Treball Final de Carrera

Tails-Group-Installer

Carlos Aguado Navarrete

Març - Juny del 2015

Una aplicació per a la distribució Tails

Índex de continguts

1 Llicència

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for

this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

2 Visió general del treball realitzat

2.1 Introducció

Avui en dia la privacitat en la comunicació a través d'internet s'ha convertit en blanc de nombrosos atacs ja sigui per la necessitat de control d'aparells de seguretat estatals (veure el degoteig constant d'informacions arrel del cas Snowden) , com per l'ambició comercial de grans proveïdors de serveis “gratuïts” i la proliferació de tot un ventall de programari maliciós. En aquest context la creació d'eines per a un ús privat i segur de la xarxa és una necessitat de primer ordre. Tot i que no existeix la seguretat absoluta i de vegades pot semblar un treball en va tenint en compte les noves formes d'atac que es van desenvolupant, sí que hi ha certes eïnes amb les que fer front de forma força solvent a la qüestió de la privacitat.

Entenem privacitat com la capacitat per escollir quina informació compartim i amb qui la compartim. En aquest sentit per poder parlar de privacitat cal ser capaços de no compartir-ne cap. Més enllà de les dades personals (nom, edat, treball, etc..) és necessari no estar obligades a facilitar a tercers ni la ubicació física de la nostra connexió, ni el contingut d'aquesta.

Per sort ja hi ha eines desenvolupades que s'acosten a aquests objectius com són el xifrat de correus i xats en quant al contingut i diverses infraestructures de xarxa com proxies o la xarxa Tor en quant a dades sobre la connexió.

Tot i això la seguretat de la comunicació no és una qüestió que depengui d'un mateix sinó que es determina per les pràctiques del conjunt humà involucrat. En aquest sentit és on avui en dia trobem més dificultats doncs la velocitat i la complexitat amb la que avancen els canvis tecnològics fàcilment fan arribar a la majoria d'usuàries al llindar de la seva capacitat per absorbir nous coneixements, establint-se una actitud de poca pro-activitat a l'hora de configurar els dispositius que fan servir. Abordar aquest problema és una tasca prioritària en l'àmbit de la seguretat informàtica. U

En aquest sentit una de les formes de privacitat més robustes i senzilles de fer servir consisteix en l'ús de distribucions live específicament dissenyades a tal efecte. Aquestes solucions però resolen alguns problemes de sèrie: anonimat en la connexió a internet, no deixar rastre a l'ordinador però en deixen altres a la configuració posterior per part de l'usuari: intercanvi de claus, creació de comptes de correu, etc. L'objectiu general del projecte és avançar en la usabilitat d'aquest tipus de tecnologies ampliant el perfil d'usuaris capaç de fer-les servir de forma plena.

2.2 Motivació

He decidit realitzar aquest treball perquè em permet aprofundir en els camps que més m'interessen de la informàtica: la criptografia i els sistemes GNU/Linux. Alhora em serveix tant per iniciar-me en la col·laboració amb projectes més amplis com per aportar una solució a un problema amb el que m'he trobat personalment com és la implantació d'eines de comunicació segures en grups humans amb coneixements informàtics bàsics.

2.3 Objectius

El treball es centrarà en l'ampliació de funcionalitats de la distribució Live TAILS (The Amnesiac Incognito Live System). Actualment la distribució ofereix “Out of the Box” l'anonimat en l'accés a internet i la creació d'una partició xifrada per a l'emmagatzemament de dades persistents quan s'executa des d'una memòria USB. La configuració del correu xifrat o del xat queden però fóra de l'abast de la distribució i cal fer-les de forma manual per part de l'usuari. D'altra banda en el cas de voler configurar un sistema de comunicació per a un grup que implementi el correu i xats xifrats mitjançant l'ús de Tails el procés a realitzar es fa força complex i extens per a un usuari normal. El programari a desenvolupar automatitzarà aquesta tasca realitzant les següents accions:

- Creació de n imatges de sistema amb l'intercanvi de claus realizat prèviament.
- Creació de n comptes de correu electrònic.*
- Configuració del client de correu electrònic per l'us xifrat i signat entre els comptes creats anteriorment.
- Configuració del client Pidgin [1] amb la funcionalitat Off The Record [2] entre els usuaris.
- Creació de volum persistent de dades per emmagatzemar arxius.

* La creació de comptes automàtics presenta alguns problemes. O és necessària la col·laboració d'algun proveïdor de correu o cal confiar en proveïdors desconeguts. Aquest aspecte està per resoldre encara i es definirà a mesura que avanci el TFC.

2.4 Planificació inicial

La planificació incial que es va realitzar va ser la següent:

1 Organitzatives i d'anàlisi

1.1. Creació de document amb la proposta clara (threat model) per a enviar als desenvolupadors de Tails.

1. 2. Un cop rebuda la resposta determinar quin tipus de llenguatge de programació i la forma de treballar (localment, repositori online, etc)

1. 3. Respondre qüestions com:

- A nivell social: Com aconseguir que el programari pugui estar avalat per desenvolupadors reconeguts en l'àmbit de la llibertat i la privacitat?
- A nivell de recursos: Quines necessitats té el projecte per al seu desenvolupament i manteniment a mig plaç?

2 Desenvolupament

2.1. Creació de l'espai de treball necessari mitjançant màquines virtuals.
https://tails.boum.org/doc/advanced_topics/virtualization/virtualbox/index.en.html

2.2. Anàlisi del funcionament de la distribució i les diferents aplicacions per determinar quins algorismes són necessaris. Especificació completa.

2.3. Desenvolupament dels scripts / funcions necesssaris per a cada un dels objectius

2.4. Creació de l'aplicació global.

2.5 Tests

3. Documentació i difusió

3.1. Documentació

3.2. Lloc web per descarregar l'aplicació.

3.3. Video-presentació.

Data	**Tasques realitzades**
5 d'Abril:	1.1., 1.2, 1.3
20 d'Abril (PAC2)	2.1, 2.2
12 de Maig	2.3
29 de Maig (PAC3)	2.4, 2.5
12 de Juny (PAC4)	3.1, 3.2
19 de Juny (Lliurament final)	3.3

Aquesta planificació no ha pogut ser duta a terme. Tant per qüestions de caràcter personal i organitzatiu (laborals, salut, etc.) com per una valoració deficient de les dificultats que tenia el projecte. La interacció amb altres programes específics de Tails i la poca documentació al respecte han fet que hagués de passar força temps mirant el codi utilitzat, i fent tasques de prova i error més semblant a la enginyeria inversa que al desenvolupament pròpiament, que tot i que ha sigut una experiència satisfactòria i m'ha permès aprendre del codi d'altres ha suposat un càrrega de treball i de tensió força elevada.

2.5 Estructura del document

El document reflectirà el treball realitzat i es dividirà a continuació en quatre parts:

- En el capítol 2 es descriurà l'aplicació a desenvolupar partint d'un model d'amenaces i descrivint les funcionalitats que haurà de tenir l'aplicació, explicant les característiques

tècniques de cadascuna i amb quins arxius de configuració del sistema caldrà interactuar.
- En el capítol 3 s'explicarà el treball de desenvolupament realitzat incloent la investigació sobre Tails i el programari més significatiu amb el que interactuar.
- En el capítol 4 es parlarà dels tests realitzats ja sigui de forma automàtica com manual.
- En el capítol 5 es farà una valoració sobre el treball realitzat, especificant els objectius aconseguits, el treball pendent i com es pensa afrontar el desenvolupament en un futur.
- En els annexos trobarem alguna informació d'interès en quant a la relació amb l'equip de desenvolupadors de Tails.

3 Aplicació a desenvolupar: Tails-group-installer.

L'aplicació a desenvolupar s'anomenarà Tails-group-installer doncs estarà destinada a la instal·lació de Tails per a grups.

3.1 Model d'amenaces.

Quan treballem en temes relacionats amb la seguretat informàtica cal definir un model d'amenaces que delimiti què volem protegir i què no i de qui. En el cas del TGI hem volgut definir un model molt restrictiu on no calgui confiar en cap proveïdor de serveis extern.

3.1.1 Què volem protegir?

Identitat dels membres del grup: No ha de ser possible esbrinar la identitat real del usuaris.

Localització dels membres del grup: des d'on es comuniquen amb la resta d'usuaris.

Contingut de les comunicacions: que s'envien per correu o per xat.

3.1.2 De qui i què?

Un observador local de la xarxa: definim observador local com algú que pugui estar connectat a la mateixa LAN o un proveïdor de serveis d'internet.

Anàlisi forense de l'ordinador: la informació manipulada durant la sessió no ha de poder ser accessible posteriorment mitjançant l'anàlisi forense de l'equip.

Recuperació de les dades emmagatzemades a la memòria USB: tota informació emmagatzemada a la memòria USB ha d'estar xifrada de forma segura.

Errors en la configuració manual del correu i la missatgeria instantània: la configuració dels programes ha de ser correcte per defecte impedint que una mala configuració d'un usuari posi en perill la resta del grup.

Proveïdors de serveis de e-mail i XMPP: no s'ha de confiar en la correcta manipulació de la informació per part dels proveïdors de serveis per ben intencionats que siguin.

3.1.3 De qui i què no?

Observador global de la xarxa: entenem com a observador global aquell que pot observar un percentatge prou elevat de la xarxa global (internet) per mitjançant l'anàlisi del moviment de paquets pugui correlacionar un usuari amb una ubicació física.

Coherció legal o violenta: no podem assegurar-nos davant estratègies de forta intimidació.

Membre del grup que actua amb traïdoria i que o bé subministra l'equip o el pendrive utilitzats per fer la instal·lació: no podem assegurar-nos davant d'un us maliciós del nostre programari o del maquinari que farem servir per executar el programa.

Els propis usuaris: trencant l'aïllament d'identitat mitjançant un mal ús del correu o el navegador d'internet: no podem assegurar a algú que deliberadament utilitza de forma clarament incorrecte el programari.

3.1.4 Proveïdors de serveis

En la versió actual del model d'amenaces es troben dins del camp de les amenaces tot i que comptem amb ells per a obtenir els serveis i valorem molt el seu treball en alguns casos, com el de riseup.net on crearem els comptes necessaris. Tot i que riseup és un proveïdor de serveis reconegut i amb una llarga trajectòria dins la lluita per la privacitat, confiar-hi plenament i sense implementar xifrat de al banda de l'usuari implica depositar la seguretat en mans d'uns equips i uns administradors de sistemes que no coneixem directament i que podrien estar compromesos en algú moment.

Incloure o no dins de les amenaces té unes conseqüències importants en quant al tipus de serveis que podrem oferir doncs el xifratge en grup està poc desenvolupat en quant a dos dels serveis que ens agradaria com son el xat o l'edició de documents compartida. Ambdós s'ofereixen via connexions https i poden implementar una contrasenya però el xifrat recau en el servidor.

3.2 Cas d'ús

Ens centrarem amb un cas d'ús concret i és la creació de n memòries USB amb Tails instal·lat i les claus de xifrat i programari de correu i xat configurats plenament per a un grup d'usuaris que es troben presencialment al menys un cop.

Tot i que ens pugui semblar un cas d'ús estrany, ho és en quant a la dificultat actual que comporta fer la correcta configuració. Però si en facilitem el procés pot ser força utilitzat en ambients on calgui crear sistemes de comunicació segurs i on les identitats dels membres no pugui relacionar-se amb altres aspectes de la seva activitat. Amb això entenem que els comptes que es facin servir han de ser només per a aquest ús.

A continuació una taula amb els passos que s'han d'executar de forma manual i el temps estimat:

Clonació de Tails a les unitats USB	7 minuts per unitat
Arrencada de la Tails a cada unitat i configuració de la	4 minuts per unitat

persistència	
Rei nici de Tails amb la persistència ja activada	4 minuts per unitat
Creació de claus GPG per cada usuari	2 minuts per usuari
Exportació de claus pública de cada usuari	2 minuts per usuari
Enviament per correu o mitjançant una memòria extraïble de les claus públiques de cada usuari a cada usuari.	2 minuts per cada parella d'usuaris.
Importació de les claus públiques	2 minuts per cada parell d'usuaris
Comprovació de l'autenticitat de les claus públiques	2 minuts per cada parell d'usuaris
Importació de les claus públiques	1 minut per cada usuari
Configuració del client de correu. (Icedove)	5 minuts per usuari
Configuració del client de xat (Pidgin)	5 minuts per usuari
Comprovació de la veracitat dels altres usuaris a Pidgin	3 minuts síncrons per usuari

En el cas de disposar d'un sol ordinador i que s'hagin de crear 10 usuaris el procediment es pot allargar durant força hores o requerir fer segons quins passos posteriorment.

3.3 Guió bàsic de funcionament

En el nostre cas el programari seguirà el següent procés:

1	Mostrar requeriments i advertències de seguretat	1 minut
2	Demanar el nombre de membres del grup i la clau de Root de Tails	1 minut
3	Demanar la clau de root de Tails	1 minut
4	Demanar al primer membre que introdueixi les dades dels comptes de correu i xmpp. En el cas de riseup permet l'ús de del mateix compte pera als dos serveis.	2 minuts
5	Demanar la frase de pas per a la clau GPG.	1 minut
6	Iteració dels punt 4 i 5	
7	Creació de les claus GPG dels usuaris i l'anell públic comú	2 minuts per usuari
8	Creació de les carpetes dels perfils de Icedove i Pidgin	1 minut per usuari
9	Demanar la memòria USB del primer usuari	1 minut
10.	Clonar Tails	7 minuts per usuari
11.	Creació volum persistent	1 minut

12	Còpia del perfil de l'usuari	1 minut
13	Iterar entre 9 i 12 fins acabar amb tots els usuaris.	

3.4 Funcionalitats

A continuació una descripció de les funcionalitats que ha de tenir el programa i la interacció que ha de tenir amb la resta d'aplicacions.

3.4.1 Interfície gràfica

Descripció: Donat que un dels interessos del treball és facilitar-ne l'ús a persones inexpertes caldrà que la interfície del programa sigui gràfica i el més senzilla possible d'utilitzar acompanyant el procés amb expilacions sobre el que s'està fent. Haurà doncs d'informar dels passos, demanar dades als usuaris, emmagatzemar-les i fer-les disponibles per a la resta de sub-programes.

Tecnologia a utilitzar: es treballarà amb `Gtk3+` per facil·litar la seva interacció i inclusió en un futur a Tals. Veure més endavant l'anàlisi de Tails a l'apartat 3.4

3.4.2 Creació de claus GPG

Descripció: caldrà crear claus GPG per a cada usuari amb la frase de pas introduïda per l'usuari. Alhora es crearà un anell de claus públiques compartit amb totes les claus de la resta d'usuaris. Aquestes claus estaran marcades com a claus de confiança.

Tecnologia a utilitzar: es treballarà amb GnuPG ja sigui directament mitjançant l'execució de comandes a una shell com mitjançant el paquet `Python-gnupg`.

3.4.3 Configuració correu xifrat: Icedove i Enigmail

Descripció: es crearan els corresponents perfils d'usuari consistents en la carpeta de perfil per defecte i les preferències amb els comptes ja configurats de forma que quan els usuaris obrin el programa no necessitin introduir més que la contrasenya per poder enviar correus de forma segura a la resta d'usuaris.

Tecnologia a utilitzar: es treballarà amb les utilitats de còpia i modificació d'arxius que disposen `Python` i `Bash`.

3.4.4 Instal·lació d'Icedove i Enigmail.

Descripció: actualment Tails ve amb Claws-mail com a client de correu. Donats els problemes que està tenint en quant a la privacitat i el rendiment està prevista la seva substitució per Icedove (la versió GNU de Thunderbird) i Enigmail (plugin per a la gestió del xifrat). Això implica que temporalment caldrà descarregar-se i instal·lar el software abans de fer servir l'aplicació. Un cop descarregada,l'aplicació s'instal·la per defecte en iniciar el sistema.[3] En cas de tenir i activada la

persistència de Apt a Tails, es pot fer servir el paquet descarregat anteriorment. Per tal d'agilitzar el procés es copiarà el paquet descarregat a totes les unitats.

Per emmagatzemar els perfils d'usuari d'aplicacions no instal·lades de sèrie a Tails cal fer servir la carpeta anomenada dotfiles on tots els arxius o directoris que s'introdueixin seran enllaçats a la home de Tails (/home/amnesia). En el cas de l'Icedove haurem de crear una carpeta anomenada ".icedove" que situarem a la carpeta dotfiles del volum persistent.

D'altrabanda cal configurar Icedove per a que faci servir Tor, això es fa modificant el seu proxy a: tipus: SOCKS5, IP: 127.0.0.1, Port: 9061. [4]

3.4.5 Creació de claus DSA per a Pidgin OTR

Descripció: Pidgin fa servir claus DSA per l'autenticació del usuaris. Cal crear un parell de claus DSA i calcular els fingerprint de les claus públiques de cada usuari.

Tecnologia a utilitzar: `openssl` permet la creació de les claus esmentades. Se'n farà ús mitjançant un script a cridar des de l'aplicació.

3.4.6 Configuració xat OTR

Descripció: cal crear la carpeta de cada perfil amb la configuració de cada usuari i especialment l'arxiu on s'emmagatzemen els fingerprints dels usuaris verificats.

Tecnologia a utilitzar: es treballarà amb les utilitats de còpia i modificació d'arxius que disposen `Python` i `Bash`.

3.4.7 Clonació del sistema

Descripció: Es tracta de clonar el sistema des d'on s'executa l'aplicació a una altra memòria. Només es clona la imatge del sistema i no els axius que pugui haver en la partició persistent.

Tecnologia a utilitzar: La clonació del sistema des d'un pendrive a un altre està contemplada actualment en Tails així que només haurem de fer servir la pròpia funcionalitat des d'un script.

3.4.8 Creació de volum persistent personalitzat.

Descripció: La creació de la partició persistent normalment es fa de forma gràfica mitjançant una GUI i només es pot fer amb el pendrive on està instal·lat el sistema. Ara bé, l'apliacatiu permet (prèvies modificacions) fer servir altres dispositius. Un cop creat el volum persistent caldrà copiar els perfils de forma correcte per a que siguin reconeguts pel sistema a l'inciar-se.

Tecnologia a utilitzar: per la creació del volum persistent caldrà interactuar amb el programari existent (Tails-persistence-setup) preferiblement de forma transparent a l'usuari. Per a la còpia dels arxius s'utilitzaran aplicacions com `rsync`.

4 Desenvolupament de l'aplicació

4.1 Configuració de l'entorn de treball

Per a la realització del treball s'han utilitzat el següent programari i configuració.

4.1.1 Eclipse + PyDev

Per tal de poder escriure l'aplicació i poder investigar altres parts del codi de Tails escrites en Python s'ha utilitat l'IDE `Eclipse` i l'extensió `PyDev`.

4.1.2 Virt-manager

La creació d'imatges o l'arrencada de sistemes Live en ordinadors són tasques força lentes per a treballar i anar provant el programari mentre es va creant. Per tal de mitigar l'impacte que té això sobre la dinàmica de treball s'ha creat un entorn de treball virtual basat en discs virtuals. Aquest entorn havi de permetre la relació entre una imatge de disc virtual amb una versió de Tails amb la persistència activada (que anomenarem (Disc1)) i un disc virtual buit en el que poder clonar i comprovar la configuració (Disc2).

D'aquesta forma s'han creat dues màquines virtuals:

TailsAB: conté els dos discs virtuals (1 i 2)

TailsB: conté només el Disc2.

Per tal de poder passar el codi actualitzat de l'entorn de programació a les màquines virtuals s'ha utilitzat un senzill script i la comanda `scp.`

Els requeriments de Tails per poder treballar amb particions persistents mitjançant màquines virtuals només els compleix actualment virt-manager aixií que s'ha optat programari en detriment d'altres opcions multi-plataforma com VirtualBox. [2]

4.1.3 Git

L'equip de Tails treballa mitjançant uns repositoris públics que es poden trobar a: https://git-tails.immerda.ch/tails|. A part del repositori general he instal·lat el que té a veure amb la persistència que serà el que possiblement em requereixi un anàlisi més profund doncs en caldrà utilitzar les seves funcions possiblement.

Per tal de poder compartir el meu propi treball i crear un wiki on poder anar explicant els passos que vaig fent i les decisions de disseny que he anat prenent, he creat un repositori a GitHub. Amb el nom de tails-group-installer. [0] S'han creat les pàgines del wiki en anglès per facilitar la col·laboració.

La idea en iniciar el treaball era poder comptar amb la col·laboració dels desenvolupadors de Tails per a la resolució de dubtes i l'auditoria del codi. Per qüestions de temps, tant els ritmes propis del

TFC com de les tasques necessàries per al manteniment de Tails i la llista de prioritats que els desenvolupadors tenen, aquesta col·laboració no ha sigut massa estreta.

Un cop finalitzada la entrega espero continuar amb el desenvoluapment i pujar la primera versió al repositori públic.

4.2 Investigació sobre el funcionament de Tails.

Per poder realitzar el programari ha sigut necessària una immersió en el codi de Tails i la seva documentació. A continuació i de forma esquemàtica els resultats d'aquesta investigació.

4.2.1 Funcionament general

En quant al funcionament general cal destacar la diversos aspectes:

- Sistema basat en Debian Live, actualment en Wheezy, tot i que es preveu la migració a Jessie en properes versions. Això implica que el programari desenvolupat hagi de ser compatible amb les versions de python i gtk instal·lades (2.7 i 3.4 respectivament).

- Es tracta d'un sistema live que tot i que permet l'ús de persistència per la configuració de diverses aplicacions (no començar de zero cada cop) no permet la modificació de la imatge del sistema ni la inclusió de forma directa de nou programari.

- Tot el trànsit de xarxa es realitza mitjançant el software Tor i un proxy intern que gestiona l'accés a la xarxa. Cal configurar les aplicacions que s'instal·lin a tal efecte.

- Per generar l'entropia necessària per l'ús de Tor inclou Haveged el que ens facil·lita alhora la creació de les claus GPG de forma automàtica per a un número elevat d'usuaris sense haver de generar-la de forma manual movent el ratolí, etc.

- Bàsicament Tails es una imatge de Debian Wheezy, una sèrie de scripts en forma de includes chroot que permeten configurar-la adequadament i algun programari extra com Tor Browser, Vidalia, i tres aplicacions desenvolupades expressament amb les que cal interactuar pel funcionament correcte de l'aplicació i que es descriuen a continuació:

4.2.2 Tails-persistence-setup

Es tracta d'una aplicació escrita en `Perl + Gtk2` encarregada de crear la partició persistent i de configurar l´´us. Per fer-ho calcula l'espai lliure, reserva una quantitat d'espai determinada per a possibles actualitzacions del sistema i crea un volum xifrat LUKS i dins seu una sistema de fitxers Ext4.

Si s'executa amb el llançador ubicat al menú d'aplicacions només permet configurar el volum a la memòria USB des de la que s'està executant. Executada des de la línia de comandes permet sobrecarregar la informació sobre a quina partició es troba el sistema i configurar un volum persistent a una altra unitat on prèviament s'hagi instal·lat Tails.

Un cop creada la partició una interfície gràfica permet configurar quines carpetes seran persistents.

Això ho realitza modificant un arxiu de configuració anomenat `persistence.conf`.

4.2.3 Tails-installer

Creat a partir d'un fork de `liveusb-creator` de `Fedora`. Escrit en `Python + Qt`. Permet la instal·lació d'una ISO a una memòria extraïble. En el cas de Tails-installer dona diverses opcions:

- Clonació del sistema
- Clonació i actualització a una memòria on ja hi ha Tails instal·lat.
- Actualitazció d'una memòria amb Tails ja instal·lat mitjançant una ISO.

Mitjançant comandes es pot llançar l'aplicació amb alguna d'aquestes opcions ja definides.

4.2.4 Tails-greeter

S'encarrega de donar la benvinguda a l'usuari i de gestionar la partició persistent mitjançant l'aplicació `live-persist`. Escrit en Python + Gtk3, més enllà de les comprovacions que fa sobre la partició persistent i que com veurem cal tenir molt en compte és interessant la implementació que fa de les finestres traduibles mitjançant classes que seria reutilitzat en versions futures de l'aplicació però que per qüestió de tempos no ha sigut possible fins ara. Al mateix temps a nivell gràfic és important mantenir una coherència dins el sistema. La seva existència ha sigut una de les raons més importants per triar Python i Gtk com a llenguatge de programació.

4.3 Llenguatge de programació

S'ha escollit Python i Gtk per permetre la futura integració amb altres aplicacions de Tails relacionades com les descrites anteriorment. Caldria veure si finalment s'accepta la inclusió de l'aplicació a Tails com s'en faria la intergració si com a programari autònom que utilitzi alguns mètodes i classes d'altres aplicacions (Tails-greeter i Tails-installer) o com ampliació de Tails-Installer.

Per crear les finestres de form àgil s'ha utilitzat Glade.

4.4 Programari desenvolupat.

4.4.1 Descripció general

S'ha realitzat un disseny orientat a objecte mitjançant els següents grups de classes:

1. **Tails-group-installer**: c lasse principal que executa Gtk i que importa la resta de classes.

2. **User**: classe model on s'emmagatzem les dades dels usuaris.

3. **Classes gestores de finestres**: una classe per a cada finestra que apareix en executar l'aplicació i que gestionen la interacció amb l'usuari: welcomewindow, userwindow, persistclonewindow,

goodbyewindow.

4. **Classes d'utilitats**: proporcionen els mètodes per a cadascuna de les funcionalitats: gpgtools, persistencetools, icedovetools, clonetools i pidgintools.

A continuació una descripció de la implementació i funcionalitat de les diferents classes i mètodes més rellevants:

4.4.2 Tails-group-installer

Arrenca l'aplicació mitjançant la importació de les diferents finestres i utilitats i inicailitza algunes variables. A continuació ens mostra la finestra de benvinguda.

4.4.3 User

Classe que emmagatzema les dades d'usuari en els següents atributs:

name = nom de l'usuari

account = compte de correu i XMPP de l'usuari

passphrase = frase de pas per a la clau GPG

gpg_key = clau GPG de l'usuari.

keyid = ID de la clau GPG

pub_key_armored = clau públic en format ASCII, utilitzada per la seva importació a l'anell públic comú.

fingerprint = fingerprint de la clau GPG

path_home = directori on s'emmagatzemarà provisionalment tots els arxius referents a l'usuari. Es construeïx mitjançant el nom d'usuari a la carpeta home de l'usuari amnesia que és des del que s'executa l'aplicació (/home/amnesia/"+name)

icedove_folder_name = nom de la carpeta del perfil de Icedove de l'usuari. Format per una cadena de 8 caràcters aleatoris de lletres minúscules i números i afegint ".default". Per generar aquest atribut també podem trobar el mètode:

create_random_id(self): crea la cadena aleatòria per a l'atribut anterior.

4.4.4 Welcomewindow

Implementa la finestra de benvinguda a través de l'arxiu de Glade corresponent. La seva funció és avisar dels requeriments i obtenir el nombre d'usuaris a configurar i la contrasenya de root que l'usuari ha d'haver introduït a l'iniciar Tails mitjançant el Tails-greeter.

Els requeriments són els següents:

- un compte de riseup per usuari: preferiblement no utilitzat amb anterioritat

- 1 memòria USB per usuari de mínim 4GB de capacitat
- Temps: 5 minuts generals i 5 minuts més per usuari.

Si no els complim ens ofereix sortir, i, en el cas de no tenir els comptes de riseup però tenir 4 invitacions ens dona la opció de clicar un botó que ens hauria de portar a una pàgina de documentació que explica com crear els comptes necessaris. (NO IMPLEMENTAT).

Un cop introduïm les dades cliquem a continuar i la finestra dona pas a la següent: Userwindow

4.4.5 Userwindow

Es tracta de la finestra mitjançant la qual es crearan els comptes d'usuari i en finalitzar les claus GPG de cada usuari i l'anell públic comú.

Les dades que demana són les següents i són comprovades les següents restriccions:

Name: nom de l'usuari, no pot estar buit ni tampoc pot coincidir amb cap usuari creat anteriorment.

Account: compte de correu de riseup.net. No pot coincidir amb cap altre anterior. Es comprova que el compte estigui introduït correctament és a dir que termini amb la cadena “@riseup.net”

GPG passphrase: frase de pas per a desbloquejar la clau GPG que es crearà en el següent pas. Ha de tenir un mínim de caràcters (actualment definit a 15) i s'ha de repetir de forma correcte.

La finestra va demanant l'entrada de dades dels usuaris fins que s'ha arribat al nombre predefinit anteriorment. En aquest moment desapareix el formulari d'entrada de dades d'usuari i se'ns informa del nombre de comptes que es crearan i es demana pulsar un botó per començar a crear les claus GPG. També s'informa de la durada estimada de la creació dels comptes.

Un cop creades les claus s'informa de que han estat creades amb èxit i es passa a la següent fase. La creació de les memòries USB.

4.4.6 Persistclonewindow

La funcionalitat consta de tres etapes:

- Clonar el sistema
- Crear la partició persistent
- Configurar la partició persistent.

Actualment per a les dues primeres opcions s'utilitzen les aplicacions natives de Tails executant-les gràficament (no incorporen cap altre possibilitat de moment ni he tingut el temps per integrar-les de forma més eficient. Tot i això la interacció amb l'usuari era necessària igualment per introduir la memòria USB).

El que si es fa és passar els paràmetres adequats per a:

- Clonar el sistema: s'executa directament el la funció Clonar el sistema (recordem que el

Tails-installer consta de tres opcions).

- Crear la partició persistent: s'executa només la part de creació de la partició doncs de la configuració s'encarrega el següent pas de forma automàtica.

A l'apartat persistencetools veurem amb més deteniment totes aquestes crides.

Al finalitzar l'operació es demana el següent usuari fins que finalitzem amb tots ells i passem a la finestra de comiat.

4.4.7 Goodbyewindow

Ens informa de que la creació de tots els usuaris ha finalitzat i que ho ha fet de forma correcte. Ens informa de què ens trobarem exactament quan arrenquem un ordinador amb la memòria creada.

4.4.8 Gpgtools

Mòdul basat en el paquet python-gnupg. En aquest mòdul trobem els mètodes necessaris per a la creació de les claus GPG. Els més rellevants són els següents:

4.4.8.1 create_gpg_profile

La creació de les claus es fa pel mètode de creació batch amb el que definim els paràmetres prèviament i els passem al mètode gpg.gen-key.

Prèviament definim la carpeta on s'emmagatzemaran les claus mitjançant el paràmetre gnupghome.

Un cop creada la clau alguns paràmetres derivats d'ella els passem a l'objecte usuari corresponent per tal de ser utilitzats posterioment.

```
    def create_gpg_profile(self, user):
        """
        Create gpg keyring for user
        """

        self.user = user
        gpg = gnupg.GPG(gnupghome=user.path_home+"/gnupg", verbose=True)
        logging.debug('Imported gnupg')
        batch_data = self.create_gpg_batch(user)
        logging.debug('Created batch data')
        user.gpg_key = gpg.gen_key(batch_data)
        user.keyid = gpg.list_keys()[0]['keyid']
        logging.debug("User keyid: "+user.keyid)
        user.pub_key_armored = gpg.export_keys(user.keyid)
        user.fingerprint = gpg.list_keys()[0]['fingerprint']
        logging.debug(user.fingerprint)
        logging.debug(user.name+" gpg profile created")
```

4.4.8.2 create_gpg_batch

Mètode per a la creació de les dades que necessitem per crear la clau.

```
    def create_gpg_batch(self,user):
        gpg = gnupg.GPG(gnupghome=user.path_home+"/.gnupg")
```

```
        user_data = {'name_real' : user.name,
                     'name_email' : user.account,
                     'expire_date' : 0,
                     'key_type' : 'RSA',
                     'key_length' : 4096,
                     'subkey_type' : 'RSA',
                     'subkey_length' : 4096,
                     'subkey_usage' : 'encrypt,sign,auth',
                     'passphrase' : user.passphrase
                     }
        batch_data = gpg.gen_key_input(**user_data)
        return batch_data
```

4.4.8.3 create_gpg_pubring

Importa les claus públiques a l'anell públic comú mitjançant la clau en format ASCII exportada anteriorment. En arrencar Tails aquest ja conté un anell públic amb les claus dels desenvolupadors. Per tal de no perdre-les utilitzem el mateix anell afegint les que hem creat.

```
    def create_gpg_group_pubring(self, users):
        """
        Modify users gpg pubring and trust keys
        """
        # Import pubkeys to pubring already existent in Tails
        gpg = gnupg.GPG(gnupghome='/home/amnesia/.gnupg')
        for user in users:
            gpg.import_keys(user.pub_key_armored)


        logging.debug("Group pubring created")
```

4.4.8.4 create_trustdb

Una de les avantatges que té compartir les claus públiques creant-les totes a l'hora amb la aplicació és que podem establir la confiança màxima sense haver de fer posteriors comprovacions. Això ens permetrà que estem segurs que les signatures que rebem amb els correus electrònics siguin vàlides i ningú està suplantant el compte.

No hi ha forma directa no interactiva per atorgar confiança a les claus a python-gnupg ni tampoc mitjançant gpg. La solució passa per utilitzar la funcio d'importació de confiança (gpg –import-ownertrust) sobre un arxiu creat amb els fingerprint i el valor de confiança de les diferents claus públiques. Hem assignat la màxima confiança (6).

```
    def create_trustdb(self,users):
        with open('/home/amnesia/.gnupg/ownertrust.txt', 'w') as f:
            for user in users:
                f.write(user.fingerprint+":6:\n")

        os.remove("/home/amnesia/.gnupg/trustdb.gpg")
        proc = subprocess.Popen(
        [
                "gpg", "--import-ownertrust",
                 "/home/amnesia/.gnupg/ownertrust.txt"
```

4.4.9 Clonetools

4.4.9.1 clone_tails

Mètode per clonar Tails. Com s'ha explicat anteriorment s'utilitza el liveusb-creator que també fa servir Tails-installer però arrencant directament amb la opció de clonar i altres paràmetres necessaris:

-u: permet executar-lo sense permisos de root. **-n:** no verifica la ISO (es la que estem executant)

-P: particiona el dispositu **-m:** reseteja la MBR

-x: deshabilita el suport per a OLPC (distribució referent a One Laptop per Child)

```
    def clone_tails(self):
        proc = subprocess.Popen(
            [
                "liveusb-creator", "-u", "-n",
                "--clone", "-P", "-m", "-x"
            ],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE
            )
        logging.debug("subprocess clone open")
        out, err = proc.communicate()
        logging.debug("out: "+out)
        logging.debug("err: "+err)
```

4.4.10 Persistencetools

S'utilitzen bàsicament mòduls per copiar i modificar directoris i la pròpia eïna de creació de persistència de Tails.

4.4.10.1 create_partition

S'ocupa de executar el creador de persistència apuntant a la memòria USB que acaba de ser clonada, per tal de fer-ho és necessari sobrecarragar el dispositiu on l'aplicació creu que hi té el sistema. Mitjançant la opció “--step bootsrap” fem que només executi la creació de la partició i no el proces de configuració.

```
    def create_partition(self):

        proc = subprocess.Popen(
            [
                "/usr/bin/tails-persistence-setup","--force",
                "--override-boot-device","/org/freedesktop/UDisks/devices/sdb",
                "--override-system-artition",
"/org/freedesktop/UDisks/devices/sdb1",
                "--step","bootstrap"
            ],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE
            )
```

```
        out, err = proc.communicate()
        logging.info("out: "+out)
        logging.info("err: "+err)
```

4.4.10.2 configure_partition

Mètode que crida tots els passos necessaris per a la configuració de la partició:

```
    def configure_partition(self,user):
        self.user = user
        logging.info("Configuring partition for user: "+self.user.name)
        self.mount_point = self.mount_point + self.user.name
        os.mkdir(self.mount_point)
        logging.debug("Created mount_point")
        self.mount_partition()
        logging.debug("Mounted partition")
        self.make_partition_writeable()
        logging.debug("Partition writeable")
        self.copy_files(self.user)
        logging.debug("Files copied")
        self.fix_permissions()
        logging.debug("Permissions fixed")
```

4.4.10.3 mount_partition

Muinta la partició persistent creada al path específic de l'usuari. Aquesta comanda necessita permisos de root i se'ls passem mitjançat un PIPE.

```
    def mount_partition(self):

        proc = subprocess.Popen(
            [
                "sudo", "-S",
                 "/bin/mount","/dev/dm-1",
                self.mount_point
            ],
            stdin=subprocess.PIPE,

                                  )
        proc.communicate(input=config.ROOT_PASSWORD + "\n")
```

4.4.10.4 make_partition_writeable

Necessària per poder copiar els arxius des de l'usuari amnesia en el pas següent:

```
    def make_partition_writeable(self):

        proc = subprocess.Popen(
            [
                "sudo", "-S",
                 "/bin/chown","amnesia:amnesia",
                self.mount_point
            ],
```

```
            stdin=subprocess.PIPE,
                                )

        proc.communicate(input=config.ROOT_PASSWORD + "\n")
```

4.4.10.5 copy_files

Copiem totes les dades necessàries per a l'usuari consistent en perfils de les aplicacions i arxius de configuració que s'han anat creant durant tot el procés o estaven predefinits a la carpeta “defaults” de l'aplicació.

```
    def copy_files(self,user):
        #Substitue user's pubring by group's pubring
        shutil.copy2("/home/amnesia/.gnupg/pubring.gpg",
user.path_home+"/gnupg/pubring.gpg")
        shutil.copytree(self.user.path_home+"/gnupg", self.mount_point+"/gnupg")
        shutil.copytree(self.user.path_home+"/dotfiles",
self.mount_point+"/dotfiles")
        shutil.copytree("defaults/apt",self.mount_point+"/apt")
        shutil.copytree("defaults/Persistent",self.mount_point+"/Persistent")
        shutil.copy2("defaults/persistence.conf",self.mount_point)
        shutil.copy2("defaults/live-additional-software.conf",self.mount_point)

        proc =subprocess.Popen(
            [
                "sudo", "-S",
                "./rsync_apt.sh", self.mount_point
            ],
            stdin=subprocess.PIPE,
                                )

        proc.communicate(input=config.ROOT_PASSWORD + "\n")
```

L'execució de rsync_apt.sh ens permet copiar els paquets i llistes d'apt descarregats per instal·lar l'Icedove i Enigmail a les mèmories del ususaris. A continuació el contingut de l'script:

```
#!/bin/bash

set -e

# Copy apt downloaded packages into new device

MOUNT_POINT=$1

rsync -av /var/cache/apt/archives/ $MOUNT_POINT/apt/cache
rsync -av /var/lib/apt/lists/ $MOUNT_POINT/apt/lists
```

4.4.10.6 fix_permissions

Per tal de que la partició persistent sigui reconeguda i accepatada pel Tails-greeter aquesta ha de complir uns requisits en quant a propietari, mode i ACL. En cas de fallar qualsevol d'ells la persistència es inhabitada per evitar possibles modificacions malicioses. Mitjançant l'script següent

s'arreglen els permisos i finalment es desmunta el volum.

```
    def fix_permissions(self):

        proc =subprocess.Popen(
            [
                "sudo", "-S",
                "./fix_permissions.sh", self.mount_point
            ],
            stdin=subprocess.PIPE,
                                )

        proc.communicate(input=config.ROOT_PASSWORD + "\n")
```

A continuació l'script cridat: fix_permissions.sh

```
#!/bin/bash

set -e

MOUNT_POINT=$1

setfacl -m m:rwx $MOUNT_POINT
setfacl -m g::rwx $MOUNT_POINT
setfacl -m u:tails-persistence-setup:rwx $MOUNT_POINT

chown root:root $MOUNT_POINT
chmod 775 $MOUNT_POINT

cd $MOUNT_POINT
chmod 600 persistence.conf
chmod 600 live-additional-software.conf
chown tails-persistence-setup:tails-persistence-setup persistence.conf
chown tails-persistence-setup:tails-persistence-setup \
live-additional-software.conf
chown amnesia:amnesia Persistent
chmod 700 Persistent
chmod 700 lost+found
chmod 700 gnupg
chmod 700 dotfiles

ls -hal
cd
umount $MOUNT_POINT
```

4.4.11 Icedovetools

Es tracta dels mètodes relatius a la creació del perfil de Icedove. Bàsicament copiem una carpeta de perfil per defecte i modifiquem les preferències amb les dades de l'usuari.

4.4.12 create_icedove_folder

Simpelment crida en ordre als mètodes necessaris

```
    def create_icedove_folder(self,user):
```

```
        logging.info("Start creation of icedove folder for user:
"+self.user.name)
        self.create_profile_folder(user)
        logging.info("profile folder created")
        self.create_ini_file(user)
        logging.info("ini file created")
        self.modify_prefs(user)
        logging.info("prefs modified")
```

4.4.13 create_profile_folder

Còpia la carpeta base a la home de l'usuari amb el nom del perfil d'icedove adequat.

```
    def create_profile_folder(self,user):
        shutil.copytree(self.icedove_profile, user.path_home +
"/dotfiles/.icedove/" + user.icedove_folder_name)
```

4.4.14 create_ini_file

Crea l'arxiu profiles.ini on l'Icedove busca el nom del perfil per defecte i la seva ubicació:

```
    def create_ini_file(self,user):
        f = open(user.path_home +  "/dotfiles/.icedove/profiles.ini","w")
        t = """[General]
StartWithLastProfile=1

[Profile0]
Name=default
IsRelative=1
Path="""+user.icedove_folder_name+"\n"

        f.write(t)
        f.close()
```

4.4.15 modify_prefs

Modifica l'arxiu prefs.js on s'emmagatzemen les dades dels comptes configurats així com els paràmetres de configuració tant del Icedove com de Enigmail. Especialment importants els relatius a el proxy local (sinó no funcionarà Icedove) i l'ús forçat del xifrat per part d' Enigmail.

Per aconseguir l'arxiu base s'ha creat una configuració correcte i comparat amb la inicial s'han trobat les línies a afegir i modificar marcant amb les etiquetes: “riseup_mail_account” i “icedove_folder_name” els valors especics a canviar.

```
    def modify_prefs(self,user):

        prefs = user.path_home + "/dotfiles/.icedove/" +
user.icedove_folder_name + "/prefs.js"

        for line in fileinput.input(prefs,1):
            print line.replace("riseup_mail_account",user.account.split("@")
```

```
[0]),

        for line in fileinput.input(prefs,1):
            print line.replace("icedove_folder_name",user.icedove_folder_name)
```

4.5 Pidgintools

No s'ha pogut implementar la part de l'aplicació relativa a Pidgin. Donada la manca de temps s'ha preferit implementar una de les dues aplicacions fins al final sabent que la implementació de l'altre no deferirà massa. Els mètodes sí que s'han definit de forma genèrica.

```
    def create_pidgin_folder(self,user):
        ''' Creates profile folder for pidgin
        '''
        self.create_otr_private(user)
        self.create_accounts_xml(user)
        self.create_otr_fingerprints(user)

    def create_otr_fingerprints(self,user):
        ''' Creates file with fingerprints of other users
        '''
        pass

    def create_otr_private(self,user):
        '''Create private profile for user
        '''
        pass

    def create_accounts_xml(self,user):
        ''' Create account file for user
        '''
        pass
```

En els anexos es pot trobar l'explicació dels arxius corresponents i

4.6 dsa_tools

Eines per a la creació de les claus corresponents a Pidgin OTR. Veure explicació anterior. Definició dels mètodes i possibles comandes necessàries:

```
   def gen_param(self, user):
        '''
        openssl genpkey -genparam -algorithm DSA out tfctestdsap.pem -pkeyopt
dsa_paramgen_bits: 1024
        '''
        pass

    def gen_key_text(self, user, params):
        '''
        Generates key with params created with gen_param
```

```
        openssl genpkey -paramfile tfctestdsap.pem out tfctestdsak.pem -text
        '''
        pass

    def get_fingerprint(self,user):
        '''
        ssh-keygen -l -f testsshkeygen.pub
        '''
        pass
```

5 Tests i gestió d'errors

Donat el caràcter de l'aplicació i la manca de temps no s'han pogut crear tests de forma automàtica. Cal dir que la introducció de dades per part de l'usuari està força limitada i que amb una correcta vigilància al respecte l'aplicació pot ser prou estable. En un futur caldria realitzar els següents tests:

5.1 Tests de les classes

Comprovar que la creació dels usuaris i dels comptes, claus i perfils els correcte. Això es pot realitzar amb el mateix ptyhon i les classes de teste que conté.

5.2 Tests de la interacció amb les aplicacions externes.

Caldria desenvolupar millor la interacció amb les aplicacions externes que es criden des de el nostre programa mitjançant subprocessos.

5.3 Tests de la interfície gràfica

Caldria comprovar la interfície gràfica amb alguna eina automàtica. Actualment Tails compta amb una suite de Tests automàtica també de la interacció amb l'usuari. Caldria veure com adaptar-ho. https://tails.boum.org/contribute/release_process/test/automated_tests/

6 Conclusions i treball futur

6.1 Objectius assolits i pendents

6.1.1 Objectius assolits:

No podem dir que s'ha assolit la creació de l'aplicació definitiva, tampoc una versió Beta a ser provada doncs manquen força detalls per pulir tant en quant a la interfície gràfica com (i sobretot a les proves i la gestió d'errors). El que sí que s'ha assolit és un prototip que implementa de inici a final l'objectiu principal del treball: la creació automatitzada (en la mesura del possible) de les instal·lacions de Tails.

El prototip realitza amb èxit:

- La creació de claus GPG per a cada usuari
- La creació d'un anell públic compartit amb les claus marcades com a claus de confiança
- La creació automàtica dels perfils d'usuari per a l'Icedove
- La configuració de Icedove i Enigmail per a l'ús amb Tails.
- La clonació i configuració persistent de Tails de forma semi-automàtica.

6.1.2 Objecius pendents

- Implementar la gestió d'errors i excepcions amb informació per a l'usuari.
- Implementar la funcionalitat de configuració del xat amb Pidgin OTR.
- Creació de tests automàtics tant dels mètodes com de la interfície gràfica.
- Afinar la interfície gràfica especialment ampliar el feedback amb l'usuari quan es realitzen operacions llargues.
- Pulir la distribució de mètodes al codi i la nomenclatura.
- Extreure totes les dades de configuració modificables a l'arxiu config.py.
- Crear una documentació online més el·laborada.

6.2 Conclusions.

La planificació ha estat poc adequada per als objectius que tenia, d'altra banda crec que hi ha aspectes que no es podien valorar amb anterioritat. La dificultat per treballar en un sistema amb moltes limitacions i poc documentat en alguns aspectes ha fet que m'hagi encallat en alguns temes que finalment he pogut solucionar però que m'han endarrerit força. Cal dir que el suport per part dels desenvolupadors ha estat més aviat escàs.. Comprensible davant el volum de feina i les prioritats que tenen.

A mesura que ha anat avançant el desenvolupament de l'aplicació m'he donat compte que el fet de crear una aplicació externa, no inclosa dins el programari que ve amb Tails fa que certs processos siguin més costosos que el que haurien de ser. Per exemple tot allò relacionat amb la instal·lació i persistència de les configuracions o amb l'execució privilegiada d'alguns mètodes que actualment s'ha hagut d'implementar passant el password de root per un pipe. D'altrabanda treballar amb una aplicació com Icedove que no està inclosa de moment però que ho estarà en un futur pròxim de forma externa també genera una sèrie de tasques que no seran necessàries (descàrrega, instal·lació i configuració).

6.3 Treball futur

Un cop finalitzada aquesta fase, i coneixent l'esforç necessari, cal analitzar amb cura les previsions de desenvolupamet de Tails i veure com afectaran els canvis previstos en el treball que pugui realitzar. Especialment en relació als següents aspectes: la inclusió de Icedove i Enigmail per defecte en Tails i la migració a Debian Jessie i a Python 3. Això ens situa en una perspectiva de futur amb dues possibilitats no excloents:

a) Acabar de pulir l'aplicació amb les funcionalitats que falten per

b) Un cop definit el prototip i la viabilitat de l'aplicació, familiaritzat jo mateix amb el sistema, proposar de forma més ferma la seva inclusió dins Tails i crear una branca de tipus Feature per al seu desenvolupament i abordar quina ha de ser la seva integració amb les altres aplicacions que treballen en els mateixos camps. Ja sigui com a opció a Tails-installer o a Tails-persistence-setup. A partir d'aquí poder definir un calendari habitable i buscar possibles col·laboracions. El desenvolupament realitzat durant els darrers mesos ha sigut una mica dur i a corre-cuita i és una cosa a evitar per treballar amb la cura necessària-

7 Annexos

7.1 Arixius de configuració

7.1.1 Icedove

La configuració de Icedove es fa mitjançant l'us de la funcionalitat "dotfiles" que permet desar arxius i carpetes a la partició persistent i enllaçar-les al arrencar de forma automàtica a la carpeta Home. En aquest cas la carpeta s'anomena ".icedove" i dins i trobem una carpeta amb el perfil anomenada "cadenarandom.default" i un arxiu profile.ini que defineix algunes variables i conté la ruta de la carpeta profile.

```
[General]
StartWithLastProfile=1
[Profile0]
Name=default
IsRelative=1
Path=randomstring.default
```

La carpta del perfil correspon a cada usuari. Es pot trobar el seu contingut explicat a la web de Mozilla.[5]

Els arxius importants per a l'autoconfiguració són els següents:

abook.mab -> conté les adreces afegides a la llibreta d'adreces.

signons.sqlite -> conté els logins emmagatzemats de forma xifrada.

key3.db -> conté la clau que es fa servir per xifrar els logins de l'arxiu signons.sqlite

prefs.js -> conté les preferències.

user.js -> conté preferències definides de forma manual que sobrecarreguen les definides a prefs.js i que permeten blocar-les. (no es poden canviar per la interfície d'usuari).

7.1.2 Pidgin

```
Arxius tipus:
```
otr.private_key:

```
(privkeys
 (account
(name "tfctest1@riseup.net/")
(protocol prpl-jabber)
(private-key
 (dsa
```

```
                                                                    (p
#00AFC5F835CD0B51B220A32D4F86183F627F867A651BD1435DC24BA8DFBE46E6B022B695CD29D63
8687EB221E6B95E7E1ADE1FB294C646E06CF3C5DDC5BE12215909193F1BF419C0483D825A2C556B8
271C5C65A1389D22826439BBCBFE0DA568A54D4A879B1AB05B3F46D4DC0A51CA512379B2DC2FB7D5
7F42C154ED873C6AEEB#)
  (q #00A1A3F38172FECC1909E2B45558329CF6CC1FA2C5#)
                                                                    (g
#0298F401B1373D23D199C79942C5762C414882E2E1C6CE437BF8282AC08532320BE1076560BA00E
2F5F14B5B6AFFBC26AE1B47BAF03CEDEFCB40F5C9676D989B09E720D87F882DE0AD478BC27851F6D
24266A809487F9B5D112C3BD868D0D169048B3724AE514BDDCDFBDB33F0A60E2E47930A22D48E0DB
20CF6F36657CD416F#)
                                                                    (y
#552FE8D8C5DE2C733EBE45CF29D84CE979D2A08BECC5043761BD6D839D566EEA9B5B28412234B33
8BEBC650676B3A0792213FED3561FDF5B9FD662657335333D32ACCED6298695B016CB62B9C241A42
5BFBAF9851BE4210083CE364004D20F4AEBE4A4FD27232BE8FA32FA8D4EE09622FBDBDEDFFB807E0
B0BFB7CF83E84B4D8#)
  (x #700077BDE6C21873994388B8A6E31B9EA3031A5A#)
  )
 )
 )
```

otr.fingerprints

```
tfctest1@riseup.net tfctest2@riseup.net/ riseup                         prpl-jabber
b3d9bfedde1b22a82410b3cc0d499e18211eb03e  smp

[cuenta propia]         tfctest1@riseup.net
[cuenta verificada]     tfctest2@riseup.net/riseup
[protocolo]             prpl-jabber
[fingerprint]           b3d9bfedde1b22a82410b3cc0d499e18211eb03e
[método verificacion]   smp  <--- socialist millionaire problem
```

accounts.xml

```
<?xml version='1.0' encoding='UTF-8' ?>

<account version='1.0'>
      <account>
            <protocol>prpl-jabber</protocol>
            <name>tfctest2@riseup.net/riseup</name>
            <password>TfcTest2_2014</password>
            <statuses>
                  <status type='available' name='Available' active='true'>
                        <attributes/>
                  </status>
                  <status type='mood' name='Feeling' active='false'>
                        <attributes/>
                  </status>
                  <status type='freeforchat' name='Chatty' active='false'>
                        <attributes/>
                  </status>
                  <status type='away' name='Away' active='false'>
```

```
                        <attributes/>
                  </status>
                  <status     type='extended_away'     name='Extended     away'
active='false'>
                        <attributes/>
                  </status>
                  <status type='dnd' name='Do Not Disturb' active='false'>
                        <attributes/>
                  </status>
                  <status type='offline' name='Offline' active='false'>
                        <attributes/>
                  </status>
            </statuses>
            <settings>
                  <setting name='buddy_icon_timestamp' type='int'>0</setting>
                  <setting name='auth_plain_in_clear' type='bool'>0</setting>
                  <setting name='custom_smileys' type='bool'>1</setting>
                  <setting name='port' type='int'>5222</setting>
                  <setting                                    name='connect_server'
type='string'>4cjw6cwpeaeppfqz.onion</setting>
                  <setting                                        name='ft_proxies'
type='string'>proxy.riseup.net</setting>
                  <setting name='check-mail' type='bool'>0</setting>
                  <setting                                name='connection_security'
type='string'>require_tls</setting>
                  <setting name='use-global-buddyicon' type='bool'>1</setting>
                  <setting name='silence-suppression' type='bool'>0</setting>
            </settings>
            <settings ui='gtk-gaim'>
                  <setting name='auto-login' type='bool'>1</setting>
            </settings>
            <current_error/>
      </account>
</account>
```

7.2 Relació amb els desenvolupadors de Tails

La presentació del treball i les converses amb els desenvolupadors es poden trobar online.

7.2.1 Presentació inicial del treball

https://www.mail-archive.com/tails-dev@boum.org/msg08576.html

7.2.2 Dubtes sobre funcionament

https://www.mail-archive.com/tails-dev%40boum.org/msg08845.html

7.3 Enllaços

Repositori a github create per a wiki:

[1] https://github.com/noidcc/tails-group-installer

Informació sobre la configuració de màquines virtuals amb Tails

[2] https://tails.boum.org/doc/advanced_topics/virtualization/virt-manager/index.en.html#index4h1

Informació sobre la instal·lació de software adicional a Tails

[3] https://tails.boum.org/doc/advanced_topics/additional_software/index.en.html

Informació sobre la configuració del proxy per a aplicacions

[4] https://tails.boum.org/contribute/design/stream_isolation/

Informació sobre els arxius del perfil de Icedove/Thunderbird

[5] http://kb.mozillazine.org/Files_and_folders_in_the_profile_-_Thunderbird

Informació sobre el protocol OTR que fa servir Pidgin.

[6] https://otr.cypherpunks.ca/Protocol-v3-4.0.0.htm

Repositoris públics de Tails

[7] https://git-tails.immerda.ch/tails

Informació sobre la suite de tests automàtics de Tails-dev

[8] https://tails.boum.org/contribute/release_process/test/automated_tests/