro isolati tra di loro e nello stesso tempo offre la comodità di un Sistema Operativo standalone. Sarebbe inutile comunque confrontarne le caratteristiche e dire qual è meglio l'uno dall'altro in quanto ricordo che sono scelte soggettive.

Tirando le somme entrambi i Sistemi Operativi sono importanti per i loro scopi:

- Tails è pensata per *essere anonimi*. Quando viene avviata in un computer fa in modo che non vengano lasciate tracce all'interno di esso, modifica il Mac Address della scheda di rete e redirige tutto il traffico all'interno di TOR.

- Qubes è pensata per *l'uso di tutti i giorni*. Il suo compito finale è di assicurare l'utente una protezione da attacchi informatici di ogni tipo. Non viene fornito di default di alcun strumento di anonimizzazione (ad eccezione dell'integrazione con Whonix) ed è necessaria una personalizzazione da parte dell'utente finale.

10.2.2 Qubes OS + Tais

Immagino che tu ti sia un attimo stizzito quando ho affermato che Qubes non è un Sistema Operativo pensato per l'anonimato e ti sarai chiesto: "ma perché ne parla?". Poi però hai letto il titolo di questo capitolo e ti sei eccitato oppure (più plausibilmente) stai ripensando alla mia affermazione: "ehi, non usare Tails sulle Virtual Machine!" costringendomi di fatti a una controaffermazione.

È vero, Tails va evitato nelle Virtual Machine e ti ho già spiegato i motivi che gli stessi sviluppatori spiegano: il più importante tra questi è la persistenza - o meglio la reminiscenza - dei dati dal Sistema Operativo che rimangono in memoria all'interno del disco. Come abbiamo avuto modo di vedere però, Qubes usa una logica di paravirtualizzazione che autodistrugge completamente i dati

che rimangono in memoria, facilitando così le operazioni di bonifica del drive. Qubes OS è quindi un ambiente adatto per la virtualizzazione di Tails e il procedimento per l'applicazione è relativamente semplice[1]. In questo modo è possibile far uso della "potenza di fuoco" di Qubes OS assieme a Tails OS: come si dice in questi casi, due piccioni con una fava!

10.2.3 Qubes OS + Whonix

L'utilizzo di Tails in Qubes ci ha permesso di comprendere la virtualizzazione di un intero Sistema Operativo all'interno di un sistema paravirtualizzato Xen, ciò tuttavia può essere considerato un limite. Questo è vero nel momento in cui si vogliano utilizzare i tools in Qubes anziché quelli virtualizzati in Tails.

Inoltre, solo nell'ambiente Tails avremo una sicurezza tale da garantire l'Anonimato. È necessario allora creare un nuovo livello (esattamente come abbiamo visto per i Network e Storage Domain) che ci permetta di veicolare il traffico in un canale di comunicazione sicuro e anonimo. Whonix è un'altra distribuzione GNU/Linux basata su Debian e Tor e fa uso di due Virtual Machine, un *Gateway* e una *Workstation*.

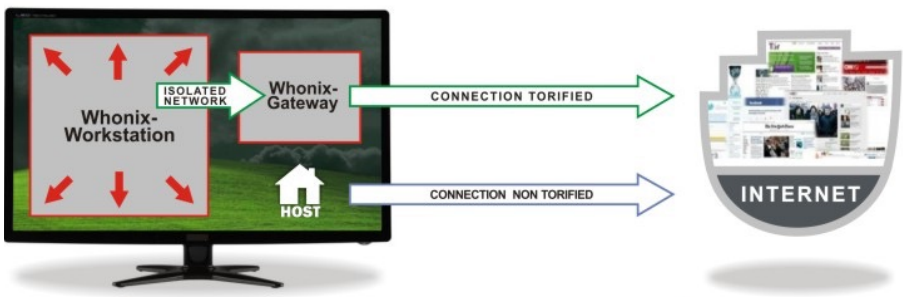


Figura 49: Grafico sul funzionamento di Whonix

[1] https://www.qubes-os.org/doc/tails/

Come vediamo in Figura 49, la *Workstation* è un ambiente che ci permette di lavorare all'interno di un'area isolata dal *Gateway*, una Virtual Machine che è già pensata per il collegamento via Tor. Fatta questa premessa è doveroso ricordare che Whonix ha gli stessi limiti di sicurezza che abbiamo affrontato nel capitolo "Tor", in più a differenza di Tails non è un sistema operativo "pronto all'uso" ma sono necessarie abilità in ambiente GNU/Linux prima di essere utilizzato.

Questa differenza si paga con *l'assenza di alcune peculiarità* che rendono Tails talvolta vantaggiosa come:

- Mancanza del Mac Spoofing pre-configurato
- Mancanza di "amnesia" del software, ovvero tutte le funzioni volte a eliminare qualunque informazione nel computer
- Mancanza di flush dei metadati
- Mancanza di una cifratura completa a livello di posta, causato dalla retrocompatibilità con il protocollo SMTP
- e tanto altro[1].

Alcune di queste lacune sono risolvibili mediante la virtualizzazione in Qubes, altre applicando alcuni accorgimenti di cui abbiamo già parlato nel documento. In ogni caso, Whonix e Qubes sono pensati per essere strumenti utilizzati da una macchina fissa e questo è un po' il prezzo che si paga preferendo l'usabilità alla sicurezza (e vi assicuro che questa bilancia si ripresenta in molte situazioni nella Sicurezza Informatica)[2].

[1] https://www.whonix.org/wiki/Warning

[2] Documentazione di installazione di Qubes + Whonix all'indirizzo : https://www.qubes-os.org/doc/whonix/install/

10.2.4 Subgraph OS

Possiamo definire Subgraph OS come l'ultimo arrivo dei Sistemi Operativi in fatto di privacy e anonimato. È ancora in versione alpha quindi prima di ogni cosa prendilo per quello che è e cioè una bozza di quello che dovrà diventare in futuro.

Gli sviluppatori assicurano che Subgraph OS sarà un Sistema Operativo rivoluzionario e in un certo senso non hanno tutti i torti: nasce come un OS veloce ed utilizzabile anche in computer obsoleti, sicuro e pensato per chi ha "paura per la propria privacy". Segue ora uno schema a Figura 50 di com'è strutturato Subgraph OS.

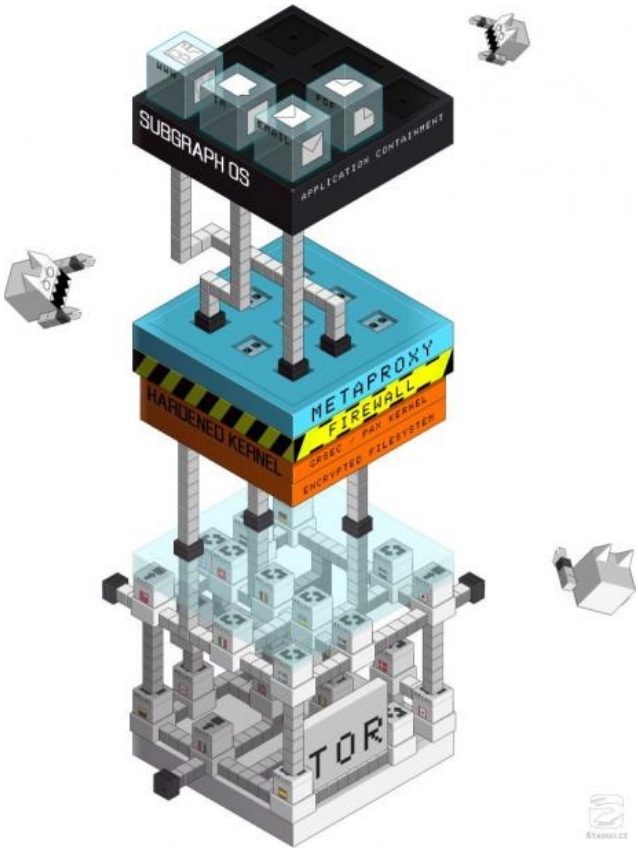


Figura 50: Grafico sul funzionamento di Subgraph OS

10.2.4.1 HARDENED COME POCHI

Subgraph OS viene distribuito di default con un **kernel** già compilato di Grsecurity[1], una serie di patch che garantiscono un alto livello di sicurezza all'intero sistema. All'interno di Grsecurity troviamo PaX, un componente in grado di rilevare all'interno dell'OS eventuali attacchi di diverso tipo come i buffer overflow grazie all'uso di una tecnologia chiamata ASLR che permette di randomizzare le allocazioni di memoria e rendere difficoltoso qualunque attacco a livello di memoria.

Subgraph OS riprende inoltre il concetto di virtualizzazione che abbiamo già visto su Qubes OS: lo scopo è quello di creare **Sandbox** isolate che non possono comunicare tra di loro. Nel caso in cui un software venisse exploitato, questo non potrebbe comunque attaccare l'intero Sistema Operativo, rimanendo così innocuo. Questo processo viene garantito da OZ, un sandbox framework pensato espressamente per Subgraph OS. Se ti stai chiedendo se Subgraph OS supporta la **cifratura del filesystem** la risposta è certo che si! E non solo, questa è addirittura obbligatoria.

La maggior parte dei tools scritti appositamente per Subgraph OS sono ad alto livello (probabilmente interpretati e non compilati) così da non poter essere soggetti ad attacchi di tipo memory; inoltre sono stati rimossi molti dei tools ritenuti superflui, in quelli fondamentali sono state applicate misure di sicurezza e in certe situazioni addirittura riscritti da zero (vedesi il client email di default).

10.2.4.2 NETWORK E ANONIMATO

Sempre sulla falsa riga di Qubes, troviamo un dominio di networking: in questo caso si chiama **Subgraph Metaproxy** affiancato da un **Firewall Software**. Mentre il Firewall consente solo alle applicazioni consentite di collegarsi al Metaproxy, quest'ultimo è configurato per collegare ogni programma a un singolo relay TOR,

[1] https://en.wikipedia.org/wiki/Grsecurity

smistando le connessioni su più canali e diminuendo le informazioni in comune sulla rete. Per farla breve, navigare sul web e scrivere una mail comporteranno l'uso di due reti TOR diverse e questa prerogativa verrà sempre garantita dal Metaproxy.

Tornando al firewall, l'utente può consentire temporaneamente o permanentemente l'accesso alla rete da parte di qualunque software, scardinando di fatti ogni possibilità a una backdoor di infettare il Sistema (a meno che questa non sia presente all'interno di un processo già whitelistato). Il whitelisting di un'app avviene sia per nome dell'applicazione che per indirizzo di destinazione; nel caso in cui un'applicazione volesse collegarsi senza essere in whitelist, il Firewall semplicemente si occuperà di killare la connessione.

Come già avrai intuito, Subgraph OS fa uso della rete TOR per comunicare con il mondo esterno: ad essere precisi, fa un uso esclusivo della rete TOR, ad eccezione di alcune situazioni dove ad esempio è necessaria una comunicazione diretta verso il portale che si sta visitando (come un captive portal in una rete wifi pubblica). Dulcis in fundo, Subgraph OS fornisce al suo interno due software custom per la sicurezza nelle comunicazioni:

- Icedove, un client basato su Thunderbird, fornito di Enigmail (PGP) e TorBirdy (Anonimato via Tor)
- CoyIM, un client XMPP riscritto completamente da zero per evitare exploiting a livello di memoria e anch'esso pensato solo ed esclusivamente per la rete TOR

10.3 Distribuzioni Pentest

Con molta probabilità conosci già le distribuzioni Linux di tipo pentest: se così non fosse, il Pentesting è il nome abbreviato di Penetration Testing, una branca della Sicurezza Informatica. Nel Penetration Testing si valuta la sicurezza generale di un apparato informatico e di ciò che lo circonda: la rete, il Sistema Operativo, i programmi e così via.

La community Linux negli anni ha dimostrato interesse verso questo mondo sviluppando distribuzioni che contengono applicazioni pre-configurate per velocizzare le operazioni di testing, offrire ambienti standardizzati e raccogliere utenti con gli stessi interessi sotto un'unica luce. Nei prossimi volumi dell'Hacklog verranno approfondite e utilizzate quando si discuterà di attacchi informatici, al momento ci limiteremo solo a stilare una lista:

- **Kali Linux**, basata su Debian (https://www.kali.org)
- **Backbox**, basata su Ubuntu (https://backbox.org)
- **Parrot Security OS**, basata su Debian (https://www.parrotsec.org)
- **DEFT**, basata su Debian (http://www.deftlinux.net/it/)
- **Pentoo**, basata su Gentoo (http://www.pentoo.ch)
- **NST**, basata su Fedora (http://networksecuritytoolkit.org/nst/index.html)
- **BlackArch**, basata su Arch Linux (https://blackarch.org)
- **Fedora Security Lab**, basata su Fedora (https://labs.fedoraproject.org/it/security/)
- **Cyborg Hawk Linux**, basata su Ubuntu (http://cyborg.ztrela.com)
- **WeakerThan**, basata su Ubuntu (http://www.weaknetlabs.com)
- **Samurai Web Testing Framework**, basata su Ubuntu (http://samurai.inguardians.com)
- **Bugtraq** (http://bugtraq-team.com)
- **Knoppix** (http://www.knoppix.org)

11. IDENTITÀ ONLINE

Arrivati a questo punto abbiamo tutti gli strumenti e le competenze necessarie per navigare in anonimato; si badi bene, ho detto navigare, non interagire! Il fatto che ci sia TOR o qualunque altra tecnologia tra le due parti non significa che siamo in una botte di ferro; al contrario, questa sensazione di protezione può essere un'arma a doppio taglio per la nostra identità reale.

11.1 Non devi MAI intrecciare le tue identità

A prescindere dall'attività che si voglia fare, sia essa in clearnet che in deepweb, è necessario essere in grado di *separare le varie attività* per evitare di creare collegamenti e creare un fingerprint della nostra identità (ti ricorda qualcosa questo termine?). Lasciare tracce delle proprie attività - email, indirizzi bitcoin, nomi, località etc… - permettono di creare un profilo più dettagliato sulla persona da ricercare. Nel caso qualcuno riuscisse ad unire le tue due identità potrebbe raddoppiare le informazioni su di te.

Torniamo a Ross Ulbricht, l'admin dell'ormai defunto Silk Road, portale che ai tempi ha permesso a lui - e a molte altre persone - di guadagnare centinaia di migliaia di dollari nel mercato dell'illegalità. Sai come l'hanno beccato? Quando ancora Silk Road non era famoso, Ross fu il primo che in clearnet chiese se qualcuno conoscesse quel mercato - si sa, lo fanno in molti per spammare i propri siti web. Da lì, assieme ad altre prove, si riuscì a risalire all'identità di Ross Ulbricht (e a cascata a molti altri membri della banda).

11.2 Non devi MAI usare gli stessi dati

Lo sanno anche i bambini che in una password non si devono mai mettere i propri dati (data di nascita, nome e cognome, località etc…) ma di utilizzare caratteri alfanumerici a caso, numeri, simboli speciali e qualunque cosa random. Puoi utilizzare diversi programmi come KeePassX (integrato in Tails), LastPass, 1Password e molti altri sia per generare nuove password che memorizzarle con una master-key in grado di sbloccarle tutte.

Per sicurezza, non usare MAI un unico portachiavi in cui memorizzi sia le password per le tue attività “normali” che per quelle “alternative”. Collegandoci a quello che abbiamo detto poc’anzi, non devi mai intrecciare le tue identità! Se nel primo caso l’avresti fatto consciamente però, in questo caso considera anche le tracce che lasci inconsciamente:

- Indirizzi IP
- Password
- Date di Nascita
- Dati di Fatturazione
- Indirizzi e Località
- Foto e Avatar simili
- Indirizzi di contatto simili
- ... qualunque cosa possa ricondurre a te o anche solo alla tua seconda identità/terza/quarta etc...

11.3 Attenzione alle abitudini

Se hai dei modi di dire che usi spesso, un dialetto particolare oppure usi Una Grafia Come Questa o ancora commetti notevoli/gli stessi errori di ortografia e grammatica tali da non lasciare alcun dubbio che tu sia quella persona… fai qualcosa!

È probabile che in molti ti abbiano fatto notare queste "particolarità"; se sei una persona particolarmente permalosa potresti non essertene neanche mai accorto ma fidati che è possibile risalire all'identità di una persona anche solo per il modo in cui si comporta.

Tempo fa avevamo un moderatore che è stato rimosso per negligenza in alcuni compiti. Questa persona si volle "vendicare" imbrattando i nostri canali di comunicazione con i poteri residui; nel farlo, usava epiteti che spesso utilizzava anche tra amici. Nonostante avevamo già la riprova che fosse lui con il confronto degli IP, in realtà ci bastò vedere come scriveva per capire chi era.

Per quanto riguarda gli orari di operatività? Sei un tipo prevedibile oppure operi H24? Ricorda che il monitoring - soprattutto a livello governativo - viene fatto maniacalmente. Qualunque cosa viene scritta viene analizzata, per ogni punto e per ogni virgola viene analizzata il carattere della persona che scrive.

11.4 Email usa-e-getta

Sono chiamati *Disposable* o *Temporary Email* quei servizi che offrono indirizzi email temporanee, vale a dire che permettono la creazione di indirizzi di posta non intestati a una persona fisica e che non richiedono alcuna registrazione.

Il loro utilizzo è pensato in parte a consentire la registrazione presso i portali che lo richiedono senza il rischio di finire in noiose mailing list che comporterebbero la ricezione di materiale spam per un tempo indefinito.

Alcune Temporary Email consentono non solo di ricevere posta ma anche di inviarla nell'anonimato più totale, ovviamente ponendo l'utilizzatore nella condizione di perdere tutto lo storico di ciò che ha inviato e ricevuto nel momento in cui la clessidra digitale esaurisca il tempo a disposizione per l'utente (e in alcuni servizi questo può essere aumentato a periodi regolari).

Tra i servizi che permettono la **sola ricezione di posta** troviamo:

- 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

mentre per quelli che permettono anche di **inviare posta** troviamo:

- https://www.guerrillamail.com
- https://mytemp.email
- http://www.yopmail.com/en/
- https://www.crazymailing.com

Lascio a te decidere in che modo possono esserti utili; l'unica cosa che posso consigliarti è di trattarli come siti usa e getta, quindi non usarli per informazioni riservate e mantieni sempre il giusto anonimato prima di accedervi.

11.5 Se gestisci un Sito/Blog/Forum

Se sei interessato all'anonimato soprattutto perché non vuoi che la tua identità venga scoperta, assicurati di non tralasciare i seguenti punti:

- Nel caso in cui tu stia lanciando un nuovo sito web potresti voler iniziare con un hosting gratuito. Ce ne sono davvero molti, alcuni pensati per avere dei CMS già preinstallati (come nel caso di wordpress.com). L'importante è che, se vuoi fare un salto di qualità, con molta probabilità ti verranno offerte delle soluzioni a pagamento. In ogni caso preferisci servizi che NON richiedono dati di fatturazione e che offrono anche metodi di pagamento anonimi come i Bitcoin (leggi il capitolo riguardante le cryptomonete).
- È raro, ma non impossibile, che il portale che stai gestendo sia compromesso. Tramite linguaggi come Javascript è possibile applicare tecniche di stilometria che permettono di effettuare un'analisi sul numero di battiti sulla tastiera al minuto e sull'uso del mouse che potrebbero in qualche modo aiutare qualcuno ad ottenere informazioni utili sul tuo modo di scrivere, sulle tue abilità, sui tuoi errori grammaticali frequenti, la punteggiatura che usi, la mano dominante e via dicendo. In questo caso potrebbe esserti utile scrivere i tuoi articoli o i tuoi post prima su un editor in locale, quindi copia-incollare il testo nel sito che stai gestendo.
- Specie nei blog puoi posticipare la pubblicazione di un articolo. Questo potrebbe essere un buon modo per depistare le tracce per chi cerca di geolocalizzarti.
- Non dimenticare di rimuovere tutti i metadati nei file che carichi, specie gli EXIF Data nelle fotografie che pubblichi. Non dimenticare di manipolare le foto (vedi il capitolo sui sensori delle fotocamere). Se sono presenti altre persone, censura i loro volti.

11.6 Cose da non fare, MAI

Segue ora una lista, se vogliamo dei comandamenti, se vuoi evitare che tutto il lavoro fatto sia invano.

- Non devi MAI navigare nel tuo sito web personale *mentre sei anonimo*
- Non devi MAI accedere al tuo account sui social network *mentre sei anonimo*
- Non devi MAI accedere ad un account che hai usato senza protezioni *mentre sei anonimo*
- Non devi MAI accedere ad un account bancario/paypal/ebay o altri siti che possono contenere le tue informazioni personali *mentre sei anonimo*
- Non devi MAI accedere a una rete Wifi aperta che non sai se viene monitorata *mentre sei anonimo*
- Non devi MAI sottovalutare il potere della crittografia *mentre sei anonimo*
- Non devi MAI confondere l'anonimato con lo pseudo-anonimato
- Non devi MAI usare la verifica telefonica *mentre sei anonimo*

12. PAGARE ONLINE

Ogni criminale che si rispetti ha il suo giro di compravendita: skimmer, tessere e documenti, sim anonime, schede di rete e via dicendo. Sarebbe però davvero un peccato se venisse beccato direttamente in casa, non ti pare? Ovviamente acquistare online in modo anonimo non dev'essere una prerogativa riservata al mondo del cyber-crimine: oggi sempre più persone acquistano sul web, inconsapevoli del fatto che ogni ordine effettuato alimenta il più grande database di analisi del mercato di sempre.

Acquistare online senza lasciare tracce nella clearnet è diventata un'operazione praticamente impossibile: gli ordini vengono memorizzati nei database dei venditori, i pagamenti vengono affidati a banche o circuiti di pagamento virtuali tracciabili, le spedizioni affidate a compagnie terze che, per motivi legali o più semplicemente scelte di politica aziendale, possono decidere di verificare il contenuto di qualunque pacco senza dover fornire alcun valido motivo.

12.1 Acquistare nella Dark Net

Come abbiamo avuto modo di vedere, con Dark Net si identifica quella porzione all'interno del Deep Web che contiene materiale considerato illegale in alcuni Paesi del mondo. I malavitosi del web usano la Dark Net per scambiarsi non solo informazioni ma anche prodotti di qualunque tipo, prodotti che ovviamente non possono essere commercializzati all'interno del mercato "normale".

Stiamo parlando non solo di droghe ma anche di armi, oggetti rubati, pornografia, strumenti per la falsificazione e duplicazione di carte di credito, farmaci che necessitano di prescrizione medica, documenti d'identità falsi, database di siti web, exploit di software 0day e molto altro.

È inutile mettere all'erta circa l'affidabilità di store e venditori all'interno della Dark Net: dal momento che lo strumento per navigare e quello per acquistare sono pensati anche per garantire l'anonimato considera che la truffa è all'ordine del giorno.

12.1.1 I Market della Dark Net

In realtà la "dark" community che gira nelle Clearnet - sto parlando di reddit[1], 4chan[2] e molti altri - rilasciano costantemente feedback e link ai nuovi nodi di vendita, soprattutto da quando *Silk Road* (noto market illegale) è stato chiuso, i nuovi canali di vendita sono decuplicati.

12.1.1.1 TIPI DI DARKNET MARKETS

Nella darknet è possibile trovare diversi tipi di market che non sempre condividono lo stesso metodo di compravendita. Nel corso degli anni abbiamo visto diversi tipi di market, così siamo riusciti a riassumerli in cinque grandi categorie:

1) **Market Centralizzati**: sono dei negozi dove compratori e venditori utilizzano un wallet di cryptovalute in comune. Sono estremamente pericolosi in quanto i gestori potrebbero decidere di bloccare il conto e truffare quindi entrambe le parti. In questa categoria troviamo il famoso Silkroad.
2) **Market De-Centralizzati**: è un nuovo tipo di market - ancora in fase di progettazione - che consiste di fare compravendita senza l'utilizzo di strumenti di navigazione esterna come TOR e altri. Al momento i progetti più ambiziosi

[1] https://www.reddit.com

[2] www.4chan.org

sono Bitmarkets[1] e OpenBazaar[2] ma ancora poco utilizzati e in fase di sviluppo.

3) **Forum Market**: sono quelli più popolari in quanto basta un software di forum per crearli e mantenerli. Sono identici in tutto e per tutto a dei forum con la differenza che contengono annunci di vendita. Quasi sempre forniti di servizi di escrow (metodi di vendita in cui si figura una terza persona che fa da arbitro nella compravendita) o acquisti di pacchetti di verifica VIP, in modo da limitare le truffe.

4) **Market Multi-Signature**: in questi tipi di market la compravendita avviene all'interno di un wallet condiviso (come per i market centralizzati) con la differenza che per poter procedere alla chiusura dell'affare due delle tre parti coinvolte (acquirente, venditore e moderatore/middleman) devono accettare lo scambio, così da poter essere (quasi) sicuri che la vendita sia avvenuta correttamente.

5) **Vendita Singola**: in questa categoria rientrano venditori freelance che tramite i loro siti web offrono servizi di qualunque tipo. Sono anche quelli che, definendo le proprie regole, truffano maggiormente nelle Dark Net.

Va considerato anche che alcuni market possono essere solo su invito, quindi non basta essere connessi tramite tramite il circuito anonimo in cui vengono distribuiti.

[1] https://voluntary.net/bitmarkets/

[2] https://openbazaar.org

12.1.1.2 DOVE TROVARE I DARKNET MARKETS

Linkare qui una lista degli store attualmente online non avrebbe alcun senso dato che questi possono durare da qualche settimana a qualche mese, mentre questo manuale (si spera) duri anni senza aggiornamenti! Bisogna inoltre considerare che, a seguito di attacchi ai vari network anonimi, i Darknet Markets tendono a spostarsi da sistema a sistema; nel caso di Silk Road ad esempio l'abbiamo ritrovato rinascere sia su TOR che su I2P.

Per questo motivo mi scuserai se non potrò aggiornare periodicamente una lista, tuttavia potrai seguire uno dei seguenti portali:

- Darkwebnews (https://darkwebnews.com/dark-web-market-list/)
- PsychonautWiki (https://psychonautwiki.org/wiki/Comparison_of_darknet_markets)

A questo proposito è interessante anche lo strumento **Grams**, un *motore di ricerca* dedicato esclusivamente ai Darknet Market (disponibile solo su rete TOR per il momento, cercare su Internet l'indirizzo .onion corretto).

12.2 Cryptomonete

Per i **pagamenti** invece sono le cryptomonete il metodo accettato dalla comunità online: sono valute riconosciute in molte realtà (anche fisiche) e con le giuste precauzioni rendono l'utilizzatore irrintracciabile.

12.2.1 Precauzioni sulle Cryptomonete

Ovviamente dipende dalla cryptomoneta ma ricordati che, in quasi tutte le strutture di questo tipo, i trasferimenti sono di dominio pubblico, quindi se si conosce il proprietario di un indirizzo automaticamente si risale ai suoi movimenti e a quelli ad egli collegati. La cosa più stupida che si possa fare è quindi sbandierare ai quattro venti un indirizzo di pagamento (chiamato wallet) che viene utilizzato per compravendite dubbie o addirittura illegali.

12.2.2 Bitcoin

Il Bitcoin è la cryptovaluta più famosa della rete che permette con le giuste protezioni di effettuare compravendite in anonimato e al di fuori del controllo di Stati e Banche; la tecnologia dei BTC si basa su una rete decentralizzata, questo per evitare possibili manipolazioni alla rete e attacchi alle infrastrutture che mantengono in memoria la cryptovaluta.

Parleremo dei Bitcoin in quanto sono socialmente i più accettati nella rete: ne esistono molti altri (ti consiglio di tener d'occhio Ethereum[1]) ognuno con le proprie peculiarità, tuttavia sarebbe insensato parlare di decine di cryptovalute che potrebbero scomparire da un momento all'altro.

[1] https://www.ethereum.org

12.2.2.1 COME FUNZIONANO I BITCOIN

I Bitcoin vengono memorizzati all'interno di un portafoglio (Wallet) e possono essere trasferiti esattamente come qualunque altro servizio di e-banking online - se hai mai usato Paypal sai di cosa sto parlando - ma non dipendono da istituti bancari, non sono tassabili e in un certo senso sono anche anonimi. Tornando ai **wallet**, questi possono essere di due tipi: software e web-based. In realtà esistono anche ibridi (come blockchain.info) che permettono di accedere al portafoglio sia da programma installato localmente che da interfaccia web. Se proprio non ci si fidasse dei wallet esterni si può sempre installare dal sito ufficiale il client ufficiale (bitcoin.org) o di utilizzare client alternativi (come electrum.org) ma sarà necessario del tempo affinché i wallet saranno allineati con la rete Bitcoin.

A ogni wallet viene assegnato un **address** (indirizzo): questo è un codice alfanumerico univoco che identifica il portafoglio in rete, una sorta di numero telefonico, a cui si invieranno o riceveranno Bitcoin. L'address viene generato alla prima installazione del programma o all'iscrizione del servizio; inoltre è possibile avere più wallet contemporaneamente e scambiare tra di loro Bitcoin a costo zero. I wallet devono essere protetti da una password e da una passphrase: questi elementi garantiscono l'utilizzo solo al legittimo proprietario e permettono di utilizzare il wallet anche solo temporaneamente all'interno di un Sistema Operativo. Si consiglia pertanto di effettuare regolarmente backup dei propri wallet e cifrarli tenendo in considerazione ciò che è stato spiegato nel capitolo riguardante la "Crittografia".

Ricordiamo alcune regole generali dei Bitcoin:

- **I Bitcoin sono digitali**: i BTC non possono essere stampati su carta (o almeno, non vengono riconosciuti ufficialmente).
- **I Bitcoin sono distribuiti**: non esistono server che gestiscono i Bitcoin.
- **I Bitcoin sono divisibili**: avere 1 BTC oggi significa avere centinaia di euro. La valuta più comunemente usata è il mBTC (che vale 0,001 BTC).

- **Bitcoin è opensource**: il codice sorgente del software è aperto a modifiche e disponibile per chiunque
- **Bitcoin è (quasi) anonimo**: tutte le transazioni sono pubbliche ma è possibile risalire solo agli indirizzi. Se si conosce la proprietà di quest'ultimi, la privacy è compromessa.

12.2.2.2 COME OTTENERE I BITCOIN

È possibile ottenere i Bitcoin fondamentalmente in due modi:

- ***Generandoli***: nel gergo informatico questo processo viene definito *Mining*. La cryptomoneta nasce come valuta distribuita e per essere creata viene appunto "minata" (da qui il termine minare). Scaricando il software di mining è possibile creare della vera e propria valuta spendibile per acquistare beni e servizi. Essendo però diventata molto popolare, sono ormai tantissimi tra esperti e aziende che puntano a generare sempre più cryptomonete, rendendo praticamente impossibile ogni forma di concorrenza.
- ***Acquistandoli***: questo è ovviamente il metodo più semplice. Esistono diversi mercati delle cryptomonete che commerciano qualunque tipo di cryptovaluta (Bitcoin, Litecoin, Anoncoin, Primecoin e via dicendo) in cambio di moneta reale. Uno dei portali più popolati è LocalBitcoins.com che permette di mettersi in contatto con altre persone della tua città (o nazione) e acquistarli tramite diversi metodi di pagamento: Postepay, Bonifico Bancario, PayPal, Wester Union e via dicendo. In alcuni forum (come inforge.net) è possibile acquistarli dai vari utenti tramite metodo di escrow.
- ***Scambiandoli***: i Bitcoin possono essere anche utilizzati come merce di scambio per altre valute (tra cui monete reali). Siti come BTC-E.com, bitstamp.net, coinbase.com e altri offrono servizi di acquisto tra cui appunto le cryptovalute. A differenza di compravendite tra privati, questi siti richiedono informazioni personali dell'utente come passaporti o licenze di guida, mettendo a serio

rischio la privacy dell'utente. Inoltre sono molti i servizi che gestiscono la piattaforma con wallet online; uno di questi nel 2014 chiuse inaspettatamente e fece perdere 387 milioni di dollari ai suoi clienti: stiamo parlando di Mt.Gox, storia in cui il CEO fu coinvolto anche in uno scandalo di bancarotta fraudolenta.

12.2.2.3 RENDERE IRRINTRACCIABILI I BITCOIN

Il Bitcoin è spesso chiamato "la moneta digitale anonima". Questa affermazione è sbagliata in due punti:

1) Non è una moneta, è una valuta
2) Non è anonima (o non lo è naturalmente)

Ogni singola transazione Bitcoin viene tracciata: per rendersene conto basta visitare il sito BlockChain.info. Se acquisti dei Bitcoin vedrai la tua transazione in chiaro sul sito. Questo chiaramente può essere un problema per la tua privacy, quindi potrebbe essere necessario nascondere le tracce dei Bitcoin.

Intendiamoci: se abbiamo un sito dove chiediamo donazioni mostrando l'address affianco ai nostri nome e cognome e poi acquisteremo marijuana dalla Dark Net... beh, tutto il mondo saprà che ci fumiamo l'erba!

Mixing Service

Uno dei modi per "pulire" i BTC è quello di utilizzare un contenitore condiviso di Bitcoin chiamato servizio di mixing (o tumbler): in pratica tutti gli utenti di un servizio mettono assieme i loro Bitcoin, quindi li fanno girare in diverse transazioni "mischiando" le carte, decidendo successivamente quanto ritirare dal portafoglio online. Il sistema funziona depositando una cifra: a questa viene applicata una tariffa che varia dall'1-3% e che compirà almeno 6 transazioni prima di essere nuovamente disponibile.

Molto più semplicemente, l'utente registrato invia dei Bitcoin a un servizio di mixing: questi si occupa di farli girare, rendendoli anonimi. Una volta fatto ciò, li riceverà a un wallet che ritiene più sicuro.

Nel web esistono diversi progetti molto interessanti che si occupano di far ciò: Helix by Grams[1], bitcoinblender.net, bitcoinmix.org, *PayShield,* bitcoin-fog.org , coinmixer.se , coinmixer.net, spacechain.io e via dicendo.

La loro qualità ed affidabilità dipende non solo dalla serietà del servizio ma anche, e soprattutto, dal modo in cui vengono utilizzati. Ogni servizio di mixing gestisce a modo suo le transazioni, le percentuali e tutto ciò che concerne i tempi e l'interfaccia grafica; ad ogni modo la logica di un "lavaggio dei Bitcoin" funziona pressappoco in questo modo:

> Mi raccomando: quando ti collegherai ai mixer userai sempre i link .onion e MAI quelli in clearnet! Dal sito (o da Internet) cerca il relativo indirizzo.

1) Creiamo un wallet nella clearnet (che chiameremo wallet#1)
2) Inviamo - acquistandoli o trasferendoli - BTC al wallet#1
3) Creiamo un secondo wallet, stavolta tramite TOR o circuiti analoghi (wallet#2)
4) Inviamo i bitcoin da wallet#1 a wallet#2
5) Inviamo i bitcoin da wallet#2 all'indirizzo Bitcoin creato dal Mixer
6) Creiamo un nuovo wallet (wallet#3) e facciamo inviare lì i nostri Bitcoin
7) (Opzionale) In caso di compravendita, possiamo inviare direttamente i bitcoin dal #2 al venditore usando il servizio mixing
8) Dal wallet#3 possiamo riprendere i nostri BTC puliti e inviarli così al wallet#1
9) Dal wallet#1 puoi usare i BTC per fare acquisti anonimi o anche in Clearnet

[1] gramshelix.com

Attenzione: Tratta i mixer service esattamente come le VPN, devi fidarti ciecamente di loro! I mixer service potrebbero loggare le tue transazioni e compromettere il tuo anonimato.

CoinJoin

Nel mondo dei Bitcoin esiste un altro modo per lavarli: il CoinJoin è un metodo di compressione di una transazione di bitcoin che è stato pensato per aumentare la privacy delle parti, rimuovendo le informazioni non necessarie della transazione. Il CoinJoin è stato ideato in quanto il Bitcoin stesso, da tempo sponsorizzato come strumento anonimo di pagamento, in realtà è tutt'altro che blindato ma anzi si potrebbe dire che è addirittura meno anonimo di una banca: almeno in questa non sono disponibili le transazioni al dominio pubblico!

Il metodo CoinJoin consiste semplicemente nel collegarsi a un server che funge da punto di ritrovo di tante persone che partecipano tutte alla stessa transazione: così facendo sarà molto più difficile analizzare tutte le valute in circolazione, in più, a differenza dei mixing services, i Bitcoin non possono essere rubati. La cryptovaluta può essere anche utilizzata per altri scopi oltre agli acquisti in anonimato: tra i metodi più accreditati vi troviamo l'evasione fiscale e il riciclaggio di denaro. Soprattutto per il primo motivo, il Bitcoin è stato messo al bando da diverse nazioni (l'Italia per il momento è esclusa) ed è al vaglio delle più grandi banche mondiali che, assieme ai governi, stanno decidendone il suo destino.

È logico pensare che in un futuro non troppo remoto questa valuta potrà essere totalmente illegale in quanto possono essere utilizzate in sostituzione della moneta ufficiale - pur non essendo la cryptomoneta del vero e proprio denaro - e ciò causerebbe il crollo dell'intero sistema bancario. Ma questa è un'altra storia…

12.2.3 Oltre i Bitcoin

Il concetto di cryptovaluta è ancora una realtà nuova e com'è giusto che sia bisogna aspettare degli anni prima di avere un reale punto di riferimento. Al momento questo sembrano essere i Bitcoin ma il mercato tende a guardare altrove, anche in vista della forte saturazione delle blockchain, la centralizzazione del mercato dei BTC e dei suoi limiti. Esistono quindi alternative con un loro certo fascino come i Litecoin, Dogecoin, Quarkcoin, Primecoin, Peercoin e via dicendo.

Il futuro dei Bitcoin sembra però essere meno fiorente del previsto: sembra proprio che i nuovi investitori si stiano concentrando sull'Ethereum[1] un nuovo modo di concepire lo scambio, risolvendo i problemi e i limiti del BTC, ottenendo di diritto la nomina del Bitcoin 2.0 . Ai fini dell'Anonimato non è necessario approfondire le cryptovalute alternative, tutte condividono parte della loro logica di funzionamento e in più non siamo sicuri se, quando e quali saranno quelle che sostituiranno i Bitcoin. Solo il tempo ce lo dirà.

[1] https://www.ethereum.org

13. SII LIBERO

Sei ora pronto a goderti la tua completa libertà nella rete, fuori dal range d'azione di qualunque organizzazione o azienda che fino a questo momento ti ha usato come carne da macello per i suoi esperimenti.

Si, forse sto esagerando, ma ritengo che in un certo modo di vedere le cose è peggio credere di essere libere che esserlo veramente. Essere liberi dalle catene della statistica, del mercato, dell'analisi, del tuo governo che non vuole ti piacciano certe cose, degli zombie attorno a te che ti guardano come un parassita che cerchi solo di essere te stesso.

Ho voluto scrivere questo ebook perché tu potessi essere libero, perché tu vivessi senza la paura costante che un giorno questo sogno finisca. Ecco perché questo libro è gratuito e lo sarà sempre.

Ora è arrivato il tuo momento: lotterai per tenerti stretta questa libertà? Cosa farai da oggi in poi affinché le cose cambino? Se vuoi combattere questa lotta con me ti chiedo allora di condividerlo con quanti più amici possibile, di farmi sapere cosa ne pensi e di supportare i progetti e le lotte che negli anni combattiamo per essere uomini liberi.

E ora vai a goderti la tua libertà. E non permettere a nessuno di fermarti.

RINGRAZIAMENTI

AUTORI E COLLABORATORI

Testi, Progettazione ed Esecuzione

Stefano Novelli

Audio Making e Recording

Mirko Marcattili

Video Making e Recording

Pasquale Giovine

Stampa del libro cartaceo

Cromaline di Leonardo Di Silvestre

Distribuito e promosso da

inforge.net - your hacks community

FONTI & RISORSE

- wikipedia.it per l'enorme quantità di informazioni, soprattutto sulle parti tecniche
- deepdotweb.com e in particolare la Jolly Roger's Security Guide for Beginners da cui ho preso spunto le storie dei vari cyber-criminali trattati
- torproject.org per le wiki che spiegano l'architettura della rete TOR
- privacytools.io per il resoconto sulla sorveglianza di massa e il riassunto dei punti fondamentali
- *Source Sans Pro*, *Oxygen* e *Ubuntu Mono* sono i font utilizzati per questo libro

SPECIAL THANKS

Il successo di questo progetto è stato possibile anche grazie ad alcuni dei più importanti portali del settore informatico che hanno messo a disposizione la loro visibilità mediatica. Senza di essi Hacklog: Volume 1 non avrebbe raggiunto questo importante traguardo. Grazie ancora.

Iffl linux freedom
News Dal Mondo Linux - Ubuntu

LEGA NERD
LEGANERD.COM

Over Security

tom's HARDWARE
THE AUTHORITY ON TECH

DONATORI

Il progetto Hacklog: Volume 1 è reso possibile grazie al contributo monetario concesso dalle persone qui presenti, dalla campagna Indiegogo[1] dell'Hacklog.

Donatori Diamond		
Lorenzo Pulcini	Roberto Talamonti	Francesco Buccoliero
Alex Fegatilli (Faustino50)	ddarix	Michele Colazzo
luca bizzotto	Kornel Roman	Mario Consorti
Francesco Pischedda	Gianluigi Frau	Andrea Sorrentino
Giorgio Vitale	Cristiano Alex Rado	Domenico Versace
Zanotti Andrea	Edoardo Piergentili	Luciano Barbato
Tony Fanara	heleentje64	Francesco Pxk
« MoMy »	Rossato Fabio	Federico Bevilacqua
Luca Baglivo	camap	Laempo L

Donatori 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	

[1] https://www.indiegogo.com/projects/hacklog-volume-1-videos-italy#/

Donatori 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 Zavanella	

Donatori 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	

Donatori 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	micheleee92	

I primi 50 Libri		
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 Vezzaro	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	