



# Un paseo por la Deep Web

**Xavier Ferrer Gimeno**

Máster Interuniversitario de Seguridad de las TIC  
TFM - INCIBE

**Tutor:** Jorge China López

**Responsable de la asignatura:** Victor García Font

Diciembre 2018



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	Un paseo por la DeepWeb
<b>Nombre del autor:</b>	Xavier Ferrer Gimeno
<b>Nombre del consultor/a:</b>	Jorge Chinaea López
<b>Nombre del PRA:</b>	Victor García Font
<b>Fecha de entrega:</b>	12/2018
<b>Titulación:</b>	Máster Interuniversitario de Seguridad de las TIC
<b>Área del Trabajo Final:</b>	TFM
<b>Idioma del trabajo:</b>	Castellano
<b>Palabras clave</b>	<i>Deep Web, Tor, Whonix</i>
<b>Resumen del Trabajo</b>	
<p>En los últimos años, hemos vivido una transformación a un mundo digital en el que parte de nuestra vida ha pasado a estar en la red. Internet acumula cada vez más datos de nuestros hábitos, gustos, costumbres, inquietudes u opiniones y en la mayoría de los casos sin que seamos conscientes de ello. Esta creciente falta de privacidad ha hecho que muchos usuarios hayan buscado otros sitios donde puedan proteger su identidad, como es el caso de la Deep Web. Pero asociado a la Deep Web, nos llega información de todo tipo y en algunos casos hablan de una zona de internet peligrosa, dónde puedes encontrar todo tipo de contenidos inmorales o dónde se guardan los mayores secretos. Este trabajo pretende dar una visión más objetiva de la Deep Web a través de un análisis técnico y práctico de las partes más populares de la misma.</p>	
<b>Abstract:</b>	
<p>In recent years, we have experienced a transformation to a digital world in which part of our life has become part of the network. Internet accumulates more and more data of our habits, tastes, customs, concerns or opinions and in most cases without us being aware of it. This growing lack of privacy has made many users have searched other sites where they can protect their identity as is the case of the Deep Web. But associated with the Deep Web we get information of all kinds, and in some cases, they speak of a dangerous internet area where you can find all kinds of immoral content or where the greatest secrets are kept. This work aims to give a more objective view of the Deep Web through a technical and practical analysis of the most popular parts of it.</p>	

# Índice

<b>1. Introducción .....</b>	<b>1</b>
1.1 Contexto y justificación del Trabajo .....	1
1.2 Objetivos del Trabajo .....	2
1.3 Enfoque y método seguido .....	2
1.4 Planificación del Trabajo .....	2
1.5 Breve resumen de productos obtenidos.....	3
1.6 Breve descripción de los otros capítulos de la memoria.....	3
<b>2. La Deep Web .....</b>	<b>4</b>
<b>3. Freenet.....</b>	<b>5</b>
3.1 Aspectos técnicos.....	6
3.2 Accediendo a Freenet.....	7
3.3 Resumen Freenet .....	9
<b>4. I2P.....</b>	<b>9</b>
4.1 Aspectos técnicos.....	10
4.2 Accediendo a I2P.....	14
4.3 Resumen I2P .....	17
<b>5. La red Tor .....</b>	<b>17</b>
5.1 Aspectos técnicos.....	18
5.2 Uso de Tor .....	23
5.3 Accediendo a Tor .....	24
5.4 Resumen Tor.....	27
<b>6. Conclusiones:.....</b>	<b>27</b>
<b>7. Referencias.....</b>	<b>30</b>
<b>ANEXO A: Instalación de la distribución Whonix.....</b>	<b>32</b>
<b>A.1 Recursos.....</b>	<b>32</b>
<b>A.2 Documentación Whonix.....</b>	<b>32</b>
A.2.1 Metadatos .....	32
A.2.2 MITM .....	33
A.2.3 Como funciona Whonix .....	34
A.2.4 Medidas de seguridad adicionales.....	35
<b>A.3 Instalación de Whonix .....</b>	<b>36</b>
<b>A.4 Conclusiones .....</b>	<b>41</b>

## Lista de figuras

Figura 1: Modo de conexión en Freenet.....	8
Figura 2: Interfaz principal de Freenet.....	9
Figura 3: Consola I2P .....	10
Figura 4: Túneles I2P .....	11
Figura 5: Garlic Routing .....	12
Figura 6: Intercambio de mensajes .....	13
Figura 7: Consola del router I2P.....	15
Figura 8: Túneles activos .....	16
Figura 9: Página I2P Wiki.....	16
Figura 10: Página echelon.i2p.....	17
Figura 11: Cifrado por capas del onion routing .....	18
Figura 12: Nodos Tor .....	19
Figura 13: Nodos <i>Authority</i> .....	20
Figura 14: Los dos tipos de Célula .....	21
Figura 15: Cell Commands.....	21
Figura 16: Alice consulta el SD .....	22
Figura 17: Establecimiento del circuito.....	22
Figura 18: Alice conecta con el destino en Internet.....	23
Figura 19: Número de conexiones de Tor .....	23
Figura 20: Numero de nodos en Tor .....	24
Figura 21: <i>Apps</i> de Android Orbot y Orfox .....	25
Figura 22: Página de prueba de Tor Browser.....	26
Figura 23: Anonimising relay monitor .....	26
Figura 24: Buscadores Ahmia y Hidden Wiki de la red Tor .....	27
Figura 25: Advertencia whonix usuarios principiantes.....	32
Figura 26: Metadatos en fotografías.....	33
Figura 27: Ataque Mitm .....	34
Figura 28: Diagrama de conexión maquinas Whonix.....	35
Figura 29: Firmas PGP de las descargas.....	37
Figura 30: Huella digital Patrick Schleizer .....	37
Figura 31: Huella digital de la página de Whonix .....	38
Figura 32: Nivel de confianza de la firma .....	38
Figura 33: Firmas archivos descargados .....	38
Figura 34: Maquinas Whonix.....	39
Figura 35: Tipo de conexión .....	39
Figura 36: Instalación finalizada .....	39
Figura 37: Actualizaciones .....	40
Figura 38: TimeSync .....	40

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

El mundo está cada vez más conectado. Las redes llegan a más sitios, son de mejor calidad y cada vez permiten ofrecer más servicios a más usuarios. Esto ha cambiado la forma en la que hacemos las cosas a todos los niveles. Internet ha permitido que no haya fronteras en este aspecto y que éste sea un fenómeno global.

Usamos Internet para todo. Incluso parte de nuestras vidas, de lo que hacemos, decimos, opinamos o imágenes nuestras, están en internet. Hay datos nuestros en todas partes y en la mayoría de los casos sin nuestro consentimiento. Las ventajas de este fenómeno son muchas pero una desventaja importante es la disminución de nuestra privacidad en la red. Actualmente toda esta información es de gran valor para grandes empresas de internet y gobiernos que explotan esos datos.

Algunos estados están buscando fórmulas legales para proteger nuestros datos personales, pero seguimos dejando una gran cantidad de datos en internet con los que se puede obtener información de los usuarios de forma directa o indirecta cruzándola con otros datos.

La privacidad, el anonimato o el derecho al olvido, son temas que cada vez nos preocupan más.

A parte del Internet que conocemos, hay otras redes que han buscado mantener el anonimato del usuario y dónde es más difícil ser rastreados, como es el caso de la Deep Web. Actualmente encontramos artículos en internet, prensa y televisión hablando de la Deep Web. Es un tema que cada vez está generando más interés. Pero asociado a la Deep Web encontramos en muchos casos cierto misticismo y nos hablan de una zona de internet peligrosa dónde puedes encontrar todo tipo de contenidos inmorales o dónde se guardan los mayores secretos [13,14,15].

En este trabajo trataremos de explicar qué es la Deep Web, aclararemos conceptos, y analizaremos las redes más importantes que forman parte de ella.

## 1.2 Objetivos del Trabajo

El objetivo principal de este trabajo es conocer y entender qué es la Deep Web. Analizar cómo funciona la red tanto a nivel técnico como práctico. Entender cómo operan los usuarios en la red tanto para ofrecer servicios como para consumirlos.

Por otra parte, también será objetivo de este trabajo el montar y probar un entorno de la distribución *Whonix* que nos debe permitir acceder a una red de la Deep Web con medidas adicionales de seguridad.

A parte de analizar el funcionamiento de la solución, estudiaremos si es un sistema fácil de implementar y si es del alcance de personas no expertas comparado con otros métodos de acceso a la Deep Web.

## 1.3 Enfoque y método seguido

En primer lugar, habrá que montar el entorno de la distribución *Whonix*, que consta de dos máquinas virtuales. Intentaremos realizar la instalación con la información o manuales que haya en la página de los desarrolladores del producto.

Una vez terminada la instalación, probaremos la solución y analizaremos las herramientas y opciones que ofrece. Documentaremos como hemos realizado la instalación y los recursos que hemos utilizado.

A continuación, realizaremos una tarea de investigación de la Deep Web empezando por las definiciones y taxonomía de conceptos básicos asociados. Buscaremos información de las redes que sustentan la Deep Web y estudiaremos sus características.

Analizaremos la parte operativa de la Deep Web. Como acceden los usuarios a la Deep Web, como se comunican y cómo funcionan los servicios que hay en ella.

Finalmente estudiaremos estas redes también desde una perspectiva funcional, analizando cuáles son las características y servicios más interesantes para el usuario.

## 1.4 Planificación del Trabajo

Para la realización del trabajo vamos a necesitar un PC que permita la instalación del software Oracle VirtualBox con capacidad suficiente para montar y ejecutar las dos máquinas virtuales de la distribución *Whonix*. También vamos a necesitar una conexión a internet estándar.

La dedicación al TFM no va a ser a jornada completa. Se destinan un total de 10h a la semana. Para representar las tareas en el diagrama Gantt, supondremos una dedicación fija de 1,5h por jornada (Tabla 1).

Actividades	Inicio	Duración (días)	Fin
PAC-1: Planificación	18/09/2018	20	08/10/2018
PAC-2: Whonix	09/10/2018	28	06/11/2018
PAC-3: Deep Web	06/11/2018	6	12/11/2018
PAC-3: Red TOR	13/11/2018	6	19/11/2018
PAC-3: Freenet	20/11/2018	4	24/11/2018
PAC-3: Red I2P	25/11/2018	4	29/11/2018
PAC-3: Entrega	30/11/2018	3	03/12/2018
PAC-4: Memoria	04/12/2018	27	31/12/2018
Entrega Video	01/01/2019	6	07/01/2019

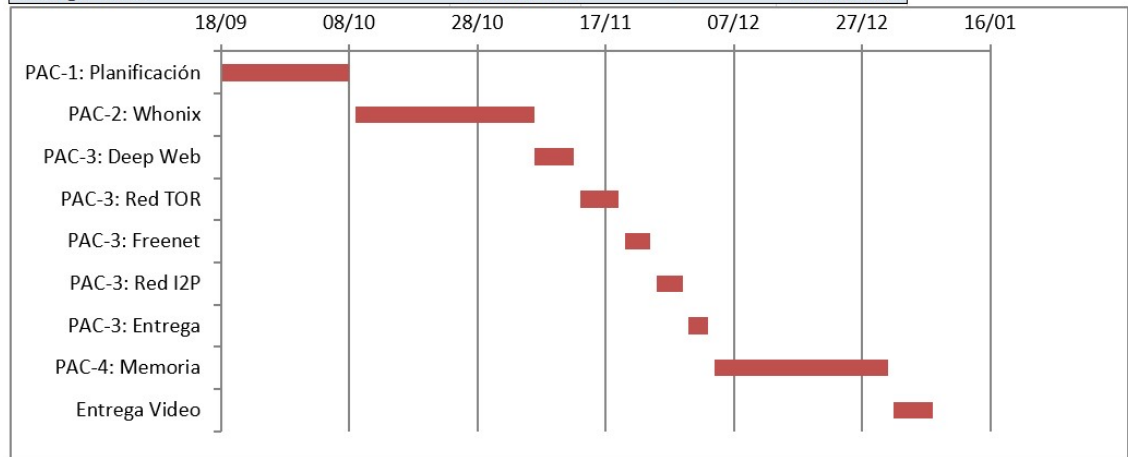


Tabla 1 : Diagrama Gantt

La inversión total estimada de tiempo es de  $80d \times 1,5h = 120h$ . Aunque en el apartado de elaboración de la memoria reservaremos 30 horas adicionales dado que en la fase final suele aumentar la carga de trabajo y es preciso cumplir con los plazos establecidos.

### 1.5 Breve resumen de productos obtenidos

En primera instancia tenemos que obtener una descripción clara de los conceptos relacionados con la Deep Web. También tenemos que obtener datos técnicos y prácticos de las redes más usadas de la Deep Web que nos permita tener una visión clara de ellas. Y finalmente tenemos que poder montar y probar la distribución de Whonix como parte de la prueba práctica de acceso a la red Tor.

### 1.6 Breve descripción de los otros capítulos de la memoria

En el capítulo 2 analizamos el contexto y los aspectos que han motivado el crecimiento de la Deep Web. Veremos en detalle los conceptos y definiciones asociados a ella.

En los capítulos 3, 4 y 5 analizamos en profundidad las redes Freenet, I2P y Tor. Primero hacemos una breve explicación del origen de la red y de sus características principales. Posteriormente entramos en los aspectos técnicos



más destacados de cada una de las redes y finalmente realizamos una descripción de la parte práctica, que hemos realizado accediendo a cada una de las redes. En este último punto se pretende analizar brevemente la experiencia de usuario, desde que instalas el software hasta que operas en la red. Es decir, tener una visión de cada red desde una perspectiva operativa, analizando aspectos como la facilidad de acceso, los servicios que hay y el interfaz del usuario.

En el capítulo 6 expondremos las conclusiones generales tanto asociadas a lo que hemos visto de la Deep Web como comparando las diferentes redes que hemos analizado.

Finalmente, en el anexo A, hacemos una descripción del proceso de instalación y prueba de la distribución de Whonix, con la que posteriormente hemos realizado las pruebas de acceso a la red Tor.

## 2. La Deep Web

La parte de Internet que esta indexada en los buscadores comunes es la que llamamos la *Surface Web* [22]. Es dónde están la mayoría de los servicios ofrecidos al público en general.

La otra parte de internet es la que se conoce como *Deep Web* [2,3,4] que engloba el resto de contenido no indexado al que no podemos acceder mediante un buscador estándar.

Los grandes buscadores de internet tienen sistemas para indexar continuamente contenido de internet. La mayoría de los usuarios usamos estos buscadores para acceder a las páginas de internet. Pero existen muchos sitios que solo están accesibles por otras vías.

Adicionalmente, dentro de la Deep Web, nos encontramos sitios a los que solo podemos acceder con un software específico. Estas redes se las conoce como *darknets* [4,5] y el contenido que ofrecen es lo que llamamos la *Dark Web* [3].

Algunas de las *darknets* más conocidas en la actualidad son Tor, I2P y Freenet. Para acceder a cada una de ellas el usuario necesita tener instalado el software específico de cada red.

Todas ellas coinciden en indicar en sus páginas oficiales que son proyectos que buscan proteger el anonimato del usuario con el objetivo de defender la privacidad y libertad de expresión de las personas.

A continuación, veremos las características más importantes de cada una de ellas. Analizaremos los aspectos técnicos más relevantes y probaremos a acceder a ellas para tener visión más real de su funcionamiento.

## 3. Freenet

Nace en 1999 y es un proyecto que busca evitar la censura y defender la libertad de expresión de los usuarios en Internet. Fue desarrollado originalmente por Ian Clarke, un estudiante de la universidad de Edimburgo, y es de software libre [8].

Es una red P2P<sup>1</sup> *inproxy*<sup>2</sup> y se basa en un sistema descentralizado de los datos de forma que todo lo que se publica en Freenet está distribuido entre los nodos. No hay servidores centrales y los usuarios ofrecen su almacenamiento y ancho de banda a la red. La red Freenet es como un dispositivo de almacenamiento enorme [16]. Además, usa un sistema de caché adaptativo de forma que los contenidos más populares, tienen una prioridad mayor y estarán disponibles más rápidamente que el resto.

Esto proporciona anonimato y evita la censura dado que el contenido, no está ubicado en un solo sitio. Si el usuario publica alguna cosa en Freenet y apaga su equipo, ese contenido sigue accesible en Freenet.

Una de las funcionalidades más interesante de Freenet es que permite crear un entorno privado de comunicación con otros usuarios de tu confianza.

Tiene una interfaz única desde dónde el usuario realiza todas las acciones. Desde allí, se gestiona el nodo permitiendo cambiar configuraciones, ver las conexiones con otros nodos u otras estadísticas. También se pueden gestionar las descargas y subidas de ficheros, y acceder a los complementos que el usuario tenga instalados, o añadir de nuevos si lo desea.

Estos son algunos de los complementos más comunes:

- **Freenet Message System:** Herramienta de mensajería.
- **Freemail:** Email entre usuarios de Freenet.
- **Sone:** Servicio de Chat.
- **Jsite:** Herramienta para subir sitios a Freenet.

Para acceder a estos servicios es necesario que el usuario tenga creada una identidad. Para hacerlo, Freenet tiene un complemento adicional llamado *WebOfTrust*, que se encarga de gestionar identidades de forma anónima.

Una característica destacada de Freenet, es la facilidad que tienen los usuarios de diseñar sus propios complementos y publicarlos. Encuentras fácilmente tutoriales con complementos de ejemplo tipo *HelloWorld*<sup>3</sup> que puedes descargar

---

<sup>1</sup> P2P: Es una red conformada por un conjunto de ordenadores conectados entre sí que actúan simultáneamente como clientes y servidores respecto a los demás nodos[1].

<sup>2</sup> Red que no tiene salida a Internet o a otras redes.

<sup>3</sup> Programa básico de muestra, que una vez se ha probado su funcionamiento, se puede usar de punto de partida para crear otro programa más elaborado.

e importar el archivo *.jar*<sup>4</sup> en una herramienta de diseño como Eclipse<sup>5</sup> para hacerte tus propios proyectos y subirlos después.

### 3.1 Aspectos técnicos

Todos los archivos que hay en Freenet tienen asociada una clave. La mayor parte de estas claves son *hashes*<sup>6</sup> de los propios archivos. Para acceder a algún tipo de información se usa este formato: <http://127.0.0.1:8888/> [Clave Freenet]

Existen cuatro tipos de clave:

- **CHK** Content Hash Keys
- **SSK** Signed Subspace Keys
- **USK** Updatable Subspace Keys
- **KSK** Keyworld Signed Keys

Las claves CHK se usan para los archivos. Contienen un hash SHA-256 del archivo al que hace referencia. La clave CHK sería de alguna forma como el identificador único *URI*<sup>7</sup> que tenemos en Freenet para llegar a un archivo concreto. Esto evita, por tanto, que haya identificadores de archivos duplicados dado que no habrá dos claves CHK iguales.

Examinamos la clave CHK de un archivo pdf que tengamos en el apartado de descargas como por ejemplo este que mostramos a continuación [24]:

<CHK@OLtg-Zbts4XKby-g4uk3Q0VEOAwFIHeOaUdepZ8XP-Q,Dbik4Fz-MNVpiW~5w85MACqbAbtBPPCu6wNE4gYqIII,AAMC--8/invisibility-report-9-0-final.pdf>

La clave CHK tiene estas tres partes:

- **OLtg-Zbts4XKby-g4uk3Q0VEOAwFIHeOaUdepZ8XP-Q:** Hash del archivo pdf.
- **Dbik4Fz-MNVpiW~5w85MACqbAbtBPPCu6wNE4gYqIII:** Clave para descifrar el archivo pdf.
- **AAMC—8:** Parámetros de encriptación.

Las claves SSK se usan para los *freesites*. En este caso, como el contenido de una web es más fácil de que cambie que el de un archivo, la clave SSK se basa

---

<sup>4</sup> Archivo JAVA.

<sup>5</sup> Plataforma para el desarrollo de software [https://en.wikipedia.org/wiki/Eclipse\\_\(software\)](https://en.wikipedia.org/wiki/Eclipse_(software))

<sup>6</sup> [https://es.wikipedia.org/wiki/Función\\_hash\\_criptográfica](https://es.wikipedia.org/wiki/Función_hash_criptográfica)

<sup>7</sup> [https://en.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier)

en la clave privada del usuario que publica la web y no en el hash de su contenido.

Por ejemplo, esta es la dirección web del buscador *Nerdageddon*<sup>8</sup>:

<http://localhost:8888/freenet:USK@tiYrPDh~fDeH5V7NZjpp~QuubaHwgks88iwlRXXLLWA,yboLMwX1dChz8fWKjmbdtl38HR5uiCOdiUT86ohUyRg,AQACAAE/nerdageddon/247/>

La clave tiene estas tres partes:

- **USK@tiYrPDh~fDeH5V7NZjpp~QuubaHwgks88iwlRXXLLWA:** Hash de la clave privada del autor.
- **yboLMwX1dChz8fWKjmbdtl38HR5uiCOdiUT86ohUyRg:** Clave de descifrado.
- **AQACAAE:** Parámetros de encriptación.

Las claves USK son un subtipo de las claves SSK que se usan básicamente para poder acceder a la última versión de un *freesite*. Así permite actualizar y crear nuevas versiones de los *freesites* fácilmente.

Y las claves KSK, permiten al usuario poner identificadores fáciles, como “KSK@miarchivo.txt”, a los archivos. Aunque en este caso, sí que se corre el riesgo de que otro usuario pueda subir otro archivo con el mismo identificador. La clave KSK también puede usarse para que apunte a una clave CHK.

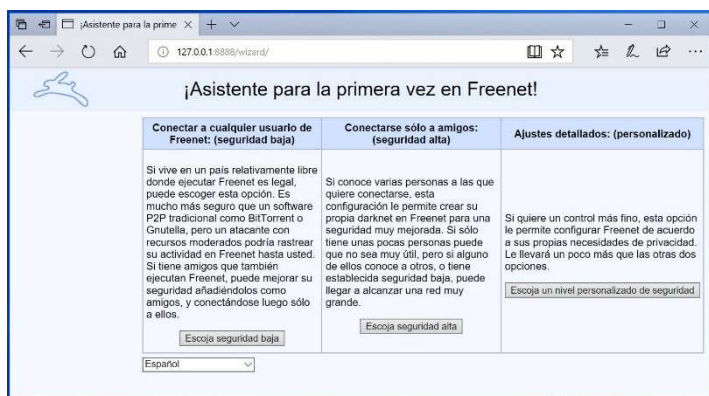
### 3.2 Accediendo a Freenet

Para acceder a Freenet necesitamos instalar un software que podemos encontrar en la página oficial [23].

Al instalar el software por primera vez, el usuario tiene opción de escoger entre dos modos de conexión como se muestra en la Figura 1. Un modo en el que te conectas a Freenet con el resto de los usuarios y otro modo “privado” en el que solo te comunicas con un grupo concreto de usuarios. Esta segunda opción, es la que junto con un mínimo de 5 usuarios, te permite crear una red totalmente aislada del resto de usuarios de Freenet (lo llama modo “darknet”).

---

<sup>8</sup> Popular índice de *freesites* con el contenido más ofensivo filtrado.



**Figura 1: Modo de conexión en Freenet**

Durante la instalación del software de Freenet, se advierte al usuario de que, por seguridad, no se use el mismo navegador para Internet y Freenet. Esto es debido a que el navegador que utilizamos habitualmente, podría tener datos de nuestra identidad, y puede que no tenga las medidas de seguridad adecuadas, para evitar que esa información se vea comprometida y por tanto, haya forma de relacionar la identidad que usemos en Freenet, con nuestra identidad real. Usando otro navegador sin datos de nuestra identidad real evitamos ese riesgo.

Como el pc del usuario va a formar parte de la red P2P, el programa de instalación pregunta por el tamaño de disco que se le asignará. Cuanto mayor es el tamaño de disco asignado, más rápido es el acceso a los contenidos desde ese nodo, y mejor es para la comunidad.

El acceso a Freenet, se hace con un navegador a la dirección *loopback*<sup>9</sup> al puerto 8888 (127.0.0.1:8888). La página que aparece al acceder es la interfaz desde la que gestionaremos nuestro nodo en Freenet. En la Figura 2, se puede ver el aspecto del interfaz de usuario que vemos al entrar en Freenet.

En la parte izquierda, enmarcado en rojo, es dónde están los apartados básicos de gestión del nodo así como los complementos que vayamos instalando. En este caso podemos ver en la Figura 2 que este usuario tiene instalado el complemento *WebOfTrust*, comentado anteriormente, para gestionar identidades. En el panel, este complemento aparece con el título de "Comunidad".

<sup>9</sup> La dirección de *loopback* es una interfaz de red virtual que apunta al propio host y de forma habitual tiene la ip 127.0.0.1.



Figura 2: Interfaz principal de Freenet

Cuando seleccionamos un apartado del cuadro de la izquierda se carga el contenido en el cuadro central del panel. En el caso de la Figura 2, el cuadro central tiene el complemento *WebOfTrust*, que te permite iniciar sesión o crear otra identidad.

En la parte inferior del panel muestra información de alertas del nodo relativas a operaciones que hayamos lanzado como descargas de archivos o instalación de complementos. Y también muestra la configuración de seguridad que tenemos y los nodos con los que estamos conectados.

### 3.3 Resumen Freenet

Freenet es una red P2P inproxy, descentralizada y diseñada en Java. Permite a los usuarios subir archivos o publicar un sitio web y simplemente queda en la red. El usuario no tiene que preocuparse de tener el pc encendido.

Un aspecto destacado de Freenet es la opción de crear un entorno totalmente privado con otros usuarios que llama modo *darknet*. También es interesante la opción de poder diseñar tus propios *plugins* en java.

La instalación del software es rápida, aunque después tarda tiempo en la sincronización del nodo. Si no tienes el nodo conectado a menudo los accesos a los contenidos serán lentos por lo que te obliga a estar mínimamente actualizado.

## 4. I2P

Es un proyecto con objetivos similares a Freenet que busca proteger el anonimato, la privacidad y la no-censura de los usuarios de internet. El usuario no oculta que está conectado a I2P, pero las actividades que realiza en la red

permanecen ocultas y la información de qué usuarios se comunican con otros es anónima.

Es una red formada por routers (*I2P node*) y el sistema de comunicación está basado en túneles (*I2P tunnel*). Los túneles se crean de forma temporal, y no son bidireccionales. Se establece un túnel para el tráfico de envío y otro distinto para el de recepción con el fin de aumentar la seguridad de la comunicación. En la red I2P, la información se envía por medio de mensajes.

De cara al usuario, presenta una interfaz única similar a la de Freenet desde la que se realizan la mayor parte de gestiones (Figura 3). Desde esta consola el usuario puede ver tanto información de su router como de servicios que tenga activos en I2P. Uno de ellos es el servicio *webserver*, dónde permite al usuario publicar su página web en I2P o *eepsite*. En este caso, a diferencia de Freenet, el servicio no es ofrecido a nivel de red, sino que está publicado en el pc del usuario y por tanto el contenido de la web está en una carpeta local.



Figura 3: Consola I2P

I2P hace una separación de las aplicaciones que se ejecutan en el propio router de las que ejecutan localmente los usuarios (los *eepsites*).

#### 4.1 Aspectos técnicos

##### Túneles

Los túneles son caminos que se establecen entre varios routers I2P. Utilizan un cifrado por capas de forma que cada router solo puede descifrar una capa y al hacerlo solo le permitirá ver la ip del siguiente router.

Existen dos tipos de túneles. Los túneles de entrada o *Inbound tunnels*, y los túneles de salida o *outbound tunnels*.

En la Figura 4, podemos ver un ejemplo de uso de los dos tipos de túneles, en el que Alice quiere enviar un mensaje a Bob.

Alice utilizará un túnel de salida (A-B-C) para enviar el mensaje, y Bob utilizará un túnel de entrada (D-E-F) para recibirlo.



Figura 4: Túneles I2P

Los routers que forman los túneles pueden adoptar estos tres tipos de roles:

- **Tunnel gateway:** Es el primer router del túnel. En la Figura 4 tendrían este rol tanto el nodo A del túnel de salida de Alice como el nodo D del túnel de entrada de Bob.
- **Tunnel endpoint:** Es el último router del túnel. En la Figura 4 tendrían este rol el nodo C del túnel de salida de Alice y el nodo F del túnel de entrada de Bob.
- **Tunnel participant:** Son los routers intermedios. En la Figura 4 serían tanto el nodo B del túnel de salida de Alice como el nodo E del túnel de entrada de Bob.

De forma predeterminada se usan túneles de 2-3 saltos, aunque se recomienda un mínimo de 3 saltos. El ejemplo de la Figura 4, sería un túnel de 2 saltos o *2-hop tunnel*.

## NetDB

NetDB es la base de datos de la red I2P. Su objetivo es almacenar y distribuir de forma segura la información necesaria para contactar con los routers (*routerinfo*) y los destinos (*leaseset*).

- **RouterInfo:** Es la información que identifica a los routers y es necesaria para que otros routers puedan llegar a él. Contiene la identidad del router.
- **LeaseSet:** Información de un destino. Por ejemplo, para llegar a un *eepsite* que tenga publicado un usuario.



La base de datos es distribuida y se almacena en un conjunto de routers de la red que se llaman *floodfill routers*. Este rol se asigna a los routers que tienen mejor ancho de banda y rendimiento.

## Garlic Routing

I2P incorpora una variante mejorada del protocolo de enrutamiento *onion routing* que utiliza la red Tor, que se llama *garlic routing*<sup>10</sup>. Este protocolo añade otra capa de encriptación adicional agrupando varios mensajes.

El mensaje que antes con *onion routing* estaba envuelto por varias capas, ahora es un mensaje que llamamos *garlic message*, que contiene múltiples mensajes o *garlic cloves*, cada uno con sus respectivas instrucciones de entrega. De esta forma, se hace más complicado que un observador relacione una cantidad de datos determinada entre dos usuarios. Ahora el mensaje que pasa por los nodos no tiene por qué formar parte totalmente de una comunicación entre dos usuarios.

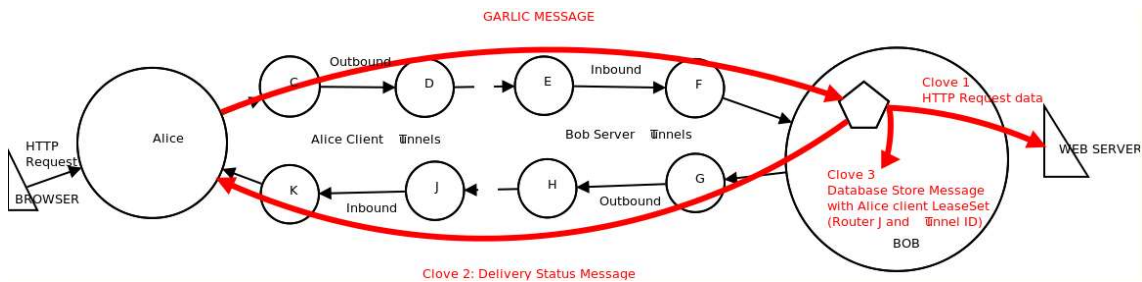


Figura 5: Garlic Routing

En la Figura 5, se ve un ejemplo de comunicación entre Alice y Bob, en una comunicación típica de 4 túneles. Uno de entrada y de salida para Alice y uno de entrada y de salida para Bob. Se envían un mensaje *garlic*, que al final se divide en 3 partes o 3 *galic cloves*.

## Intercambio de mensajes en I2P

Antes de intercambiar mensajes, los usuarios deben construir sus respectivos túneles de entrada y de salida. Para hacerlo, primero contactan con NetDB para obtener la lista de routers (*routerinfo*). Para construir un túnel, se envía un mensaje *tunnel build message* a uno de los routers que lo propaga a otros routers hasta formar el túnel.

En la Figura 6 se muestra un ejemplo con 4 usuarios que son Alice, Charlie, Bob y Dave que tienen un único destino. En este caso todos los túneles son de 2 saltos.

Suponemos que Alice quiere comunicarse con Bob. Alice envía un mensaje por uno de sus túneles de salida en dirección a uno de los túneles de entrada de

<sup>10</sup> <https://geti2p.net/mg/docs/how/garlic-routing>

Bob. Para hacer esto, Alice ha consultado previamente la base de datos NetDB, y por eso tiene información del *leaseset* de Bob y de los *gateways* de los túneles de entrada de Bob.

Antes de enviar el mensaje, el Gateway del túnel de salida de Alice realiza el cifrado por capas. Cifrará el mensaje tantas veces como saltos queden hasta el final del túnel. A medida que el mensaje pasa por los routers, estos quitan una capa de cifrado hasta llegar al último router *Endpoint*, que obtiene el mensaje y las instrucciones para reenviar el mensaje al túnel de entrada de Bob. En ese punto, el mensaje aún tiene una capa adicional de encriptación que se establece extremo a extremo. Alice ha cifrado el mensaje con la clave pública del Destino, obtenida del *leaseset* de Bob, de forma que esta última capa solamente la podrá quitar Bob con su clave privada. Este cifrado en I2P se llama *garlic encryption*.

Cuando el mensaje se entrega al Gateway del túnel de entrada de Bob, se realiza de nuevo el cifrado por capas hasta llegar al router *Endpoint* del túnel que finalmente entrega el mensaje a Bob.

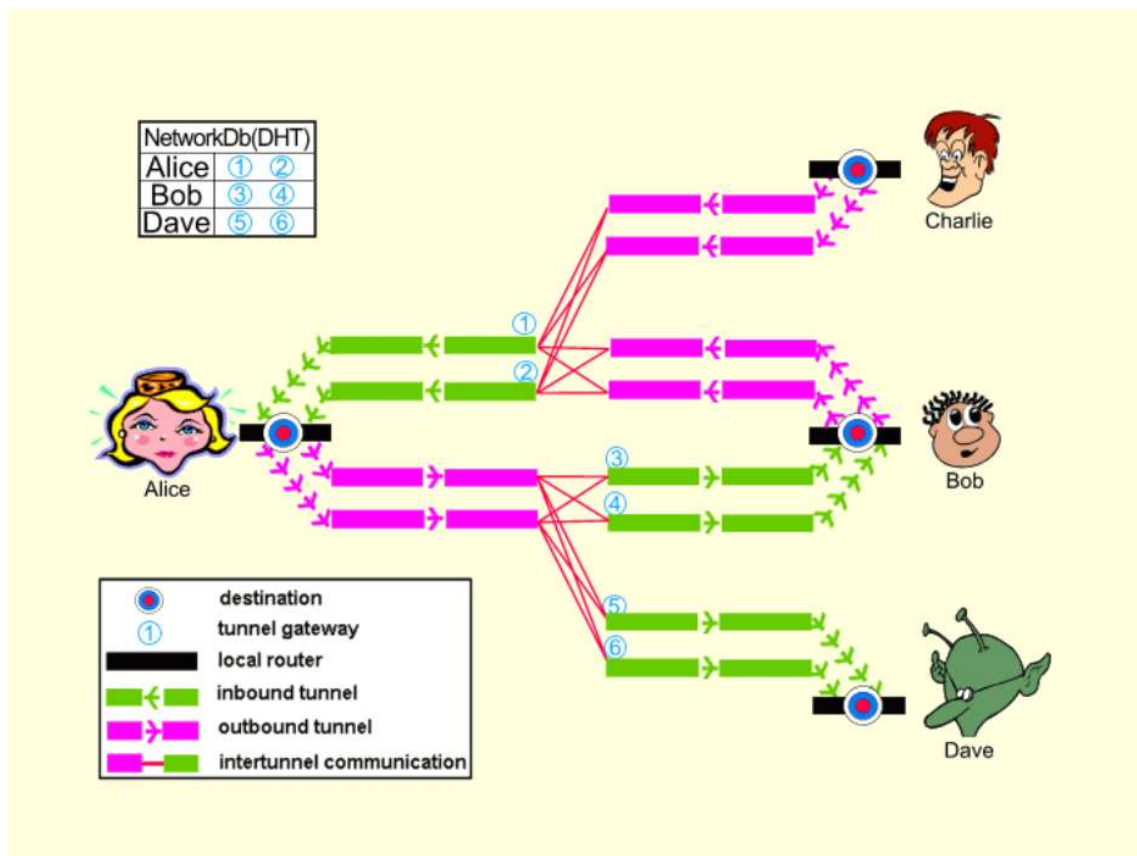


Figura 6: Intercambio de mensajes

## Protocolo de transporte

I2P empezó utilizando TCP, pero actualmente usa un protocolo basado en UDP, que se llama *Secure Semireliable UDP* (SSU). El concepto de *semi reliable* o semi fiable, viene de que los mensajes que no se han recibido completamente, se retransmitirá solo un número máximo de veces. Después el mensaje se descarta.

Los retos de este protocolo eran principalmente garantizar la seguridad de los mensajes, control de congestión, capacidad de afrontar los obstáculos en la red con NATs o firewalls y ser capaz de mover eficientemente volúmenes de datos grandes.

## Protocolos de encriptación

I2P usa un protocolo de criptografía simétrica (AES<sup>11</sup>), uno para criptografía asimétrica (2048bits ElGamal<sup>12</sup>), uno para firma (ECDSA<sup>13</sup>) y otro para hashes (SHA256<sup>14</sup>).

Se usan individualmente o combinados como por ejemplo en el cifrado extremo a extremo *gralic* que usa ElGamal/AES+Session.

### 4.2 Accediendo a I2P

El software de instalación se descarga de la página oficial de I2P en la URL <https://geti2p.net/en/download> . Es necesario tener JAVA instalado para que la aplicación funcione.

Una vez instalado el software, se arranca el servicio y hay que abrir el navegador e ir a la dirección de *loopback* 127.0.0.1 puerto 7657. Aparece la consola principal de I2P como podemos ver en la Figura 7.

---

<sup>11</sup> [https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)

<sup>12</sup> [https://en.wikipedia.org/wiki/ElGamal\\_encryption](https://en.wikipedia.org/wiki/ElGamal_encryption)

<sup>13</sup> [https://en.wikipedia.org/wiki/Elliptic\\_Curve\\_Digital\\_Signature\\_Algorithm](https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm)

<sup>14</sup> <https://en.wikipedia.org/wiki/SHA-2>



Figura 7: Consola del router I2P

En la parte izquierda, están los apartados a los que podemos acceder y en la parte central, se visualiza el contenido. En este sentido es bastante parecido a la interfaz de Freenet. Tenemos apartados de información del router, servicios, configuraciones, una sección de ayuda y una parte dónde vemos los pares (*peers*) y túneles (*tunnels*) activos.

Como servicio relevante, está el servicio de correo *susimail*. Llama la atención que, al crear una nueva cuenta, el administrador nos avisa de que no usemos la cuenta para actividades ilegales o nos la deshabilitarán. En la mayoría de los servicios de la *Deep Web*, no suelen poner limitaciones.

En la parte de servicios de la consola, también podemos crear nuestro propio *eepsite*. En el apartado *Servidor Web* nos explica en qué carpeta de nuestro pc debemos poner los archivos de la página web, y como publicar el servicio iniciando el túnel del servicio oculto. También nos comentan como publicar el sitio en la libreta general de direcciones de I2P y asignarle un nombre fácil (de tipo *mieepsite.i2p*) que se asocia a nuestro Destino Local.

Otra opción es publicar el sitio solo a un grupo concreto de usuarios con *leasesets* cifrados solo accesibles para los usuarios que conozcan una clave de descifrado.

En la consola también podemos cambiar configuraciones del router. En la parte de red, permite definir la longitud de los túneles tanto entrantes como salientes. Con una configuración de pocos saltos, tienes mejor rendimiento y peor seguridad, y en una configuración de más saltos sucede lo contrario. Por defecto se recomiendan túneles de 2 saltos.

También podemos ver los túneles de entrada y de salida que tenemos activos con los diferentes nodos como podemos ver en la Figura 8.

← → ↻ 127.0.0.1:7657/tunnels#Pi3g

## COMPENDIO DE TÚNELES DE I2P

### INFORMACIÓN DEL ROUTER IP

Versión: 0.9.37-0  
Duración De La Conexión: 24 horas

### ANCHO DE BANDA ENTRANTE/SALIENTE

3 S: 0.78 / 1.03 Kbps  
5 Min: 3.01 / 2.64 Kbps  
Total: 1.10 / 1.03 Kbps  
Usados: 5.41 MB / 5.01 MB

Red: Bloqueado por cortafuegos (firewall)

### SERVICIOS I2P

Correo electrónico  
Torrents  
Servidor web

### CONFIGURACIÓN INTERNA DE I2P

Libreta de direcciones Gráficas Ayuda  
Administrador de servicios ocultos  
Registros (logs)  
NetDB (base de datos de red)  
Pares (peers) Perfiles Túneles

### AYUDA Y PREGUNTAS FRECUENTES

Configuración avanzada  
Registro de cambios Configuración

### TÚNELES EXPLORATORIOS

↓	Caducidad	Uso	Pasarela (gateway)	Participantes	Extremo
↓	3 min	4 KB	gf i B 2974016850	am9r 3802340839	local 2712568712
↓	5 min	0 KB	RJ4H 2565619526	QRmE 2970890187	local 2001651292
↑	6 min	0 KB	local 3061540940	GWCT 793533938	XT r P 3386084073
↑	6 min	1 KB	local 1679545292	HoX1 2544460782	Bq f O 2030756942

Uso de ancho de banda del total de vida: 569 KB entrada, 1,38 MB salida

### TÚNELES DE CLIENTE PARA CLIENTES COMPARTIDOS

↓	Caducidad	Uso	Pasarela (gateway)	Participantes	Extremo
↓	6 min	313 KB	vSSD 1194187482	YRtv 3497028884	yBHR 3520564641
↓	6 min	21 KB	VeXH 3601780898	YRtv 1566202573	yBHR 813469496
↑	73 s	397 KB	local 1672291896	VeXH 2141882548	bdPL 2836130894

Creación de túnel en curso: 1 Saliente

Uso de ancho de banda del total de vida: 2,84 MB entrada, 1,94 MB salida

### TÚNELES DE CLIENTE PARA EEPSITE

↓	Caducidad	Uso	Pasarela (gateway)	Participantes	Extremo
↓	72 s	0 KB	W0FS 512157149	89r e 3117839318	yBHR 2209131466

Figura 8: Túneles activos

Para poder navegar por los eepsites, es necesario revisar la configuración de proxy de nuestro navegador y verificar que tenemos puesta la dirección 127.0.0.1 y puerto 4444 y 4445 para https.

Una vez cambiada la configuración de proxy vemos que ya podemos acceder a los eepsites. Probamos algunos sitios del apartado “servicios ocultos de interés” que aparecen en la consola como la página i2pwiki.i2p (Figura 9).

Figura 9: Página I2P Wiki

Y probamos también a acceder a la página echelon.i2p , un sitio de descarga de software (Figura 10):

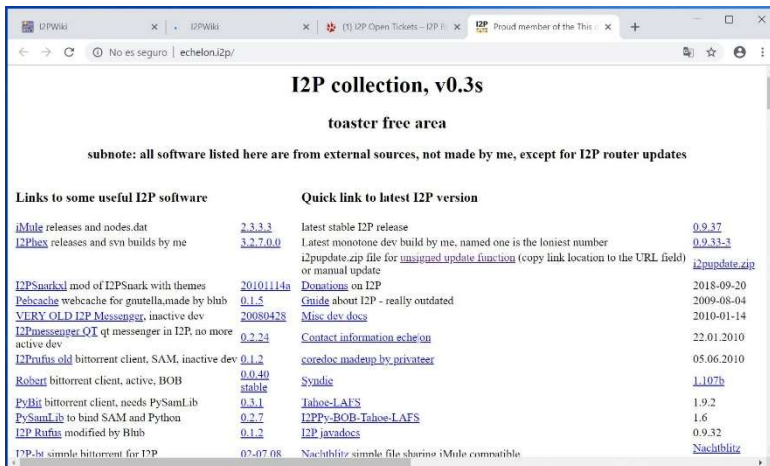


Figura 10: Página echelon.i2p

### 4.3 Resumen I2P

I2P es una red P2P básicamente inproxy programada en Java con algunas similitudes con Tor. Tiene la base de datos distribuida a diferencia de Tor, un protocolo de enrutamiento más avanzado que el *onion routing* de Tor y un modelo de construcción de túneles que da mucha flexibilidad a la red. Esto también permite ofuscar más eficientemente las comunicaciones para evitar detectar patrones de comunicación entre usuarios de la red.

En cuanto a la instalación, también es fácil de instalar y la consola principal es muy cómoda para trabajar en la red. Muy parecida a la de Freenet. Aunque por otra parte no tienes la flexibilidad que te ofrecen los navegadores de Tor (Tor Browser o Orfox) que incorporan funciones adicionales de seguridad para dejar la mínima información posible cuando navegas por Internet.

## 5. La red Tor

A mediados de los años 90, en el laboratorio de investigación naval de los Estados Unidos (U.S. Naval Research Laboratory), nace el concepto de *onion routing* [9]. Buscaban proteger las comunicaciones del departamento de inteligencia, que se realizaran desde cualquier parte del mundo, a través de internet. Unos años más tarde, en 1997, DARPA<sup>15</sup> continúa con su desarrollo.

En 2002 Paul Syverson, Roger Dingledine y Nick Mathewson crean la primera versión de Tor, con el nombre de *The Onion Routing Project*. Poco tiempo después, el *Naval Research Laboratory* libera el código de Tor que pasa a ser

<sup>15</sup> DARPA: Agencia de proyectos de Investigación avanzados de defensa de los Estados Unidos.

software libre. La EEF<sup>16</sup> apoya económicamente a Dingledine y Mathewson para que sigan con el desarrollo del proyecto, hasta que en 2006 se acaba fundando *The Tor Project* , una organización sin ánimo de lucro que se encarga del mantenimiento y gestión de la red Tor.

El *onion routing*<sup>17</sup> , a diferencia del enrutamiento convencional, busca mantener el anonimato y la privacidad de los usuarios. Los paquetes se envían a través de varios nodos forma aleatoria, creando un nuevo túnel para cada salto, con el objetivo de no revelar las identidades del emisor y del receptor. Este concepto de añadir “capas” en los saltos dio nombre al *onion routing* por entender que la comunicación tiene diferentes capas una sobre otra como una cebolla [9,10,11].

Podemos verlo de una forma gráfica en la Figura 11.

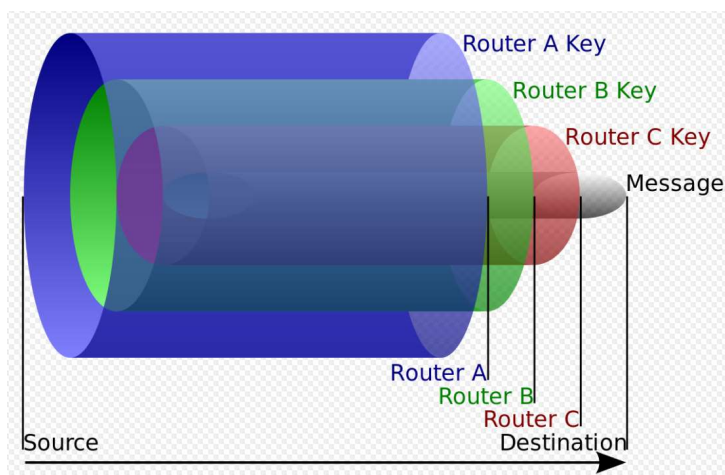


Figura 11: Cifrado por capas del onion routing

## 5.1 Aspectos técnicos

### Onion Routers

La red está formada por nodos llamados *onion routers*, que se comunican entre ellos mediante una conexión TLS. Estos nodos son pcs de voluntarios de todo el mundo que ejecutan el software de Tor. Actualmente hay más de 6000 nodos en todo el mundo (<https://metrics.torproject.org/networksize.html> ).

Tipos de nodo:

- **Exit Relay:** Nodo de salida de la red Tor. Es el último salto en los circuitos de salida (*exit circuit*).

<sup>16</sup> La *Electronic Frontier Foundation* (EEF), es una organización sin ánimo de lucro que lucha por proteger los derechos de libertad de expresión. Se mantiene a base de donaciones y su sede está en San Francisco, California.

<sup>17</sup> [https://en.wikipedia.org/wiki/Onion\\_routing](https://en.wikipedia.org/wiki/Onion_routing)

- **Non-exit Relay:** Nodos de la red Tor que envían y reciben solamente de otros nodos y clientes Tor.
- **Entry Relay:** Primer salto de un circuito Tor. Puede ser tanto un *Guard Relay* como un *Bridge*.
- **Guard Relay:** Relay que usa el cliente de entrada a la red Tor.
- **Bridge:** Relay no publicado en *consensus*<sup>18</sup>.
- **Directory cache:** Relay que descarga información de directorio de los nodos autoridad de directorio y permite a los clientes consultarlo.
- **Rendezvous point:** Relay que conecta el cliente y el *hidden service*. Cada parte construye un circuito de 3 saltos hasta este relay, con el objetivo de que no haya una comunicación directa entre el cliente y el *hidden service*.

En la Figura 12, podemos ver una comunicación habitual, que realiza un usuario que tiene un cliente de Tor y accede a un sitio web de Internet pasando por la red Tor. En ella se pueden ver tres tipos de nodos diferentes y el servicio de directorio [9,11].

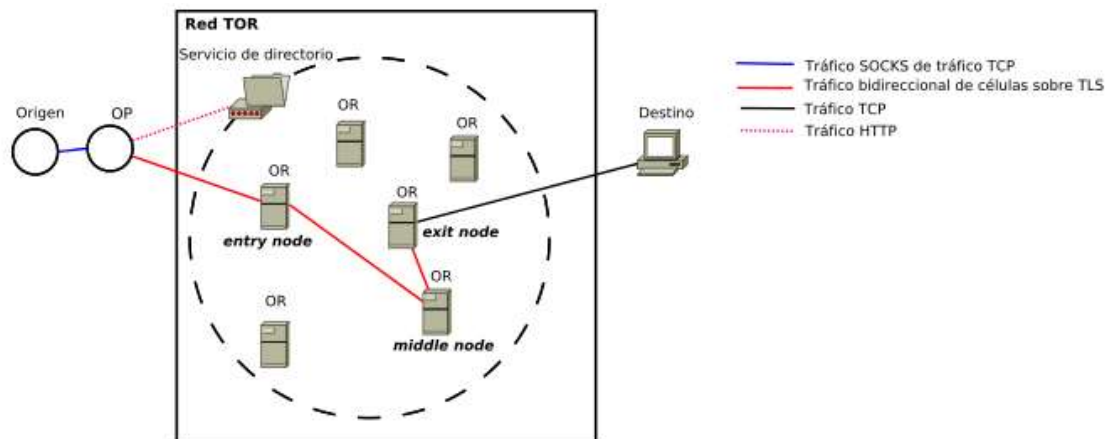


Figura 12: Nodos Tor

## Las Autoridades

En Tor, hay un pequeño grupo de nodos que tienen roles especiales y son de vital importancia para el funcionamiento de la red. Estos nodos especiales reciben el nombre de *Authorities* [11]. Es un rol que se asigna a los nodos mejor valorados por lo que no siempre podría estar asignado a los mismos nodos.

Existen estos tres tipos de autoridades:

<sup>18</sup> *Consensus* es un documento público de la red Tor que contiene la lista de nodos de la red entre otros datos.



- **Directory Authority (DA):** Hay nueve nodos con este rol. Publican información del estado de la red Tor y de la lista de nodos.
- **Bridge Authority:** Hay un nodo con este rol y publica datos de estado de la red, pero para nodos tipo *bridge*.
- **Fallback Directory Mirror:** Ayudan a los clientes a conectar a la red.

En la página de monitorización de la red Tor, se pueden ver los nodos con el *flag* de autoridad como vemos en esta la Figura 13.

<https://metrics.torproject.org/rs.html#search/flag:Authority>

Nickname <sup>†</sup>	Bandwidth	Uptime	Country	IPv4	IPv6	Flags	Add. Flags	ORPort	DirPort	Type
● <a href="#">dizum</a> (1)	3.03 MiB/s	9d 19h		194.109.206.212	-			443	80	Relay
● <a href="#">Serge</a> (1)	1.23 MiB/s	17d 59m		66.111.2.131	2610:1c0:0:5::131			9001	9030	Relay
● <a href="#">moriai</a> (1)	500 KiB/s	5d 18h		128.31.0.34	-			9101	9131	Relay
● <a href="#">tor26</a> (1)	75 KiB/s	14d 4h		86.59.21.38	2001:858:2:2:aabb:0:563b:1526			443	80	Relay
● <a href="#">bastet</a> (1)	50 KiB/s	41d 23h		204.13.164.118	2620:13:4000:6000::1000:118			443	80	Relay
● <a href="#">maatuska</a> (8)	50 KiB/s	26d 1h		171.25.193.9	2001:67c:289c::9			80	443	Relay
● <a href="#">dannenber</a> (1)	40 KiB/s	5d 5h		193.23.244.244	2001:678:558:1000::244			443	80	Relay
● <a href="#">Faravahar</a> (1)	40 KiB/s	83d 13h		154.35.175.225	2607:8500:154::3			443	80	Relay
● <a href="#">gabelmoo</a> (1)	40 KiB/s	8d 20h		131.188.40.189	2001:638:a000:4140::ffff:189			443	80	Relay
● <a href="#">longclaw</a> (1)	38 KiB/s	4d 19h		199.58.81.140	-			443	80	Relay
<b>Total</b>	<b>5.07 MiB/s</b>									

Figura 13: Nodos Authority

## Hidden Service

Es un servicio publicado por un equipo cualquiera de la red Tor, que permite mantener la ip del equipo que lo ofrece, de forma anónima. Para acceder a ellos se hace mediante el navegador y son nombres con extensión *.onion*.

## Circuito

Camino que se establece entre varios nodos de la red Tor, en el que se negocian claves criptográficas entre cada nodo y el siguiente. Existen dos tipos de circuito. El *exit circuit* es el circuito que conecta a los clientes fuera de la red Tor y el *Internal circuit* es el que se establece dentro de la red Tor, y no sale de ella.

## Células

Cuando se establece la conexión TLS entre el OP y el nodo de entrada OR, la información que se envían se realiza por paquetes de tamaño fijo llamados células (*cells*<sup>19</sup>). En la Figura 14 se muestra la estructura de las células de control y de relay.

<sup>19</sup> <https://svn.torproject.org/svn/projects/design-paper/tor-design.html>

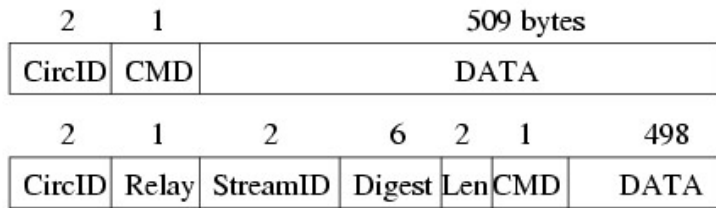


Figura 14: Los dos tipos de Célula

Los nodos usan *cell commands*<sup>20</sup> para operar con los circuitos (CREATE, DESTROY, RELAY, RELAY\_EXTEND..). Hay comandos distintos para las células de relay y de control. En la Figura 15 aparecen los comandos que se usan en una célula de relay (a la izquierda) y de control (a la derecha).

### 6.1. Relay cells

Within a circuit, the OP and the end node use the contents of RELAY packets to tunnel end-to-end commands and TCP connections ("Streams") across circuits. End-to-end commands can be initiated by either edge; streams are initiated by the OP.

End nodes that accept streams may be:

- \* exit relays (RELAY\_BEGIN, anonymous),
- \* directory servers (RELAY\_BEGIN\_DIR, anonymous or non-anonymous),
- \* onion services (RELAY\_BEGIN, anonymous via a rendezvous point).

The payload of each unencrypted RELAY cell consists of:

Relay command	[1 byte]
'Recognized'	[2 bytes]
StreamID	[2 bytes]
Digest	[4 bytes]
Length	[2 bytes]
Data	[PAYLOAD_LEN-11 bytes]

The relay commands are:

1 -- RELAY_BEGIN	[forward]	
2 -- RELAY_DATA	[forward or backward]	
3 -- RELAY_END	[forward or backward]	
4 -- RELAY_CONNECTED	[backward]	
5 -- RELAY_SENDME	[forward or backward]	[sometimes control]
6 -- RELAY_EXTEND	[forward]	[control]
7 -- RELAY_EXTENDED	[backward]	[control]
8 -- RELAY_TRUNCATE	[forward]	[control]
9 -- RELAY_TRUNCATED	[backward]	[control]
10 -- RELAY_DROP	[forward or backward]	[control]
11 -- RELAY_RESOLVE	[forward]	
12 -- RELAY_RESOLVED	[backward]	
13 -- RELAY_BEGIN_DIR	[forward]	
14 -- RELAY_EXTENDED2	[forward]	[control]
15 -- RELAY_EXTENDED2	[backward]	[control]

32..40 -- Used for hidden services; see rend-spec-{v2,v3}.txt.

The 'Command' field of a fixed-length cell holds one of the following values:

0 -- PADDING	(Padding)	(See Sec 7.2)
1 -- CREATE	(Create a circuit)	(See Sec 5.1)
2 -- CREATED	(Acknowledge create)	(See Sec 5.1)
3 -- RELAY	(End-to-end data)	(See Sec 5.5 and 6)
4 -- DESTROY	(Stop using a circuit)	(See Sec 5.4)
5 -- CREATE_FAST	(Create a circuit, no PK)	(See Sec 5.1)
6 -- CREATED_FAST	(Circuit created, no PK)	(See Sec 5.1)
8 -- NETINFO	(Time and address info)	(See Sec 4.5)
9 -- RELAY_EARLY	(End-to-end data; limited)	(See Sec 5.6)
10 -- CREATE2	(Extended CREATE cell)	(See Sec 5.1)
11 -- CREATED2	(Extended CREATED cell)	(See Sec 5.1)
12 -- PADDING_NEGOTIATE	(Padding negotiation)	(See Sec 7.2)

Variable-length command values are:

7 -- VERSIONS	(Negotiate proto version)	(See Sec 4)
128 -- VPADDING	(Variable-length padding)	(See Sec 7.2)
129 -- CERTS	(Certificates)	(See Sec 4.2)
130 -- AUTH_CHALLENGE	(Challenge value)	(See Sec 4.3)
131 -- AUTHENTICATE	(Client authentication)	(See Sec 4.5)
132 -- AUTHORIZE	(Client authorization)	(Not yet used)

The interpretation of 'Payload' depends on the type of the cell.

VPADDING/PADDING:

Payload contains padding bytes.

CREATE/CREATE2: Payload contains the handshake challenge.

CREATED/CREATED2: Payload contains the handshake response.

RELAY/RELAY\_EARLY: Payload contains the relay header and relay body.

DESTROY: Payload contains a reason for closing the circuit.

(see 5.4)

Figura 15: Cell Commands

## Consensus

Es un documento de la red Tor, que actualizan los nodos DA cada hora, con información actualizada de los elementos de la red. Los DA procesan información de todos los nodos (rendimiento, estabilidad, ancho de banda...) y realizan una votación para elegir los roles de los nodos [21].

## Funcionamiento de la red

Suponemos que el usuario Alice, solicita acceder a una página de Internet como se muestra en la Figura 16. Antes de iniciar la comunicación, el software cliente (OP) de Tor, obtiene una lista de los nodos Tor de un servidor de directorio (SD) [9,11].

<sup>20</sup> <https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt>

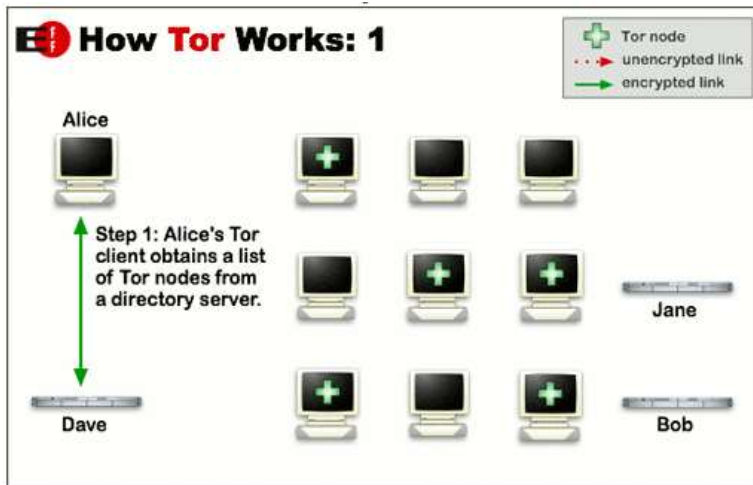


Figura 16: Alice consulta el SD

A continuación, escoge un circuito por dónde pasarán los paquetes, que como mínimo será de 3 nodos diferentes (Figura 17).

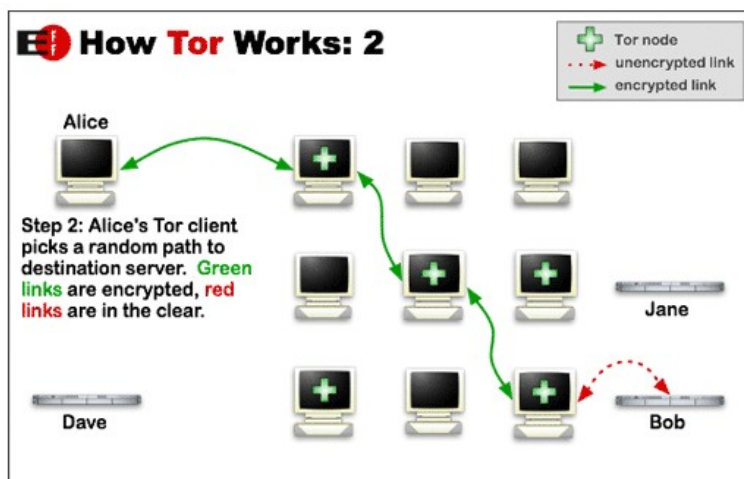


Figura 17: Establecimiento del circuito

Después negocia claves de cifrado AES256 con cada nodo por el que va a establecer comunicación. Utilizará claves simétricas distintas para cada sentido de la comunicación. Cada nodo OR solo conoce su nodo antecesor y predecesor por lo que ninguno de ellos puede saber el circuito completo. También hay que mencionar que los circuitos tienen una vida de 10 minutos. Pasado ese tiempo, se escoge un nuevo circuito.

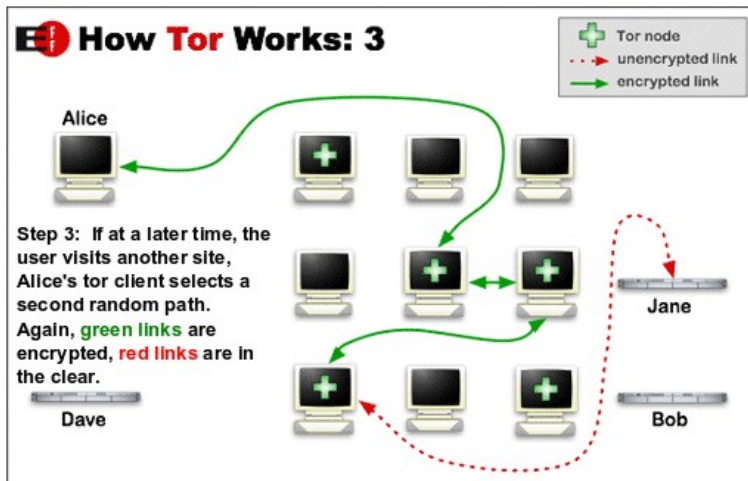


Figura 18: Alice conecta con el destino en Internet

Si el usuario ha conectado a un sitio de la *clearnet*, su comunicación dejará de estar encriptada por parte de la red Tor en cuando salga por el último nodo (*exit relay*) como se muestra en la Figura 18 . Por esa razón es preferible que el usuario solo use conexiones encriptadas extremo a extremo.

## 5.2 Uso de Tor

El número de usuarios conectados a TOR sufrió un aumento importante en octubre de 2013 por varias noticias que aparecieron en prensa relacionadas con el propósito de la NSA de vigilar Internet [28].

A partir de entonces en España el número de conexiones ha bajado al mismo nivel que en 2013 y en el resto del mundo ha sufrido algunas fluctuaciones manteniéndose ahora en unas 2M de conexiones como se puede ver en la Figura 19 y con alrededor de 6m nodos como se observa en la Figura 20.

Los datos están obtenidos de la página de monitorización de Tor: <https://metrics.torproject.org/networksize.html>.

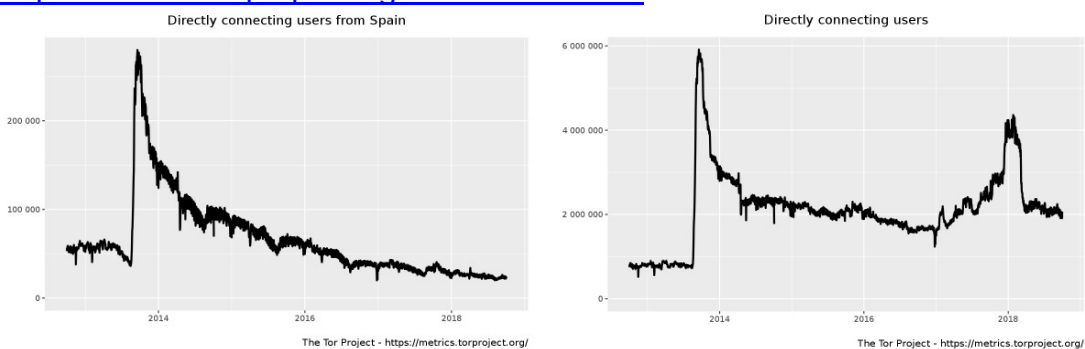
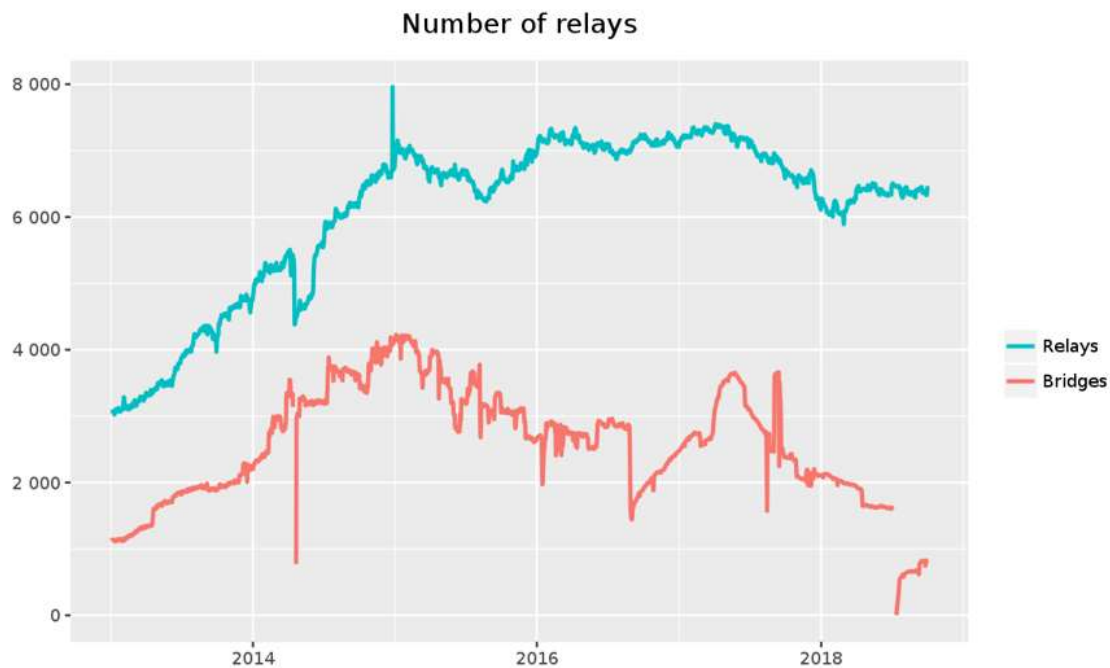


Figura 19: Número de conexiones de Tor



The Tor Project - <https://metrics.torproject.org/>

**Figura 20: Numero de nodos en Tor**

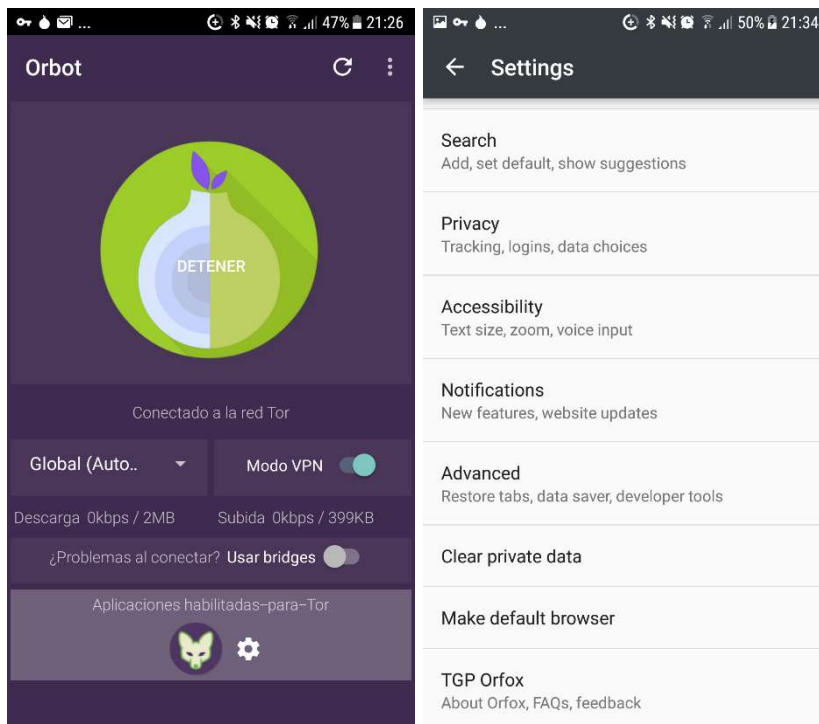
### 5.3 Accediendo a Tor

La red Tor es la *darknet* más popular y la que más opciones tiene el usuario para conectarse a ella. La forma más sencilla es instalando el navegador Tor Browser<sup>21</sup>. También es muy fácil y cómoda de instalar la versión para Android. Se instala la *app* Orbot<sup>22</sup> que hace la conexión a la red Tor y la *app* Orfox<sup>23</sup> que es el navegador que se conecta con Orbot. El aspecto de las dos aplicaciones podemos verlo en la Figura 21.

<sup>21</sup> Tor Browser: Es un navegador basado en Firefox, compatible con los sistemas operativos más populares que conecta directamente a la red Tor. Además, permite algunas opciones de seguridad adicionales diseñadas especialmente para proteger la privacidad del usuario.

<sup>22</sup> <https://en.wikipedia.org/wiki/Orbot>

<sup>23</sup> <https://guardianproject.info/apps/orfox/>



**Figura 21: Apps de Android Orbot y Orfox**

Por otra parte, hay soluciones que se integran con Tor y ofrecen un sistema operativo aislado y diseñado de base para mantener la privacidad del usuario. Tor, por su naturaleza de red *outproxy* y por ser la *darknet* más popular ha motivado proyectos de este tipo, que ofrecen medidas adicionales de seguridad.

Una de estas soluciones es la distribución de *whonix*, que se integra directamente con la red Tor y que está diseñada para agregar medidas de protección a nivel de sistema operativo y de red. El solo hecho de usar un sistema operativo aparte, y además virtualizado, ya hace que el usuario no tenga aplicaciones instaladas con datos que puedan revelar su identidad. En cuanto a la parte de red, trabaja con dos máquinas enmascarando la ip de la máquina que se usa para navegar, de forma que ni esta misma conoce con qué ip pública se conecta la otra máquina, que se encarga de realizar la conexión a la red Tor.

Adicionalmente, hay otras medidas de seguridad que whonix y este tipo de proyectos recomiendan, pero que son relativas al conocimiento y acciones del propio usuario. De alguna forma indican que ellos pueden proporcionar la herramienta adecuada para navegar por Tor, pero el conocimiento y comportamiento del usuario son fundamentales para no revelar ningún tipo de información relativa a la identidad de este.

Para probar a acceder a Tor con Whonix, primero realizaremos la instalación de la distribución de Whonix que está explicada en el ANEXO A.

Una vez tengamos el entorno de trabajo, abrimos el navegador Tor Browser que esta instalado por defecto en la máquina Workstation de la distribución de Whonix.

Pulsamos en el link IP-Check y en caso de estar conectados nos tiene que aparecer una página como la de la Figura 22.

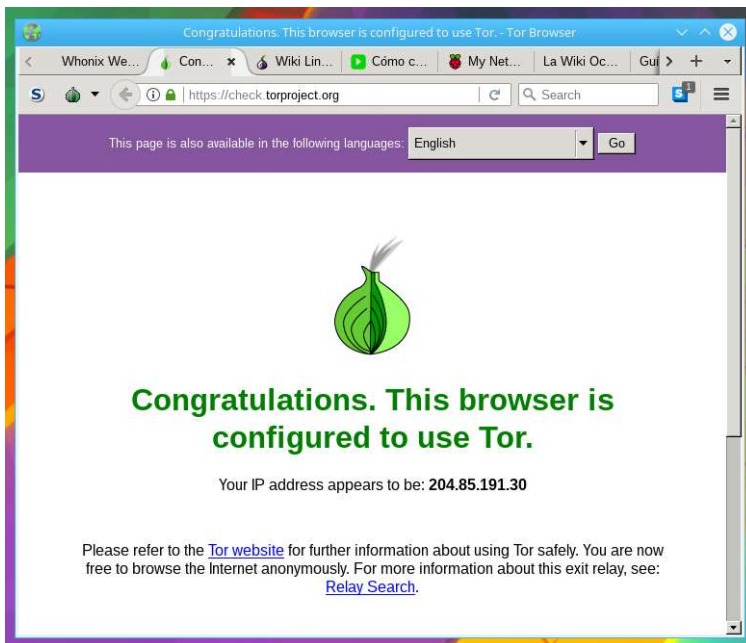


Figura 22: Página de prueba de Tor Browser

Usaremos el comando *whonixcheck* en el Gateway, que nos proporciona información acerca de si el gateway esta conectado a la red Tor y nos indicará si la distribución whonix esta actualizada.

```
user@host:~$ whonixcheck
[INFO] [whonixcheck] Connected to Tor.
[INFO] [whonixcheck] Whonix is produced independently of, with no guarantee from, The Tor Project. Whonix is experimental software. Do not rely on it for strong anonymity. https://www.whonix.org
[INFO] [whonixcheck] Whonix APT Repository: Enabled.
When the Whonix team releases STRETCH updates, they will be AUTOMATICALLY installed (when you run apt-get dist-upgrade) along with updated packages from the Debian team. Please read https://www.whonix.org/wiki/Trust to understand the risk.
If you want to change this, use:
  sudo whonix_repository
[INFO] [whonixcheck] Debian Package Update Check: Checking for software updates via apt-get... ( Documentation: https://www.whonix.org/wiki/Update )
[INFO] [whonixcheck] Debian Package Update Check Result: No updates found via apt-get.
user@host:~$
```

La herramienta *anonimising relay monitor* (Figura 23), proporciona información en tiempo real, del tráfico que esta pasando por el gateway.

```
arm - host (Linux 4.9.0-8-amd64) Tor 0.3.4.8 (recommended)
Relaying Disabled, Control Socket: /var/run/tor/control GroupWritable RelaxDirModeCheck
cpu: 20.0% tor, 1.6% arm mem: 39 MB (1.0%) pid: 6575 uptime:
page 1 / 5 - s: menu, p: pause, h: page help, q: quit
Bandwidth (limit: 8 Gb/s, burst: 8 Gb/s):
Download (20.9 Kb/sec):
Upload (0.0 b/sec):
Events (TOR/ARM NOTICE - ERR):
19:31:19 [ARM NOTICE] Unable to prepopulate bandwidth information (insufficient uptime)
19:31:19 [ARM_WARN] The torrc differs from what tor's using. You can issue a sighup to reload the torrc values by pressing x.
- torrc value differs on line: 6
- configuration values are missing from the torrc: DisableNetwork, RunAsDaemon
19:31:19 [ARM NOTICE] Unneeded torrc entries found. They've been highlighted in blue on the torrc page.
- entry matches its default value: \include (line 5)
19:31:19 [ARM NOTICE] Tor is presenting system utilities like netstat and lsof from working. This means that arm can't provide you with connection information. You can change this by adding 'DisableDebuggerAttachment 0' to your torrc and restarting tor. For more information see...
https://trac.torproject.org/3339
19:31:19 [ARM NOTICE] No armrc loaded, using defaults. You can customize arm by placing a configuration file at '/home/user/arm/armrc' (see the armrc.sample for its options).
```

Figura 23: Anonimising relay monitor

En la máquina Workstation, es dónde el usuario tiene el navegador Tor Browser y es dónde probaremos a conectar a algún sitio .onion como el popular buscador *Ahmia*<sup>24</sup> o la *Hidden Wiki*<sup>25</sup> (Figura 24):

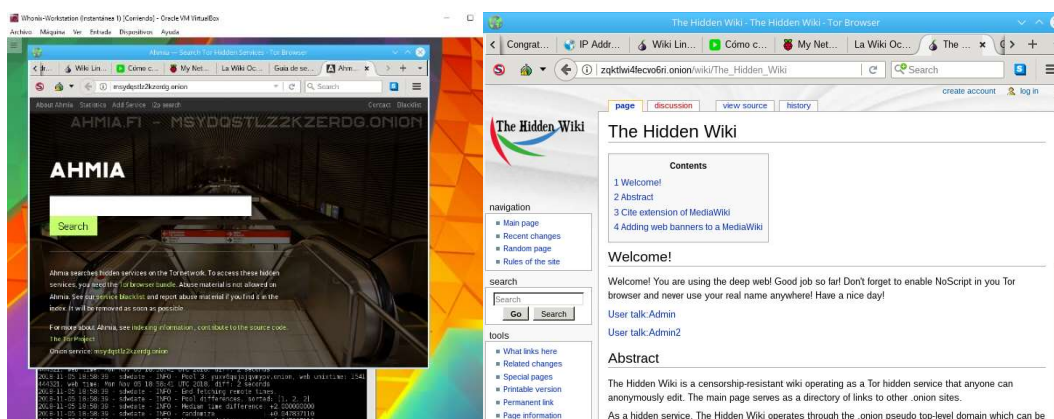


Figura 24: Buscadores Ahmia y Hidden Wiki de la red Tor

## 5.4 Resumen Tor

La red Tor, es la *darknet* más popular actualmente y la que tiene mayor número de nodos distribuidos por el mundo. Esto hace que tenga más recursos y documentación que el resto de *darknets*. La instalación de las herramientas es fácil y rápida. En pocos minutos instalas el Tor Browser en cualquier sistema operativo o la versión para Android y puedes estar navegando por Internet de forma anónima. La instalación de la distribución de Whonix, es un poco más laboriosa, pero su modelo con máquinas virtuales también facilita la instalación. Técnicamente tiene algunas mejoras posibles en el modelo centralizado de las autoridades de directorio y en los túneles que I2P ha mejorado. También te encuentras muchos enlaces de los *hidden services* que a veces no funcionan.

## 6. Conclusiones:

Cada vez usamos más internet. En la mayoría de las cosas que hacemos a diario necesitamos internet. Tenemos más servicios en la red y con ello empezamos a dejar parte de nuestros hábitos y de nuestra vida en internet. Estos datos que dejamos en la red, preocupan cada vez a más usuarios que quieren mantener su privacidad. Además, hay países donde los gobiernos han empezado a analizar el tráfico de los usuarios aplicando sistemas de filtrado para censurar ciertos contenidos [27]. Esto ha motivado a ciertos colectivos a buscar soluciones para luchar por la libertad de expresión de los internautas. Han aparecido muchos proyectos apoyados por organizaciones sin ánimo de lucro, que trabajan desarrollando redes y herramientas que protejan a los internautas.

<sup>24</sup> <https://en.wikipedia.org/wiki/Ahmia>

<sup>25</sup> [https://en.wikipedia.org/wiki/The\\_Hidden\\_Wiki](https://en.wikipedia.org/wiki/The_Hidden_Wiki)



Unas de las redes más conocidas actualmente de la Deep Web, que tienen este objetivo común, son Tor, I2P y Freenet. Todas ellas son de software libre, y buscan proteger la privacidad del usuario que consideran un derecho fundamental. Cierto es, que estas redes también pueden ser aprovechadas para realizar actividades ilícitas, pero como hemos visto no es ni mucho menos la motivación principal de sus desarrolladores, y por tanto no es del todo correcto que, al nombrarlas, se las asocie únicamente con actividades de este tipo.

La red Tor es la más conocida y usada de las *darknets*. El *onion routing* fue una revolución aplicando el cifrado por capas, que permitía a los usuarios no dejar información en la red del origen y destino de la comunicación. El hecho de que haya muchos usuarios en Tor, ha hecho que haya muchos nodos y por tanto ha facilitado su propagación. También ha permitido, que haya más documentación y distribuciones que se integran con Tor, como *Whonix*<sup>26</sup>, *Tails*<sup>27</sup> o *Qubes*<sup>28</sup>. Incluso la versión que hay para Android, que es muy fácil de usar y de instalar. Por otra parte es sencillo para el usuario montar sus propios *hidden services* y por su característica de red *outproxy* navegar directamente a internet es muy fácil y rápido. En cuanto a los puntos débiles de Tor, se le ha criticado que no tuviera los roles de las *authorities* descentralizados, y que todo el tráfico pasara por circuitos bidireccionales facilitando el análisis de patrones de comunicación entre dos puntos de la red.

En el caso de I2P, hemos visto que mejora algunos aspectos de Tor como es la descentralización de la base de datos de la red y el *garlic routing* que podría considerarse una versión mejorada del *onion routing*, dado que añade una capa adicional de encriptación en el mensaje permitiendo que éste contenga mensajes para varios destinatarios. Por otra parte, I2P establece múltiples circuitos unidireccionales, permitiendo mayor flexibilidad y diversificación de las comunicaciones frente a Tor. Por el contrario, no tiene tantos nodos y no hay tanta documentación ni desarrollos. Y la navegación del usuario a internet no es su punto fuerte. Al no tener tantos nodos no es tan buena como en Tor y no es tan sencillo para el usuario.

Y podríamos decir que Freenet, tiene como aspectos destacados la modalidad *darknet* que permite al usuario crear un entorno privado entre varios usuarios y la ventaja de que el contenido que sube el usuario o una página que cree, se queda en la red. No obliga al usuario a tener esos datos en su pc, ni que éste tenga que estar encendido. Es un concepto interesante, por ejemplo para usuarios que estén en países en los que se aplique algún tipo de censura, ya que permite al usuario no tener estos contenidos localmente, ni tiene que tener el equipo encendido para publicar una web por ejemplo. Como parte negativa, podríamos decir que obliga al usuario a tener su nodo suficientemente sincronizado para poder acceder a los contenidos a una velocidad razonable. Por otra parte, el usuario tiene una alta dependencia de su nodo teniendo allí sus archivos descargados e identidades.

---

<sup>26</sup> <https://www.whonix.org/>

<sup>27</sup> <https://tails.boum.org/index.en.html>

<sup>28</sup> <https://www.qubes-os.org/doc/torvm/>

En definitiva, podríamos decir que para navegar directamente por internet de forma anónima destacaríamos Tor. Para publicar anónimamente un servicio interno es una buena opción hacerlo con I2P, y Freenet es una buena red para crear un entorno privado con otros usuarios.

Pero son proyectos que están en continua evolución. Es probable que sigan creciendo y que aparezcan nuevos proyectos basados en estas redes o con nuevas tecnologías. Un ejemplo puede ser la red *Kovri* de Monero<sup>29</sup> que se basa en I2P y que busca mantener la privacidad de los usuarios en sus transacciones con criptomonedas Monero [26]. Investigar estos proyectos o analizar la evolución de estas tecnologías podrían ser temas para futuros proyectos de investigación complementarios a este trabajo.

Finalmente, en cuanto a los contenidos que hemos encontrado en las tres redes, podríamos decir que es similar. Generalmente hay documentación técnica asociada a las propias *darknets*, temas de cifrado, programación, seguridad, *pentesting*<sup>30</sup>, *hacking* y temas similares. Suele estar en foros o páginas web internas. También hay páginas y foros de opinión que te recuerdan los principios de internet con páginas muy básicas.

En relación al contenido ilícito que encuentras, es cierto que sin mucho esfuerzo puedes llegar a algunas referencias de libros electrónicos, música o películas y algunos links a páginas de drogas. Pero en general, es contenido que tienes que buscar, por lo que es algo residual. Por tanto, podemos concluir que no es correcto asociar de forma general la Deep Web a este tipo de contenidos o actividades.

---

<sup>29</sup> Criptomoneda de código abierto descentralizada y que prioriza la privacidad basada en el protocolo *Cryptonote*.

<sup>30</sup> [https://en.wikipedia.org/wiki/Penetration\\_test](https://en.wikipedia.org/wiki/Penetration_test)

## 7. Referencias

- [1] <https://es.wikipedia.org/wiki/Peer-to-peer> | 10/12/2018
- [2] [https://es.wikipedia.org/wiki/Internet\\_profunda](https://es.wikipedia.org/wiki/Internet_profunda) | 18/12/2018
- [3] <https://www.xataka.com/basics/que-es-la-dark-web-en-que-se-diferencia-de-la-deep-web-y-como-puedes-navegar-por-ella> | 20/12/2018
- [4] [https://documents.trendmicro.com/assets/wp/wp\\_below\\_the\\_surface.pdf](https://documents.trendmicro.com/assets/wp/wp_below_the_surface.pdf) | 23/12/2018
- [5] <https://es.wikipedia.org/wiki/Darknet> | 10/12/2018
- [6] <https://geti2p.net/es/docs/how/intro> | 21/11/2018
- [7] <https://www.redeszone.net/2012/09/07/i2p-red-segura-y-anonima-para-navegar-chatear-y-descargar-archivos/> | 20/12/2018
- [8] <https://en.wikipedia.org/wiki/Freenet> | 20/12/2018
- [9] [https://es.wikipedia.org/wiki/Tor\\_\(red\\_de\\_anonimato\)](https://es.wikipedia.org/wiki/Tor_(red_de_anonimato)) | 20/12/2018
- [10] <https://www.genbeta.com/seguridad/como-funciona-la-red-tor> | 20/12/2018
- [11] <https://www.torproject.org/docs/documentation.html.en> | 20/12/2018
- [12] <https://www.welivesecurity.com/la-es/2014/09/05/mitos-realidades-deep-web/> | 20/12/2018
- [13] <https://geekland.eu/mitos-de-la-internet-profunda-o-deep-web> | 20/12/2018
- [14] [https://es.wikipedia.org/wiki/Internet\\_profunda](https://es.wikipedia.org/wiki/Internet_profunda) | 20/12/2018
- [15] <https://expansion.mx/tecnologia/2014/03/10/las-profundidades-del-mar-de-internet> | 20/12/2018
- [16] <https://freenetproject.org/pages/documentation.html> | 23/12/2018
- [17] <https://geti2p.net/es/download#windows> | 20/12/2018
- [18] <https://es.wikipedia.org/wiki/I2P> | 20/12/2018
- [19] <https://geekytheory.com/que-es-y-como-funciona-la-red-tor> | 20/12/2018

- [20] <https://resources.infosecinstitute.com/hacking-tor-online-anonymity/#gref> | 24/12/2018
- [21] <https://jordan-wright.com/blog/2015/05/14/how-tor-works-part-three-the-consensus/> | 24/12/2018
- [22] [https://es.wikipedia.org/wiki/Internet\\_superficial](https://es.wikipedia.org/wiki/Internet_superficial) | 23/12/2018
- [23] <https://freenetproject.org/pages/download.html> | 23/12/2018
- [24] <https://github.com/freenet/wiki/wiki/Content-Hash-Key> | 23/12/2018
- [25] <https://freenetproject.org/pages/documentation.html> | 24/12/2018
- [26] <https://src.getmonero.org/resources/moneropedia/i2p.html> | 24/12/2018
- [27] [https://es.wikipedia.org/wiki/Censura\\_en\\_Internet](https://es.wikipedia.org/wiki/Censura_en_Internet) | 24/12/2018
- [28] [https://www.bbc.com/mundo/noticias/2013/10/131031\\_eeuu\\_nsa\\_espionaje\\_tecnicas\\_az](https://www.bbc.com/mundo/noticias/2013/10/131031_eeuu_nsa_espionaje_tecnicas_az) | 24/12/2018

# ANEXO A: Instalación de la distribución Whonix

## A.1 Recursos

Para preparar el entorno de trabajo hemos usado un equipo con procesador i7 de cuatro núcleos, 16MB de RAM y un disco de 500GB.

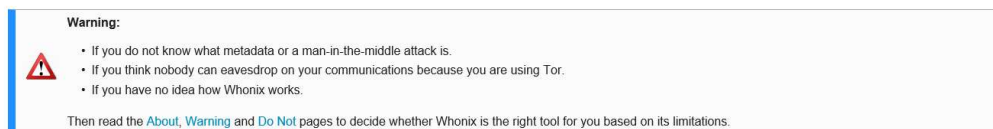
El sistema operativo base es un *Windows 10 enterprise* de 64 bits que tiene instalado Oracle VM Virtualbox v 5.2.18.

Para conectarnos a internet usaremos una conexión FTTH de 100Mbps casera con ip dinámica.

## A.2 Documentación Whonix

Como hemos comentado anteriormente, Whonix es una distribución diseñada para mejorar la seguridad y privacidad del usuario. Pero en la propia documentación te advierte de que te replantees usar esta distribución, si no sabes qué son los metadatos, un ataque *man-in-the-middle*, si piensas que por estar en Tor nadie puede escuchar tus comunicaciones o si no tienes idea de cómo funciona Whonix como aparece en la Figura 25.

### First Time Users



**Figura 25: Advertencia whonix usuarios principiantes**

Esto pretende evitar que un usuario sin conocimientos tenga una falsa sensación de seguridad solo por el hecho de usar Whonix.

Whonix es una herramienta que te puede ayudar a mantener tu privacidad en este sentido, pero al final tu seguridad y anonimato dependerá de tus acciones y de las configuraciones que hayas puesto.

Veamos de forma general estos conceptos, que recomienda Whonix conocer, antes de usar su distribución y porqué son importantes para mantener nuestra privacidad.

### A.2.1 Metadatos

Los metadatos se definen como datos que describen otros datos. Hay que ser conscientes de que los archivos que creamos pueden contener datos como por ejemplo del autor, fecha de creación, idioma, o posición GPS de una foto del sitio dónde fue tomada. El usuario debe conocer qué metadatos tienen sus archivos, especialmente los que vaya a publicar o compartir con otros.

Existen herramientas para ver o eliminar metadatos de los archivos. Por ejemplo:

- Metadata Anonymisation Toolkit (incluido en Whonix)
- Mat2<sup>31</sup>

En este artículo de la BBC<sup>32</sup>, se comenta el caso de un hacker al que el FBI estaba investigando. Fue descubierto por una foto que publicó, que contenía metadatos del sitio preciso dónde fue tomada la foto (Figura 26).

La foto había sido tomada con un iPhone y a w0rmer se le escapó el hecho de que las fotos tomadas con este dispositivo contienen en su metadata información sobre qué tipo de aparato las tomó, con qué configuración y, lo más importante, dónde se tomó indicando la latitud exacta.

De este modo, los agentes pudieron averiguar que ésta había sido tomada por un teléfono en Australia. Como verán más adelante, esta información sería crucial para inculpar al sospechoso.

“

Ya es tiempo de que los gobiernos admitan a la gente que gobiernan que los datos que almacenan no están a salvo, ni seguros. Algo debe hacerse

w0rmer, pirata informático

Figura 26: Metadatos en fotografías

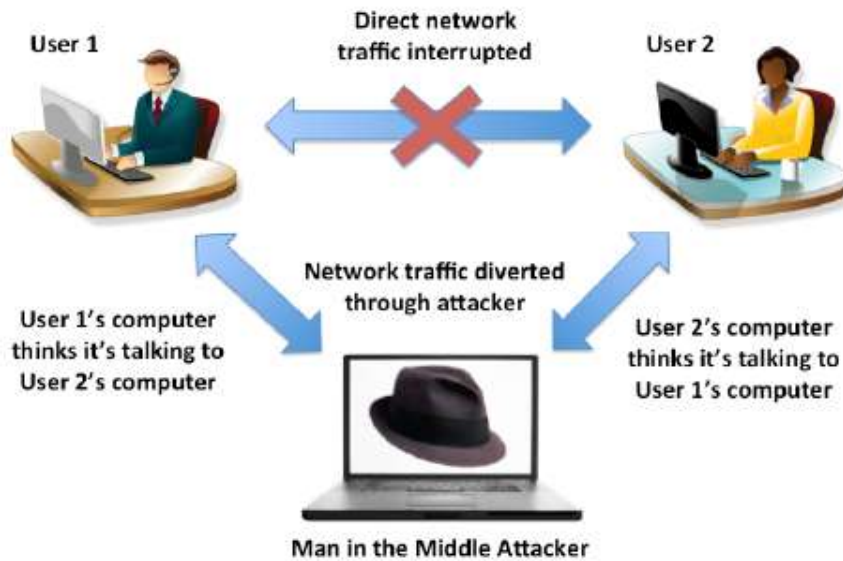
### A.2.2 MITM

En un ataque *man-in-the-middle*<sup>33</sup>, el atacante consigue tener la capacidad de leer, insertar o modificar los datos que se envían entre dos víctimas (Figura 27).

<sup>31</sup> <https://mat.boum.org>

<sup>32</sup> [https://www.bbc.com/mundo/noticias/2012/04/120413\\_tecnologia\\_pechos\\_atraparon\\_hacker\\_aa](https://www.bbc.com/mundo/noticias/2012/04/120413_tecnologia_pechos_atraparon_hacker_aa)

<sup>33</sup> [https://www.whonix.org/wiki/Warning#Man-in-the-middle\\_attacks](https://www.whonix.org/wiki/Warning#Man-in-the-middle_attacks)



**Figura 27: Ataque Mitm**

En la documentación de Whonix, te advierte en este sentido de que si te conectas a la red Tor y luego navegas por páginas de la *clearnet*, tu tráfico puede ser interceptado. Si el último nodo Tor, esta comprometido y tu comunicación no está cifrada extremo a extremo, tu comunicación puede ser interceptada a pesar de que estés conectado a la red Tor.

### A.2.3 Como funciona Whonix

La distribución de whonix se basa en dos máquinas. Una máquina se usa para hacer de puente con la red Tor y la otra es la estación de trabajo del usuario.

La primera máquina, llamada Gateway, tiene dos interfaces de red. Uno de ellos conecta a la red pública y se usa para conectar a la red Tor. El otro interfaz está en una red interna.

La segunda máquina, o Workstation, solo tiene una interfaz de red que está conectada a la misma red interna que el Gateway. Es decir, esta máquina no está conectada directamente a la red pública como se puede ver en la Figura 28.

# Whonix

## Anonymous Operating System

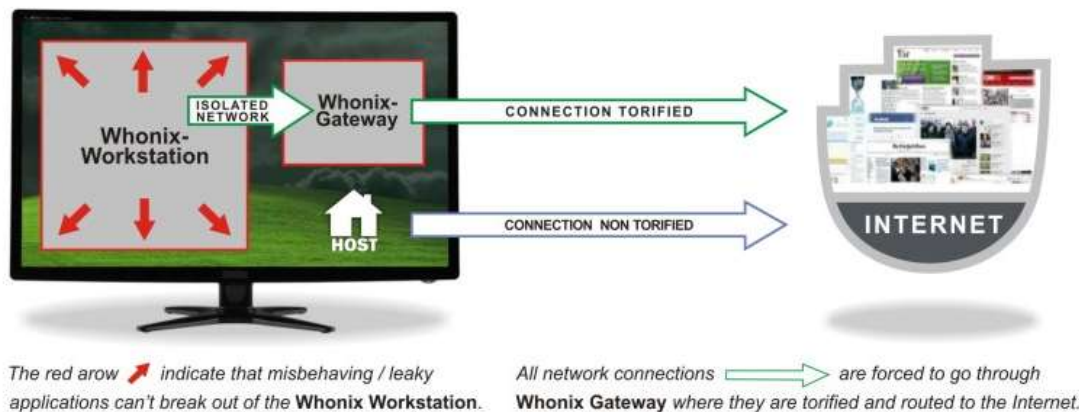


Figura 28: Diagrama de conexión maquinas Whonix

Una conexión que haces pasar por la red Tor, también se llama *conexión torificada* como también se muestra en la Figura 28.

### A.2.4 Medidas de seguridad adicionales

A parte de los conceptos básicos que recomienda whonix conocer, encontramos varias recomendaciones que nos pueden ayudar a mantener nuestro anonimato en cualquier red.

**Javascript:** Desactiva Javascript en el navegador.

**Metadatos exif:** Son metadatos más difíciles de borrar y no todas las herramientas los borran. Están en archivos tipo jpg, jpeg, tif y wav.

**VPN:** Si pensamos que nuestra conexión a internet no es segura o nos preocupa lo que pueda ver nuestro operador ISP, podemos usar una VPN para proteger nuestra comunicación y ocultar a qué sitios conectamos.

**Cookies:** Hay que tener cuidado con las Cookies de rastreo de algunos sitios<sup>34</sup>.

**Identidad en la DeepWeb:** Si queremos mantener el anonimato por completo, y que nadie pueda cruzar datos para obtener nuestra identidad real, es recomendable adoptar identidades diferentes en la red Tor. No debemos conectar con esta identidad, a sitios de la *clearnet* a los que conectamos con nuestra identidad real, para que no puedan relacionarse ambas identidades (por

<sup>34</sup> Han aparecido documentos que revelan que la NSA ha podido usar *cookies* de navegación para obtener información de PCs que tenían bajo sospecha (<https://cacm.acm.org/news/170468-nsa-uses-google-cookies-to-pinpoint-targets-for-hacking/fulltext> | 20/11/2018).



ejemplo no entraremos en servicios de correo , chat o redes sociales de nuestra identidad real cuando naveguemos por la DeepWeb).

**Patrones de comportamiento:** En la red Tor, vigilarémos la forma que tenemos de escribir a otros usuarios, como construimos las frases , las faltas que cometemos , nuestro argot , horario de actividad , rutina , idioma , y en definitiva qué personalidad presentamos para evitar que alguien pueda cruzar datos nuestros y averiguar información de nuestra identidad real.

**Cifrado:** Intentar no escribir en texto plano. Usar PGP para cifrar los archivos, emails o mensajes...

**Clearnet:** Si salimos a la internet abierta desde la red Tor, debemos usar comunicaciones cifradas extremo a extremos como HTTPS. Hay que vigilar porque algunos protocolos de encriptación ya no son seguros .

**Servicios en la red Tor:** Aunque los servicios *.onion* no están en la internet abierta, y por defecto las comunicaciones están cifradas , el servidor siempre puede verse comprometido. Por tanto hay que vigilar lo que hacemos en ese servidor y qué información nuestra tiene.

**Seguridad física:** Si no queremos dejar rastro de nuestras actividades en el disco duro local, podemos arrancar y usar la distribución en USB. Podemos guardar las claves privadas PGP<sup>35</sup> en un USB o memoria que llevemos siempre encima. Así si roban nuestro equipo no podrán acceder a nuestros datos cifrados sin las claves privadas.


### A.3 Instalación de Whonix

Para instalar las dos máquinas, primero descargamos las imágenes de la página oficial de la distribución de Whonix en <https://www.whonix.org>. Son dos archivos .ova de 1GB aproximadamente cada uno.

Descargamos también los archivos de firma PGP y la *Signing Key* del autor ( Figura A5) que validaremos a continuación con otra herramienta.

---

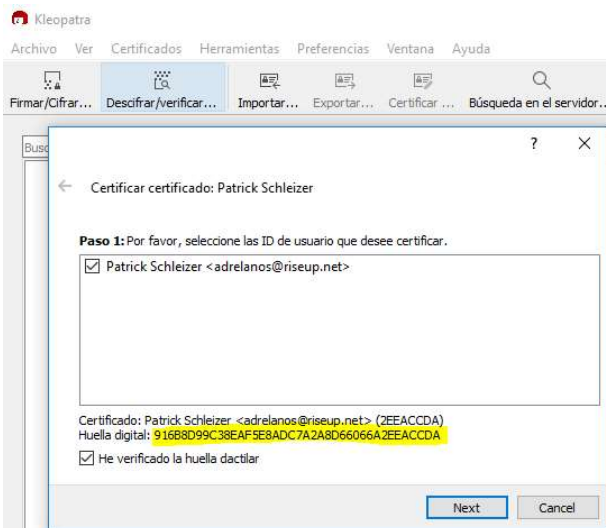
<sup>35</sup> [https://en.wikipedia.org/wiki/Pretty\\_Good\\_Privacy](https://en.wikipedia.org/wiki/Pretty_Good_Privacy)

	Whonix-Gateway (850 MB)	Whonix-Workstation (1.1 GB)	Anonymous Download Possible <sup>[1]</sup>	Download Security without Verification	Download Security with Verification
 <a href="https://">https://</a>	Download (v3 Onion)	Download (v3 Onion)	Yes <sup>[1]</sup>	Medium	High <sup>[2]</sup>
	OpenPGP Signature (v3 Onion) ( sha512, sig)	OpenPGP Signature (v3 Onion) ( sha512, sig)	Yes <sup>[1]</sup>	-	-
	Verify the images using the <a href="#">Signing Key</a>		Yes <sup>[1]</sup>	-	-

**Figura 29: Firmas PGP de las descargas**

Para validar la firma PGP instalaremos la aplicación Gpg4win. Descargaremos el software de la página oficial <https://www.gpg4win.org>.

Una vez realizada la instalación, abrimos la aplicación gráfica “Kleopatra”. Importamos la clave *signing key* que descargamos antes. Vemos en la Figura 30 que es de Patrick Schleizer.



**Figura 30: Huella digital Patrick Schleizer**

Verificamos que la huella es la misma, que la de la página original del software como vemos en la Figura 31.



## Key Transition

### Signed by New Key

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

My old key was:

pub 4096R/0x9C131AD3713AAEEF 2012-03-02
Key fingerprint = 9B15 7153 925C 303A 4225 3AFB 9C13 1AD3 713A AEEF
uid [ultimate] adrelanos <adrelanos@riseup.net>
sub 4096R/0xFEFB3583794279C4 2012-03-02

My new key is:

pub 4096R/0x8D66066A2EEACDDA 2014-01-16 [expires: 2015-01-16]
Key fingerprint = 9168 8D99 C3BE AF5E 8ADC 7A2A 8D66 066A 2EEA CCDA
uid [ unknown] Patrick Schleizer <adrelanos@riseup.net>
sub 4096R/0x3B1E6942CE998547 2014-01-16 [expires: 2015-01-16]
sub 4096R/0x10FDAC5311983FD6 2014-01-16 [expires: 2015-01-16]
sub 4096R/0xC88D508B778B3C48 2014-01-16 [expires: 2015-01-16]
```

Figura 31: Huella digital de la página de Whonix

Una vez importado, verificamos que Kleopatra nos da la firma como correcta (Figura 32).

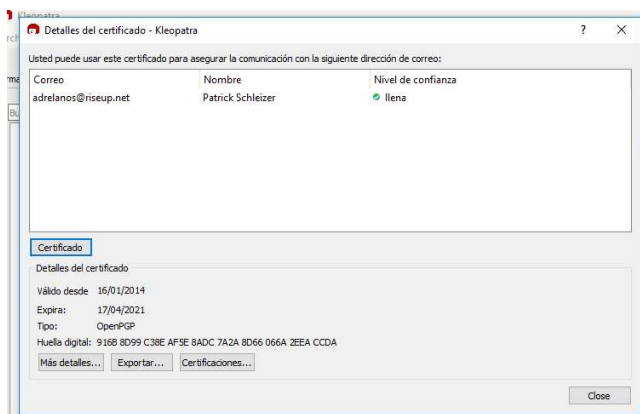


Figura 32: Nivel de confianza de la firma

Verificamos ahora las firmas de los dos archivos .ova (Figura 33).

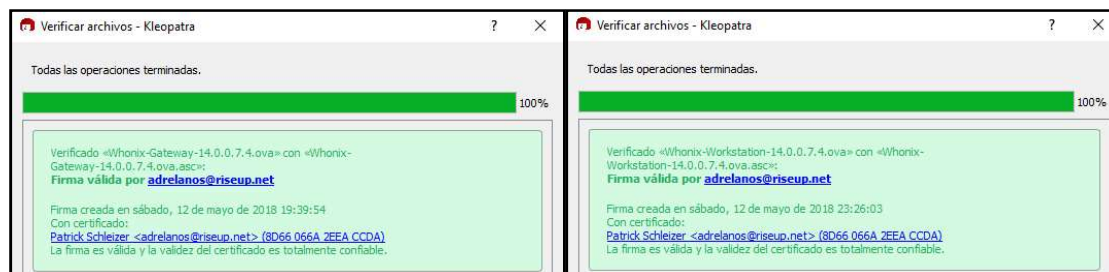


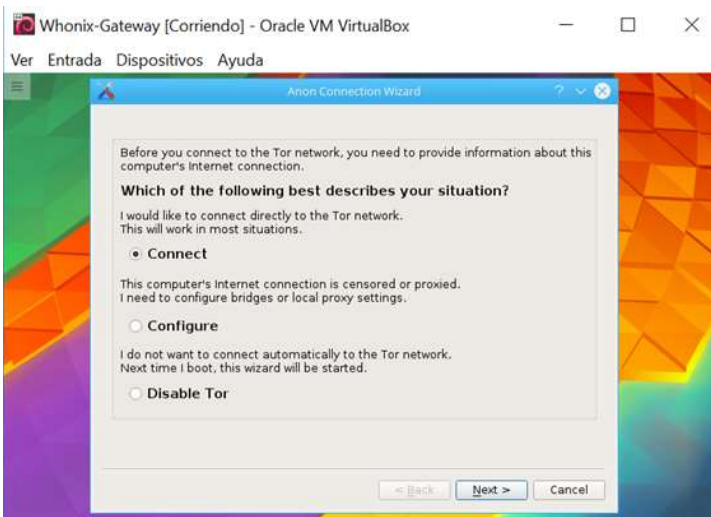
Figura 33: Firmas archivos descargados

Una vez validadas las firmas, importaremos las máquinas en Oracle VM Virtualbox. Nos deben aparecer estas dos máquinas que aparecen en la Figura 34.



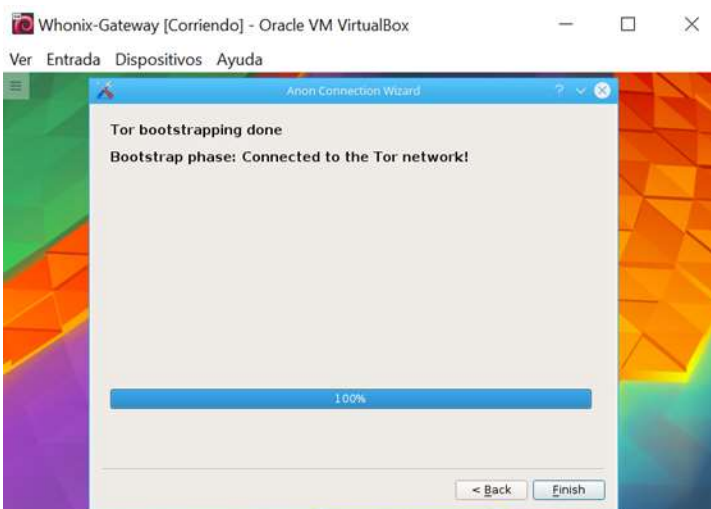
**Figura 34: Maquinas Whonix**

A continuación , vamos a la configuración de los recursos de las máquinas, y asignamos 4GB de RAM y 2 procesadores. Arrancaremos la máquina Gateway. Escogemos la opción “connect” que nos recomienda por defecto en caso de tener una conexión directa a internet (Figura 35).



**Figura 35: Tipo de conexión**

Cuando terminamos con la instalación nos indica que la máquina esta conectada a la red Tor (Figura 36).



**Figura 36: Instalación finalizada**

Una vez hemos terminado con el Wizard de instalación, nos indica que instalemos las últimas actualizaciones (Figura 37).

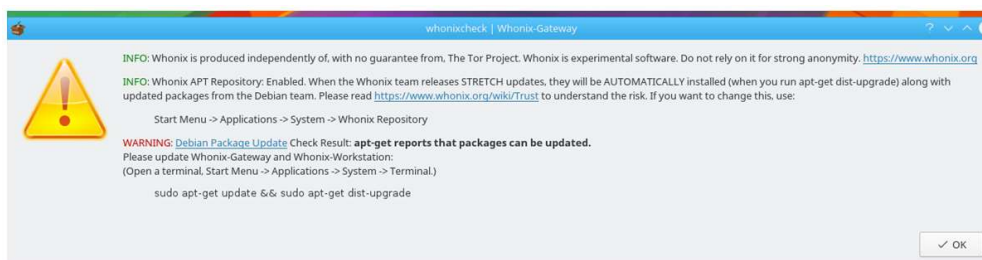


Figura 37: Actualizaciones

A continuación abrimos una consola y ejecutamos:

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

```
Please update Whonix-Gateway and Whonix-Workstation:  
(Open a terminal, Start Menu -> Applications -> System -> Terminal.)  
sudo apt-get update && sudo apt-get dist-upgrade  
user@host:~$ sudo apt-get update && sudo apt-get distr-update
```

En cuanto a ajustes de usuario podemos seleccionar el teclado en español, pero no ajustaremos la configuración de hora dado que es un parámetro que nos recomiendan no modificar como se muestra en la Figura 38.

Introduction [\[edit\]](#)



**Warning:** The system clock inside Whonix is set to UTC to prevent against time zone leaks. This means it may be a few hours ahead or behind the user's host system clock. It is strongly recommended not to change this setting.

It is recommended to read the [TimeSync Technical Design](#) page together with this chapter, although it is a difficult topic. Interested users, developers and auditors should also review the footnotes for additional information, or to explore design elements and the reasoning for this entry.

Figura 38: TimeSync

Para terminar, siguiendo las recomendaciones de seguridad anteriores, generaremos un par de claves PGP con la herramienta GPG, para una identidad que usaremos de ejemplo, que se llamará *LordByron*. Esto nos permitirá cifrar archivos o mensajes que queramos compartir dentro de la red de forma segura.

Ejecutamos:

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

```

user@host:~$ gpg --gen-key
gpg (GnuPG) 2.1.10-1 Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: LordByron
Email address: LordByron@email.com
You selected this USER-ID:
  "LordByron <LordByron@email.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/user/.gnupg/trustdb.gpg: trustdb created
gpg: key 0x4A5782A5D7CE5E44 marked as ultimately trusted
gpg: directory '/home/user/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/user/.gnupg/openpgp-revocs.d/EA75CCE22072857B5EE5A30D4A5782A5D7CE5E44.rev'
public and secret key created and signed.

pub  rsa2048/0x4A5782A5D7CE5E44 2018-11-01 [SC] [expires: 2020-10-31]
     EA75CCE22072857B5EE5A30D4A5782A5D7CE5E44
uid  LordByron <LordByron@email.com>
sub  rsa2048/0x0FBCFADA57C7518C 2018-11-01 [E] [expires: 2020-10-31]

```

Exportamos nuestra clave pública que sería la que daríamos a nuestros contactos.

```

$ gpg --armor --output public-key-lordbyron.txt --export 'LordByron'
$ cat public-key-lordbyron.txt

```

```

user@host:~$ gpg --armor --output public-key-lordbyron.txt --export 'LordByron'
user@host:~$ cat public-key-lordbyron.txt
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBFvazEsBCADZ81/xUA6r91atMKZwtpnwcAo4K8f71WuwmS1eBKFxRCAJakCy
EiCHd8L3BkkjLJWamhYXjSjK4FyJHPfyN3vKsE47UnDy2iKPq426tzKI1C7ZMMw9
p0S7nQvgDzjidqp/mdCA9oc0SvdXc/buy4TLiZEX4XflrGNXqG7mDfVYjffSHOIff
l8s+UMarAc1SvARWjjoIdjLP8ucVUly3hJRQtr9KWNK0a1EtUOLCAi9t1h6hVklX
6d+b55nMR7GRp2phORndXzjA6+474+vL0tWh5Y5GxacLWBFqJnTLVLF2D/LmGX
yvbxdmL641pvfLQdtqdxTEaR12016475uhQ/ABEBAAGOHoxvcmRceXJvbiA8Tg9y
ZEJScm9uQUVtYWlsLnNvbT6JAVQEEwEKAD4WIQTdcziIHKFe171ow1Kv4K1185e
RAUQW9rMSwTbAwUJ8JnAAULCQgHawUVcgkIcWUAgMBAAIeAQI.XgAAKCRBKV4K1
185eRFdqB/wIbKJBO+i1LLZ0gbNG9vD0dqXqQ6JhFpFPtWpJvod3M2465wrrH10x
riIWBj09lqJy93ZhzKlqFkR0GqnJC1mhLwI/SgtQ37vVPz5S2iYbLQgQHTPoQxF
GITHReDcoIo2W6J4p7IYYWqW409vdbWgKo4Q2tnVMcSbTM8+4whPLppriMLgcJwT
mAngRGS/58W82mLJfgXr1MeXf/DK4I1r47v0H0f6Vwzt3yeLehEpLZUKcCOaq+54
3fiTf1Lnw5q1V+Qkjvniog18ebnY8vWouP8Jjx7fiVsono4j5EkdqSk/V4MjL3
VQ5DLxPa/ecN1QbMwLaY2MdTB45rtAruQENBFvazEsBCAC0Ky+9RAZC/5F419
FTGovtUoPeAT86jhRzwpzeZD81/UY0/+XoXpS1dLPqJiW1wtyH0iwoLLUvEYaeR
UFZ0veag2FiANSzyuMoAYhmouC1WhCVjFGodIvSpkHj1i7v8z00ejmyY6qETfL4s
qeMp4hji1WJEppzGGLFAWuj5eJh++u2EbKjU6DuYXVf2GHv5rLYRbKvCqa/wQX0
roR3Ffp417NRHP33uVCKLsJ9QXpm6PBnaCj/EebZrRoDPKvSX1ini6L/kWVEH
BOPNkb1VXXEvCTzypuYn9z7TSa7k+GEFA+5wpBGM39X3t0AcqsK3AMWmepobUt
SL9pABEBAAGJATvEGAeAKACyWIQTdcziIHKFe171ow1Kv4K1185eRAUQW9rMSwTb
DAUJ8JnAAAKCRBKV4K1185eRNUVCAC0XlF63yhJSpPvvnZ1FwLkRnAW2WmVcqd
Shrv9UQLeyyAJqPDRNspYPhYmrGgay61TcsvKw9G3kgYmIDI9DZCHG9503uu9Q5
Np15Jriw24773Rj/7pJQ3C9byLdcBX621r+je8RUS7G27xiCeE7+N0k9wCA5TBH
gZXLcViMkNFw3mMQApocFFACOnHOVzqIjMacIU7tKa+xMIkbu8SKf/bJay3XFf
gPpWQtVJSFwoi1JPSHX0hZyQTzpgT9CIUypSpvYquV5PoEuoP1u7brkwZB5bXiH
v1Lb9MatZkbl1CNwL9s8MBYGNhc1T63g5oe8v+x486N744Q1VqaU
=VIg1
-----END PGP PUBLIC KEY BLOCK-----

```

## A.4 Conclusiones

Whonix es una distribución fácil de montar. Sobre todo facilita mucho las cosas disponer de las máquinas en formato .ova . Funciona bien en general, no requiere muchas configuraciones y los *updates* han funcionado bien durante las pruebas. A veces el servicio *sdwdate* indica que la hora no está sincronizada y te recomienda apagar la máquina virtual y arrancarla de nuevo.

Por otra parte, creo que es una distribución para un perfil de usuario experto. Hay mucha documentación técnica que tienes que leer relacionada con configuraciones y medidas de seguridad adicionales. No es recomendable plantearte usar Whonix sin tener unos conocimientos mínimos de seguridad informática y si no conoces cómo funciona la red Tor mínimamente. Pero por otra parte consigue que llegues a tener un nivel de paranoia muy alto trabajando con claves PGP, siendo consciente de no navegar en internet a sitios de tu identidad real, configurar bien el navegador, no generar patrones de comportamiento en la red, etc.. Me ha parecido una experiencia muy interesante en este sentido.