Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Grado en Ingeniería Informática

Comparativa entre diferentes herramientas para la publicación de páginas web en la Dark Web

Trabajo fin de estudio presentado por:	Miguel Trigueros Muñoz
Línea de investigación:	Ingeniería en la Web, servicios Web y Seguridad en la Web.
Director/a:	Dr. Sánchez Rubio, Manuel
Fecha:	Julio de 2022

Resumen

Varias razones llevan a los internautas a optar por el anonimato a la hora de navegar o publicar. Estas razones no siempre son criminales.

Debido a que Internet es utilizado por una gran parte de la población mundial, existe una gran diversidad en cuanto a sus usuarios, cuyas necesidades y habilidades técnicas son muy diversas.

Existen ciertas herramientas que permiten a estos usuarios publicar páginas web anónimas mediante servicios Onion. Estas herramientas son variadas tanto en características como en los conocimientos técnicos que se requieren para su uso.

Este trabajo examina el uso de tres de estas herramientas a través de la creación de tres sitios web diferentes utilizando los servicios Onion para proporcionar una comparativa entre ellos. Además, se diseñará una aplicación destinada a visitar las páginas web diseñadas con el objetivo de extraer automáticamente los enlaces a los sitios web oscuros que estas páginas tienen.

Palabras clave: Dark Web, Servicios Onion, Red Tor, Publicación, Alojamiento

Abstract

Several reasons lead Internet users to choose anonymity when surfing or publishing. These reasons are not always criminal.

Since the internet is used by a huge amount of the Earth's population, there is great diversity in terms of its users, whose needs and technical skills are highly diverse.

There are certain tools that allow these users to publish anonym web pages as Onion services. These tools are varied in characteristics as well as in the technical knowledge that is required for their use.

This work examines the use of three of these tools through the creation of three different websites using Onion services in order to provide a comparative assessment between them. In addition, an application will be designed aimed at visiting the designed web pages with the objective of automatically extract the links to the dark web sites they have.

Keywords: Dark Web, Onion services, TOR, Publish, Hosting

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción

Desde los orígenes de Internet el anonimato nunca ha sido uno de los requisitos de esta tecnología. Ya sea en los fundamentos militares de la Red, que podemos situar en la famosa Arpanet, como en los científicos, cuyo exponente podría ser el origen de la World Wide Web en el Cern, el anonimato no era necesario. Los usuarios que accedían a la red se suponían autorizados y autenticados.

Con la apertura de Internet y la web a la sociedad ésta tomó la Red como un lugar anónimo. La web 1.0 en realidad era unidireccional, de manera que los visitantes tan solo consumían pasivamente la información que otros habían publicado. Los autores necesitaban algún tipo de autorización para utilizar un servidor en el que publicar sus contenidos.

Más tarde, durante la era del Blogging, la web 2.0 modificó de alguna manera este paradigma inicial, de manera que ahora la barrera entre los autores como generadores de contenido y los visitantes como consumidores comenzó a difuminarse. El anonimato nunca fue totalmente real, pues siempre quedaba rastro de las direcciones IP de autor y visitante. Finalmente, los usuarios de Internet comenzaron a ser conscientes de la ausencia de anonimato de la que adolecía el sistema. Las interacciones en las redes sociales o la publicidad dirigida son dos claros ejemplos de ello.

Para muchos usuarios de Internet esta ausencia de anonimato puede ser un peaje que están dispuestos a pagar a cambio de disfrutar del enorme abanico de posibilidades que la Red ofrece. Otros, sin embargo, preferirían tener la certeza de que sus datos no son registrados ni utilizados por las compañías o el gobierno. Algunos de ellos no pueden permitirse una navegación no anónima porque se exponen a represalias en sus países, debido a que algunos estados prohíben la publicación o el acceso a determinados tipos de contenidos, por razones políticas, religiosas o de otra índole.

Evidentemente, los criminales también tienen razones para desear el anonimato en Internet.

Estas razones han influido para desarrollar variantes de la tecnología que sostiene Internet que permitan la publicación de contenido y el acceso al mismo con las mayores garantías de anonimato posibles.

La red TOR es la variante de Internet que más ha avanzado en la búsqueda de esta ansiada característica. Basada en la utilización de nodos especiales en la red y en el enrutamiento por capas, esta tecnología es ampliamente utilizada por usuarios de Internet en todo el mundo.

Muchos de estos usuarios no tienen grandes conocimientos técnicos o residen en lugares donde el acceso a estos conocimientos no es sencillo. Para acceder como visitante y consultar información de manera anónima, en la actualidad no es necesario más que instalar el navegador TOR Browser, de la misma manera que se instala cualquier otra aplicación. Sin embargo, para publicar contenido la situación no es tan sencilla.

Dado el interés de estos usuarios por el anonimato y las dificultades a las que se enfrentan es importantes facilitar la publicación de contenido. Para ello, en el presente trabajo se van a abordar distintas formas de publicar sitios web anónimos.

1.1. Justificación del tema elegido

Como se ha comentado en el punto anterior existe una gran necesidad de herramientas que permitan la publicación de contenidos en Internet de manera anónima.

El presente trabajo pretende recoger las herramientas más ampliamente utilizadas para la publicación de páginas web en la Dark Web de manera anónima.

Una vez recopiladas, se llevará a la práctica la creación y publicación de una página web sencilla con cada una de las herramientas, poniendo especial énfasis en las dificultades técnicas encontradas para utilizar cada una de ellas.

Con esta información, se realizará una comparativa en la que se contratarán las características de cada herramienta, como facilidad de uso y seguridad.

1.2. Objetivos del TFE

Este trabajo de fin de grado se marca como objetivo principal investigar las dificultades técnicas que supone la publicación de información en la Dark Web.

Para lograr este objetivo se construirán tres páginas web simples que se publicarán como servicios Onion utilizando para cada una de ellas una herramienta distinta.

Además, se llevará a cabo un análisis comparativo de las tres herramientas destacando para cada una de ellas sus ventajas e inconvenientes.

2. Marco teórico

El avance imparable de las tecnologías de la información está transformando el mundo tal y como lo conocemos. La mayoría de los seres humanos hacemos uso de Internet para comunicarnos y obtener información a diario. En los últimos diez años, el crecimiento del número de dispositivos conectados que forma parte de Internet de las Cosas ha hecho aumentar todavía más rápidamente el uso de Internet (Hasan, 2022).

El tráfico de datos generado por el conjunto de usuarios de Internet se distribuye entre la Surface Web, la Deep Web y la Dark Web. En los siguientes puntos se explican cada uno de estos conceptos y el funcionamiento de las principales tecnologías utilizadas para navegar a través de la Dark Web.

2.1. Internet profunda e internet oscura

Debido al hecho de que la mayoría del tráfico de Internet se concentra en la Deep Web y en la Dark Web, habitualmente se propone la metáfora que sugiere que Internet es como un iceberg. En la parte superficial del mismo se encuentra la información accesible libremente y que es indexada por los buscadores. En la parte sumergida del iceberg tendríamos la Deep Web y la Dark Web, escondidas bajo la superficie (Hatta, 2020).

En la figura 1 se representa esta visión de Internet. En la superficie, como la parte visible del iceberg, se encontraría la información accesible sin restricciones a cualquier usuario de Internet y que, por tanto, puede ser indexada por los buscadores. En realidad, de las redes sociales solo se encontraría en este nivel los datos que pueden ser leídos sin necesidad de identificarse en la plataforma. Bajo la superficie, se encuentra toda la información accesible a través de Internet solo después de haberse autenticado: datos bancarios, foros privados, bases de datos, etc. En la zona más profunda del iceberg se situaría la Dark Web, únicamente accesible a través de tecnologías apropiadas (TOR, P2P, Freenet) que permiten mantener el anonimato tanto de los autores de la información como de los visitantes.

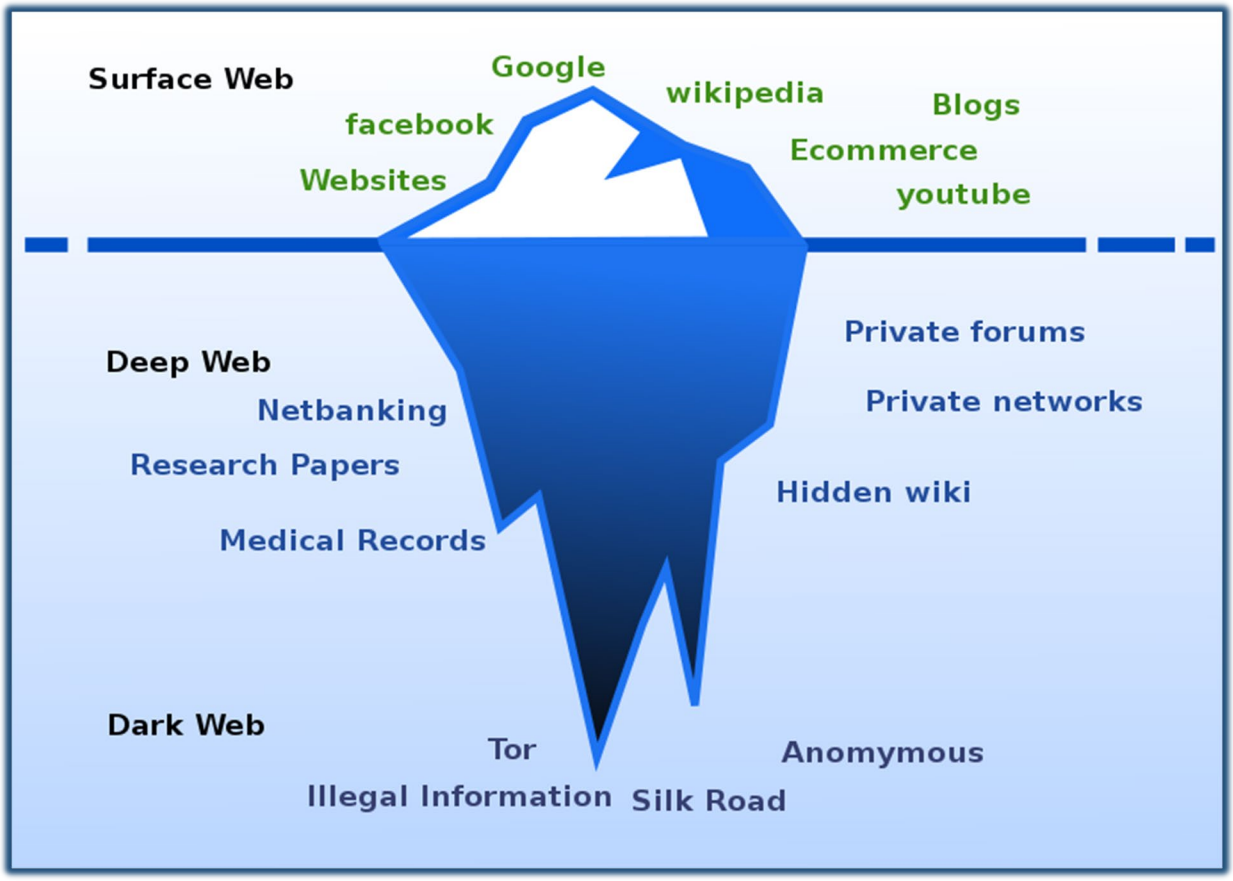

Figura 1: Distribución de la información en internet según la metáfora del iceberg. La Deep Web y la Dark Web son mayores que la Surface web. Fuente: Wikimedia.

2.1.1. Deep Web (Internet profunda)

Para organizar, seleccionar y dar acceso a la mayoría de los usuarios a la ingente cantidad de información que acumula la Web, los buscadores indexan el contenido de las páginas rastreando con sus crawlers las páginas que permiten el acceso libre y que desean ser indexadas (Guía detallada sobre cómo funciona la Búsqueda de Google, s. f.).

Sin embargo, este rastreo no puede llegar a multitud de información que, principalmente, no es de libre acceso. De esta manera, las páginas que cambian su contenido según la IP del visitante o si este se ha identificado, las páginas dinámicas generadas como respuesta a la información que introduce el usuario en un formulario o con acceso restringido, el contenido que no es texto y no puede ser analizado, las páginas a las que no apunta ningún enlace en otras páginas, el contenido de las redes P2P y, por supuesto, la información escondida en la Dark Web no pueden ser indexada por los buscadores.

Por tanto, para acceder a esta parte de la web se debe contar con esas credenciales de acceso, conocer la dirección URL del servicio porque previamente ha sido compartida por otros medios, formar parte de las redes P2P o, en el caso de Dark Web, utilizar las tecnologías y herramientas necesarias para navegar por la misma.

2.1.2. Dark Web (Internet oscura)

Los usuarios de Internet pueden tener distintas razones para aspirar a poder realizar una navegación anónima.

Tanto los proveedores de servicios que permiten a los usuarios finales el acceso a Internet, como los responsables de todos los nodos intermedios que canalizan el tráfico y, por supuesto, las autorizades, pueden analizar el tráfico de la red y recabar información sobre nuestros intereses y actividades en la red. Por otro lado, los sitios web que visitamos tienen un gran interés en conocer la mayor cantidad posible de detalles sobre nuestra identidad y necesidades para ofrecernos publicidad que sea de nuestro interés o, tal vez, para vender la información recabada.

También es cierto que, en gran cantidad de países donde la libertad de expresión e información está restringida, el acceso a ciertos contenidos está directamente prohibido. Por ejemplo, China bloque el acceso a multitud de sitios web, como redes sociales, plataformas de blogging, de correo electrónico, buscadores, periódicos o plataformas de streaming (Perunicic, s. f.). No solo está prohibido el acceso, también es habitual que los gobiernos espíen el tráfico generado en Internet en busca de evidencias de comportamientos ilegales en esos países. La persecución por motivos políticos, religiosos o de cualquier otra índole es una práctica común.

Sin embargo, también ocurre que muchos delincuentes buscan el anonimato para realizar actividades manifiestamente ilegales en la mayoría de los países: compraventa de armas y drogas, pornografía infantil, etc. (Bertram, 2015).

Todas estas razones han llevado a muchos usuarios de Internet a aspirar a poder realizar una navegación totalmente anónima, bien sea buscando su derecho a la privacidad o bien tratando de escapar del acecho de las autoridades.

Los aspectos que se deben camuflar para conseguir la navegación anónima son principalmente dos: la dirección IP del usuario y el contenido del tráfico que genera. Algunas

tecnologías proporcionan exclusivamente enmascaramiento de IP y otras añaden, además, la posibilidad de ocultar el tráfico del usuario.

2.1.2.1. Accediendo al contenido de manera anónima

A continuación, se presentan cuatro tecnologías que persiguen mantener el anonimato de los usuarios (Winkler & Zeadally, 2015):

Los **reenviadores de mensajes** (Anonymous Remailers) permiten ocultar el remitente de un correo electrónico mediante la utilización de una red, que cuente con al menos un nodo intermedio, que mezcla todos los mensajes que se desean enviar en un momento dado. El mensaje no llega directamente al remitente, sino que da varios saltos. En este proceso, se elimina la información del remitente y los mensajes son homogeneizados para evitar ataques de análisis de tráfico. Los reenviadores de tipo III proporcionan un anonimato satisfactorio.

En la figura 2 se contrastan dos aproximaciones. En la primera aproximación el reenviador utiliza el protocolo SMTP para enviar los mensajes, con lo cual un observador externo puede saber que el tráfico contiene correos electrónicos. En la segunda aproximación, los mensajes se transmiten utilizando un protocolo binario propio, con lo cual un observador del tráfico no puede saber que se están enviando mensajes. Tan solo el último nodo de la red es un servidor de correo estándar.

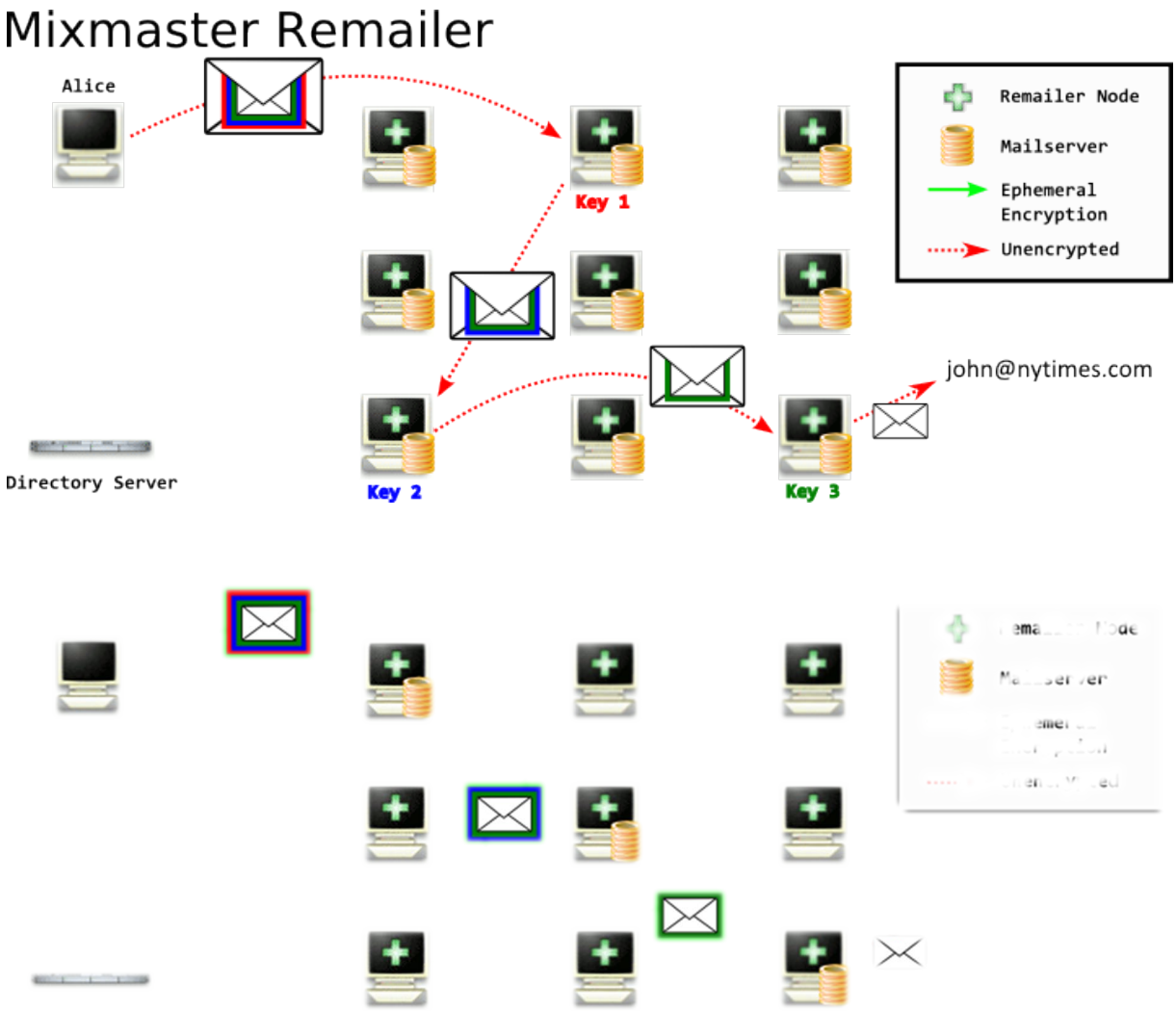

Figura 2: Dos aproximaciones diferentes a los reenviadores de correo electrónico. Fuente: Ritter.vg

Los **rewebbers** permiten la navegación a través de la web de manera anónima. También utilizan uno o varios proxys http, que eliminan los detalles del tráfico que pudieran permitir la identificación del usuario. El anonimato proporcionado por los rewebbers es menos robusto. En la figura 3 se muestra una metáfora del funcionamiento de la cadena de servidores.

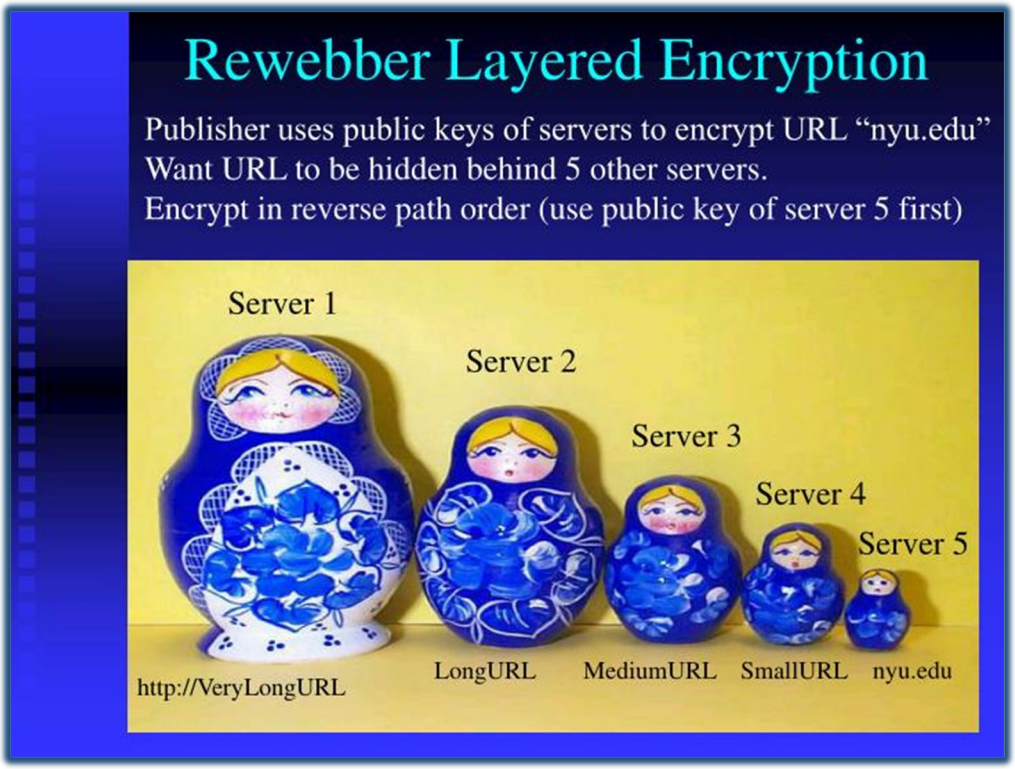

Figura 3: La URL solicitada se enmascara progresivamente a través de la red de servidores.

Fuente: Marc Waldman, New York University.

El **enrutamiento cebolla** (la red TOR) canaliza el tráfico generado por el usuario a través de una red anónima de servidores que, voluntariamente, sostienen esta tecnología. El anonimato se consigue gracias a que cada uno de los nodos que reenvía la información únicamente conoce la identidad del nodo que le remitió el paquete y la identidad del nodo al que debe remitirlo. La confidencialidad de la información y de la identidad del resto de nodos se asegura mediante varios cifrados sucesivos con criptografía asimétrica. Esta tecnología destaca por su disponibilidad y facilidad de uso.

En la figura 4 se aprecia cómo el tráfico que genera el cliente (Alice) es canalizado a través de una red de nodos TOR que transfieren la información encriptada y que solo conocen la identidad de los nodos adyacente. Solamente es pública la conexión entre el último nodo del circuito TOR y el servidor que se desea visitar (Bob).

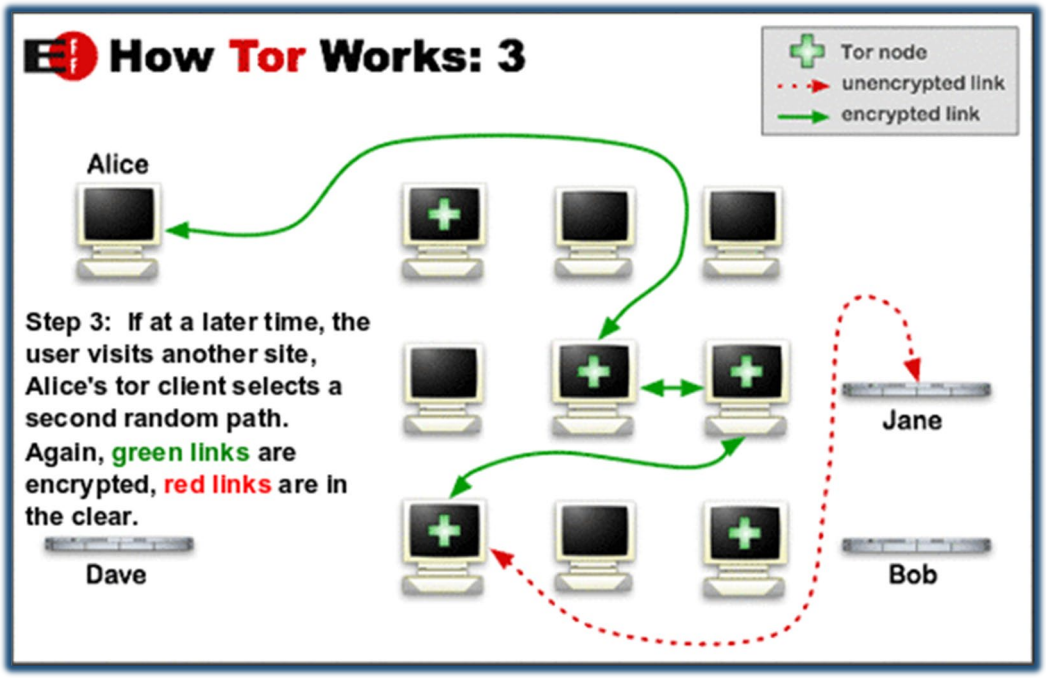

Figura 4: Circuito TOR para proporcionar anonimato. Fuente: The Tor Project, Inc.

La tecnología **I2P** pretende ser una alternativa a TOR, con la que comparte filosofía. Pare ello proporciona una serie de mejoras en ciertos aspectos. I2P es más distribuido y auto organizado. Los túneles tienen una vida corta, reduciendo la exposición a ciertos ataques. Es más eficiente en la creación de circuitos. Sin embargo, su uso está menos extendido. (Comparación de I2P con Tor - I2P, s. f.).

2.1.2.2. Creación de contenido de manera anónima

Las mismas razones por las que los visitantes de Internet pueden desear mantener el anonimato se pueden aplicar a los creadores de contenidos. La vigilancia y persecución por parte de las autoridades sobre los responsables de la publicación de contenidos y la censura ejercida sobre los mismos son habituales en muchos países.

Por ello, los usuarios que deseen publicar contenido sujeto a este tipo de persecución necesitan disponer de una tecnología que les permita anonimizar el mismo. El objetivo es impedir que se puedan hallar evidencias que conecten el contenido con el responsable de su publicación.

Una primera aproximación sería la que engloba al contenido situado en la Deep Web, escondiendo el acceso a la información de la vista de los buscadores. Proteger el acceso al contenido por medio de contraseñas o filtros IP no es garantía de anonimato suficiente. Las contraseñas se pueden filtrar o romper y los filtros IP se pueden saltar. Una vez localizado el contenido en un servidor, conectarlo con el responsable es técnicamente sencillo.

Siendo la dirección IP la base tecnológica para la identificación mutua entre los equipos que forman parte de Internet, la publicación anónima debe cumplir con los siguientes requisitos mínimos:

1. Que se pueda acceder a la información publicada desde cualquier parte de la red
2. Que sea posible el acceso a la información publicada sin necesidad de conocer la dirección IP del autor/servidor

La red TOR se creó con el objetivo de construir una herramienta, disponible para todos los usuarios de Internet, que cumpla con los dos requisitos anteriores.

2.2. La red TOR (The onion router)

TOR es una red que funciona por encima de internet y que permite navegar de manera anónima a través de la web.

Aplicado a la navegación a través de la Surface Web y de la Deep Web, TOR proporciona anonimato a los lectores, que pueden acceder a la información sin que los sitios web ni cualquier nodo intermedio puedan conocer su identidad. De esta manera, tampoco las autoridades pueden conocer la actividad de los ciudadanos. Utilizado para acceder a la Dark Web, con los llamados servicios Onion, TOR permite además la publicación de contenido evadiendo la censura, sin que sea posible en principio conectar el contenido de los sitios web con los autores.

Desde que comenzaron las primeras propuestas para el desarrollo de un sistema que permitiese ocultar la información de enrutamiento (Goldschlag et al., 1996) hasta la actualidad, la promoción de la navegación anónima y la lucha contra la censura ha pasado por distintas fases. En todo momento se han mantenido los valores del software libre y de código abierto.

En 2004 la Electronic Frontier Foundation comienza a financiar el proyecto que desarrollaba la herramienta y, en 2006, se funda el Tor Project, inc., la organización sin ánimo de lucro que promueve el desarrollo de Tor.

El desarrollo del navegador TOR Browser popularizó el uso de la red TOR, formando parte de acontecimientos históricos recientes como la primavera árabe o la filtración de documentos clasificados protagonizada por Edward Snowden (MacAskill et al., 2013).

Actualmente, la red TOR es una forma sencilla y fiable de navegar a través de Internet de forma privada (El Proyecto Tor, s. f.).

En la figura 5 puede verse un mapa en el que se muestra el uso de la red TOR en los distintos países del mundo medido en número de usuarios diarios de TOR por cada 100.000 usuarios estándar diarios de Internet. Destaca el uso en Francia, España, Italia y Alemania, superior al de Estados Unidos y Rusia.

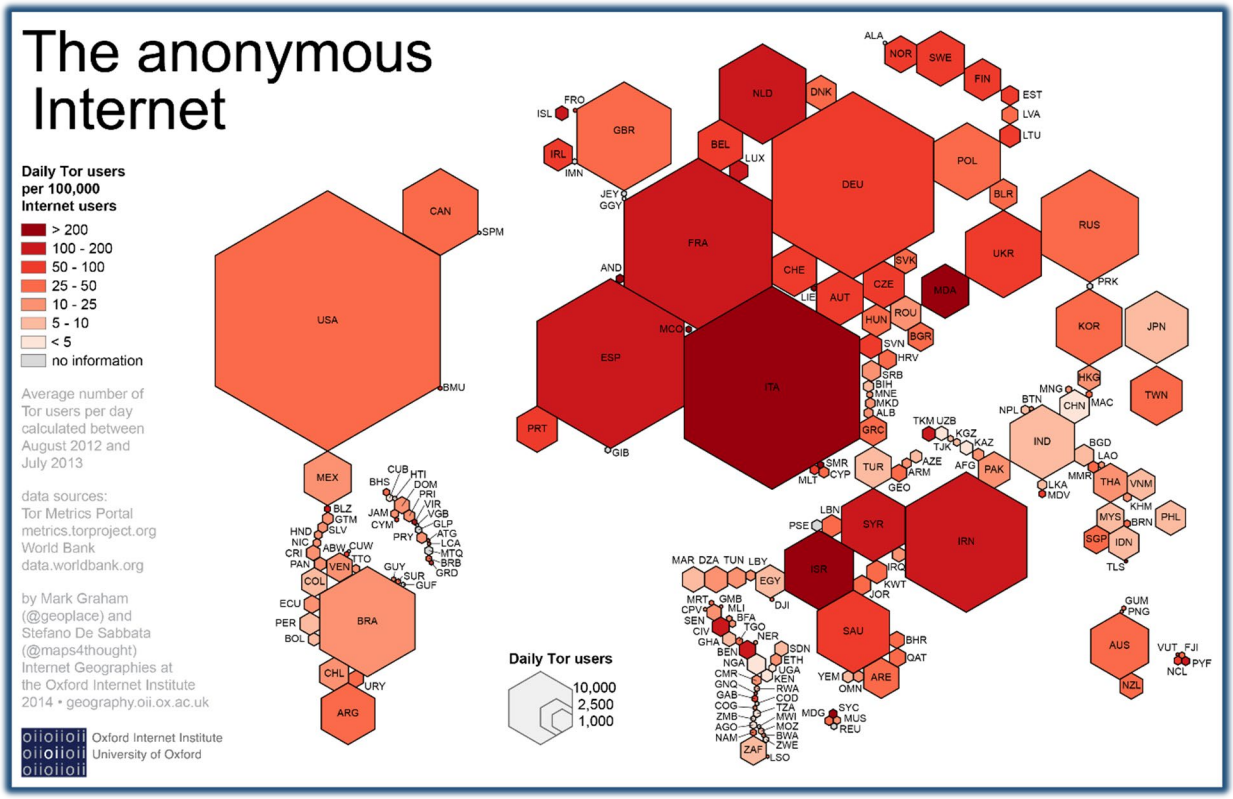

Figura 5: Uso de TOR en distintos países. Fuente: University of Oxford.

2.2.1. Una alternativa a la red clásica

Para comprender el funcionamiento de la red TOR, es conveniente entender previamente la manera en que funciona la Internet que utilizamos habitualmente.

Cuando un usuario desea acceder a una página web tan solo debe indicar en un navegador web la dirección URL donde se aloja la información que desea consultar. El equipo que utiliza el usuario en el que corre el navegador se conoce como cliente. La computadora en la que se encuentra la información que se desea consultar se conoce como servidor. Al más alto nivel, el cliente solicita un archivo al servidor y éste devuelve una copia del archivo al cliente, que lo visualiza.

Para concretar un poco más este funcionamiento debemos disponer, además de nuestro cliente y nuestro servidor, de:

- Un canal de comunicación, que es la conexión de red entre ambos equipos.
- Unos lenguajes comunes, que son:
    - El protocolo TCP/IP, que permite la localización de los equipos y el establecimiento de comunicación entre ellos.
    - El protocolo HTTP, que proporciona la capacidad de negociar las peticiones y envíos de archivos.
- El sistema DNS, que permite localizar a los equipos a lo largo de la inmensa red.

A continuación, se muestran los pasos que se suceden cuando un usuario desea visitar el contenido de una página web (Cómo funciona la web, s. f.):

En primer lugar, el usuario introduce en su navegador la dirección de la página web que desea visitar (URL). El usuario puede conocer la dirección de antemano u obtenerla en el momento. Esta dirección puede estar almacenada en el cliente: el usuario puede utilizar el navegador u otro archivo como agenda de direcciones (bookmarks) o puede obtenerse de una búsqueda realizada en un buscador u otra página web (hiperenlaces).

En cualquier caso, es necesario averiguar la localización real del servidor en el que se encuentra la información que se desea consultar. Los equipos conectados a Internet están identificados mediante su dirección IP. Así, el navegador consulta al servidor DNS, que almacena en una base de datos jerarquizada las correspondencias entre los nombres de

dominio (que son la parte principal de las direcciones URL) y las direcciones IP. El servidor DNS responde con la dirección IP correspondiente.

En este momento el cliente realiza la petición al servidor, remitiéndola a su dirección IP. El servidor responde con el archivo solicitado remitiendo la información a la dirección IP del cliente.

La información viaja dividida en paquetes que son reagrupados al llegar al receptor. Los paquetes pueden atravesar multitud de nodos de la red, todos los cuales leen la dirección IP de destino para reenviar el paquete. Este enrutamiento puede ser distinto para cada paquete.

En este esquema de funcionamiento básico de la Internet estándar podemos destacar algunas vulnerabilidades en las que se pone en riesgo el anonimato, la confidencialidad, la integridad y la disponibilidad de la comunicación:

1. En la consulta al servidor DNS, éste puede recopilar información sobre los servidores que un cliente determinado desea visitar. Esta debilidad es inherente al diseño básico del sistema.
2. Todos los nodos que van reenviando los paquetes entre el remitente y el destino conocen ambas identidades, por lo que también pueden realizar la recopilación de esta información.
3. Sabiendo la dirección IP, es posible localizar geográficamente los equipos.
4. Si la información no viaja cifrada, todos los nodos por los que pasa pueden escuchar la comunicación entre las partes.
5. Si se puede acceder al contenido de la información, se puede manipular.

Estas debilidades en cuanto anonimato y privacidad son las que trata de superar la red TOR.

2.2.2. Funcionamiento de la red TOR

Para hacer frente a las vulnerabilidades anteriores, la red TOR intercala entre los nodos de la Internet clásica al menos tres nodos que pertenecen a la propia red TOR. El cliente no se comunica directamente con el servidor, sino que planifica las comunicaciones de manera que se asegura de que la información pasa por estos nodos. Al primer nodo solo le informa de la identidad del segundo nodo. El segundo nodo solo conoce la identidad del primer y último nodo. El último nodo únicamente sabe la identidad del segundo nodo y del servidor que se desea consultar.

De esta manera, solo el cliente conoce los detalles de la ruta que seguirá la información y ninguno de los nodos, ni por supuesto el servidor, pueden determinar quién está solicitando la información.

Además, tanto la información como las cabeceras viajan cifradas en varias capas sucesivas, lo que impide el análisis de cabeceras y el espionaje de la información (The Tor Project, Inc, s. f.).

Para poder intercambiar información mediante esta tecnología, TOR utiliza varias técnicas que se describen a continuación: el enrutamiento TOR y el cifrado en capas.

2.2.2.1. Enrutamiento TOR

Cualquier usuario que se conecta a la red TOR puede permitir que su equipo ejerza altruistamente como nodo de la red. Así, otros usuarios pueden utilizar su equipo como parte de la ruta hacia el servidor que desean visitar.

Existen cuatro tipos de nodos en la red TOR. Cada uno de ellos tiene distintas funciones, especificaciones e implicaciones legales (Tor Project, s. f.).

Los nodos de **entrada**, también conocidos como nodos guardianes (Guard Relays), permiten la entrada a la red TOR. No requieren configuraciones especiales, únicamente se les solicita cierta estabilidad y ancho de banda.

Los nodos **intermedios** pueden ser utilizados para trazar una ruta entre un nodo de entrada y un nodo de salida. Solo se comunican con otros nodos, no envían ni reciben tráfico desde el exterior. Tampoco requieren especial cuidado en la configuración ni gran ancho de banda.

Tanto los nodos de entrada como los nodos intermedios no suelen recibir quejas por abuso.

Los nodos de **salida** son los que, finalmente, envían las solicitudes hacia el servidor destino y retornan la respuesta. Expuestos a ser utilizados para cometer ilegalidades, habitualmente se encuentran alojados en universidades u otras instituciones que promuevan la privacidad. Estas organizaciones tienen mayor capacidad para lidiar con las quejas que puedan recibir los administradores de nodos de salida.

Los nodos **puente** son publicados de manera limitada en la lista pública de nodos de la red TOR, de manera que proporcionan servicio a usuarios que se encuentren en zonas donde se impide el acceso a la red TOR mediante el bloqueo de las IP de los nodos TOR públicos. Los

requisitos de ancho de banda para un nodo puente son mínimos y las implicaciones legales habitualmente inexistentes.

En la figura 6 se pueden distinguir los tres nodos básicos de un circuito TOR: el nodo de entrada (el más cercano al cliente Alice), el nodo intermedio y el nodo de salida (el más cercano al cliente Bob). Fuente: The Tor Project, inc.

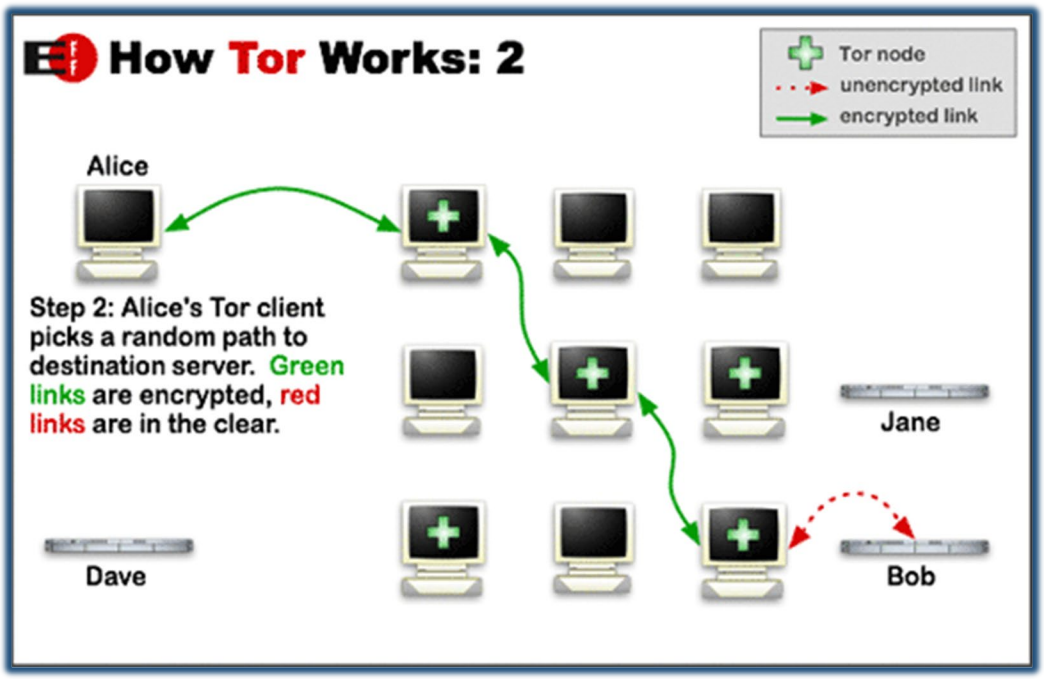

Figura 6: Circuito TOR constituido por un nodo de entrada, un nodo central y un nodo de salida. Fuente: The Tor Project, Inc.

Cuando un usuario desea utilizar la red TOR en primer lugar debe seleccionar un circuito a través del cual realizar la conexión. Un circuito está constituido por la concatenación de un nodo de entrada, un nodo intermedio y un nodo de salida. A veces se puede utilizar un nodo puente.

Para obtener los nodos disponibles se consulta la lista pública de repetidores de TOR, en la que se muestran todos los nodos de entrada, intermedios y de salida. Para construir el circuito se negocian una serie de parámetros y se validan los nodos elegidos mediante distintos criterios, para evitar la introducción de nodos maliciosos (path-spec.txt - torspec - Tor's protocol specifications, s. f.).

2.2.2.2. Cifrado en capas

Para que el sistema funcione correctamente el cliente debe asegurarse de que cada nodo de la red puede acceder a la información que debe conocer y no a la que es dirigida al resto de nodos o al servidor.

Supongamos que los nodos utilizados en un circuito TOR generado con el sistema de enrutamiento anteriormente descrito son tres, que es el número mínimo exigido en el diseño. El nodo de entrada, al que llamaremos A, el nodo intermedio, al que llamaremos B y el nodo de salida, al que llamaremos C.

El cliente obtiene de la lista pública de nodos disponibles las credenciales de cada uno de ellos, llegando a obtener, entre otra información, la clave pública para el cifrado asimétrico correspondiente a cada nodo. Con estas claves, el cliente va cifrando el paquete de información correspondiente a cada nodo con su clave pública.

- Para el nodo C se cifra la capa más interna con la clave pública de C el paquete que contiene el mensaje (que incluye la petición y la dirección del servidor que se desea consultar).
- Para el nodo B se cifra la capa intermedia con la clave pública de B el paquete anteriormente cifrado, indescifrable para B, y la dirección de C.
- Para el nodo A se cifra la capa externa con la clave pública de A el paquete anteriormente cifrado, indescifrable para A, y la dirección de B.

El cliente envía el paquete triplemente cifrado al nodo A.

El nodo A utiliza su clave privada A para descifrar la primera capa más externa del paquete y obtener la parte del paquete que debe remitir a B y la dirección de B.

El nodo A envía el paquete doblemente cifrado al nodo B.

El nodo B utiliza su clave privada B para descifrar la segunda capa intermedia del paquete y obtener la parte del paquete que debe remitir a C y la dirección de C.

El nodo B envía el paquete simplemente cifrado al nodo C.

El nodo C utiliza su clave privada C para descifrar la última capa del paquete y obtener la parte del paquete que debe remitir al servidor que el cliente desea consulta (la petición) y la dirección del servidor.

El nodo C envía la petición al servidor y remite la respuesta devuelta por el servidor al cliente a través del circuito TOR seleccionado.

En la figura 7 puede verse un diagrama en el que se esquematiza la estructura del paquete cifrado en tres capas. Cada una de las capas envuelve sucesivamente la información interior, de manera que nadie puede descifrar completamente el paquete, excepto el nodo de salida. Por ello, el nodo de salida no conoce la identidad del cliente.

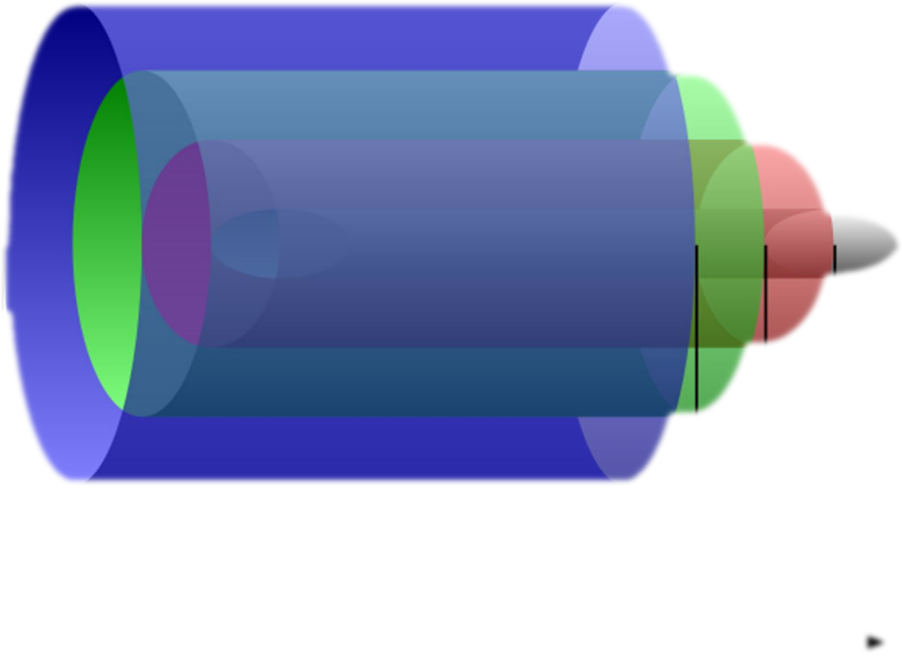

Figura 7: Paquete cifrado en capas, origen del nombre de Onion Router. Fuente: Wikimedia

De la misma manera que sucede con el enrutamiento y otras funcionalidades de la red TOR, el sistema de cifrado utilizado es un campo de estudio en constante evolución y en el que se sigue trabajando (Catalano et al., 2017).

La red TOR utilizada de esta manera permite al cliente acceder a servidores sin que estos servidores conozcan quien es el verdadero cliente.

En la figura 8 se muestra la página principal de Google visitada utilizando en navegador TOR Browser. Como el circuito TOR generado en este caso utiliza un nodo de salida situado en Suiza, el servidor devuelve la versión de la página escrita en alemán. El servidor entiende que

la petición ha sido realizada desde Suiza y, en principio, no puede saber que en realidad la petición proviene de España.

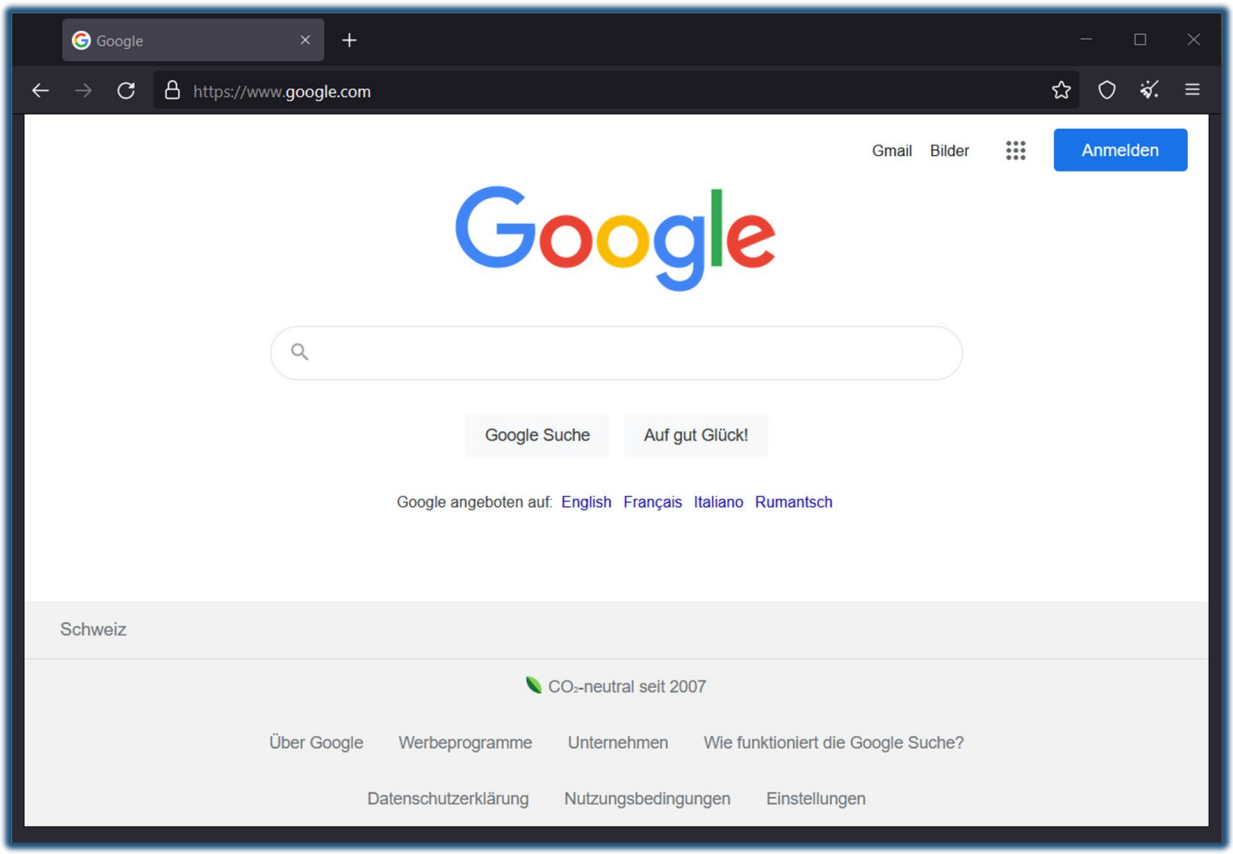

Figura 8: Página principal de Google visitada utilizando el navegador TOR Browser. Fuente: autor.

2.3. Servicios Onion

En el punto anterior se ha descrito cómo la tecnología utilizada en la red TOR ayuda a los usuarios de Internet a mantener la privacidad y el anonimato cuando navegan por la red, accediendo a servidores de la Surface Web sin que estos puedan conocer ni el origen ni la identidad de los usuarios.

Sin embargo, como se ha explicado previamente, en el otro lado, los autores de contenidos también tienen poderosas razones para utilizar tecnologías que les proporcionen privacidad y anonimato.

En esta situación es donde entran en juego los servicios Onion, antes conocidos como servicios ocultos o Hidden Services (Johnson, 2015).

Los servicios Onion proporcionan privacidad y anonimato porque cumplen con el requisito de ocultación de la IP del servidor y, por tanto, de su localización, dificultando los posibles intentos de ataque.

Además, los servicios Onion cifran la comunicación entre cliente y servidor de extremo a extremo.

Por último, el propio diseño de las direcciones .onion que son utilizadas por los servicios Onion para identificarse, evita la necesidad de un servidor tipo DNS y los ataques tipo man in the middle (SERVICIOS CEBOLLA, s. f.).

En la figura 9 se muestra una página web montada en un servicio Onion visitada con el navegador TOR Browser. Destaca la dirección .onion V3 escrita en la barra de direcciones, con sus 56 caracteres.

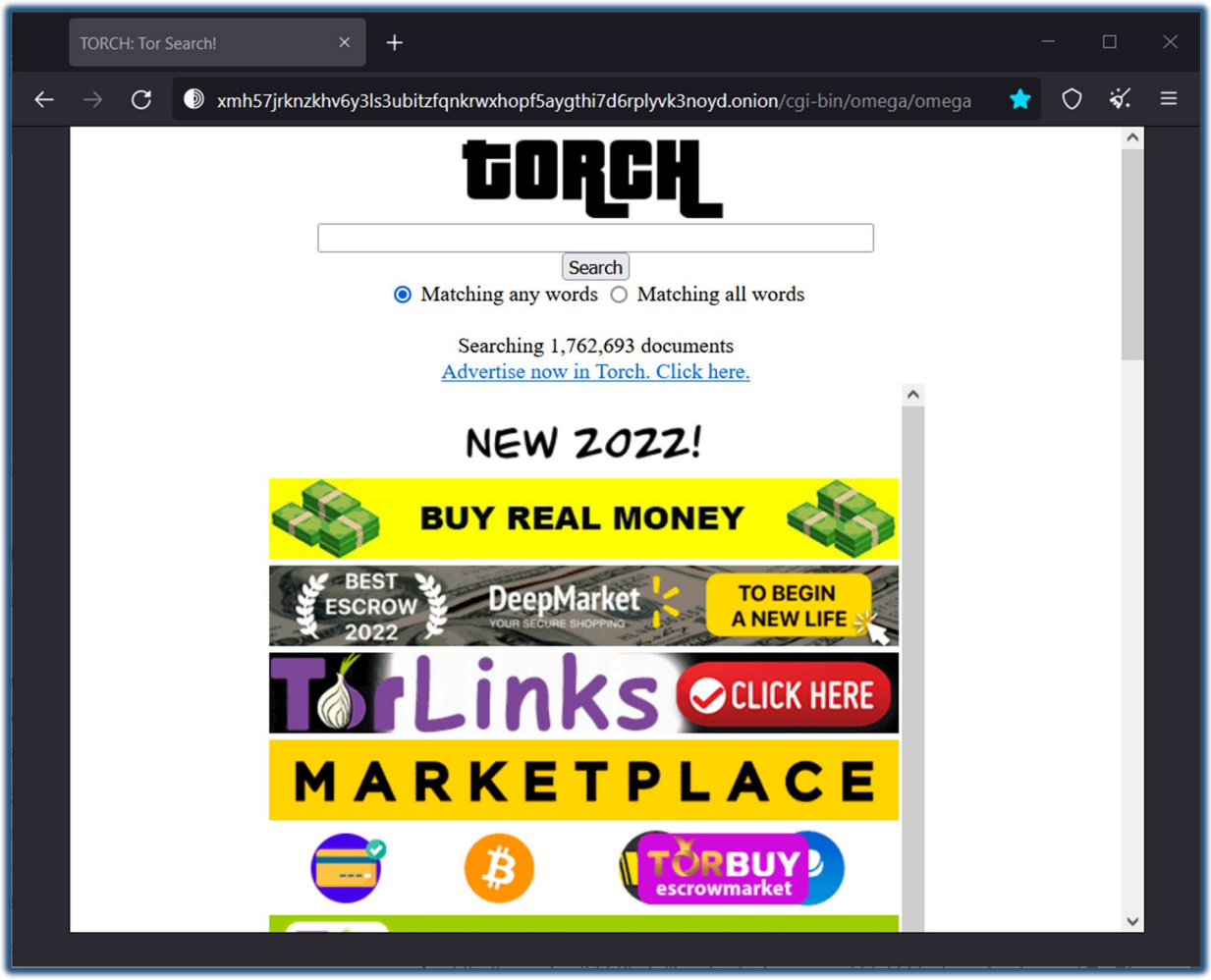

Figura 9: Página web publicada en un servicio Onion visitada utilizando el navegador TOR Browser. Fuente: autor.

2.3.1. Funcionamiento de los servicios Onion

Para establecer la comunicación entre el cliente y el servicio Onion se necesita la participación de varios nodos pertenecientes a la red TOR que realizarán funciones de asistencia: un nodo punto de introducción (introduction point), un nodo de cita (rendezvous point) y la base de datos distribuida de servicios Onion (que sustituye de alguna manera al servicio DNS).

A continuación, se describen los pasos que se deben dar para establecer la comunicación con un servicio Onion (The Tor Project, Inc, s. f.-a):

En primer lugar, el servicio Onion inicializa distintos puntos de entrada. Para ello selecciona algunos nodos para que funcionen como puntos de introducción. El servicio Onion les comunica su clave pública a través de un circuito de la red TOR. De esta manera, el punto de introducción no conoce la verdadera dirección IP del servicio Onion, únicamente su clave pública. Figura 10.

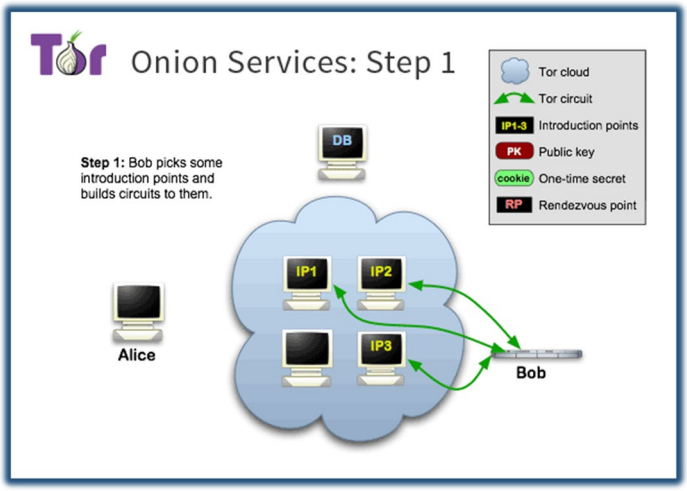

Figura 10: El servicio Onion selecciona los puntos de entrada. Fuente: The Tor Project, Inc.

En segundo lugar, el servicio Onion publica en la base de datos distribuida su propio descriptor. El descriptor contiene la clave pública del servicio Onion y la lista de puntos de entrada que pueden usarse para establecer la comunicación con el servicio. El descriptor se firma con la clave privada del servicio. Figura 11.

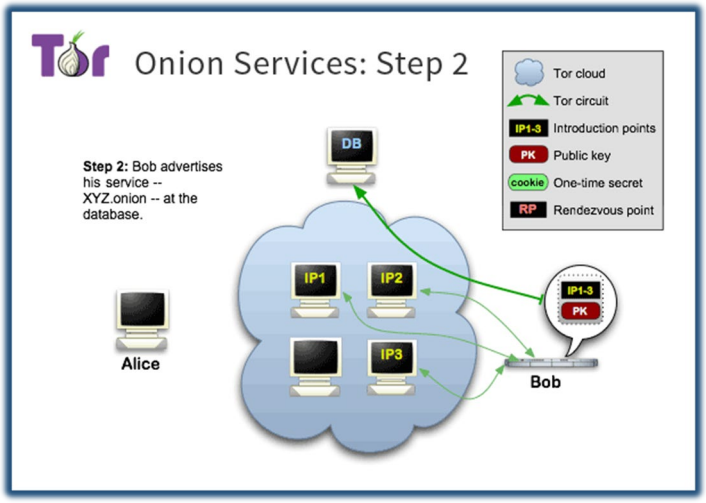

Figura 11: El servicio Onion publica su descriptor y los puntos de entrada. Fuente: The Tor Project, Inc.

A continuación, cuando un cliente desea conectar con un servicio Onion, comienza obteniendo el descriptor utilizando para ello la dirección .onion que el cliente debe conocer. La dirección .onion del servicio se deriva de la clave pública del mismo, de manera que se puede comprobar la relación entre una dirección .onion concreta y su correspondiente clave pública. El cliente ahora conoce los posibles puntos de entrada y la clave pública del servicio Onion. En la figura 12 se muestra la consulta del cliente a la base de datos de servicios Onion.

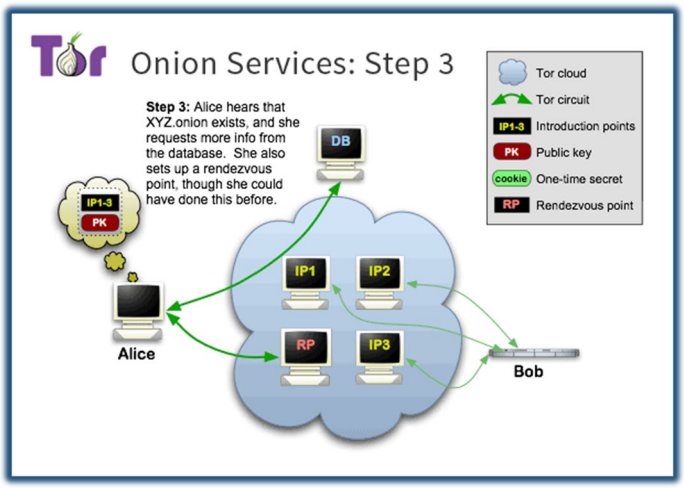

Figura 12: El cliente consulta descriptor y puntos de entrada del servicio Onion en la base de datos. Fuente: The Tor Project, Inc.

Por otro lado, el cliente crea un circuito hasta otro nodo que actuará como punto de cita.

Para negociar el inicio de la conexión, el cliente prepara un mensaje para el servidor. Este mensaje contiene la dirección del punto de cita y una clave de un solo uso. El mensaje se cifra con la clave pública del servicio Onion. El cliente envía el mensaje al servicio a través de uno de los puntos de introducción. El cliente permanece anónimo porque utiliza un circuito TOR para conectar con los puntos de introducción. En la figura 13 se muestra el uso de los puntos de introducción.

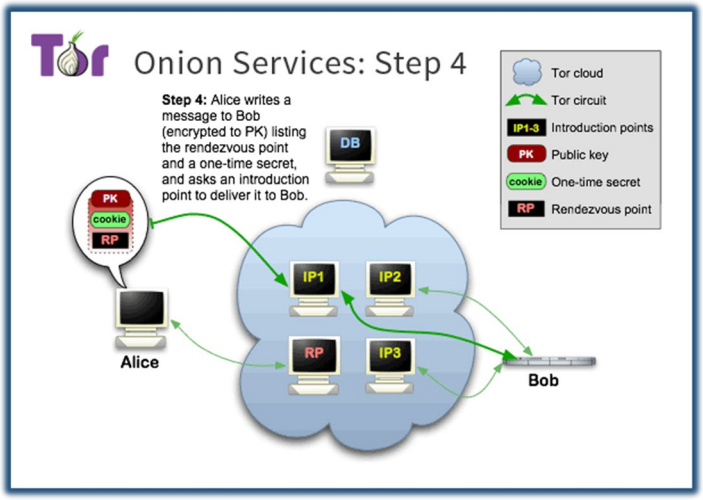

Figura 13: El cliente se pone en contacto con el servidor a través de uno de los puntos de entrada. Fuente: The Tor Project, Inc.

El servicio Onion recibe el mensaje de solicitud de conexión de parte del punto de introducción. Al descifrar el mensaje con su clave privada, el servicio ahora conoce la dirección del punto de cita y la clave de un solo uso. Ahora, el servicio remite al punto de cita la clave de un solo uso. En la figura 14 se muestra el uso del nodo punto de cita.

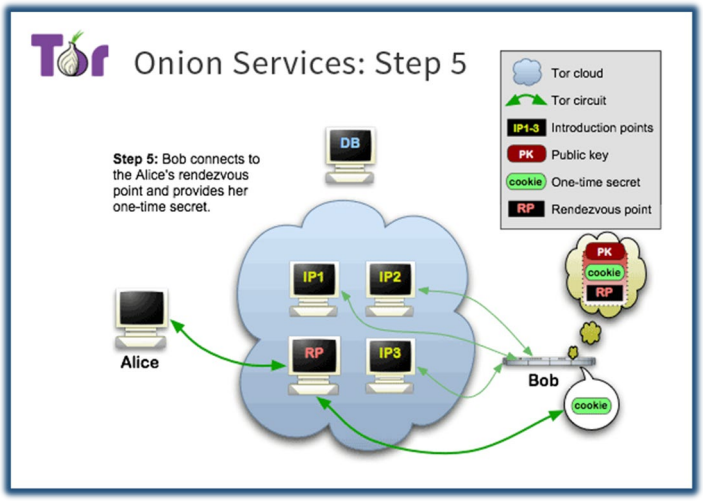

Figura 14: El servicio Onion confirma el establecimiento de la conexión a través del punto de cita. Fuente: The Tor Project, Inc.

Por último, en la figura 15 se representa cómo el punto de cita confirma al cliente que la conexión ha sido creada.

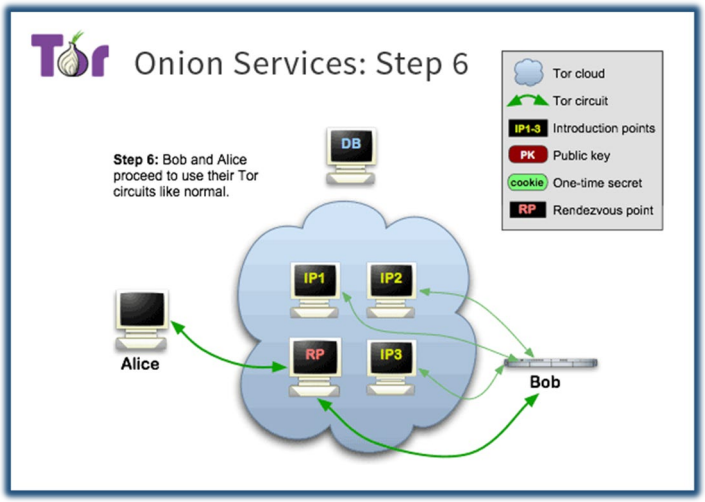

Figura 15: El punto de cita confirma al cliente la conexión con el servidor. Fuente: The Tor Project, Inc.

Para comunicarse, ahora tanto el cliente como el servicio Onion utilizan sus propios circuitos independientes para transferir la información al punto de cita, que se encarga de remitirla a cada uno de ellos.

De esta manera, cliente y servicio Onion pueden establecer una comunicación sin que ninguno de ellos conozca la IP, ni por tanto la localización, del otro.

Así, utilizando este protocolo, se pueden evitar las vulnerabilidades propias de la navegación clásica a través de Internet, analizadas en el punto 2.2.1.

- El servidor DNS es sustituido por una base de datos distribuida que no conoce, gracias al uso de circuitos TOR, quien la consulta.
- Los nodos que transmiten la información no conocen a la vez las identidades del cliente y del servicio Onion ni, por tanto, los circuitos utilizados para la comunicación.
- La información viaja en todo momento cifrada, por lo que no se puede conocer el contenido de los mensajes.

En la figura 16, se muestran el cliente, Alice, el servicio Onion, Bob, la base de datos con los descriptores, DB, los puntos de entrada seleccionados por el servicio Onion y el punto de cita seleccionado por el cliente, RP. Es necesaria la participación de al menos seis nodos de la red TOR.

Las comunicaciones entre todos estos nodos especiales se realizan siempre a través de circuitos TOR completos, tal y como se describe en el punto 2.2.2.1 Enrutamiento TOR.

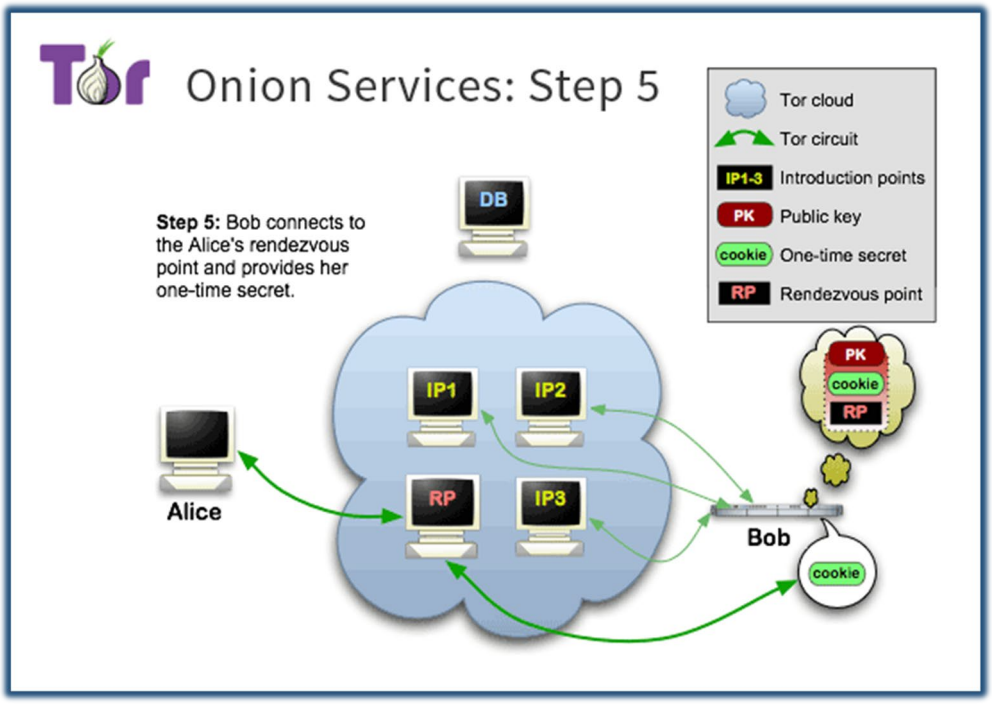

Figura 16: Doble circuito TOR necesario para garantizar el anonimato del cliente y del servidor. Fuente: The Tor Project, inc.

2.3.2. Razones y ventajas de las direcciones .onion

El aspecto de las direcciones .onion es por diseño difícil de recordar y poco amigable. Esta apariencia no es casual sino consecuencia de la manera en la que se construyen estas direcciones y las funciones que deben cumplir.

Evidentemente, una dirección .onion debe permitir establecer la comunicación con el servicio Onion que se desea utilizar.

Además, la dirección .onion permite en todo momento confirmar que se está comunicando con el servicio Onion deseado y que nadie se está interponiendo en la conexión.

Para construir la dirección .onion es necesario disponer de la clave pública del servicio Onion. Como se ha visto, estas claves están disponibles en los descriptores que los servicios Onion publican en la base de datos pública de servicios Onion, de manera que, además del propio servicio Onion, cualquiera puede generar la dirección .onion correspondiente al servicio si dispone de la clave pública del mismo. Esta relación entre la clave pública y la dirección .onion permite comprobar en todo momento que se está comunicando con el servicio correcto.

Para generar la dirección, se aplica una función hash a la clave pública. Se toma una parte del resultado y se codifica en Base 3. La dirección obtenida tiene siempre 56 caracteres en la versión 3, utilizada actualmente (Castillo, 2015).

Al activar un servicio Onion, se genera un par de claves privada y pública. Por tanto, es posible personalizar en cierta manera los caracteres de la dirección .onion. Sin embargo, es computacionalmente costoso, pues de deben ir generando pares de claves pública y privada y comprobando la dirección resultante hasta dar con una que coincida con el patrón deseado (Scaife, s. f.).

Cuando el proyecto TOR se inició, el protocolo pronto estableció las direcciones .onion versión 2 como estándar. Estas direcciones tenían 16 caracteres. En el año 2020 ya era conocido por los responsables del proyecto que el cifrado utilizado para generar las direcciones v2 había sido superado por el avance de las técnicas criptográficas y que exponía a los servicios Onion a ataques de bloqueo, denegación del servicio y enumeración o localización (Goulet, 2020).

Por tanto, se desarrolló la versión 3 del protocolo. El cambio más notable es que la propia dirección v3 es una clave de cifrado de manera que puede ser utilizada para cifrar y/o firmar

los mensajes intercambiados. De esta manera, la base de datos distribuida que contiene los descriptores de todos los servicios Onion almacena la información cifrada, que puede ser descifrada por el cliente con la propia clave contenida en la dirección. Para evitar la recolección automática de direcciones v3, sus descriptores se cifran con una determinada combinación de la clave pública contenida en la dirección y la fecha actual (V3 onion services usage, s. f.).

Entre septiembre de 2020 y octubre de 2021 se produjo la transición de las direcciones v2 a las direcciones v3. Finalmente, las direcciones v2 fueron desactivadas y actualmente ya no pueden ser utilizadas, ni para crear nuevos servicios Onion ni para visitar antiguos (Onion Service version 2 deprecation timeline, s. f.).

En la figura 17 se puede comparar la cantidad de direcciones v2 y v3, viendo cómo ha evolucionado su número. Destaca el aumento explosivo del número de direcciones v3 a partir de septiembre de 2021 y la desaparición paulatina de las direcciones v2.

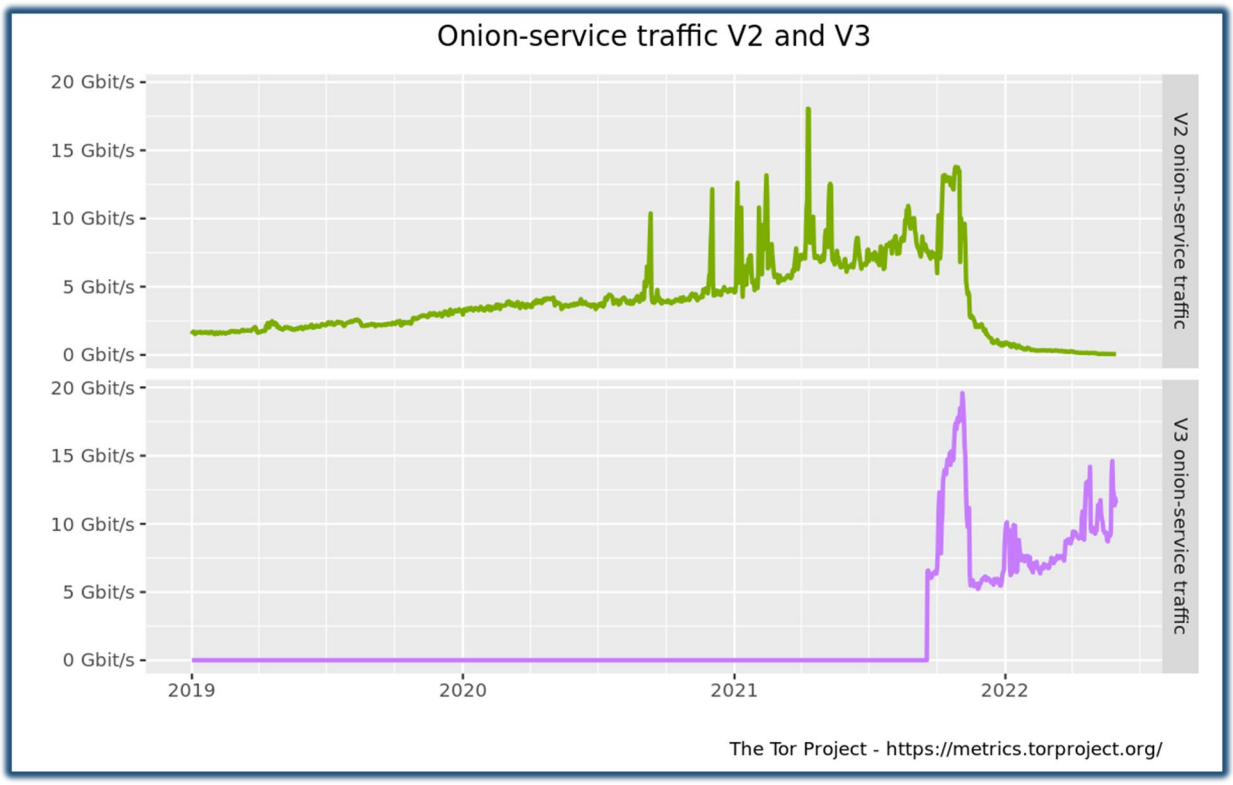

Figura 17: Comparativa entre la evolución del número de direcciones onion de tipo v2 y v3.: Fuente: The Tor Project, Inc.

2.3.3. Vulnerabilidades de los servicios Onion

Dado el interés de las autoridades por determinar la localización de los servicios Onion y, en última instancia, a los autores responsables de los contenidos y/o servicios en ellos alojados, el análisis de vulnerabilidades y el diseño de potenciales ataques contra la red TOR es un campo de estudio en desarrollo.

Cuando se trata de identificar a un cliente que navega a través de la red TOR es preciso comprometer tanto el nodo de entrada como el nodo de salida que ha seleccionado para generar su circuito TOR. En ese momento, es posible realizar un análisis del tráfico generado para tratar de asociar la dirección IP del cliente con la dirección IP del servidor visitado.

Sin embargo, para atacar un servicio Onion tan solo es necesario comprometer el nodo de salida, ya que el nodo de entrada puede ser manejado por el propio atacante (Tor, servicios ocultos y desanonimización, 2015).

En 2014 se sucedieron una serie de extraños comportamientos en algunos nodos de la red TOR que llevaron a los responsables de la red a desactivar esos nodos. Se desconoce el alcance real de la información que llegaron a recopilar. Se cree que se usó una combinación de ataque de análisis de tráfico y ataque Sybil.

El ataque Sybil permite controlar una gran cantidad de nodos de la red, probablemente porque el atacante los pone en marcha, aunque también es posible tomar el control infectando nodos (Wikipedia contributors, s. f.). Una vez controlados suficientes nodos, el ataque de análisis de tráfico se concretó en la inyección de un mensaje codificado en las cabeceras del protocolo de TOR. Este mensaje se introducía en la red cuando los nodos controlados eran seleccionados como parte del directorio de servicios Onion. El análisis de tráfico consistía en seguir el mensaje desde el directorio, hacia el nodo de entrada y hasta el nodo de salida (Tor security advisory: «relay early» traffic confirmation attack, s. f.).

3. OBJETIVOS Y METODOLOGÍA

Una vez establecidas las bases de funcionamiento de la red TOR, tanto para su uso en la navegación anónima como para sostener la Dark Web, mediante la utilización de servicios Onion, se propone la realización del siguiente trabajo:

3.1. Objetivos

3.1.1. Objetivo general

El principal objetivo que se persigue con este trabajo de fin de grado es determinar de qué maneras es posible en la actualidad publicar información en forma de página web en la Dark Web.

Además, se pretende analizar las herramientas disponibles, utilizándolas en la práctica, para extraer información sobre sus características y poder realizar una comparativa entre ellas.

3.1.2. Objetivos específicos

- Investigar los alojamientos (hosting) disponibles que permitan la publicación de páginas web en la internet oscura mediante servicios Onion.
- Publicar una página web en un hosting utilizando la tecnología de servicios Onion.
- Investigar el funcionamiento del sistema Onion Share para compartir documentos de manera anónima a través de la red TOR y publicar páginas web en servicios Onion.
- Publicar una página web mediante el sistema Onion Share.
- Estudiar la tecnología necesaria para la publicación de una página web con servicios Onion desde un servidor web instalado en un PC doméstico.
- Publicar una página web mediante un servicio Onion que corra en un servidor web instalado en un PC doméstico.
- Realizar un análisis comparativo entre la facilidad de uso, el riesgo, el coste y la vulnerabilidad a la censura de las tres herramientas.

3.2. Metodología

Para llevar a cabo el presente trabajo se empleará una metodología consistente en la investigación y la puesta en práctica.

En primer lugar, se estudiarán las bases tecnológicas que permiten el funcionamiento de la red TOR y la Dark Web.

A continuación, se investigarán las distintas herramientas existentes que permiten la publicación de páginas web con el uso de servicios Onion.

Después, se procederá a la publicación de una página web en un servicio Onion utilizando cada una de las herramientas investigadas en el punto anterior.

Por último, se evaluarán las herramientas desde el punto de vista del autor del contenido a publicar, realizando una comparativa entre las mismas.

Además, se desarrollará una herramienta que visite automáticamente páginas web publicadas en servicios Onion y se visitará con ella los tres sitios web publicados.

4. PROPUESTA METODOLÓGICA

A continuación, se resume el trabajo de investigación y análisis realizado. Se detallan los pasos necesarios para utilizar cada una de las herramientas con el fin de poder evaluar la facilidad de uso.

4.1. Análisis de opciones y publicación de los servicios web

Dada la ausencia de privacidad que actualmente se vive en Internet, ya que la mayoría de los sitios públicos a los que accedemos recaban información (o pueden hacerlo) sobre nuestra identidad y comportamiento, es natural que ciertos usuarios reclamen la posibilidad de navegar de la manera más anónima posible.

Además, en algunos países se prohíbe abiertamente la consulta y/o publicación de ciertos contenidos.

Debido a lo anterior, las personas y organizaciones que desean publicar sus contenidos también pueden desear disponer de la posibilidad de realizar esta publicación de manera anónima.

Para la publicación de contenidos en la Dark Web es posible crear servicios web ocultos que utilicen la red TOR, actualmente conocidos como servicios Onion. Los servicios Onion pueden utilizarse para publicar herramientas con otros fines, como un servidor de correo o un servicio de intercambio de archivos.

Este trabajo se centra en la publicación de páginas web estáticas mediante servicios Onion.

Para crear estos servicios existen principalmente tres alternativas técnicas:

- Opción 1: publicar el contenido en una web alojada en un servicio de alojamiento de terceros
- Opción 2: publicar el contenido en una web alojada en un equipo propio mediante la aplicación Onion Share
- Opción 3: publicar el contenido en una web alojada en un equipo propio mediante una aplicación que ejerza de servidor y trabaje de manera conjunta con la aplicación TOR

A continuación, se expone cómo se debe preparar un equipo para acceder a la Dark Web con seguridad. Después, se desarrollarán y analizarán las tres opciones anteriores.

4.1.1. Preparación del equipo

En primer lugar, describiremos las herramientas mínimas necesarias para acceder y publicar información en la internet oscura.

4.1.1.1. Instalar TOR Browser

La primera aproximación a la navegación anónima bien sea en la Surface Web o en la Dark Web consiste en la instalación del navegador TOR Browser.

Este proceso es tan sencillo como instalar cualquier otro programa, si bien, en este caso especialmente, se recomienda verificar la firma del paquete descargado para evitar una posible suplantación de identidad.

Una vez instalado tan solo debemos ejecutar TOR Browser, conectar a un circuito TOR y podremos comenzar a navegar de forma anónima.

A través de esta aplicación es posible utilizar servicios web con la seguridad de que el anonimato está garantizado siempre que no filtremos nosotros mismos información que nos pueda identificar: autenticarnos en ciertos servicios web, completar formularios, etc. El idioma seleccionado en las visitas que realizamos también puede ayudar a desanonimizar nuestra navegación, ya que gran parte de los usuarios de Internet utilizan el idioma inglés, si elegimos idiomas menos hablados, como el alemán o el holandés estaremos revelando potencialmente información personal.

En la figura 18 se muestra el navegador TOR Browser listo para visitar páginas web manteniendo el anonimato. Se ha generado el circuito TOR, que se muestra al pulsar sobre el icono del candado. La dirección IP del nodo de entrada o guarda se ha enmascarado ya que éste se mantiene durante dos o tres meses para dificultar algunos de los ataques de ruptura de anonimato más conocidos.

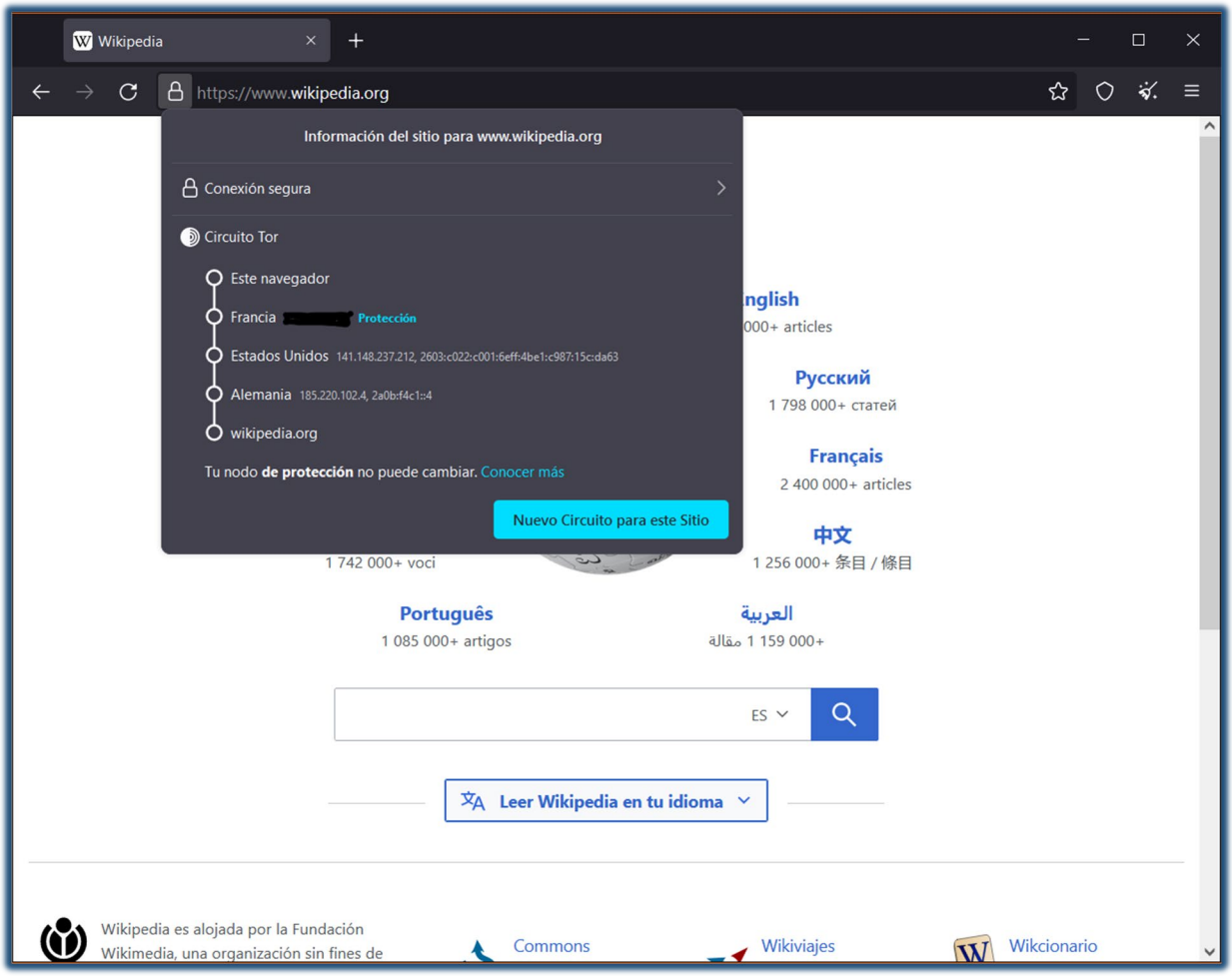

Figura 18: Navegador TOR Browser mostrando el circuito TOR que va a ser utilizado. Fuente: autor.

4.1.1.2. Cuenta de correo electrónico anónima

Para contactar con otros usuarios de la Dark Web o contratar servicios como el hosting para nuestra página web puede ser necesario disponer de una cuenta de correo electrónico anónima.

Las cuentas de correo más populares suelen solicitar gran cantidad de información personal para ser creadas. Sin embargo, las cuentas anónimas tan solo solicitan la aportación de un nombre de usuario y una contraseña.

Existen varios proveedores de correo electrónico anónimo, entre los que encontramos ProtonMail y Mail2tor (Hayes, 2022). Todos ellos permiten el acceso a través de servicios Onion. Para este trabajo se ha creado una cuenta de Mail2tor.

En la figura 19 se muestra la página principal de la interfaz de Mail2Tor y la página de autenticación. Mail2Tor utiliza el servidor de correo SquirrelMail. La interfaz es muy sencilla pues, para mantener las recomendaciones de seguridad de The Tor Project, no utiliza JavaScript (Complementos, extensiones y JavaScript, s. f.).

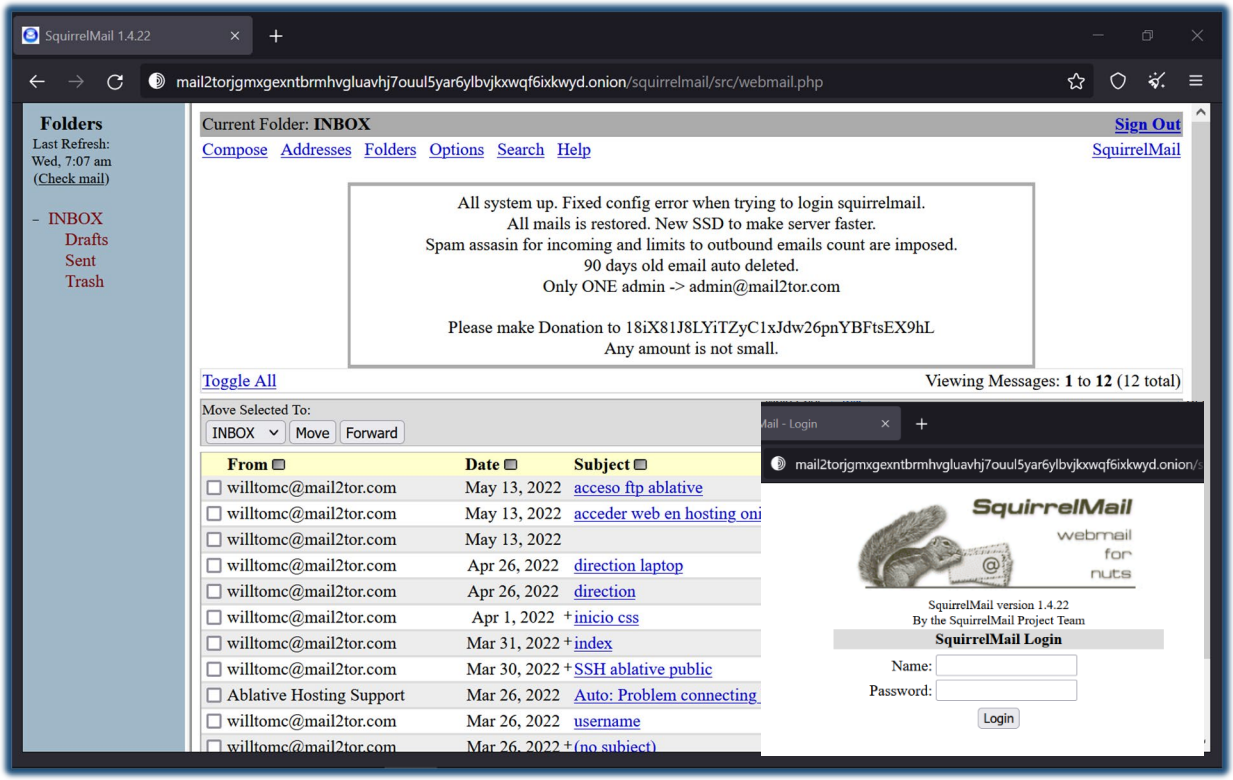

Figura 19: Interfaz del servicio de correo anónimo Mail2tor. Fuente: autor.

4.1.1.3. Instalar SO Linux Tails

La instalación del TOR Browser proporciona navegación anónima siempre que desde nuestro equipo se utilice este navegador, pero no proporciona automáticamente garantías durante la ejecución del resto de aplicaciones del equipo.

De esta manera, si utilizamos otro navegador distinto para acceder a Internet o cualquier otra aplicación se conecta a Internet podría desvelar información que permitiese nuestra identificación y/o localización.

Es posible, en general, configurar nuestro equipo para que utilice la aplicación TOR como proxi para la conexión a Internet, de manera que todo el tráfico que generemos tenga la garantía de anonimato que ofrece la red TOR. Sin embargo, esta configuración no es trivial y siempre

puede ocurrir que parte del tráfico generado por nuestro equipo se envíe directamente a internet sin pasar por la red TOR.

Por todo ello, si queremos tener la seguridad de que nuestro equipo únicamente comparte información en Internet a través de la red TOR, la mejor manera de hacerlo es utilizando la distribución TAILS del sistema operativo Linux.

Sin embargo, la garantía de anonimato utilizando Tails no es total. Es posible sufrir un ataque tipos man-in-the-middle, ya que la conexión entre el nodo de salida y el servidor web circula fuera de la red TOR. Además, la gestión de las contraseñas utilizadas en nuestros equipos y en los servicios a que nos suscribimos siempre es un punto débil (Derechodelared, 2019).

Esta distribución es de tipo Live, de manera que se instala en una memoria USB y se puede ejecutar en cualquier ordenador.

Los pilares de la filosofía de Tails son: anonimato, no persistencia, herramientas de seguridad, uso exclusivo de la red TOR, código abierto y gratuidad (Tails - Cómo funciona Tails, s. f.).

En este trabajo se ha utilizado una copia de Tails para poder utilizar la herramienta OnionShare para la publicación de páginas web en servicios Onion.

En la imagen 20 se muestra el sistema operativo Tails utilizado para la realización de este trabajo. Se puede apreciar la interfaz GUI de OnionShare y el circuito TOR utilizado por el navegador Tor Browser para la conexión a la red Tor. El circuito consta de 3 nodos ya que en ese momento se estaba consultando una página web de la Clear Web.

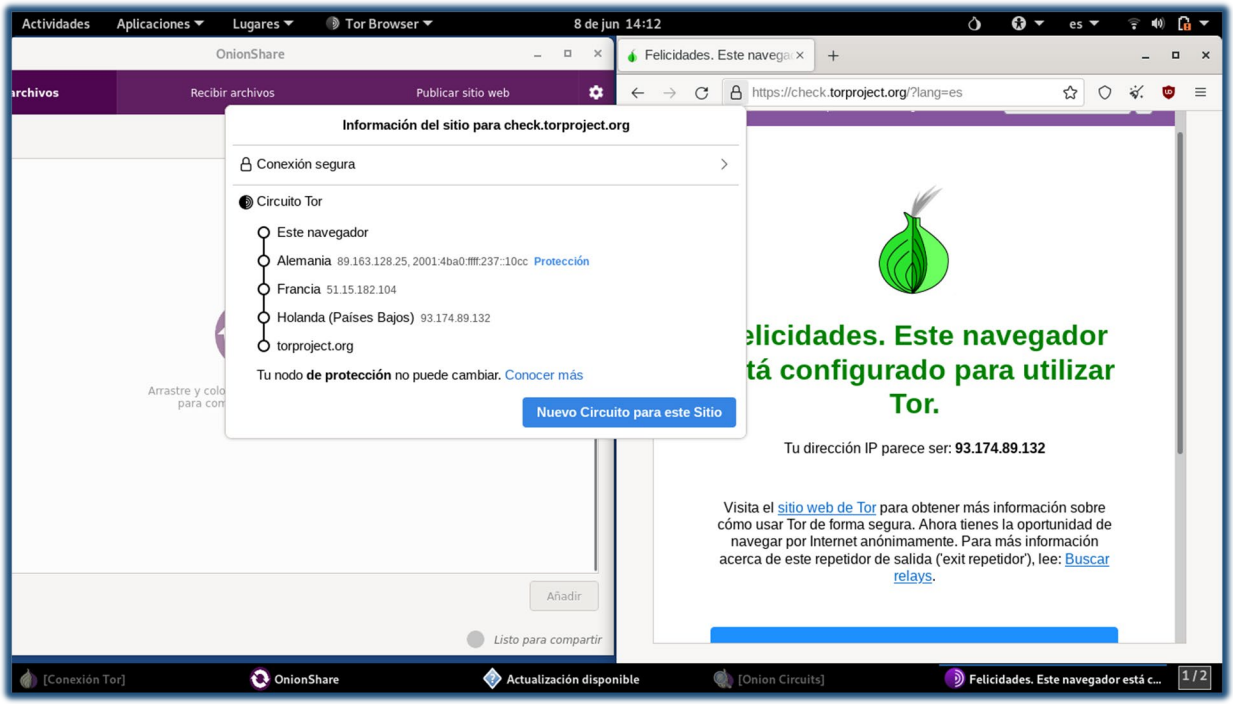

Figura 20: Interfaz del sistema operativo Tails, basado en Linux. Fuente: autor.

4.1.1.4. Almacenamiento persistente

El sistema operativo Tails es de tipo Live y, a priori, no permite el almacenamiento persistente de datos. Sin embargo, esta característica, que aumenta la seguridad y el anonimato, en ocasiones puede dificultar la realización de ciertas tareas.

Para poder almacenar datos de configuración, como contraseñas wifi, claves SSH, preferencias de idioma, etc., es posible configurar el almacenamiento persistente. Para ello se utiliza parte del espacio de almacenamiento disponible en la memoria USB que se haya utilizado para instalar TAILS.

Para la realización del presente trabajo ha sido imprescindible la activación del almacenamiento persistente en TAILS. Una de sus funciones es guardar las claves SSH que permiten la identificación del equipo, por ejemplo, para conectar a un servidor remoto o ante un servidor de hosting para una conexión `sftp`. Para conectar con el servidor de Ablative Hosting utilizado en el punto 4.1.2 se verá que es necesario utilizar una clave SSH permanente.

En la figura 21 se muestra la configuración del volumen persistente de la copia de TAILS utilizada. Se ha seleccionado la persistencia de la configuración de usuario, los datos personales, los marcadores del navegador, las conexiones de red y del cliente SSH.

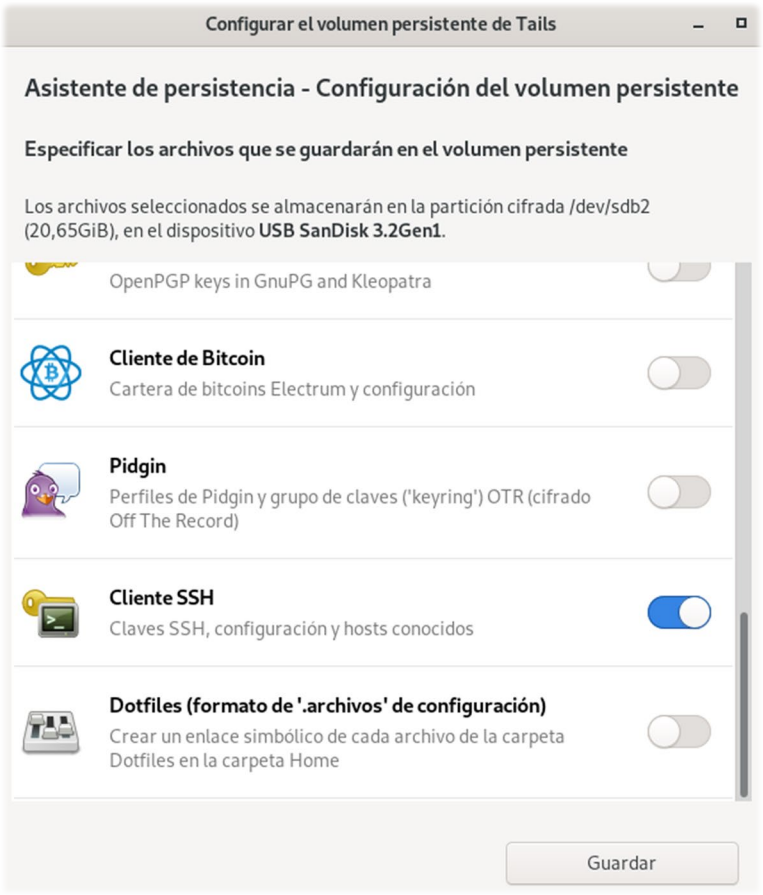

Figura 21: Configuración del volumen persistente en el SO Tails. Fuente: autor.

4.1.2. Publicación de una página web en la Dark Web mediante servicios Onion en un alojamiento externo

Tanto en la Clear Web como en la Dark Web, cuando utilizamos un servicio externo que se encargue de alojar nuestra página web delegamos en un tercero la responsabilidad de la puesta en marcha y mantenimiento del equipo.

A cambio, cedemos al servicio de alojamiento nuestro contenido y le otorgamos la posibilidad de detener el funcionamiento de nuestro servicio web en cualquier momento. Además, siempre es posible que el servicio de alojamiento comunique a las autoridades nuestras actividades y, si dispone de ella, nuestra identidad.

4.1.2.1. Alojamientos que ofrecen publicación de servicios web en la internet oscura

Como se ha descrito previamente, la combinación de las tecnologías TOR para la navegación anónima y el uso de los servicios Onion para la publicación de servicios web permite, por un

lado, el alojamiento y publicación de páginas web de manera que los autores no puedan ser rastreados y, por otro, el acceso a los contenidos alojados por parte de lectores que estén interesados en acceder a ellos de manera anónima.

Esta forma de proceder permite, con ciertas garantías, que los autores de contenido evadan la censura a la que puedan estar sometidos, bien sea ejercida por los gobiernos o por otros grupos de presión. También permite, a su vez, que los lectores accedan a contenidos que podrían estar prohibidos o cuyo consumo podría estar perseguido en determinados países.

Se han localizado tres servicios que ofrecen alojamiento para servicios Onion. Se trata de Impreza, Ablative Hosting y OnionLand.

Impreza ofrece alojamiento en servidores compartidos y también en servidores dedicados. Cada una de estas categorías ofrece diferentes planes dependiendo de la potencia de los equipos en que se alojan, del tráfico que soportan y de las tecnologías que admiten.

OnionLand oferta diferentes planes específicamente diseñados para el empleo de tecnologías de construcción web como WordPress, Joomla y Drupal. En este sentido es más flexible que Ablative Hosting. Sin embargo, no podemos olvidar que estas tecnologías facilitan la fuga de información que podría permitir la identificación de autores y visitantes.

Ablative Hosting también ofrece distintos planes que varían según las características de los servidores utilizados y si son compartidos o no. Para la creación de la cuenta y el acceso al contenido del servidor Ablative Hosting extrema las medidas para mantener el anonimato de los usuarios, bastando con proporcionar una clave SSH. Con esta clave se autenticará en su servidor ftp el equipo desde el que deseemos enviar los archivos de la web. Esta compañía propone una modalidad gratuita, por lo que es la elegida para alojar nuestro servicio Onion en un servidor externo.

En la figura 22 se muestran los servicios ofertados por Impreza en su web. Se aprecia cómo Impreza diferencia entre servicios con servidores compartidos y servicios con servidores dedicados. Impreza no presta servicios de alojamiento gratuitos.

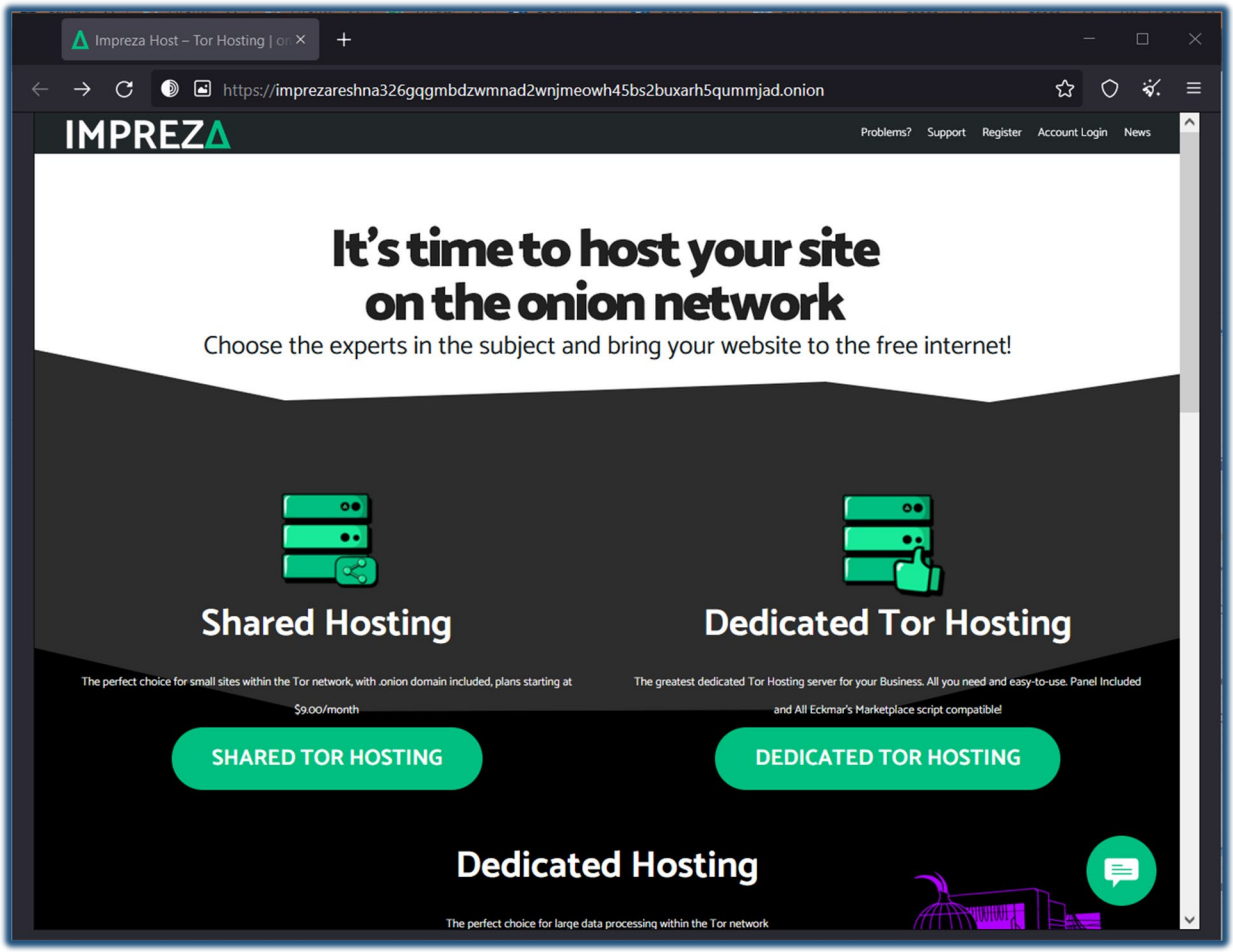

Figura 22: Ejemplo de servicios de hosting ofertados por Impreza. Fuente: Impreza.

Para la publicación de página web en primer lugar se debe realizar el registro en el servicio de alojamiento externo. Una vez dados de alta podremos enviar mediante conexión segura SFTP el contenido de nuestro sitio web al espacio compartido en servidor.

4.1.2.2. Registro en el servicio de alojamiento externo de Ablative Hosting

Para registrarnos y obtener nuestro espacio en el servidor es necesario, como mínimo, disponer de la clave SSH del equipo desde el que nos conectaremos vía SFTP al alojamiento y la contraseña que utilizaremos para autenticarnos. El nombre de usuario es generado automáticamente por el servicio para evitar que, descuidadamente, se propongan nombres de usuario que puedan suponer una fuga de información.

Opcionalmente se puede proporcionar una dirección de correo electrónico en la que el servicio de atención al cliente podrá atendernos en caso de necesidad. Para garantizar la confidencialidad y firmar la comunicación a través de esta vía, se solicita una clave publica PGP

con la que se cifrarán los mensajes que nos sean enviados para que solamente nosotros como receptores legítimos, utilizando nuestra clave privada PGP, podamos descifrar estos mensajes.

En la figura 23 se muestra el formulario de registro en el que se solicitan la clave SSH, la contraseña elegida, la dirección de correo electrónico y la clave pública PGP para contratar el servicio de hosting con la compañía Ablative Hosting.

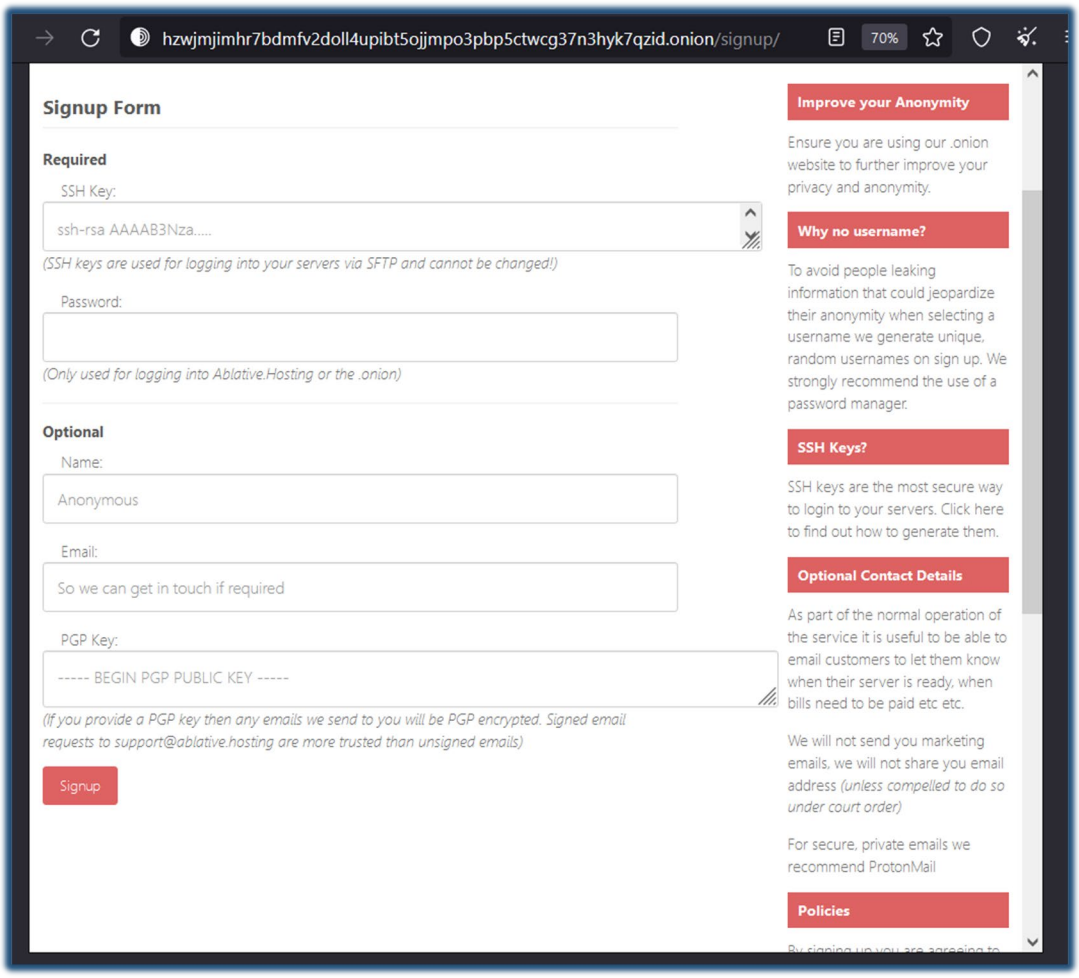

Figura 23: Formulario para dar de alta en Ablative Hosting. Fuente: autor.

Una vez registrado, se nos proporciona el nombre de usuario generado por el sistema. Este nombre de usuario se puede considerar en sí mismo una contraseña, pues su diseño dificulta su memorización.

Los nombres de usuario ofrecidos por Ablative Hosting son de la siguiente forma:

```
ad696e0d-8558-47a2-adb3-8c1fe331ada2
```

Al acceder a la cuenta creada, después de aproximadamente treinta minutos, nuestro espacio reservado en el servidor está disponible y se muestran nuestra dirección tipo Onion V3 y el usuario del servicio SFTP con el que se podrá conectar mediante dicho protocolo y transferir los archivos de nuestro sitio web.

4.1.2.3. Transferencia y publicación de la página web en el servidor

Para conectar con el servidor mediante SFTP es necesario utilizar la misma clave SSH que se aportó durante el registro, el nombre de usuario proporcionado por la compañía de alojamiento y la dirección IP del servidor.

Una vez establecida la conexión es posible navegar por el directorio asignado y transferir al mismo los archivos que formarán parte de nuestra página web. Se deben configurar los permisos de los archivos para que se puedan leer por todos los usuarios, para evitar errores de tipo 403 forbidden. Este error es frecuente cuando los archivos que formarán el sitio web han sido creados en un sistema operativo diferente a Linux. Mediante el comando `chmod` se configuran los permisos en el SO Linux.

En la figura 24 se muestra la conexión SFTP realizada con el servidor de Ablative Hosting. Mediante los comandos `lcd` y `cd`, se navega hasta los directorios entre los que se quiere intercambiar archivos. Con el comando `put` se cargan los archivos desde el equipo local al servidor.

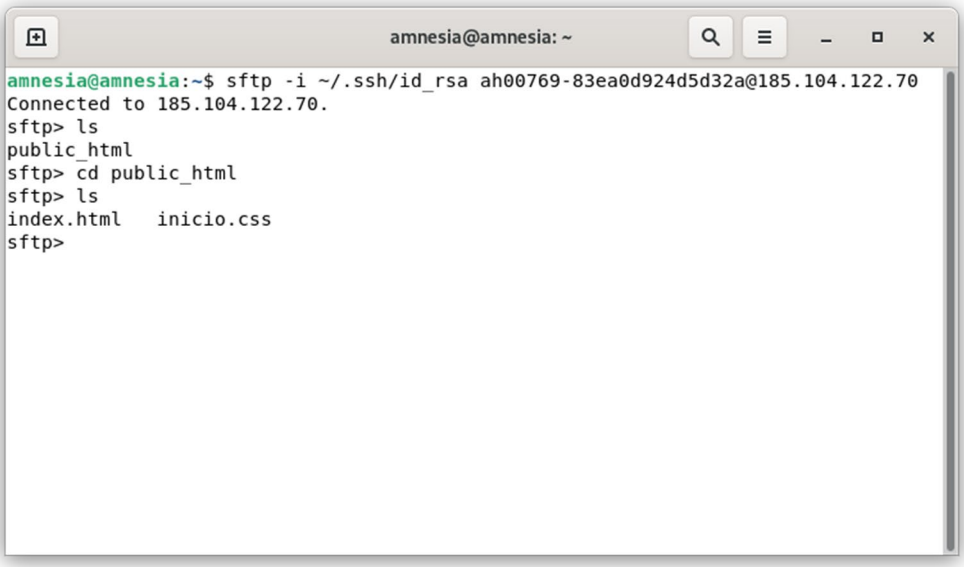

Figura 24: Conexión SFTP con el servidor de Ablative Hosting. Fuente: autor.

Una vez transferidos los archivos que constituyen nuestro sitio web ya es posible acceder al mismo utilizando el navegador web TOR y visitando la dirección .onion v3 que nos fue proporcionada por el servicio de alojamiento.

El servicio proporcionado por Ablative Hosting no permite la personalización de la dirección .onion que tendrá nuestro servicio web. La dirección proporcionada por el servicio de hosting, con la que es posible acceder a la página web mediante el navegador TOR Browser es:

---

http://ve75ty4n7iiuv2nt4wwxngb6kvswpqrj2vpu4caqimu3p5kxmh3qdayd.onion/

En la figura 25 se muestra el sitio web publicado en el alojamiento, visitado utilizando el navegador TOR. También se muestra la topología del circuito TOR utilizado, con 6 nodos intermedios en este caso, ya que se está visitando una página web alojada en un servicio Onion. Ver punto 2.3.1.

Figura 25: Sitio web del presente TFG alojado en el servicio de hosting de Ablative Hosting. Circuito TOR de 6 nodos utilizado. Fuente: autor.

4.1.3. Publicación de una página web en la Dark Web mediante la solución OnionShare

OnionShare es una aplicación multiplataforma que permite el intercambio de archivos entre distintos equipos, la comunicación mediante mensajería instantánea y la publicación de páginas web mediante servicios Onion (How OnionShare Works — OnionShare 2.5 documentation, s. f.).

La utilización de la red TOR proporciona las habituales garantías de anonimato. Estas garantías no son totales y dependen, en parte, de la responsabilidad del usuario de la aplicación a la hora de configurar su equipo y compartir la información.

4.1.3.1. Instalación de OnionShare

La aplicación está disponible para sistemas operativos Windows, MacOS y Linux. Dispone de una versión con interfaz gráfica y otra versión funcional a través del terminal de Linux.

En la distribución de Linux Tails, OnionShare viene instalado por defecto. Esta distribución ofrece mayores garantías de anonimato ya que canaliza todo el tráfico generado hacia internet a través de la red TOR. Cuando se utiliza OnionShare en otro sistema operativo distinto a Tails, el riesgo de fuga de información es más alto.

En la figura 26 se muestra la versión de OnionShare 2.2 instalada en Tails. Esta aplicación cuenta tanto con una interfaz de tipo GUI, Graphic User Interface, como con una interfaz de usuario a través del terminal.

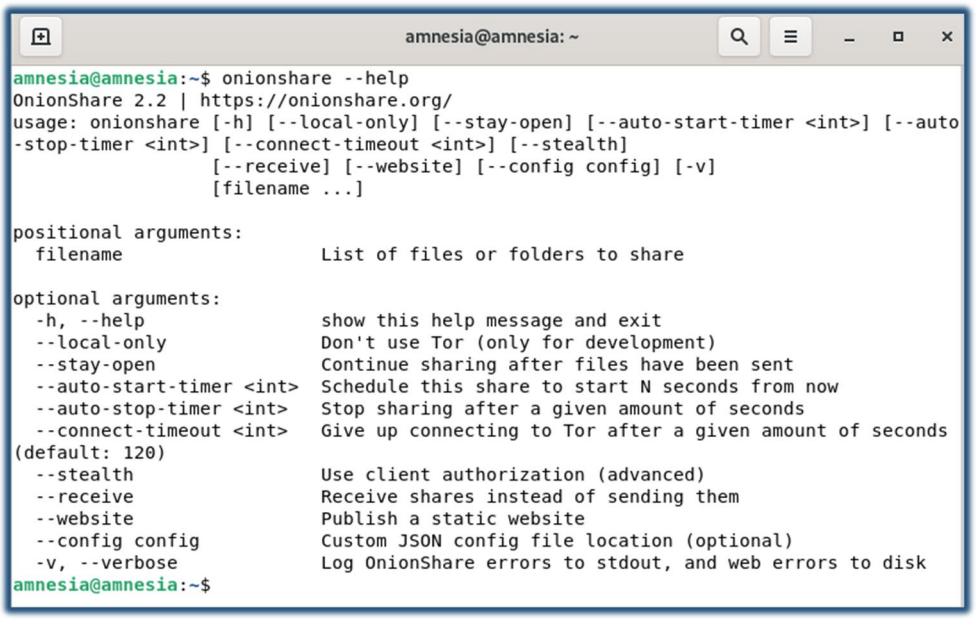

Figura 26: Interfaz en el terminal Linux de la aplicación OnionShare. Fuente: autor.

4.1.3.2. Publicación de la página web mediante OnionShare

A través de la interfaz gráfica de la aplicación OnionShare, en primer lugar, es necesario seleccionar la pestaña Publicar sitio web.

A continuación, se deben añadir los archivos que formarán parte del sitio web. Si se desea que una de las páginas servidas actúe como página principal de la web, debe nombrarse como `index.html`.

Por último, se pulsa el botón `Comenzar a compartir`. Una vez que se ha comenzado a servir el sitio web, se nos muestra la dirección .onion v3 a través de la cual podremos acceder al mismo.

En la figura 27 se muestra el sitio web publicado a través de OnionShare, visitado utilizando el navegador TOR. Se muestra también la interfaz de OnionShare, con el espacio para arrastrar y soltar los archivos que forman la página. Se aprecia la dirección v3 que propone la aplicación y que se introduce en la barra de direcciones del navegador TOR Browser.

http://335b7gmxd2zv3oute5dh7h22as2a6xc3gl4mhyjtl4acocze6gwaqpad.onion

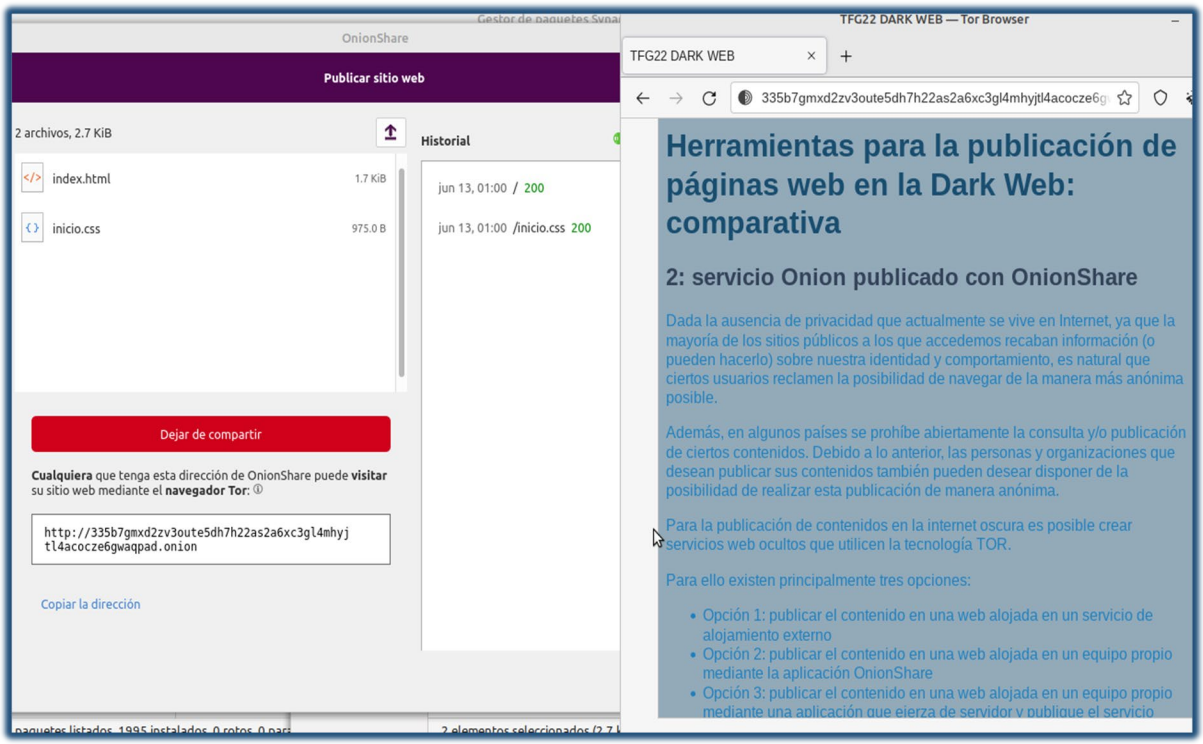

Figura 27: Sitio web del presente TFG alojado en la aplicación OnionShare. Fuente: autor.

4.1.3.3. Persistencia de la dirección .onion generada con OnionShare

Por defecto la dirección web .onion generada por la aplicación para poder acceder al sitio web se renueva cada vez que reiniciamos el programa.

En el caso de que deseemos que la dirección sea persistente, se debe configurar esta preferencia en el menú de configuración de la aplicación. La persistencia de la dirección tan solo es configurable en las instancias de OnionShare que corran en sistemas operativos que no sean Tails.

Debido al diseño propio de esta distribución Linux, de forma predeterminada todas las configuraciones y archivos guardados durante una sesión de Tails son eliminados cuando esta sesión se cierra. Utilizando el almacenamiento persistente se puede guardar parte de la configuración del sistema operativo y los archivos que vayamos creando. Sin embargo, no es posible guardar la dirección .onion generada por OnionShare.

Por todo ello, si deseamos mantener un sitio web con cierta estabilidad, para utilizar OnionShare con una dirección persistente será necesario utilizar un sistema operativo diferente a Tails, con las consecuencias para la seguridad que ello conlleva.

La razón para este comportamiento por defecto de la aplicación OnionShare radica en que la volatilidad de la dirección .onion utilizada en nuestro servicio web favorece el anonimato a la hora de publicar contenidos. Evidentemente, esta dirección .onion constantemente renovada ha de ser constantemente compartida con los potenciales visitantes de nuestro servicio web.

4.1.4. Publicación de una página web en la Dark Web mediante un servidor instalado en un PC doméstico

La última alternativa que se va a estudiar consiste en instalar un servidos NGINX en un ordenador doméstico y configurarlo de manera que sirva la página web a través de un servicio Onion en TOR (The Tor Project, Inc, s. f.-b).

4.1.4.1. Instalación y configuración de NGINX y TOR

La instalación del servidor NGINX resulta bastante sencilla. Tras la actualización de rigor se procede a invocar el gestor de paquetes apt.

En la figura 28 se muestra la instalación del servidor NGINX. La instalación básica es muy sencilla, aunque requiere del uso del terminal.

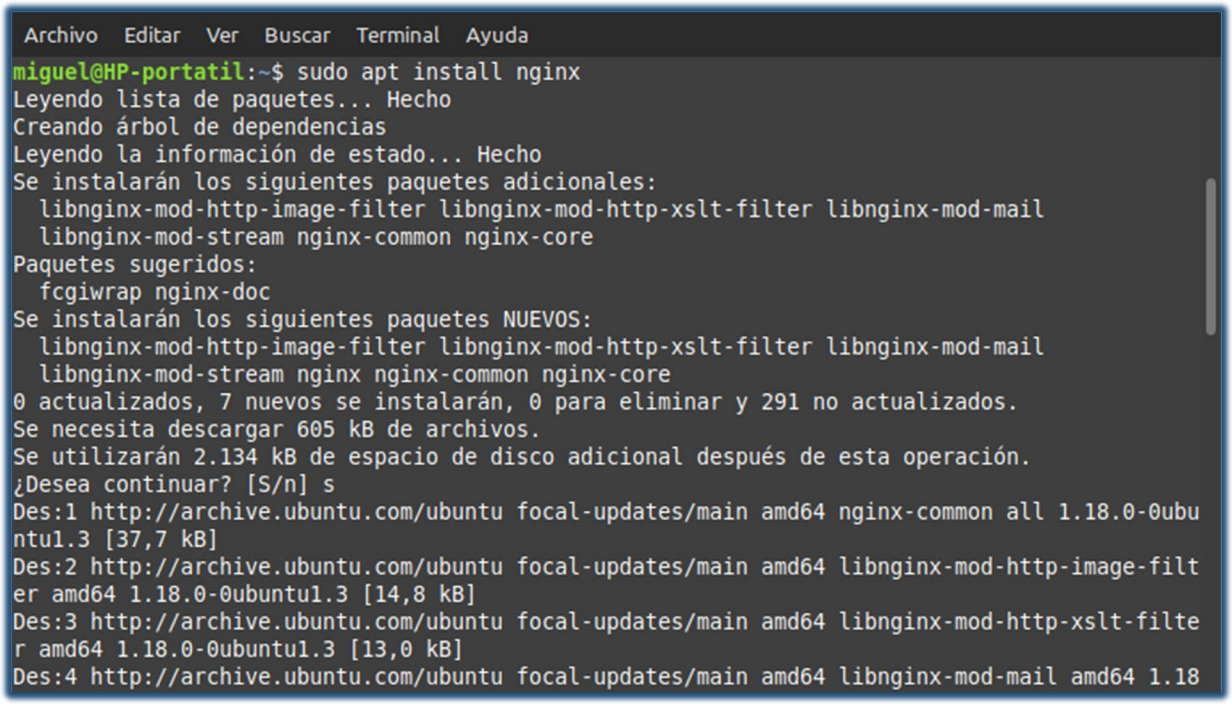

Figura 28: Instalación de la aplicación servidor NGINX. Fuente: autor.

A continuación, debemos instalar TOR en el caso de que no esté instalado. Una vez dispongamos de TOR debemos configurar el servicio Onion modificando el archivo de configuración que se encuentra en `/etc/tor/torrc`.

En este archivo de configuración torrc se deben localizar las líneas `HiddenServiceDir` y `HiddenServicePort`. Por defecto se hayan comentadas. Para activar el servicio Onion se deben descomentar ambas.

En la primera línea, se indica el directorio en el que se almacenará la clave privada del servicio Onion y, además, la clave pública y un archivo, llamado `hostname`, con la URL de tipo Onion que tendrá nuestro servicio.

En la segunda línea, se indica el puerto a través del cual llegarán las conexiones a nuestro servicio Onion.

En la figura 29 se muestra la dirección v3 generada inicialmente por la aplicación TOR. Para conocerla se ha de consultar el contenido del archivo `hostname`.

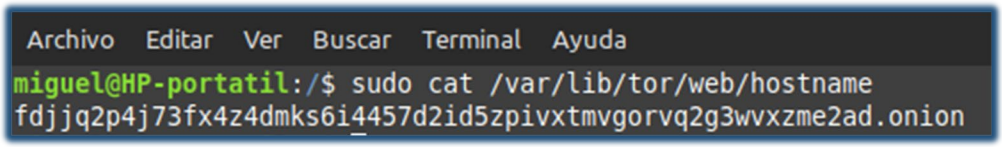

Figura 29: Dirección v3 generada inicialmente por la aplicación TOR. Fuente: autor.

En la figura 30 se muestra la sección del archivo de configuración actualizado para activar el servicio oculto. Se debe buscar la sección titulada: "`this section is for location hidden services`". La primera línea indica que las claves se encuentran en el directorio `web` y la segunda que el puerto seleccionado es el puerto 80.

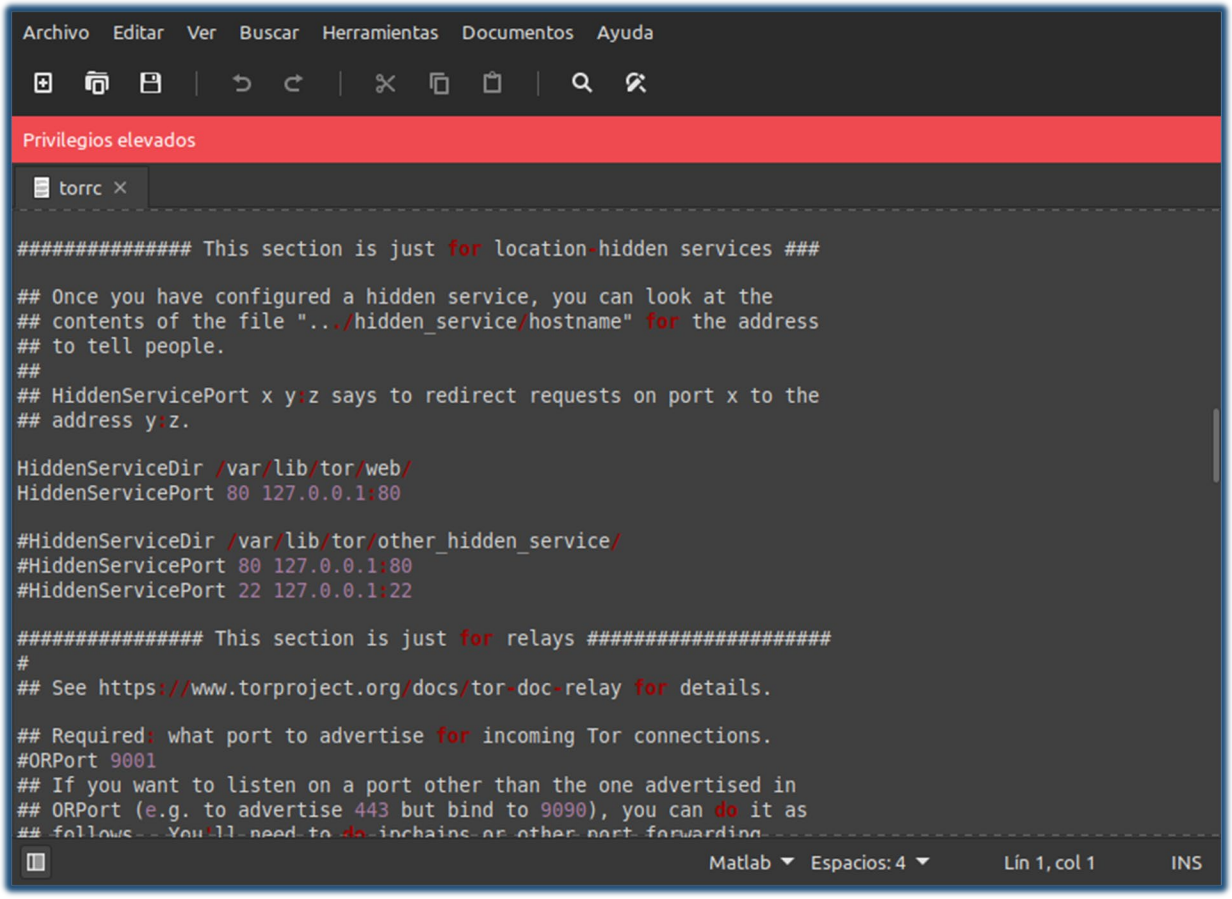

Figura 30: Archivo de configuración de la aplicación TOR. Fuente: autor.

4.1.4.2. Generación y configuración de una dirección .onion personalizada

Cuando activamos por primera vez el servicio Onion, TOR genera automáticamente un par de claves pública y privada y su URL correspondiente. Habitualmente las direcciones tienen un

aspecto poco amigable, recordemos que todas ellas son una sucesión de 56 caracteres seguidos del sufijo .onion.

El diseño de las direcciones no permite tampoco la utilización de letras mayúsculas ni los números 1, 8, 9 ni 0. Sin embargo, es posible, hasta cierto punto, personalizar el aspecto de la dirección de nuestro servicio web, seleccionando alguno de los caracteres que formarán parte de ésta.

Las claves pública y privada y la dirección URL están relacionadas criptográficamente. Por ello, no es posible elegir una dirección .onion totalmente personalizada. Los generadores de direcciones personalizadas van generando claves aleatoriamente y comprueban las direcciones resultantes hasta que una de ellas cumple con los requisitos impuestos. Este proceso requiere de una gran cantidad de cálculo, de manera que, cuantos más caracteres deseemos fijar más lento será el proceso.

En la figura 31 se muestra la velocidad con la que crece el tiempo necesario para generar una dirección .onion v3 según aumentamos los caracteres que deseamos fijar. Se observa un crecimiento de carácter exponencial. Por ejemplo, para obtener una dirección con los 4 primeros caracteres personalizados, son necesarios solo 30 segundos. Sin embargo, para personalizar 8 caracteres ya se necesitaría 1 año. Los datos del ejemplo se han calculado con un equipo bastante modesto, pero el crecimiento exponencial se mantendría con cualquier otro equipo.

```
# Roughly estimated Onion v3 vanity address generation times using
mkp224o on a 5-node Raspberry Pi Cluster
# mkp224o: https://github.com/cathugger/mkp224o
# Raspberry Pi Cluster: https://www.jamieweb.net/projects/computing-stats/

Vanity Characters : Approximate Generation Time
1  : <1 second
4  : 30 seconds
5  : 16 minutes
7  : 11.5 days
8  : 1 year
9  : 32 years
10 : 1,024 years
11 : 32,768 years
12 : 1 million years
20 : 1 quintillion years
30 : 1 decillion years
40 : 1 quindecillion years
50 : 1 vigintillion years
51 : 32 vigintillion years
52 : 10^66 years
53 : 10^69 years
54 : 10^72 years
55 : 10^75 years
56 : 10^78 years
```

Figura 31: Aumento exponencial del tiempo necesario para personalizar una dirección .onion ve. Fuente: Jamie Scaife, www.jamieweb.net.

Para personalizar la dirección del servicio Onion alojado en el servidor del presente trabajo se ha utilizado la herramienta para generación de direcciones v3 `mkp224o`. Esta herramienta se debe compilar en Linux.

El proceso se lanza con el comando `mkp22o` al que se le añaden usualmente dos modificadores. Con `-f` se indica un fichero de texto en el que se introducen las palabras con las que deseamos que comience nuestra dirección. En lugar de indicar un solo conjunto de caracteres, cuanto mayor sea el número de palabras propuestas más sencillo y rápido será conseguir que una dirección cumpla con los requisitos. Con el modificador `-d` se indica el directorio donde queremos que se vayan almacenando las claves generadas.

En la figura 32 se muestra la herramienta `mkp224o` generando direcciones y claves que cumplan las restricciones introducidas en el archivo `filtro2`, que se listan en una columna al comenzar la ejecución. Se aprecia como, tras 60 segundos de procesamiento, se obtiene la primera dirección que cumple los requisitos.

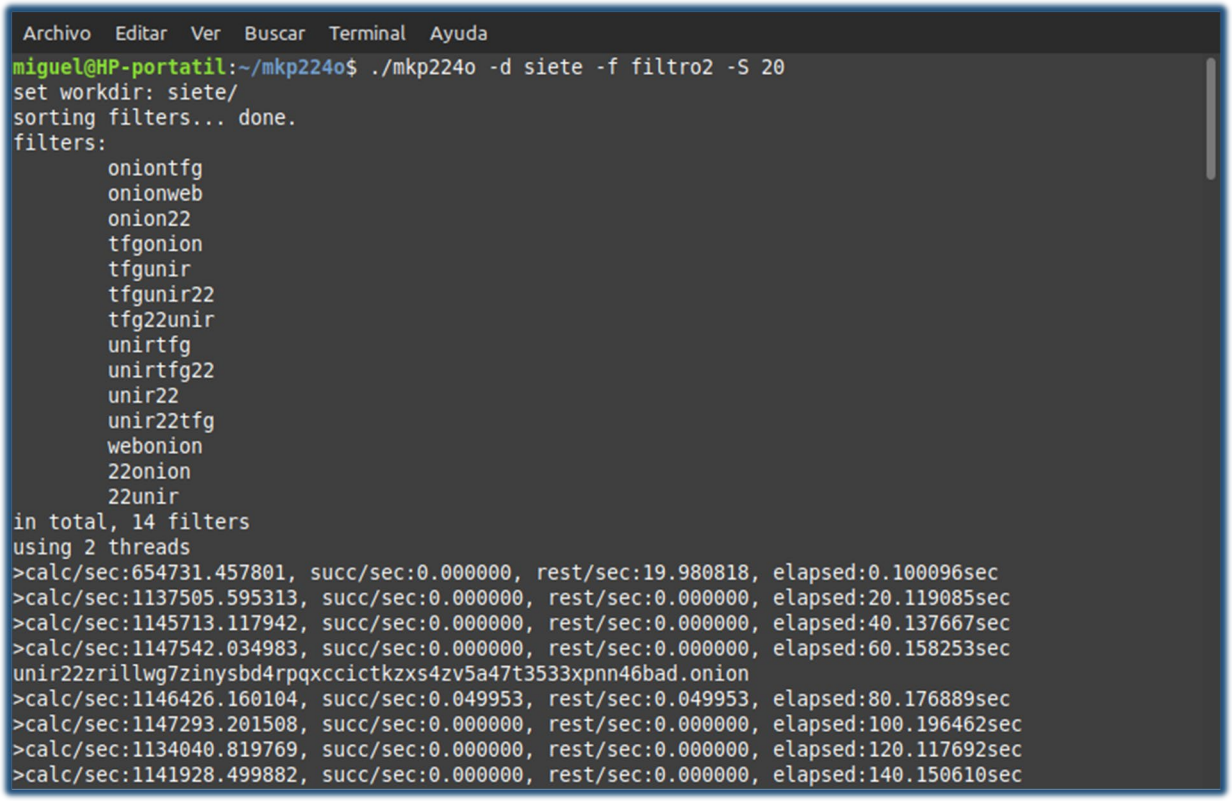

Figura 32: Aplicación mkp224o personalizando direcciones .onion v3. Fuente: autor.

La dirección URL v3 seleccionada para la página web es la siguiente:

unir22n6hjmgi2gx6kmcpknzbpqisydoxt7khjosezhxy4l32ikwxiad.onion/

En la figura 33 se muestra el servicio Onion tipo sitio web servido mediante el servidor NGINX y TOR, visitado mediante el navegador TOR BROWSER. En la barra de direcciones se aprecia la dirección .onion v3 personalizada generada con la herramienta mkp224o.

Figura 33: Sitio web del presente TFG alojado en el servidor propio. Fuente: autor.

4.1.5. Análisis del contenido de los sitios web publicados mediante la aplicación crawler desarrollada

Como se expone en el anexo I, la aplicación desarrollada permite la descarga del contenido íntegro de una página web publicada en un servicio Onion. A continuación, este contenido es analizado en busca de enlaces a otros servicios Onion. Por último, todos los enlaces localizados se exportan a archivos de texto.

Para demostrar el funcionamiento de la herramienta en un entorno controlado se ha ejecutado introduciendo como dirección Onion v3 inicial la correspondiente a la versión de la página web publicada en un alojamiento externo:

http://ve75ty4n7iiuv2nt4wwxngb6kvswpqrj2vpu4caqimu3p5kxmh3qdayd.onion/

En cada una de las páginas webs creadas se ha introducido un enlace al resto de páginas, de manera que el programa visitará las tres páginas sucesivamente.

En la figura 34 se muestra la página publicada en un alojamiento externo con el enlace a la página publicada mediante OnionShare.

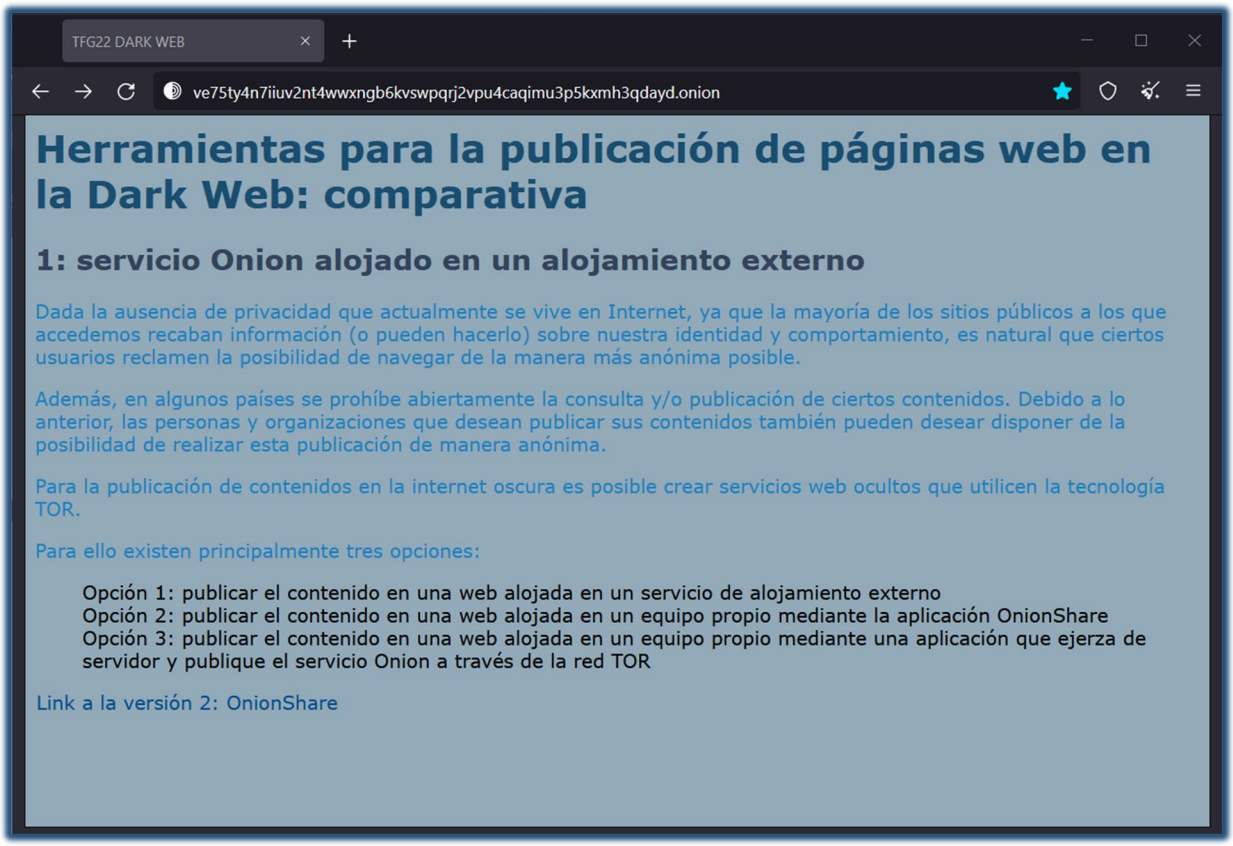

Figura 34: Versión de la web publicada en un alojamiento externo con un enlace a la versión publicada con OnionShare. Fuente: autor.

Para lanzar el programa se deben indicar como argumentos el número de páginas que se desea visitar y la dirección Onion v3 de la página por la que se desea comenzar.

En la figura 35 se muestra el resultado del programa después de visitar las tres páginas publicadas. Cabe destacar la llamada al programa, con los argumentos correspondientes y el encadenamiento de resultados. Tras analizar la primera dirección, (http://ve75...) se ofrece como resultado la segunda (http://335b...). Después de analizar la segunda dirección, se ofrece como resultado la tercera (http://unir...).

Además, los resultados se exportan a archivos de texto identificados con la fecha y la hora en que se realizó el análisis, como se muestra en la figura 36.

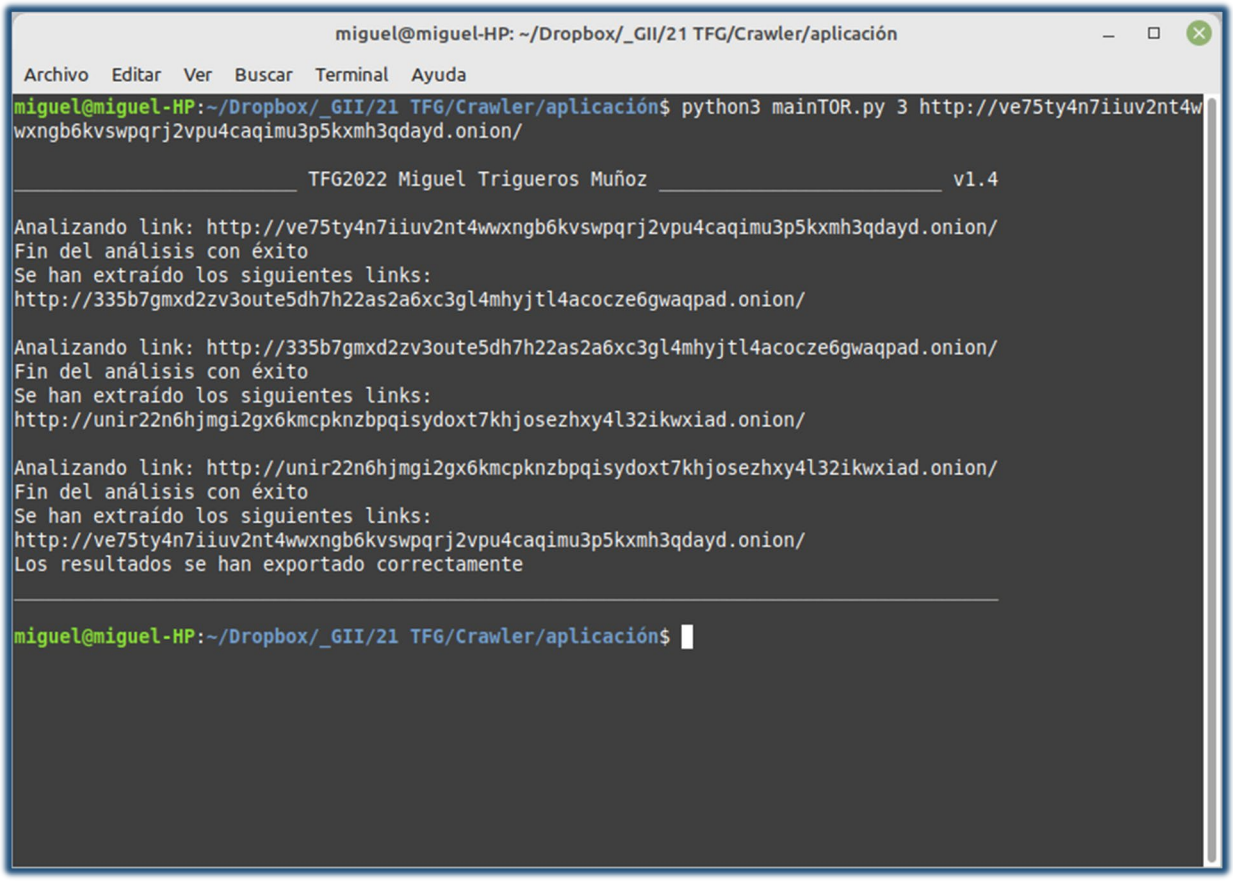

Figura 35: Resultados del programa tras visitar las tres páginas publicadas en el presente trabajo. Fuente: autor.

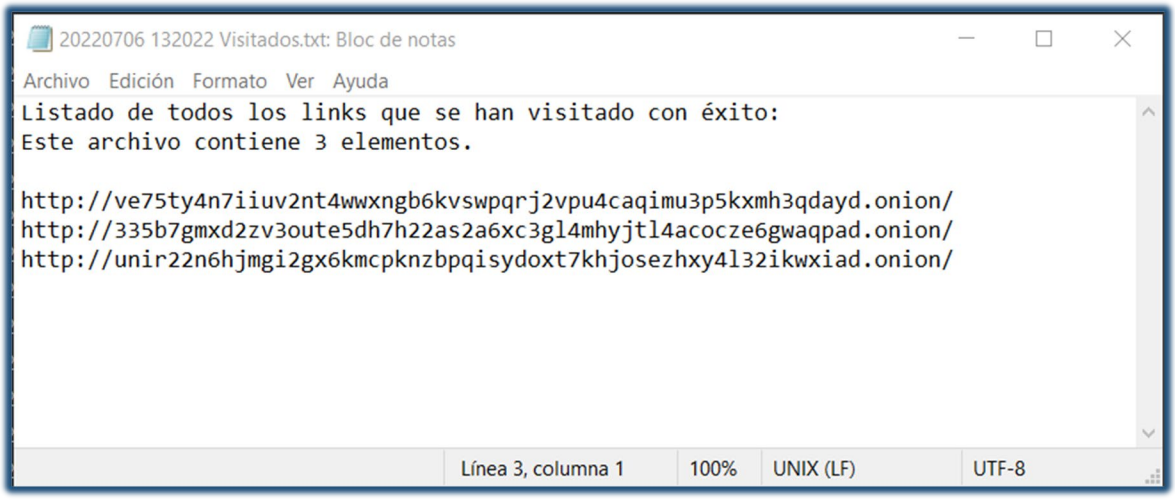

Figura 36: Contenido del archivo de texto en el que se exportan las direcciones de las páginas web visitadas.

4.2. Evaluación

4.2.1. Comparativa entre los tres métodos analizados

Una vez establecidas las distintas opciones para la publicación del servicio oculto se procede a comparar los tres métodos descritos según los siguientes criterios: localización de los datos, limitaciones al contenido, coste, riesgo y dificultad.

En cuanto a la localización de los datos, en el método 1, utilizando un alojamiento de terceros, los archivos y, en general, los datos se alojan en los equipos de la organización que presta el servicio. Con el método 1 no es necesario que el promotor del servicio oculto almacene el contenido es su propio equipo ni que mantenga su propio equipo permanentemente encendido y conectado.

Tanto en el método 2, utilizando OnionShare, como en el método 3, utilizando un servidor propio, los datos sí permanecen alojados en el equipo del autor y éste deber permanecer permanentemente encendido y conectado.

Si pensamos en las limitaciones sobre el contenido, en el método 1, la organización prestadora de los servicios de alojamiento puede poner las limitaciones que considere al contenido que puede ser alojado en sus servidores.

En los métodos 2 y 3 no existen limitaciones al tipo de contenido que el promotor del servicio web decida mostrar en él.

Los tres métodos, en general, pueden ser gratuitos.

En el método 1, los servicios de alojamiento de terceros pueden ser de pago. Habitualmente, las prestaciones ofrecidas por el servicio de pago suelen mejorar con el precio.

En los métodos 2 y 3, en principio no hay que realizar ningún pago, si bien es cierto que los costes energéticos al mantener un servidor permanentemente activo pueden ser importantes.

También debe analizarse el riesgo que se corre en el caso de alojar contenidos ilegales o incriminatorios.

En el método 1, los únicos enlaces necesarios entre el alojamiento y el autor del servicio web son el nombre de usuario y la clave SSH utilizados para, a través de la conexión SFTP, transferir los archivos al servidor. Si durante una investigación forense se encontrasen el nombre de

usuario o la clave SSH en un equipo del autor, estas podrían ser evidencias suficientes para implicar al autor en la publicación del servicio oculto.

En los métodos 2 y 3, dado que el autor debe mantener los datos en su propio equipo, las evidencias que le relacionan con los datos son mucho más directas.

Además, como el equipo permanece siempre conectado a la red es posible la realización de un ataque de análisis de tráfico que termine por revelar la localización del servidor.

Por último, debemos analizar la dificultad técnica que implica para un usuario con unos conocimientos medios de informática la puesta en marcha de cada una de estas opciones.

El método 1 se considera la opción de mayor dificultad, debido a la necesidad de generar y gestionar la clave SSH y utilizar SFTP para subir los datos al servidor.

El método 2 sería la opción más sencilla, ya que OnionShare cuenta con una interfaz gráfica bastante simple y amigable que permite la publicación del contenido del servicio oculto con solo arrastrar y soltar los archivos.

El método 3 puede considerarse una opción intermedia ya que, como mínimo, se ha tener unos conocimientos básicos de Linux para instalar el servidor a través de línea de comandos y editar el archivo de configuración de TOR para activar los servicios ocultos.

En la tabla 1 se resume la comparativa realizada entre los tres métodos analizados, según los criterios de localización de archivos, limitaciones al contenido, coste, riesgo y dificultad técnica.

Tabla 1: Comparativa entre los tres métodos analizados. Fuente: autor.

Método	Localización de archivos	Limitaciones al contenido	Coste	Riesgo	Dificultad técnica
1 Alojamiento	Servidor de terceros	Según propietario del alojamiento	Gratuito / de pago	Bajo	Alta
2 OnionShare	Propio equipo	Ninguna	Gratuito	Medio	Baja
3 Servidor propio	Propio equipo	Ninguna	Gratuito	Medio	Media

5. Conclusiones y trabajo futuro

5.1. Conclusiones

La tecnología de servicios Onion proporciona a los usuarios de la Red altas garantías de anonimato y privacidad. Tanto los autores de contenido como los visitantes que quieran acceder a la información pueden utilizar los servicios Onion y el navegador TOR Browser para, a través de la red TOR, publicar contenido y consultarlo evadiendo la posible vigilancia de estados y empresas.

Actualmente existen tres herramientas disponibles para cualquier usuario de Internet que pueden ser utilizadas para publicar páginas web a través de servicios Onion en la Dark Web.

La aplicación OnionShare es la herramienta que ofrece la menor dificultad técnica para publicar contenido. Además, al estar disponible para varios sistemas operativos el acceso a la misma es más sencillo para cualquier usuario de Internet. Sin embargo, al permanecer los datos en el equipo del autor de la página web, presenta un riesgo medio de que una investigación encuentre evidencias que lo inculpen.

El uso de un servidor de terceros que opere en la Dark Web representa la mayor dificultad técnica, pero, a cambio, el riesgo de que el autor pueda ser relacionado con el contenido del sitio web es menor. Además, con esta herramienta derivamos la responsabilidad sobre la disponibilidad de la página a un tercero. Finalmente, el responsable del servicio puede censurar el contenido de nuestro sitio web.

La construcción de un servidor en un equipo propio proporciona las mismas ventajas que la aplicación OnionShare. Sin embargo, la dificultad técnica es bastante mayor. El factor diferencial más positivo de esta herramienta es que ofrece la posibilidad de personalizar la dirección URL de nuestro sitio web.

En resumen, se han construido las tres páginas web que se marcaron como objetivo y se ha podido evaluar las ventajas y los inconvenientes de las tres herramientas estudiadas.

Además, se ha comprobado automáticamente el funcionamiento de las tres páginas mediante el análisis de enlaces realizado con la herramienta tipo crawler desarrollada.

5.2. Trabajos futuros

Este trabajo se ha centrado en evaluar la facilidad de uso de las tres herramientas, analizando si un usuario medio de la Red podría utilizarlas para publicar un sitio web en las Dark Web.

En el futuro, sería muy interesante realizar una comparación entre las distintas herramientas analizando el rendimiento que alcanzan en cuanto a número de visitas, velocidad de conexión, etc.

También podrían ser expuestos a distintos tipos de ataques, como ataques DDOS, para evaluar su capacidad de respuesta.

Por último, las páginas web utilizadas para poner a prueba las herramientas se han construido únicamente con tecnología html y css. Otra línea de investigación futura podría tratar de comprobar las tecnologías que estas herramientas permiten utilizar: JavaScript, php, MySql, etc.

Referencias bibliográficas

Bertram, S. K. (2015). The Tao of open source intelligence (IT Governance Publishing, Ed.). IT Governance Publishing.

Castillo, P. (2015, junio 29). Servicios ocultos en Tor: Cómo logran esconderse. SecurityInside.info. https://securityinside.info/servicios-ocultos-en-tor-como-pasan-desapercibidos/

Catalano, D., Fiore, D., & Gennaro, R. (2017). A certificateless approach to onion routing. International Journal of Information Security, 16(3), 327-343. https://doi.org/10.1007/s10207-016-0337-x

Cómo funciona la web. (s. f.). Mozilla.org. Recuperado 1 de junio de 2022, de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/How_the_Web_works

Comparación de I2P con Tor - I2P. (s. f.). Geti2p.net. Recuperado 1 de junio de 2022, de https://geti2p.net/es/comparison/tor

Complementos, extensones y JavaScript. (s. f.). Torproject.org. Recuperado 8 de junio de 2022, de https://tb-manual.torproject.org/es/plugins/

Derechodelared, P. (2019, noviembre 16). «Tails», privacidad para cualquier persona en cualquier lugar. Derecho de la Red. https://derechodelared.com/tails/

El Proyecto Tor. (s. f.). Torproject.org. Recuperado 1 de junio de 2022, de https://www.torproject.org/es/about/history/

Goldschlag, D. M., Reed, M. G., & Syverson, P. F. (1996). Hiding Routing information. En Information Hiding (pp. 137-150). Springer Berlin Heidelberg.

Goulet, D. (2020, junio 15). Onion Service v2 Deprecation Timeline. Torproject.org. https://lists.torproject.org/pipermail/tor-dev/2020-June/014365.html

Guía detallada sobre cómo funciona la Búsqueda de Google. (s. f.). Google Developers. Recuperado 31 de mayo de 2022, de https://developers.google.com/search/docs/advanced/guidelines/how-search-works?hl=es

Hasan, M. (2022, mayo 18). State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally. IoT Analytics; IoT Analytics GmbH. https://iot-analytics.com/number-connected-iot-devices/

Hatta, M. (2020). Deep web, dark web, dark net: A taxonomy of “hidden” Internet. Annals of Business Administrative Science, 19(6), 277-292. https://doi.org/10.7880/abas.0200908a

Hayes, T. (2022, marzo 23). Dark web email service providers. Darkweblinkssites.com. https://darkweblinkssites.com/dark-web-email-service-providers/

How OnionShare Works — OnionShare 2.5 documentation. (s. f.). Onionshare.org. Recuperado 3 de junio de 2022, de https://docs.onionshare.org/2.5/en/features.html

Johnson, A. (2015). A proposal to change hidden service terminology. Tor Project. https://lists.torproject.org/pipermail/ tor-dev/2015-February/008256.html.

MacAskill, E., Dance, G., Cage, F., Chen, G., & Popovich, N. (2013, noviembre 1). NSA files decoded: Edward Snowden's surveillance revelations explained. http://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decoded

Onion Service version 2 deprecation timeline. (s. f.). Torproject.org. Recuperado 3 de junio de 2022, de https://blog.torproject.org/v2-deprecation-timeline/

path-spec.txt - torspec - Tor's protocol specifications. (s. f.). Torproject.org. Recuperado 1 de junio de 2022, de https://gitweb.torproject.org/torspec.git/tree/path-spec.txt

Perunicic, K. (s. f.). The Complete List of Blocked Websites in China & How to Access Them. Vpnmentor.com. Recuperado 31 de mayo de 2022, de https://www.vpnmentor.com/blog/the-complete-list-of-blocked-websites-in-china-how-to-access-them/

Scaife, J. (s. f.). Tor Onion v3 vanity address. Jamieweb.net. Recuperado 2 de junio de 2022, de https://www.jamieweb.net/blog/onionv3-vanity-address/

SERVICIOS CEBOLLA. (s. f.). Torproject.org. Recuperado 2 de junio de 2022, de https://tb-manual.torproject.org/es/onion-services/

Tails - Cómo funciona Tails. (s. f.). Boum.org. Recuperado 8 de junio de 2022, de https://tails.boum.org/about/index.es.html

The Tor Project, Inc. (s. f.-a). Tor: Onion Service Protocol. Torproject.org. Recuperado 2 de junio de 2022, de https://2019.www.torproject.org/docs/onion-services.html.en

The Tor Project, Inc. (s. f.-b). Tor project: Onion service configuration instructions. Torproject.org. Recuperado 3 de junio de 2022, de https://2019.www.torproject.org/docs/tor-onion-service.html.en

The Tor Project, Inc. (s. f.-c). Tor Project: Overview. Torproject.org. Recuperado 1 de junio de 2022, de https://2019.www.torproject.org/about/overview.html.en

Tor project. (s. f.). Torproject.org. Recuperado 1 de junio de 2022, de https://community.torproject.org/onion-services/setup/

Tor Project. (s. f.). Torproject.org. Recuperado 1 de junio de 2022, de https://community.torproject.org/es/relay/types-of-relays/

Tor security advisory: «relay early» traffic confirmation attack. (s. f.). Torproject.org. Recuperado 3 de junio de 2022, de https://blog.torproject.org/tor-security-advisory-relay-early-traffic-confirmation-attack/

Tor, servicios ocultos y desanonimización. (2015, enero 7). INCIBE-CERT. https://www.incibe-cert.es/blog/tor-servicios-ocultos-desanonimizacion

V3 onion services usage. (s. f.). Torproject.org. Recuperado 2 de junio de 2022, de https://blog.torproject.org/v3-onion-services-usage/

Wikipedia contributors. (s. f.). Ataque Sybil. Wikipedia, The Free Encyclopedia. https://es.wikipedia.org/w/index.php?title=Ataque_Sybil&oldid=138389706

Winkler, S., & Zeadally, S. (2015). An analysis of tools for online anonymity. International Journal of Pervasive Computing and Communications, 11(4), 436-453. https://doi.org/10.1108/ijpcc-08-2015-0030

Índice de acrónimos

IP: Internet protocol

TOR: The Onion Router

P2P: Peer to peer

I2P: Invisible Internet Protocol

SO: Sistema Operativo

URL: Unirversal Resouce Locator

Anexo A. Documentación de la aplicación crawler desarrollada

La aplicación desarrollada tiene el objetivo de recopilar direcciones de páginas web situadas en la Dark Web. Para ello, dada una dirección inicial, visita la página correspondiente y extrae los enlaces a otras páginas que también se encuentren en la Dark Web. Después, puede ir visitando de la misma forma todos los enlaces recopilados.

A continuación, se exponen los requisitos de la aplicación, el diseño y algunas partes del código.

REQUISITOS

Se proponen los siguientes requisitos funcionales y no funcionales para la aplicación.

RF1: Obtener los enlaces a páginas de la Dark Web existentes en una página web situada en la Dark Web

RF2: Visitar los enlaces obtenidos y obtener sus enlaces

RF3: Recolectar los enlaces que no funcionan

RNF1: Exportar los resultados a archivos de texto

ANÁLISIS

Para poder analizar el contenido de las páginas web necesitaremos disponer del código html que las constituye en un formato analizable. Para obtener el código html la aplicación debe contar con un módulo que sea capaz de realizar las peticiones al servidor y recuperar el código. Estas peticiones deben ser dirigidas a la red TOR.

A continuación, otro módulo deber recorrer el código en busca de enlaces, identificados con las etiquetas html correspondientes. Los enlaces obtenidos se deben almacenar para su posterior análisis.

Una vez finalizado el análisis, los datos almacenados en memoria deben se exportados a archivos de texto para su persistencia.

En la figura 37 se muestra un diagrama de secuencia en el que se representa el trasiego de direcciones entre las listas de links pendientes de visitar, `LinksPendientes`, links visitados

con éxito, LinksVisitadosOk, y de links que no han respondido a la petición, `LinksNoResponden`.

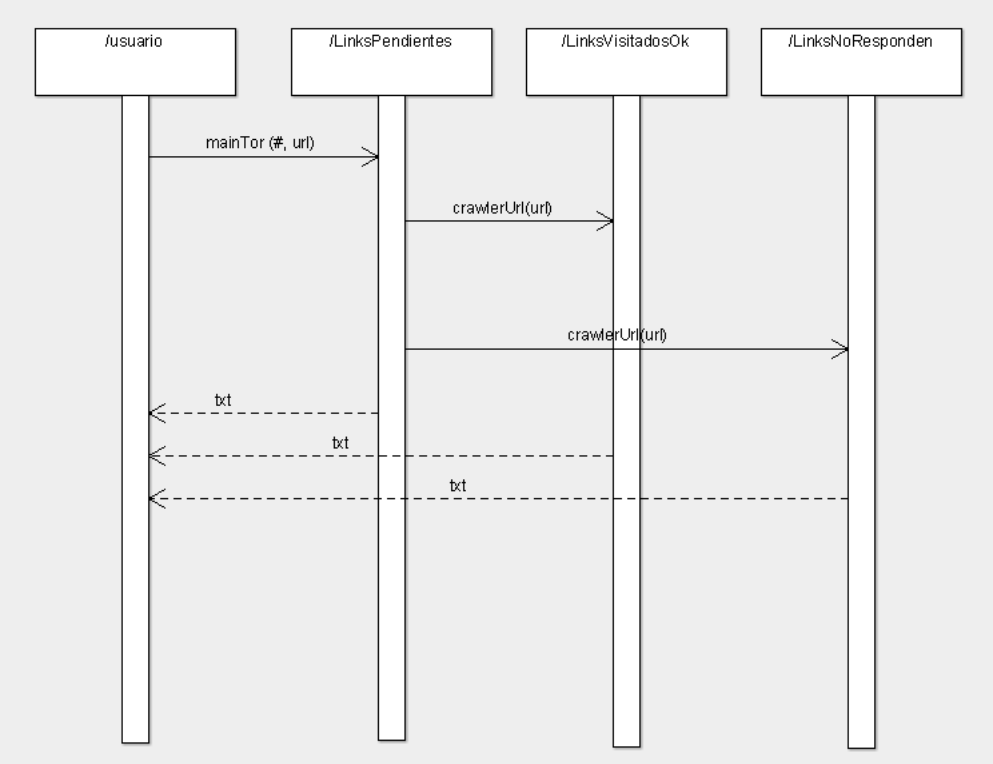

Figura 37: Intercambio de links entre las distintas listas. Fuente: autor.

DISEÑO

Para la construcción de la aplicación se ha decidido utilizar el lenguaje de programación Python en el sistema operativo Linux con el sistema TOR instalado. TOR proporciona un proxi para dirigir las peticiones http a la red TOR.

Python cuenta con la librería `requests` que permite la realización de las peticiones http de manera muy sencilla. Además, permite dirigir estas peticiones a través del proxi de TOR.

El resto de las funciones de la aplicación se han escrito para el presente trabajo.

En la figura 38 se muestran las interacciones entre las distintas funciones que ejecutan las acciones del programa.

`CrawlerUrl` llama a `webToStringTOR` que realiza la petición HTML y convierte la respuesta a string. A continuación, llama a `findLinks` que analiza el código HTML en

formato string en busca de links. Para ello llama a, `findLinkStart`, `findLinkEnd`, `sliceLink` y `linkToList`.

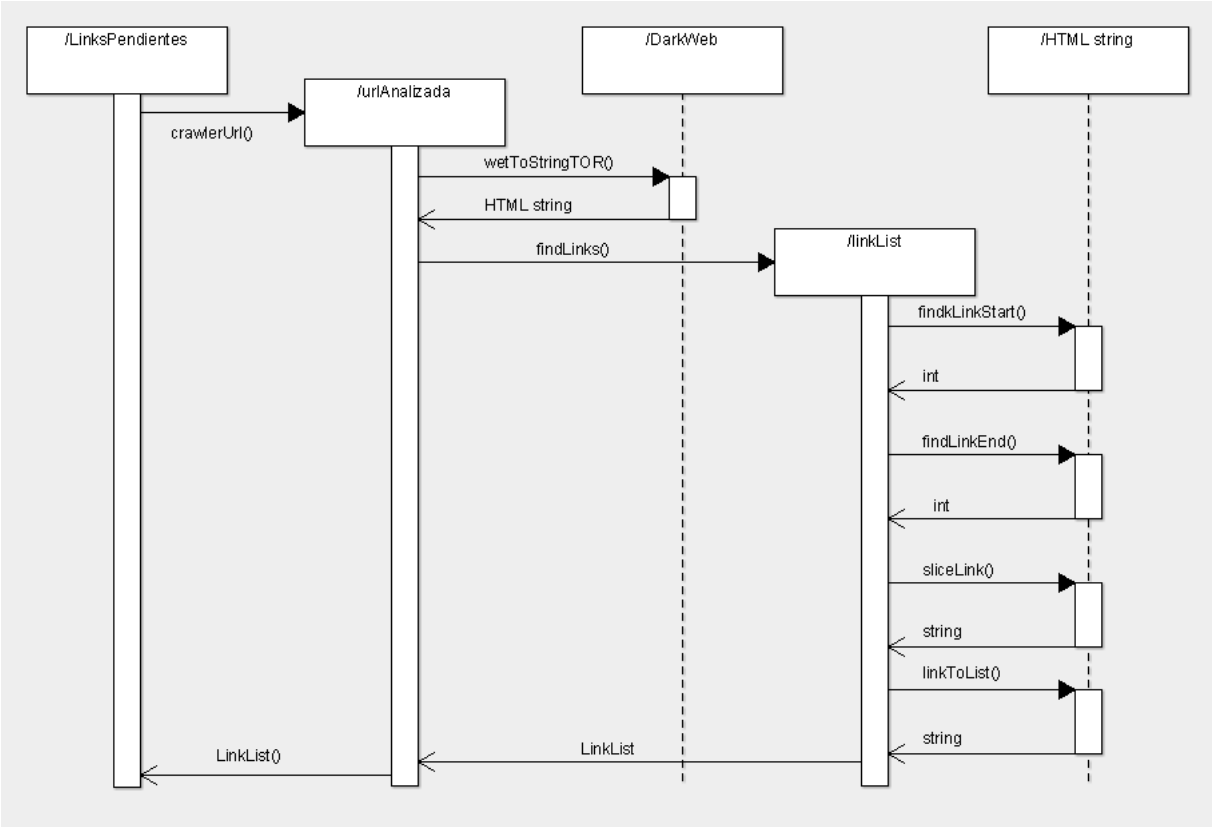

Figura 38: Interacciones entre las distintas funciones que forman parte de la aplicación. Fuente: autor.

A continuación, se resume su funcionamiento y la coordinación entre las distintas funciones:

`crawlerUrl`

Esta función es una interfaz que permite realizar la llamada desde el control principal del programa (main) de una manera más limpia.

Agrupa las llamadas a las funciones `webToStringTOR` y `findLinks`.

`webToStringTOR`

Esta función es la que utiliza la librería `requests` de Python para configurar el uso del proxi de TOR y realizar la petición http a la url indicada. La respuesta recibida es convertida al tipo string.

findLinks

Esta función recibe el código de la página almacenado en una variable de tipo string y lo analiza en busca de enlaces. Para ello utiliza las funciones `findLinkStart`, `findLinkEnd` y `sliceLink`. Los enlaces localizados se van almacenando en una lista usando la función `linkToList`.

findLInkStart

Se encarga de recorrer el código de la página html, a partir de una posición dada, en busca de la marcha (`href")` que indica el comienzo de un enlace.

findLinkEnd

Esta función busca la marca de final de enlace (") que se encuentra justa a continuación de la marca de inicio de enlace localizada por `findLinkStart`.

sliceLink

Utilizando las posiciones inicial y final proporcionadas por `findLinkStart` y `findLinkEnd`, esta función extrae la cadena con el enlace.

linkToList

Esta función es la encargada de almacenar el enlace proporcionado por la función `sliceLink` en una lista. Previamente comprueba que se trata de una enlace a la Dark Web buscando la subcadena ".onion" en él.

listToTxt

Para almacenar los enlaces obtenidos en el disco duro esta función utiliza la librería `pathlib` para generar las rutas de los archivos txt en los que se exportará cada una de las listas de enlaces. Cada archivo se marca con un código de tiempo que lo distingue de otros archivos generados en ejecuciones previas de la aplicación.

codigoTiempo

Proporciona un código tipo string de la forma AAAAMMDD HHMMSS.

PYHTON DOC

A continuación, se muestra la documentación generada automáticamente por Python extrayendo y dando formato a los comentarios iniciales de las funciones:

codigoTiempo()

```
genera un código de tiempo con la forma AÑOMESDÍA HORAMINUTOSEGUNDO

Parameters
----------

Returns
------
string con el código generado
```

crawlerUrl(url)

```
Inicia el análisis de una determinada página en busca de enlaces

Parameters
----------
url : string
    dirección de la página que se desea analizar

Returns
-------
    Lista con los links extraídos de la página
```

findLinkEnd(webString, initialLink)

```
Localiza la posición de la etiqueta de final de Link (") posterior a
una posición initialLink

Parametes
---------

webString : string
    el string del código html de la página
initialLink : int
    posición en la que se ha localizado un inicio de etiqueta link y
se ha de comenzar a buscar el final

Returns
-------
int con la posición en la que finaliza la etiqueta link
```

findLinkStart(webString, initialPos)

```
Localiza la posición de una etiqueta de inicio de Link <a href... a p
artir de la posición initialPos dada
Parametes
---------

webString : string
    el string del código html de la página
initialPos : int
    posición en el string a partir de la cual se ha de comenzar a bus
car

Returns
-------
int con la posición en la que comienza una etiqueta de enlace
```

findLinks(webString, url)

```
Obtiene una lista con los links de la página
Parametes
---------

webStrign : string
    el string del código html de la página
url: string
    la dirección de la página visitada

Returns
-------
lista con los links encontrados
```

linkToList(link, linksList, url)

```
Filtra y añade el link a la lista

Parametes
---------

link : string
    link que se va a añadir a una lista
linkList : list
    lista a la que se quiere añadir un link
url : string
    dirección de la página de la que procede el link para no añadir d
uplicados
    #PENDIENTE DE IMPLEMENTAR

Returns
-------
list con la dirección añadida a su contenido previo
```

listToTxt(linkList, name, lineaInicial='Lista de links:')

```
exporta una lista a un txt

Parameters
----------
linkList : any
    lista que se desea exportar
name : string
    nombre para el txt
lineaInicial: string
    Primera línea explicativa del archivo
Returns
------
    nothing
```

sliceLink(webStrign, initialLink, endLink)

```
Obtiene una subcadena con el link entre dos posiciones int

Parametes
---------

webString : string
    el string del código html de la página
initialLink : int
    posición en el string a partir de la cual se ha de comenzar corta
r
endLink : int
    posicón el el string en la que se ha de terminar de cortar

Returns
-------
string con el enlace extraído
```

webToString(url)

```
Conecta con el servidor, solicita página y la codifica como string

Parametes
---------

url : string
    Dirección de la página que se desea solicitar y convertir en stri
ng

Returns
-------
    string con el código html
```

webToStringTOR(url)

```
Conecta con el proxi de TOR, pide página y devuelve el contenido en u
n string

Parametes
---------

url : string
    Dirección de la página que se desea solicitar y convertir en stri
ng

Returns
-------
    string con el código html
```

En la figura A3 se muestra el cógido ge la función webToStringTOR(). Cabe destacar la primera sección con un comentario formateado para la lectura generación automática de la documentación de Python. A continuación, se crea la sesión que permite encauzar el tráfico de la petición a través del proxi de TOR. Por último, se realiza la petición y se transforma la respuesta en un string codificado en utf-8.

```python
def webToStringTOR(url):
    """Conecta con el proxi de TOR, pide página y devuelve el contenido en un string

    Parametes
    ---------

    url : string
        Dirección de la página que se desea solicitar y convertir en string

    Returns
    -------
        string con el código html"""

    session = requests.session()
    session.proxies["http"] = "socks5h://localhost:9050"
    session.proxies["https"] = "socks5h://localhost:9050" #en realidad las peticionse suelen ser siempre http

    #petición
    respuesta = session.get(url)
    respuesta.encoding = 'utf-8'
    return(respuesta.text)
```

Figura 39: Código de la función webToString(). Fuente: autor.