UN PASEO POR LA DEEP WEB

Mario Lobo Romero

MISTIC
Director: Jorge Chinea
INCIBE

Universitat Oberta de Catalunya
Enero 2018

"*Mucha gente pequeña,*
en lugares pequeños,
haciendo cosas pequeñas,
puede cambiar el mundo"
(Eduardo Galeano)

A mi Familia, mis amigos y mi compañera a la distancia…
Mi sustento…

ABSTRACT

This document seeks to present a technical and inclusive perspective to some of the interconnection technologies developed in DeepWeb, first from a theoretical point of view and then with a brief practical introduction.

The demystification of the processes developed under the DeepWeb provides tools to the users to clarify and build new paradigms of society, knowledge and technology that contributes to the responsible development of this type of networks and helps to the growth of the still young community of enthusiasts behind this world.

RESUMEN

Este documento busca presentar una mirada técnica e inclusiva a algunas de las tecnologías de interconexión desarrolladas en la *DeepWeb*, primero desde un punto de vista teórico y después con una breve introducción práctica.

La desmitificación de los procesos desarrollados bajo la *DeepWeb*, brinda herramientas a los usuarios para esclarecer y construir nuevos paradigmas de sociedad, conocimiento y tecnología que aporten al desarrollo responsable de este tipo de redes y contribuyan al crecimiento de la aún joven comunidad de entusiastas detrás de este mundo.

Tabla de Contenidos

Tabla de Figuras

1. Introducción a las redes anónimas

Hoy nos encontramos ante un mundo lleno de peligrosos contrastes. Por un lado, en la sociedad se desarrolla una tendencia compulsiva y acelerada hacia la interconexión, todos somos clientes y productores de noticias en tiempo real y el concepto de verdad está siendo fácilmente inducido. Aquí pocos detalles se escapan a la divulgación y el escrutinio público y los gobiernos aprovechan las frágiles barreras de la privacidad que hemos interpuesto para perfilarnos y estudiarnos.

Paradójicamente, mientras por un lado existe un exceso de información, en otras latitudes hay un control extremo. Aún se mantienen sistemas de gobierno que coartan a sus ciudadanos de recibir información libremente y les niegan su derecho de pensar y opinar distinto, así como de comunicarlo al resto de la humanidad.

Mucho se habla de las redes anónimas, de su necesidad y de los peligros que encierran. La *Deep Web* se ha convertido en un término de moda entre aficionados, profesionales y especialistas en tecnología que buscan un mayor nivel de privacidad o anonimato. Pero, más allá de la fantasía y las historias que envuelven crimen organizado, redes de traficantes, grupos terroristas, etc. De manera técnica y documentada poco se sabe sobre realmente de estas redes y de los servicios que puede prestar a la sociedad.

Para entender la Deep Web, el primer paso es entender la Web (www), o la idea general que se tiene de ella. La Internet, de manera genérica y universalmente aceptada, se define como una "red de redes". Internet fue concebida para comunicar redes distantes y compartir datos entre equipos terminales, en un principio militares, pero con el paso del tiempo de casi todos los individuos del planeta.

En el transcurso de este documento se intentará aclarar algunos conceptos relacionados con las tecnologías detrás de la Deep Web, desmitificar los preconceptos creados alrededor de su uso y alertar sobre los peligros reales para el usuario.

1.1 Historia de Internet

Las primeras nociones de lo que podría llegar a ser Internet comenzaron en los años 60s en el MIT[1]. En cabeza del sicólogo J.C.R. Licklider se desarrolló la idea de "Red galáctica". Este visionario ideó esta red como "el medio principal y esencial de interacción normativa para gobiernos, instituciones, corporaciones e individuos[2]", Una idea muy aproximada a lo que es Internet hoy en día.

Licklider pasó a ser director de la agencia estadounidense DARPA[3], allí, a partir de los conceptos de "Red Galáctica" y de importantes desarrollos en las comunicaciones usando paquetes (conmutación de paquetes), de científicos como Leonard Kleinrock y Lawrence G. Roberts, entre otros, se construyó la antecesora de la Internet actual, ARPANET.
Esta Internet arcaica terminó de afinarse teóricamente en 1968 y en 1969 se puso en marcha enlazando solo 4 nodos: UCLA[4], SRI[5], UC Santa Bárbara y la Universidad de Utah. En los años siguientes se incorporaron muchísimas más computadoras de centros de investigación, laboratorios, centros

1 Massachusetts Institute of Technology

2 (Radical Evolution: The Promise and Peril of Enhancing Our Minds, Our Bodies -- and What It Means to Be Human)

3 Defense Advanced Research Projects Agency

4 University of California - Los Angeles

5 Standford Research Institute

militares, etc. Durante este periodo se crearon los primeros protocolos de comunicación efectiva host-host y ya se buscaba la forma de listar los servicios de cada nodo de ARPANET para el mejor aprovechamiento de la red.

Como parte de los mecanismos de aprovechamiento de la red, se buscó desarrollar un programa que permitiera la coordinación efectiva entre el personal de cada nodo conectado, allí surgió el correo electrónico en 1972 de la mano de Ray Tomlinson. Si bien el concepto de correo electrónico ya se había usado antes para transmisión de mensajes entre casilleros, Tomlinson consiguió reunir varias de las principales características, agregar unas nuevas como las direcciones con el universal "@", indexar, leer, escribir y hacerlo funcional en la nueva red. La Internet moderna comenzaba a tomar forma.

Uno de los puntos clave para que Internet tuviera el éxito rotundo que tiene hoy en día, fue la introducción del concepto de red abierta. Una red abierta, permite la escalabilidad sin depender de la topología o infraestructura de sus nodos, además, los protocolos, interfaces y estándares no son de ningún propietario[6], están disponibles para el desarrollo de la comunidad en general. Robert Kahn introdujo este concepto a mediados de 1972 como parte de uno de sus proyectos en DARPA, lo llamó "*Internetting*"[7]. Para que este precepto se pudiera cumplir en la red apenas naciente, Kahn diseño junto con Vint Cerf un protocolo que permitiera llegar tanto a las maquinas directamente conectadas, como a las subredes que podrían eventualmente ir naciendo en cada nodo. Así surge la idea del protocolo TCP/IP como estándar de transporte de paquetes en toda la red, iniciando sus pruebas desde 1973 para su implementación definitiva en 1983.

En el transcurso de este documento se intentará abordar de forma teórica y práctica, algunos conceptos un poco más profundos sobre la Deep Web y su contenido, establecer la diferencia con otras redes, aclarar algunos mitos e historias creadas entrono a ella, para de esta manera, aproximar al lector a su uso en diferentes plataformas, con diferentes clientes, y contribuir al aprovechamiento que pueda hacer de esta red.

1.2 Diferenciación de las redes

1.2.1 Surface web

En el proceso de evolución de Internet, surgió un momento crítico que marcó para siempre la historia de la red y de la sociedad en general, este hito ocurrió en marzo de 1989 con la propuesta por parte de Tim Berners-Lee[8] de lo que sería la *World Wide Web* -desde ahora- (WWW) y sus conceptos de hipertexto e indexación del contenido, para que pudiera estar disponible a todo el público. Esta tecnología se convierte en la piedra angular para entender la diferencia entre la *Surface Web* y la *Deep Web*.

Con este nuevo sistema de indexación y el creciente número de computadores personales entrando en el mercado, las posibilidades de expansión de Internet aumentaban considerablemente. El paradigma del tamaño y costo de los computadores se rompió y cada individuo podría tener uno en su casa u oficina. Internet estaba lista para que todas esas pequeñas fuentes de información se unieran y compartieran sus datos, pero con esta escalada de nuevos miembros y ráfagas de información

6 http://www.umich.edu/%7Earchive/linguistics/bigdummysguidetotheinternet
7 Internet: A Historical Encyclopedia, Volume 2
8 Berners-lee, Tim. Weaving the Web, The original design and ultima destiny of the world wide web.

desordenada, comenzó a requerirse algún sistema para poder encontrar la información exacta en el lugar correcto y de esta manera optimizar el uso de la red.

Esta nueva necesidad trajo consigo la creación de un nuevo termino, motor de búsqueda. Archie[9] para muchos se instauró como el primer motor de búsqueda en la historia, o por lo menos el primero usado masivamente, indexaba archivos descargables vía FTP y guardaba sus vínculos en una base de datos que permitía la consulta para los usuarios interesados. Los primeros motores de búsqueda seguían el principio de buscar palabras clave dentro de los sitios almacenados, guardarlas, clasificarlas de acuerdo a la cantidad de veces que se repetía esta palabra, y ofrecer una interfaz amigable para que el usuario pudiera encontrar lo que deseaba de acuerdo a esos criterios.

Conforme crecía la información en la WWW los motores de búsqueda cobraban mayor relevancia, a mediados de los noventa algunos lograron gran notoriedad como *Lycos, Altavista, Hispavista, AOL* o Excite. Casi todos con patrones de búsqueda similares y con poca diferenciación, exceptuando la cantidad de sitios indexados que manejaba cada uno.

Fue hasta agosto de 1996 que en la Universidad de Standford Sergey Brind y Lawrence Page tuvieron la idea de diseñar un motor que exhibiera los resultados de acuerdo a un sistema de jerarquías y rankings, nació en la academia *BackRub*[10], semilla del motor que 2 años más tarde se convertiría en *Google*. Con la llegada de *Google*, el panorama de los motores de búsqueda cambio drásticamente, pasaron de ser simples indexadores a ser clasificadores y analistas de un sin número de información de todos los usuarios de la WWW y a cobrar por ello.

Muchas aplicaciones y sitios, comenzaron a utilizar estos métodos para recopilar mayor información acerca de sus usuarios. Se comenzaron a crear perfiles de los visitantes, sus gustos, tendencias, preferencias, discrepancias, influencias, etc. Eran registradas. Ahora las campañas publicitarias podían enfocarse en objetivos mucho más específicos, y la información podría dirigirse a grupos más interesados en ella; se podría individualizar el receptor. Con la recopilación de esta información, la privacidad comenzó a presentarse como una preocupación para los usuarios.

Paralelo al desarrollo de los buscadores, las redes sociales fueron ganando espacio, en 1994 nació *GeoSites* considerada la primera de su tipo en llegar a un considerable número de usuarios. Con el tiempo fueron llegando sitios como *AOL* y más famosos como *Myspace*, muy cercano al concepto de red social actual.

La explosión definitiva de las redes sociales llegó en febrero del 2004 con la publicación de *The Facebook*. Esta red, de 2004 hasta la fecha ha logrado recolectar más de 2 billones de usuarios[11], y una cantidad inigualable de datos sobre ellos. Con el éxito de *Facebook* se marcó una tendencia desbordada a la recolección de información que los suscriptores voluntariamente entregan en varias redes. *WhatsApp, Instagram, Twitter, LinkedIn*, son solo algunos ejemplos.

Como se puede apreciar, todo este universo de datos en la WWW sobre un sinnúmero de temas e individuos, se encuentran públicos, con ciertos filtros de privacidad, pero indexados en bases de datos gigantescas que alimentan lo que denominamos la "*Surface Web*". Un conjunto de sitios, bases de datos, servicios, mensajería y ecosistemas sociales dispuestos para el acceso y el conocimiento del

9 http://archie.icm.edu.pl/archie-adv_eng.html
10 http://infolab.stanford.edu/~backrub/google.html
11 https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/

público en general, gobiernos, proveedores de servicio, cuerpos de seguridad, sistemas de mercadeo, etc.

Quien interactúa en la *Surface Web* no tienen ningún problema en exponer un poco, o mucho de su privacidad, de acuerdo a sus hábitos de navegación, a cambio de información que pueden encontrar con mayor facilidad navegando un poco. Todo aquello que pueda ser indexado y mostrado se encuentra en este espacio dentro de Internet.

La información en la *Surface Web* está soportada sobre Internet, utiliza su infraestructura, sus protocolos y estándares de comunicaciones. Es una red más dentro de la "Red de Redes", pero no la única.

1.2.2 Deep Web

Los inicios de la *Deep Web* se remontan a la creación misma de ARPANET. En los años 70's se comenzó a denominar redes profundas u oscuras (*Deep* o *Dark* en la época no se había establecido una diferencia puntual) a algunas redes que recibían datos de ARPANET pero que por motivos de seguridad no aparecían en las listas proporcionadas a todos los nodos, lo cual no permitía su localización y las mantenía en las "sombras".

Definimos como *Deep Web* todo aquel contenido de Internet: redes, sitios, bases de datos, mensajería, sistemas de intercambio de ficheros, etc. Que por voluntad propia o como consecuencia de alguna tecnología aplicada, no permite que sus contenidos sean indexados en los motores de búsqueda de la *Surface Web*. La *Deep Web* no es una red física separada, si no una capa de aplicación y protocolos montada sobre las redes existentes[12]. La *Deep Web* también es parte de Internet.

Esta red no necesariamente almacena solo información sensible, secreta o cifrada. Desde el inicio de Internet bases de datos académicas, bibliotecas de artículos especializados, grupos de lectura, grupos de hacking, activistas o simplemente individuos con intereses específicos utilizaron tecnologías diferentes a las de la WWW para alojar y compartir sus contenidos con usuarios determinados, lo que la hace inalcanzable para un usuario común de la *Surface Web*.

Con el crecimiento exponencial de la WWW se cambió trascendentalmente la forma de concebir las relaciones sociales y el funcionamiento de la sociedad en general. La vida cotidiana se trasladó a un espacio donde los mayores protagonistas del esquema social debían confluir y la privacidad comenzó a jugar un papel fundamental para los usuarios.

Esta nueva concepción de la sociedad mitificó en cierto sentido el papel de la *Deep Web*, debido a su anonimato, y la rotuló como un espacio criminal de facto, sin advertir las bondades que puede ofrecer para grupos especializados que quieren tener un filtro mayor para sus visitantes.

1.2.3 Dark web

Ante la inminente migración de la sociedad a un nuevo paradigma digital, los gobiernos, cuerpos de seguridad, sistemas de mercadeo, criminales, etc, tuvieron que reaccionar y acoplarse a las mudanzas.

12 http://msl1.mit.edu/ESD10/docs/darknet5.pdf

El activo más importante de la sociedad actual se encontraba ahora por cantidades desbordadas en la WWW, de manera relativamente fácil de recopilar y analizar con los medios adecuados. La posibilidad de supervisar directa o indirectamente la información o el actuar de los usuarios; llámense ciudadanos, clientes o víctimas, delega un poder desproporcional que comenzó a hacerse visible y con el que los actores involucrados se empezaron a preocupar y a tener mayor cautela.

Como se puede intuir, no todo el mundo está conforme en el *"statu quo*", o simplemente no están dispuestos aceptar la posibilidad de entregar un poco de su privacidad a cambio de algo. Gobiernos de muchos países censuran la información y bloquean los accesos a medios de comunicación. Así es que quien quiera anular o por lo menos disminuir la posibilidad de ser monitoreado, encontrado o indexado, debe diseñar algún sistema, llámese red o tecnología, que permita cubrir con una o varias capas la información que se transporta por Internet, la forma de transportarla y quién o quienes la transportan. Esto llevó a la fama a la *Dark Web* y disparó su desarrollo.

Un concepto que es necesario clarificar en este escenario es el de las *DarkNets.* Aunque es parte de la *DeepWeb*, la *DarkWeb* está formada por redes exclusivas cuyo acceso se encuentra restringido a aplicaciones específicas creadas para cada una de ellas, estas redes son conocidas como *DarkNets*. Se deben diferenciar estos dos conceptos que si bien son cercanos, no son iguales. Tor, I2P, FreeNet, Resilio, GnuNet, etc. son *DarkNets*.

La *Dark Web* se desarrolló en paralelo con Internet. Hace parte de la *Deep Web* ya que su contenido tampoco se puede indexar, pero agrega un punto de complejidad que la diferencia; La capa de aplicación donde se desarrolla implementa sistemas de cifrado y enrutamiento particulares y robustos, cuya finalidad es propender por el anonimato de sus miembros y la confidencialidad de los datos que intercambian.

La aparición del concepto de *Onion Routing* en 1996[13], clarificó la posibilidad de tener un escenario real lejos de la censura, supervisión o monitoreo del tráfico existente en Internet. Se implementó un sistema basado en nodos para anonimizar el flujo de información que pasa a través de ellos, a partir de esta tecnología nació Tor.

Desafortunadamente en los últimos años esta red ganó notoriedad en los medios gracias a los mercados negros que se desarrollan aprovechando su infraestructura, el caso de Skil Road fue icónico y encendió las alarmas de las autoridades quienes comenzaron a buscar maneras de monitorear lo que sucede en estos terrenos.

Pero no es nuevo el uso criminal que se le da a las redes, según Jamie Bartlett en su libro The Dark Net[14], el primer caso de lo que podría ser una transacción de tráfico de drogas se llevó a cabo en 1972 por parte de algunos estudiantes del *MIT* y La Universidad de *Stanford* usando la cuenta de ARPANET del laboratorio de inteligencia artificial para intercambiar Marihuana. El delito es innato al usuario e independiente de la red donde se desarrolle.

Afortunadamente la *Deep Web* ofrece un gran abanico de posibilidades más allá del Cibercrimen. En el 2008 se presentó Bitcoin por parte de un grupo de trabajo bajo el seudónimo *Satoshi Nakamoto[15]*, y sirvió de base para la explosión de las criptomonedas en el mundo. A partir de allí, se comenzó a dar

13 https://www.onion-router.net/Publications/IH-1996.pdf
14 Barlett Jamie, The Dark Net, Random House, 2014. ISBN 1473506034
15 https://bitcoin.org/bitcoin.pdf

una nueva mirada al concepto de *BlockChains*, esta tecnología promete cambiar de raíz la mayoría de paradigmas de nuestra sociedad, transformar por completo el sistema financiero y desde un punto de vista ambicioso, tal vez hasta nuestro concepto mismo del Estado.

Con la tecnología *Blockchains*, el tercero de confianza o entidad central en las transacciones, es reemplazado por un robusto sistema de cifrado en bloques que se distribuye entre cada miembro, así la transacción se encuentra presente y es avalada por toda la red, pero sin posibilidad de réplica y sin conocer el origen ni el destino de la misma. Este principio ofrece infinitas posibilidades en contrataciones, certificaciones, transacciones, etc. Dota de un poder real a cada usuario y descentraliza el existente en las instituciones normales.

En múltiples conflictos alrededor del mundo se han usado redes anónimas para poder acceder a material censurado por los gobiernos y para hacer saber al mundo lo que ocurre en los países. Periodistas y activistas han encontrado en la *DarkWeb* una forma de evadir los cercos informativos de los regímenes políticos imponen a sus ciudadanos. De acuerdo a la *Tor project Fundation*[16], los países con una mayor censura informativa en el mundo se muestran en la figura 1.

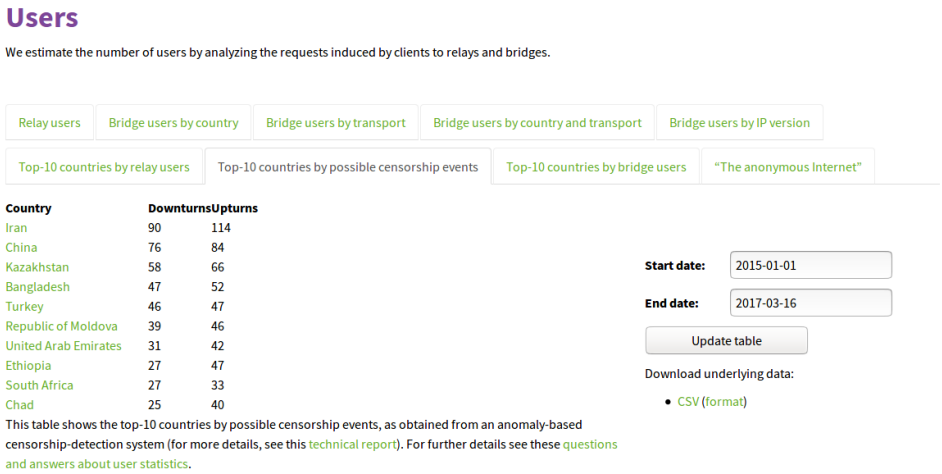


Figura 1. Top países con posible censura informativa

En la actualidad se conocen ya cientos de redes y aplicaciones de la *Dark web* para diversos propósitos, intercambio de archivos, mensajería de texto, correo electrónico, intercambios financieros, grupo científicos, etc. En el siguiente capítulo se abordarán las redes anónimas más usadas y se realizará un acercamiento a las distribuciones y clientes más utilizados.

16 https://metrics.torproject.org/userstats-censorship-events.html

2. Redes Centralizadas (Tor)

Por definición, las redes centralizadas son aquellas que focalizan en algún punto el gerenciamiento de sus tareas o servicios. En este modelo, todas las comunicaciones de un dispositivo a otro pasan por un punto central por medio de rutas o circuitos predefinidos[17].

En este orden de ideas, puede parecer paradójico que una red anónima logre adaptarse a este modelo de trabajo. Si Tor fuese totalmente centralizada supondría saber la ubicación exacta de los servidores y los clientes, pero esto no es así, realmente es una red anónima. ¿Qué pasa entonces? La verdad es que Tor hace un gran esfuerzo por cumplir con los preceptos de una red descentralizada, pero no llega a serlo totalmente gracias a algunos de sus nodos. Estos nodos, si bien son muchos y están distribuidos alrededor del mundo, le dan un hilo común de gerenciamiento a la red. Además, existen instancias llamadas autoridades de directorio encargadas de mantener el listado actualizado de dichos nodos, lo que centraliza en ellas la administración los mismos.

Tor nace a partir de estudios realizados en el Laboratorio de Investigación Naval de los Estados Unidos por David M. Goldschlag, Michael G. Reed, and Paul F. Syverson en el año 1996[18]. En el estudio se describe la forma de anonimizar el tráfico de una red aplicando un concepto de cubrimiento por capas, como las de una cebolla, para ocultar el origen y destino de las comunicaciones. A esta técnica se le llamó *Onion Routing* (enrutamiento de cebolla), de allí el nombre de *The Onion Router (TOR).*

Tiempo después Tor se conformó como una organización sin ánimo de lucro denominada *The Tor Project* quien coordina todo el desarrollo y mantenimiento de la Red Tor. Para el 2002 se lanzó la primera versión del conocido navegador al público.

Es necesario entender el funcionamiento de Tor para saber por qué se le cataloga como una red centralizada y descubrir cómo logra su nivel de anonimato. A continuación, se intentará explicar los principales conceptos sobre esta red.

2.1 Nodos (*Relays*)

Tor se puede describir como una red bastante flexible ya que brinda la posibilidad de usar servicios dentro de la propia red (servicios ocultos) o comunicarse con otras redes (como proxy). Gracias a su tecnología de enrutamiento, usando Tor y casi sin darnos cuenta de la diferencia podemos acceder a buscadores, portales, redes sociales, etc. En la *Surface Web* o navegar en sitios. *onion* de la *Dark web*.

El uso extendido de Tor se realiza a través de la aplicación Tor Browser, esto ya que en principio el tráfico o servicio principal que viaja a través de la red Tor es https. Aun así, Tor no se limita a navegar vía http y permite anonimizar cualquier servicio usando algunas aplicaciones específicas que se describirán más adelante en este documento.

La tecnología de enrutamiento de cebolla, desarrollada para garantizar el anonimato de las partes. Está basada en el uso de nodos distribuidos alrededor del mundo que se comunican entre sí para formar circuitos virtuales. Diferentes tipos de Nodos brindan diferentes servicios en la red *onion*. Algunos

17 Network Design, Second Edition: Management and Technical Perspectives, Teresa C. Piliouras, Taylor & Francis, 2004

18 David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. \Hiding Routing Information," Workshop on Information Hiding, Cambridge, UK, May, 1996.

sirven de proxy de entrada, otros de proxy de salida y otros como nodos de enrutamiento (*relays* o repetidores).

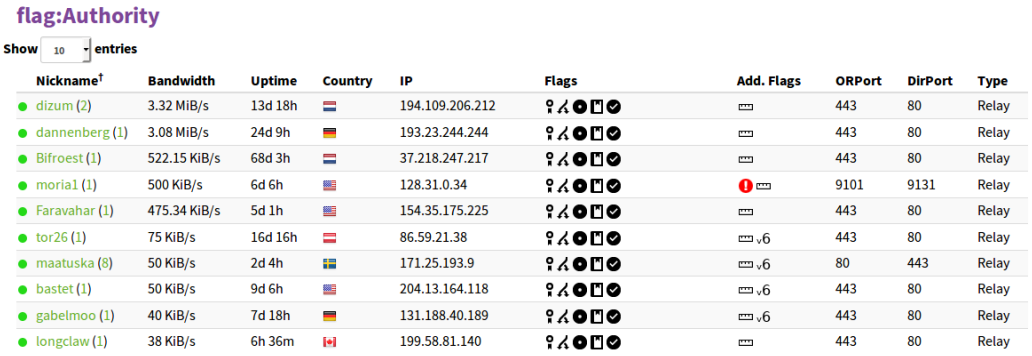
Los nodos de la red Tor están compuestos por voluntarios anónimos de diversas partes del mundo, esto hace parte de las ventajas de la red. Cuanto mayor es la cantidad de nodos que componen la red Tor, mayor es su velocidad, con la complejidad de la red la capacidad de camuflaje también aumenta. En una red robusta es más difícil intentar hallar rutas o analizar el tráfico para identificar algún equipo, por equivalencia la posibilidad de mantener el anonimato es mejor.

Para entender correctamente la diferencia de las tareas de estos nodos, es importante conocer el concepto de proxy o pasarela. Un proxy es un servicio, programa o dispositivo que hace la labor de intermediario y controla el tráfico entre dos equipos[19]. El uso de los *proxies* en las comunicaciones es variado, algunos se utilizan para filtrar contenidos, otros para agilizar la navegación entregando información previamente almacenada y otros como elementos de anonimato reemplazando el dispositivo que realiza la petición original.

2.1.1 Autoridades de Directorio

El motivo por el cual la red Tor se define como una red centralizada es debido a las autoridades de directorio. Estas entidades son nodos especiales que listan y almacenan los nodos presentes en la red, realizan una revisión general y generan un acuerdo de consenso sobre el tipo y la participación de los nodos, que posteriormente es replicada a los clientes.

En la actualidad existen 10 autoridades de directorio[20]. Por defecto Tor Browser tiene almacenadas estas autoridades para evitar que sean suplantadas y usadas para extraer información de la red. Si alguien controla estas entidades, la red puede colapsar.


flag:Authority

Show 10 entries

Nickname†	Bandwidth	Uptime	Country	IP	Flags	Add. Flags	ORPort	DirPort	Type
dizum (2)	3.32 MiB/s	13d 18h		194.109.206.212			443	80	Relay
dannenberg (1)	3.08 MiB/s	24d 9h		193.23.244.244			443	80	Relay
Bifroest (1)	522.15 KiB/s	68d 3h		37.218.247.217			443	80	Relay
moria1 (1)	500 KiB/s	6d 6h		128.31.0.34			9101	9131	Relay
Faravahar (1)	475.34 KiB/s	5d 1h		154.35.175.225			443	80	Relay
tor26 (1)	75 KiB/s	16d 16h		86.59.21.38		v6	443	80	Relay
maatuska (8)	50 KiB/s	2d 4h		171.25.193.9		v6	80	443	Relay
bastet (1)	50 KiB/s	9d 6h		204.13.164.118		v6	443	80	Relay
gabelmoo (1)	40 KiB/s	7d 18h		131.188.40.189		v6	443	80	Relay
longclaw (1)	38 KiB/s	6h 36m		199.58.81.140			443	80	Relay

Figura 2. Autoridades de Directorio

El protocolo de autoridades evolucionó desde sus primeras versiones buscando aumentar la eficiencia de la red, en un principio los descriptores completos eran descargados, para las últimas versiones solo son actualizadas ciertas *flags* con información específica que las autoridades validan con una votación colectiva, después de obtener un consenso son transmitidas a los clientes.

19 Network Design, Second Edition: Management and Technical Perspectives, Teresa C. Piliouras, Taylor & Francis, 2004 p 345

20 https://atlas.torproject.org/#search/flag:Authority

En el transcurso del proceso, las diversas selecciones que se realizan por parte de los clientes están sujetas a la información provista por estos nodos. Si un atacante lograra obtener el control de más de la mitad de las autoridades, podría colapsar y desvelar a los usuarios ya que podría influenciar los consensos e inducir al cliente a realizar conexiones que se pudieran monitorear.

2.1.2 Nodos de entrada (*Entry Guard*)

Cuando un usuario usa el cliente Tor se encuentra con dos opciones: acceder a un servicio dentro de la red Tor (servicio oculto u *onion service*[21]) o utilizar Tor para anonimizar su navegación en la Surface Web. Para cualquiera de estos dos casos, el primer paso es ingresar a la red.

Cuando el usuario ingresa a la red Tor utiliza un "Proxy in" o nodo de entrada (*entry guard*), que básicamente es un nodo que se encarga de llevar las peticiones de los clientes hacia la red Tor. En un esquema normal cuando un cliente realiza una petición de un servicio oculto dentro de la red podemos observar el siguiente comportamiento:

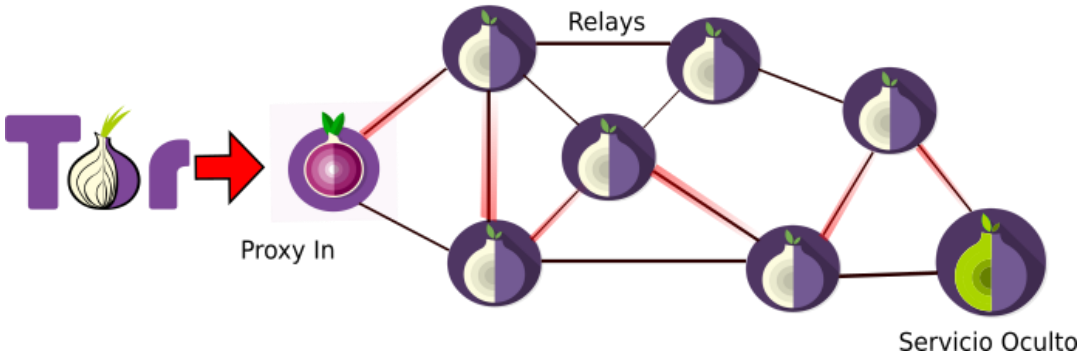


Figura 3. Esquema servicio oculto onion

En la figura 3 se puede observar que el ingreso a la red se realiza a través del nodo de entrada. El cliente antes de iniciar el recorrido del paquete, selecciona el nodo de entrada y escoge algunos nodos intermedios (de los muchos disponibles), para armar un circuito virtual que se convertirá en el camino cifrado por el que se transmiten tanto las peticiones como la información.

El nodo de entrada cumple una labor trascendental en el esquema de funcionamiento de la red Tor; al ser el elemento que tiene contacto directo con el cliente, es el único que sabe realmente su IP. Esta característica lo ha hecho objetivo de diversos ataques desde los inicios de la red. Para disminuir la posibilidad de exposición de los clientes, pasaron a convertirse de nodos normales a *entry guards*, con algunas cualidades específicas y requisitos por parte de la autoridad.

Cuando el cliente solicita la entrada a la red Tor se escoge un grupo de nodos *guard* de la lista dispuesta por las autoridades de directorio, esta selección no se realiza al azar entre todos los nodos, sino únicamente entre la lista de los catalogados con la bandera *guard.* En este momento ese grupo está constituido por 3[22] nodos por cliente. Una vez formado este grupo, el cliente selecciona solo un nodo *guard* para iniciar el circuito virtual; este nodo *guard* no es aleatorio, ni cambia con cada conexión, por

21 https://metrics.torproject.org/glossary.html#onion-service
22 https://consensus-health.torproject.org/consensus-health.html#consensusparams

el contrario, se mantiene estable en un periodo estipulado por el consenso; para la fecha de este escrito está estipulado de 30 a 60 días[23], con 60 como valor por defecto.

Aunque parezca extraño, esta estabilidad en el primer nodo del circuito tiene varias ventajas de seguridad. La primera es que al hacer que la selección sea controlada por el consenso, los nodos de entrada van a permanecer con un balance de carga tal que impida la concentración de todo el tráfico por alguno de ellos, así como alguna sobrecarga. Además, permite que los nodos que se unan como e*ntry guard*, no sean subutilizados por la red.

El mayor problema con el que lidia una red como Tor, es permitir que un atacante controle nodos tanto de entrada como de salida. En este escenario, comparando el tráfico entrante versus el saliente, el atacante eventualmente podría correlacionar coincidencias en la cantidad de flujo de datos y crear perfiles de navegación, esto, acompañado de un análisis concienzudo llevaría a develar la identidad de los clientes.

El hecho de permanecer esa cantidad de días con el mismo nodo *entry guard*, pretende evitar la posibilidad de encontrar un mayor número de nodos maliciosos en cada cambio de circuito. Es claro que esto no elimina la probabilidad de que el nodo que permanezca activo sea malicioso, pero como el resto del circuito virtual si es aleatorio, reduce la posibilidad de encontrar parejas de nodos entrada y salida controlados por el atacante[24].

La disminución de la probabilidad de encontrar un nodo malicioso también está apoyada por el filtro que se realiza para asignar la bandera de *guard* por parte de la autoridad de directorio. Niveles altos de ancho de banda, estabilidad y eficiencia deben caracterizar a estos nodos para ser catalogados como *entry guard.*

2.1.3 Nodos intermedios (*relay*)

El Nodo de entrada es el primero en la estructura del circuito virtual, todos los siguientes antes de llegar al nodo de salida se catalogan como nodos intermedios.

El éxito de la topología de la red Tor se basa en la existencia de gran cantidad de estos nodos dispersos alrededor del mundo, conectados lógicamente entre sí y disponibles para ser parte de un circuito virtual cuando un cliente lo solicite. Un circuito virtual Tor está compuesto mínimo por 3 nodos para garantizar el anonimato, pero actualmente el protocolo está parametrizado para 6 saltos en servicios .onion.

El protocolo de enrutamiento de cebolla permite que cada nodo conozca únicamente quien le entrega el paquete y el siguiente nodo al que lo debe entregar. Esto supone que cuantos más nodos intermedios existan, mayor será la dificultad para intentar encontrar una correlación entre ellos.

2.1.4 Nodos de salida (*exit relay*)

Como se mencionó anteriormente, la red Tor cuenta con sus propios servicios ocultos y también se puede utilizar para acceder anónimamente a servicios provistos en la *Surface Web*, a este esquema de conexión se le denomina "*proxy out*", figura 4.

23 https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt
24 https://www.freehaven.net/anonbib/cache/wright03.pdf

El nodo de salida es el único elemento de la red Tor que tiene contacto directo con el destino, por tanto, es el único que sabe realmente su IP. Debido a esta particularidad, este tipo de nodos han sido blanco de numerosos ataques buscando controlar la mayor cantidad de ellos y así desvelar servicios.

La información que pasa del nodo de salida al destino es vulnerable, esto, debido a que las capas de cifrado se terminaron y las peticiones y cruce de datos entre este nodo y el servidor se realiza directamente. Por esta razón, para asegurar que además de anónima la conexión sea confidencial, se recomienda que, si la información a transmitir es sensible, vaya cifrada.

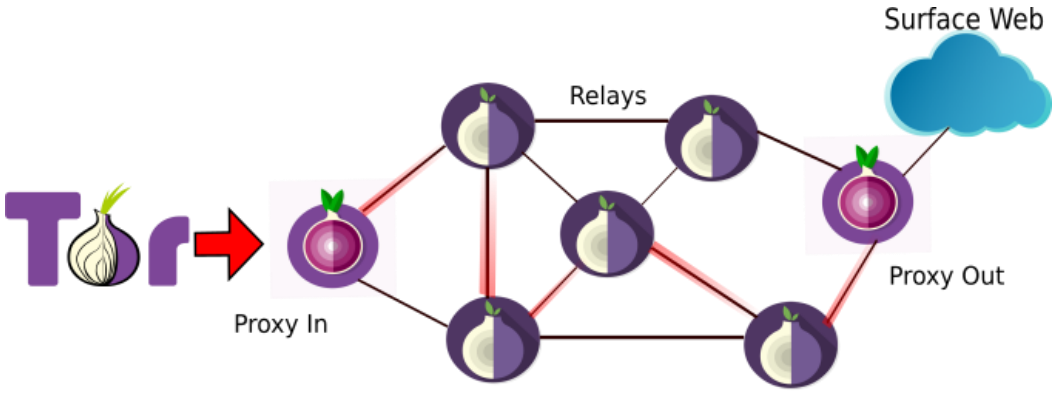


Figura 4. Esquema Proxy out

Convertirse en un nodo de salida conlleva una alta dosis de riesgo y responsabilidad, la comunidad de desarrollo de Tor es consciente de esto y alerta a quien quiera serlo. Como punto de salida de los datos, si por allí circula contenido relacionado con algún crimen de cualquier tipo, el primer dispositivo sobre el que recaerá alguna investigación será el nodo de salida.

2.2 Circuitos

Para poder ser transmitida a través de la red Tor, la información pasa por un proceso de empaquetamiento y cifrado por capas. Al momento de armar un paquete *onion* (célula), se cifra cada capa con una llave perteneciente a cada uno de los nodos participantes en el circuito; en orden: desde el último hasta el primero.

Todas las conexiones en la red Tor están cifradas por medio de llaves TLS efímeras, que aparecen de acuerdo a la formación de los circuitos. La creación de los circuitos se hace de forma incremental, así el cliente va enviando peticiones *create cell* para negociar las llaves y recibe adherida una capa nueva en cada turno, este proceso se repite hasta completar el circuito.

En la figura 5 se puede observar una analogía de una célula *onion.* Antes de ser enviada, la información es cifrada con la llave del *relay* D, después con la del C, luego la del B, y por último del A, es decir el primer nodo. El mensaje está en el centro de todas las capas.

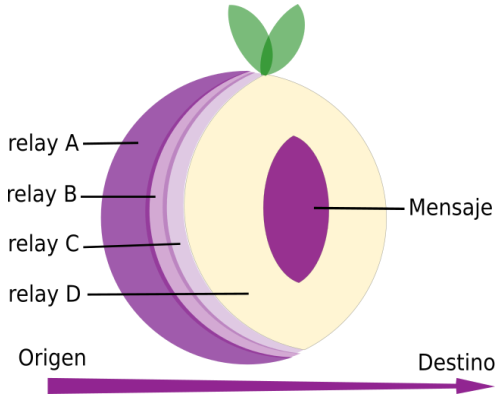


Figura 5. Analogía célula onion

Al llegar la célula *onion* al *relay* A, este descifra su capa y entrega el resto del paquete al *relay* B. El *relay* A únicamente sabe que el paquete viene del cliente y va ser entregado al *relay* B, así sucesivamente el B sólo sabe que viene del A y va para el C, y el C sabe que viene del B y va para el D. Esta lógica se repite hasta el último nodo del circuito, que es el único que conoce el destino.

En este orden de ideas, el último nodo del circuito conoce la IP del destino, pero nunca la del origen, solo la del nodo anterior. El nodo de entrada conoce la IP del cliente, pero nunca la del destino, únicamente la del nodo siguiente. De este modo se cumple la premisa de anonimato durante toda la conexión.

Una célula *onion* tiene un tamaño fijo de 512 bytes, este tamaño no varía nunca durante la conexión. A pesar de retirar las llaves de cifrado correspondientes en cada nodo, el tamaño no cambia ya que este espacio es rellenado con bits aleatorios y enviado al siguiente nodo. La intención al realizar esta tarea, es evitar que se relacione el tamaño del paquete con su lugar dentro de la conexión.

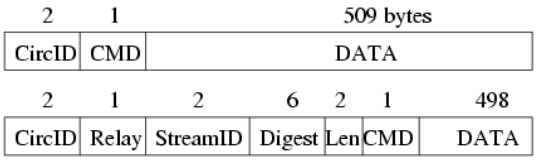


Figura 6. Célula Onion

Existen dos tipos de células *onion* diferenciadas según el contenido del campo CMD, este comando indica si la célula es de control, las cuales son interpretadas por todos los nodos, o si son del tipo *relay* y llevan datos a través del circuito. Estas últimas son interpretadas únicamente por el nodo *relay* que tiene las llaves.

El campo CircID corresponde al identificador de circuito; existe ya que por una misma conexión TLS[25] pueden transitar paquetes de varios circuitos, este campo es necesario para distinguir cada uno de ellos y es común para todas las células *onion*.

Como se observa en la figura 6, las células *relay* tienen algunos campos adicionales, conocidos como campos de retransmisión:

- *StreamID*: Identificador de Flujo, para saber el orden de transmisión.

25 Transport Layer Security

- *Relay*: Un token de retransmisión.
- *Digest*: Suma de comprobación.
- *Len*: La longitud de la carga útil.

Cada vez que la célula es procesada por un nodo *relay*, tanto el encabezado de retransmisión como la carga útil son cifrados nuevamente, utilizando AES-128[26], y enviados al siguiente nodo, hasta llegar al nodo de salida.

2.3 Servicios Onion (Servicios Ocultos)

En principio la red Tor está diseñada para manejar tráfico del tipo TCP[27] https, ya que su sistema de cifrado entre nodos es del tipo TLS[28] y los puertos por defecto son para servicios web, incluso su principal cliente es el Tor Browser. Pero esto no restringe a la red de poder llevar cualquier otro servicio TCP como SSH o FTP utilizando las herramientas adecuadas para encaminarlo.

Un servicio *onion* antes llamado servicio oculto se define como aquel que únicamente es accesible por medio de la red Tor[29]. Como se comentó antes, se supone que la mayoría de servicios *onion* en la red Tor son sitios web, pero como el tráfico siempre es del tipo TLS no hay certeza de ello. El número de servicios ocultos cada vez es mayor, en la figura 6 se puede observar los datos de los últimos 3 meses a la fecha de este escrito, se superan los 2 Gbit/s de tráfico de estos servicios.

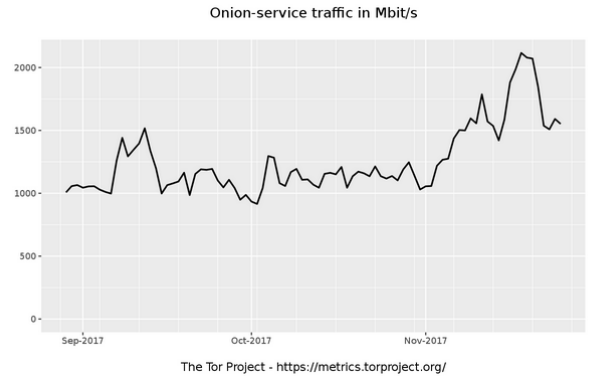


Figura 7. Tráfico Onion Services

Algunos de estos servicios se utilizan con fines delictivos, pero muchos otros son utilizados en países donde la libertad de expresión es mínima, también por grupos perseguidos con ideas políticas contrarias a los regímenes, grupos que realizan investigaciones confidenciales o simplemente por personajes cansados de la poca privacidad en la *surface web*.

En general, para crear un servicio oculto se debe primero crear un servicio normal. Por ejemplo, si se quiere publicar una web oculta, entonces el primer paso será crear un servidor apache y su contenido de la misma manera como se publicaría en la *surface web*.

26 https://svn.torproject.org/svn/projects/design-paper/tor-design.html
27 Transmission Control Protocol
28 http://www.rfc-base.org/rfc-5246.html
29 https://metrics.torproject.org/glossary.html#onion-service

Una vez se tiene el servicio corriendo en la maquina local, se le debe indicar a Tor que el servicio que está activo en el puerto 80, por ejemplo, se debe procesar como un *onion service.* Para esto, se agregan dos líneas al archivo de configuración *torcc* que contienen el HiddenServiceDir y HiddenServicePort.

Si se publicara este servicio en la *surface web* debería asignársele un dominio, es decir, un nombre con el que sería identificado en Internet. Este nombre es consultado por el cliente en los DNS[30] y ellos retorna una dirección IP para realizar la conexión cliente-servidor. Para la red Tor existe algo parecido, una analogía mucho más elaborada llamada dirección .onion.

La dirección .onion es un identificador o *hostname* formado por 16 caracteres alfanuméricos seguidos de la extensión .onion, ejemplo: xmh57jrznw6insl.onion (Buscador Torch).

La creación de una dirección incluye varios pasos de cifrado[31]. Para poder publicar un servicio, se debe crear un par de llaves RSA-1024[32], una vez creadas, Tor calcula el hash SHA-1[33] de llave pública usando DER *encoding*, el valor resultante tiene 160 bits de los cuales se toma la primera mitad (80), por último, se le aplica una función Base32[34] para obtener un hash con 16 caracteres.

No todos los caracteres y números están disponibles en el alfabeto utilizado para la codificación Base32.

Table 3: The Base 32 Alphabet

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	9	J	18	S	27	3
1	B	10	K	19	T	28	4
2	C	11	L	20	U	29	5
3	D	12	M	21	V	30	6
4	E	13	N	22	W	31	7
5	F	14	O	23	X		
6	G	15	P	24	Y	(pad)	=
7	H	16	Q	25	Z		
8	I	17	R	26	2		

Figura 8. Tabla de caracteres BASE32

Ahora se ha creado el *hostname* .onion para el servicio. Pero si no hay un servidor DNS, ¿cómo sabe la red Tor que ese servicio está funcionando, y cómo se puede llegar a él?

La ubicación de un servidor DNS en una red como Tor claramente iría en contra vía de sus objetivos de anonimato. Por eso, para anunciar la existencia de un servicio *onion*, el servidor selecciona aleatoriamente varios relays a los que les comparte su llave publica, además, les indica que pasen a funcionar como *introductions points*[35] o puntos de introducción. El número máximo puntos de introducción por servicio es de 10.

El servidor inicia un circuito virtual con cada uno de los puntos de introducción, así crea una ruta, pero se asegura que ningún punto intermedio sepa la IP real de donde se encuentra. Anexo a esto, el servidor ensambla un descriptor *onion*, que contiene su clave pública y un resumen de los puntos de

30 Domain Name Server
31 https://trac.torproject.org/projects/tor/wiki/doc/HiddenServiceNames#Howare.onionnamescreated
32 https://tools.ietf.org/html/rfc3447
33 https://www.ietf.org/rfc/rfc3174.txt
34 https://tools.ietf.org/rfc/rfc4648.txt
35 https://www.torproject.org/docs/onion-services.html.en

introducción, este descriptor se firma con su llave privada y se coloca en una tabla de hashes distribuidos.

Cuando un cliente quiere acceder a determinada dirección .onion consulta la tabla distribuida y en caso de encontrar una coincidencia, descarga ese descriptor *onion.* Con este descriptor el cliente conoce ahora la llave pública y los nodos intermedios que se comunican con el servicio. Paralelamente, el cliente escoge un *relay* aleatorio para que sirva como punto de encuentro o *rendezvous point,* crea un circuito con él y le comparte un secreto de una sola vez.

Una vez se ha establecido el circuito con el punto de encuentro y el cliente cuenta con el descriptor del servicio, envía una petición (mensaje de introducción) a cualquiera de los puntos intermedios para que el servicio sea entregado al punto de encuentro. Esta comunicación con los puntos intermedios también se realiza a través de un circuito virtual, por lo que la IP del cliente se mantiene oculta.

El servidor toma la petición del cliente, la descifra y crea un circuito con el punto de encuentro, a quien también le entrega un secreto de una sola vez, pero esta vez en un mensaje de encuentro. Si las llaves del servidor coinciden, el punto de encuentro informa al cliente que se estableció una conexión y el punto de encuentro pasa a funcionar como una pasarela de mensajes cifrados entre los circuitos del cliente y el servidor.

2.4 Ventajas de la Red Tor

- Una mayor cantidad de usuarios permite que la red sea más compleja y existan más nodos disponibles para las conexiones.
- Es controlada por una fundación encargada de direccionar el proyecto, lo que permite mayor fluidez y rapidez en las adaptaciones y mejoras.
- Tiene una mayor y mejor documentación dada su temprana expansión en ambientes académicos.
- Posee una mayor cantidad de desarrolladores e inversores en el proyecto.
- El control centralizado reduce el ancho de banda y esfuerzo requerido en los demás nodos.
- Está optimizada para la función de proxy de salida a la *surfaceweb*.
- Su desarrollo en C ha logrado una mayor eficiencia en la utilización de la memoria en los clientes.
- Ha logrado perfeccionar el sistema de escalado, por lo que su expansión no es traumática.

2.5 Desventajas de la Red Tor

- Su topología de gerenciamiento centralizado permite que quien controle la mayoría de sus autoridades de directorio pueda controlar el consenso de la red.
- Por su arquitectura de conexión, la velocidad de respuesta de los servicios ocultos no es tan buena como en la red I2P.
- Está optimizado para tráfico TCP y especialmente https, para anonimizar otros servicios se requiere herramientas anexas.
- La totalidad del tráfico de la conexión pasa por un único circuito virtual.
- Falta de redundancia y balanceo de carga en sus circuitos.

3. Redes Descentralizadas o Distribuidas (I2P)

Una red descentralizada se define como una red en la que su administración no se centra en algún punto en particular, si no que permite que todos los dispositivos se comuniquen directamente entre sí[36].

De acuerdo a Teresa C. Piliouras, Una de las principales ventajas que brinda la adopción de un sistema distribuido es reducir el impacto de una falla puntual que afecte el funcionamiento de la red como un todo, proporcionando diversidad respaldo y recuperación rápida de desastres. De esta manera se intenta evitar la vulnerabilidad presentada en la red Tor con la centralización de las autoridades de directorio.

La red I2P (Internet Invisible Project) nació a partir del protocolo de baja latencia creado para el proyecto Freenet y en el 2003[37] se convirtió en un proyecto independiente con sus primeras *releases* para los primeros meses del 2006.

Según la documentación del proyecto[38], "I2P es una capa de red auto organizada, resistente, anónima, dentro de la cual pueden funcionar un gran número de aplicaciones con diferentes modelos de seguridad y anonimato. Cada una de estas aplicaciones pueden usar sus propios niveles de seguridad y sus propios niveles de rendimiento, sin preocuparse por crear una implementación apropiada de una red libre, permitiéndoles mezclar sus actividades con un gran número de usuarios anónimos que ya utilizan I2P".

La red I2P se propone como una capa superior de red que permite la anonimización de muchos servicios comunes de la *SurfaceWeb*, no se define como un proyecto especialmente académico, de investigación, gubernamental o comercial. Si no simplemente, un esfuerzo de ingeniería enfocado a ofrecer un nivel decente de anonimato en Internet.

3.1 Network Database (netDb)

Las redes anónimas distribuidas aparecieron como una evolución de la red Tor, buscando solventar algunas de sus vulnerabilidades más conocidas. En este orden de ideas, muchos de los conceptos utilizados en la red Tor fueron incorporados en en las otras redes y se fueron agregando mejoras graduales con el paso del tiempo.

Una de las mayores diferencias entre la red I2P y la red Tor es precisamente lo que las define como centralizadas y distribuidas: su sistema de gerenciamiento. La red I2P presenta una base de datos totalmente distribuida, buscando subsanar las vulnerabilidades presentadas con las autoridades de directorio centralizadas de la red Tor.

La netDb, como es conocida esta base de datos, contiene únicamente dos tipos de metadatos: Información de contacto de los nodos I2P o "*routers*" (*RouterInfos*) e información de contacto del destino (*LeaseSets*). Cada tipo de dato es firmado por su propietario y verificado por cada nodo donde es almacenado.

36 Network Design, Second Edition: Management and Technical Perspectives, Teresa C. Piliouras, Taylor & Francis, 2004,p 144
37 https://geti2p.net/_static/pdf/i2p_philosophy.pdf
38 https://geti2p.net/es/docs

Los metadatos tienen un tiempo de vida útil en cada *route*r lo que permite que la red sea totalmente dinámica y los datos irrelevantes o caducados sean descartados y reemplazados por nuevas entradas. Esta característica disminuye notablemente la posibilidad de ataques basados en el estudio de la estructura de la red.

Los datos *RouterInfo* y *LeaseSet* son almacenados en la netDb mediante nodos dedicados llamados *floodfill nodes* o nodos de inundación, estos nodos por lo general se caracterizan por tener altos anchos de banda y tiempo de disponibilidad superior a los demás. Usando esta técnica, la red es continuamente actualizada de forma que ningún nodo tiene la visión la totalidad de la red en tiempo real[39].

Podría parecer que este conjunto de nodos realiza la misma labor que las autoridades de directorio en la red Tor, pero la verdad es que su esencia es muy diferente. A diferencia de la red Tor, los nodos de inundación no permanecen estáticos ni son verificados por ninguna autoridad, es decir, cualquier nodo podría eventualmente convertirse en un *floodfill.*

Al no tener una entidad central que administre y controle las funciones de los nodos, se implementó un mecanismo para evitar suplantaciones y de esta manera asegurar que los perfiles asignados a cada *router* de la red correspondan y estén de acuerdo a sus capacidades.

La información acerca de los nodos de la red I2P es inherentemente no confiable, por esto, para evitar que la información sea deliberadamente modificada por alguno de ellos, se incluyó un sistema de "*peer profiles*" o perfiles iguales en donde los nodos que interactúan entre si van creando estadísticas sobre parámetros de ancho de banda y disponibilidad de sus pares, entre otros. Estos datos ayudan a la red en la toma de decisiones.

La netDb utiliza el protocolo *Kademlia*[40] para saber la proximidad de los nodos y formar la tabla de hashes. I2P realizó una pequeña modificación al protocolo original agregando al hash del nodo, la fecha en formato UTC y haciendo que cambia diariamente a media noche, con esto se intenta minimizar el riesgo de ataques sobre las tablas de hash. En la tabla se almacena algo similar a: *SHA256(clave+yyMMdd)*.

3.1.2 Metadatos *RouterInfo*

Para la comunicación entre dos nodos o *routers* en la red I2P, se requiere conocer algunos parámetros contenidos en la estructura *RouterInfo* almacenada en la *netDb,* esta información ayuda a los nodos a tomar decisiones de conexión; como los nodos que utilizará para formar túneles y para descartar nodos que no cumplen con el umbral mínimo de ancho de banda.

La clave de cifrado del paquete está dada por su hash SHA256. La estructura de metadatos contiene la siguiente información:

- La identidad del *router,* utilizando una firma digital y un certificado realizado bajo el algoritmo ElGamal 2048.

39 https://petsymposium.org/2017/papers/hotpets/i2p-looking-for-group.pdf
40 https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf

- La dirección con la que puede ser accedido
- Cuando fue publicado
- Un set de opciones de texto
- La firma del propio *RouterInfo*

3.1.3 Metadatos *LeaseSet*

La segunda estructura de datos almacenada en la netDb contiene los "*Leases"* o contratos. Estos datos informan sobre cuáles entradas de túneles corresponden a algún cliente en particular de la red I2P. Con esta información se lleva a cabo la negociación para la utilización de túneles de entrada y salida para la transmisión de datos.

Además de lo anterior, contiene las identidades y firmas que permiten el cifrado punto a punto de la red. La estructura contiene los siguientes datos:

- La identidad del *gateway* del túnel
- El ID del túnel por el que se envían los mensajes (4 bytes)
- Tiempo de caducidad del túnel
- La firma y el certificado en cifrado ELGamal 2048 del destino, que se generan cada vez que el *router* inicia un nuevo proceso de conexión.
- Clave pública adicional para el cifrado g*arlic* o ajo
- La firma del propio *LeaseSet*

3.2 Nodos (*Routers*)

3.2.2 Nodos de Salida (*Output Gateway*)

El concepto de nodo en esta red también juega un papel fundamental, la existencia de miles de nodos voluntarios distribuidos alrededor del mundo, asegura la robustez y complejidad de la red que deriva siempre en la dificultad para desvelar cualquier comunicación interna o analizar cualquier tipo de tráfico. Los nodos en la red I2P son llamados *routers*.

De la misma manera que Tor, la red I2P incorpora servicios internos ocultos a los que únicamente clientes de la red pueden acceder, además, puede ser usada para anonimizar la navegación en la *SurfaceWeb*, puntualizando que su punto fuerte de desarrollo está enfocado hacia los servicios internos.

La red I2P también utiliza un concepto similar al de los circuitos virtuales, con algunas modificaciones para sus conexiones; en esta tecnología se pasan a llamar túneles de entrada o salida dependiendo su función. Estos túneles son temporales y unidireccionales, los datos salen o entran, desde o hacia el cliente. El concepto de túnel será estudiado con detalle más adelante.

El primer nodo en cualquiera de los túneles, bien sea de entrada o de salida se denomina *Gateway*. El hecho que sean de entrada o salida está determinado por la dirección del túnel.

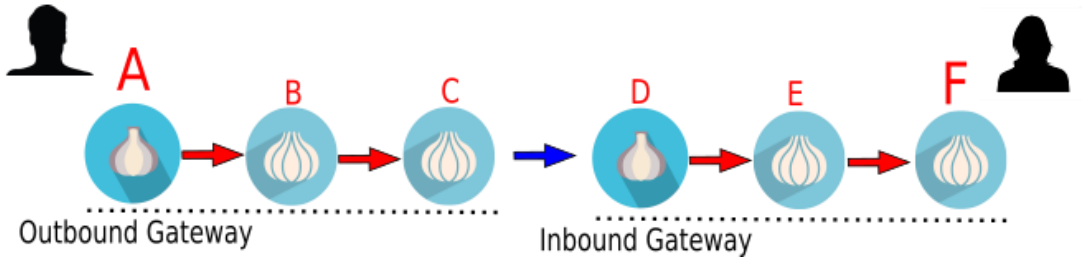


Figura 9. Túnel Garlic

En la figura 9 se puede observar la topología típica de una conexión en la red I2P, la conexión se realiza en sentido de transmisión del nodo A hasta el F. El *gateway* de salida, en este ejemplo, es el primer nodo del túnel de salida ya que envía datos hacia el nodo F, es decir, los saca de su nodo usando un túnel de salida.

Una de las funciones de los *gateways* es acumular y pre-procesar los mensajes I2NP para que puedan transitar cifrados y con un tamaño fijo de 1024 bytes por los túneles.

Después del pre-procesado de los mensajes hacia una carga con datos de relleno, el *gateway* construye un valor de IV (vector de inicialización) aleatorio de 16 bytes, cifrando a este y al mensaje de túnel iterativamente tanto como se necesite, y reenvía la tupla { tunnelID, IV, mensaje de túnel cifrado } al siguiente salto[41].

Los *gateways* de salida introducen en la red a siguiente información[42]:

- Clave de cifrado de túnel: Una clave privada AES para cifrar los mensajes, y las instrucciones para el siguiente salto.
- Clave IV del túnel: Una clave privada AES para cifrar doblemente el IV hasta el siguiente salto.
- Clave de repuesta: Una clave privada AES para cifrar la respuesta a la petición de construcción del túnel.
- IV de respuesta: El IV para cifrar la respuesta de la petición de construcción del túnel.
- Siguiente salto: Cual es el siguiente *router* en el camino (a no ser que sea un túnel de 0 saltos, y la puerta de salida sea también el punto final)
- ID del siguiente túnel: La ID del túnel en el siguiente salto

41 https://geti2p.net/es/docs/tunnels/implementation
42 https://geti2p.net/en/docs/how/tunnel-routing

3.2.3 Nodo de entrada (*Inbound Gateway*)

Aplicado la misma lógica utilizada para identificar el nodo de salida; el nodo de entrada es el primer nodo presente en un túnel de entrada. El nodo de entrada se encargará de encaminar los datos que vienen desde un túnel de salida hasta el nodo receptor.

Además de la información suministrada por el nodo de salida, el nodo de entrada adiciona un identificador de túnel de 4 bytes en los metadatos *LeaseSet.* De esta manera los túneles de salida identifican el túnel de entrada que podría encaminarlos a un nodo receptor final en particular.

3.2.4 Nodos Intermedios

Todos los *routers* participantes en la comunicación entre túneles, exceptuando los *gateways in/out* (Nodos A y D*)* y los puntos finales (Nodos C y F) del túnel son denominados como intermedios. Los metadatos aportados a la *LeaseSe*t son los mismos del nodo de salida.

3.2.5 Nodos Finales

Estos *routers* terminan los túneles tanto de entrada como de salida, para el caso de la figura 9, se tratan de los nodos C y F.

El *router* final de salida u "*Outbound Endpoint"* es el último punto del túnel de salida que conecta con algún *gateway* de entrada, de esta manera se une un túnel de salida y uno de entrada y se forma el circuito completo de conexión.

Como es de esperarse el *"Inbound Endpoint"* o *router* final de entrada será el receptor del paquete de datos, quien es el último nodo del túnel de entrada.

3.3 Túneles

Teniendo en cuenta las definiciones anteriores, se puede estructurar el concepto de túnel como los caminos virtuales, temporales y unidireccionales formados por paquetes en una secuencia de *routers* que cumplen funciones especiales para la conexión de dos puntos, en procura del anonimato de la fuente y el receptor de los mensajes.

Existen tres tipos de roles de túnel:

- Túnel de entrada*:* Es el túnel que forma un nodo para que puedan entrar los datos desde la red I2P.
- Túnel de salida*:* Es el túnel que forma un nodo para sacar datos hacia la red I2P.
- Túnel Intermedio: Es un túnel que hace la labor de pivote para conectar túneles de salida y de entrada.

El tamaño de los túneles, es decir el número de nodos que lo componen puede ser seleccionado por el cliente. Ha de tenerse en cuenta que el número de nodos es directamente proporcional a la complejidad del circuito, pero también a la latencia y la probabilidad de fallo por lo que se debe estudiar los requerimientos de la aplicación y el mensaje para esta toma de decisión.

Para reducir la probabilidad de ataques por análisis de tráfico, se recomiendan 3 nodos ya que después de este número no se logran mejoras significativas en el nivel de protección. La longitud por defecto para los túneles de entrada y salida es de 3[43].

inbound.length		número desde 0 hasta 3	0 a 7	3	Longitud de túneles de entrada.
outbound.length		número desde 0 hasta 3	0 a 7	3	Longitud de túneles de salida..

Figura 10. Opciones de I2CP

Como se mencionó en el capítulo anterior, la red Tor usa el protocolo de enrutamiento *onion* para la creación de los circuitos virtuales y el cifrado por capas de la información. La red I2P introduce una variante que se denominó "*garlic routing"* o enrutamiento ajo[44].

El funcionamiento de la técnica de enrutamiento *garlic* es muy similar al *onion* en sus conceptos más básicos: crear caminos virtuales y mover mensajes cifrados por capas, pero presenta diferencias substanciales en cuanto a la operatividad y capacidad del sistema.

En primer lugar, en el sistema *garlic* el mensaje puede ser agrupado con otros similares de la misma aplicación, cifrado por capas y transmitido. En la analogía cada mensaje sería como un diente que forma un bulbo de ajo con varias capas.

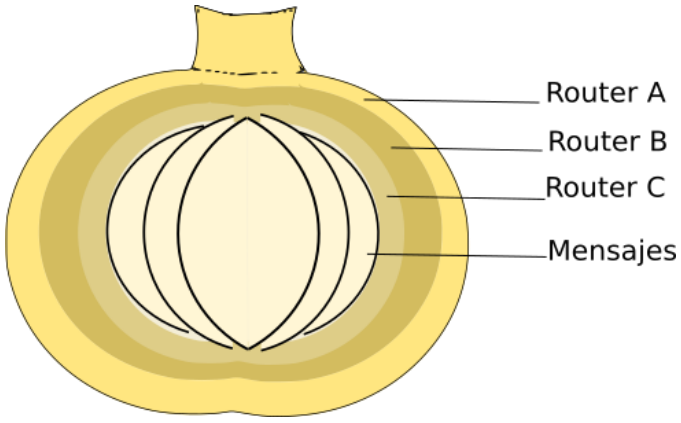


Figura 11. Analogía Mensaje Garlic

La otra diferencia sustancial es el concepto de túnel como camino unidireccional por el que solo pueden salir o entrar datos, estos túneles se pueden agrupar y cada *router* puede crear los que considere necesarios para su aplicación y para balancear la carga de los mismos. En el enrutamiento *onion* los datos pueden retornar por el mismo circuito, en la red I2P se deben usar túneles diferentes para cada sentido de la comunicación.

Para una comunicación simple entre dos *routers* se deben crear 4 túneles: un par de entrada y salida por cada punto de la comunicación. En la Figura 12 se puede observar la topología de una comunicación I2P básica.

43 https://geti2p.net/es/docs/protocol/i2cp#options
44 https://www.freehaven.net/doc/freehaven.pdf

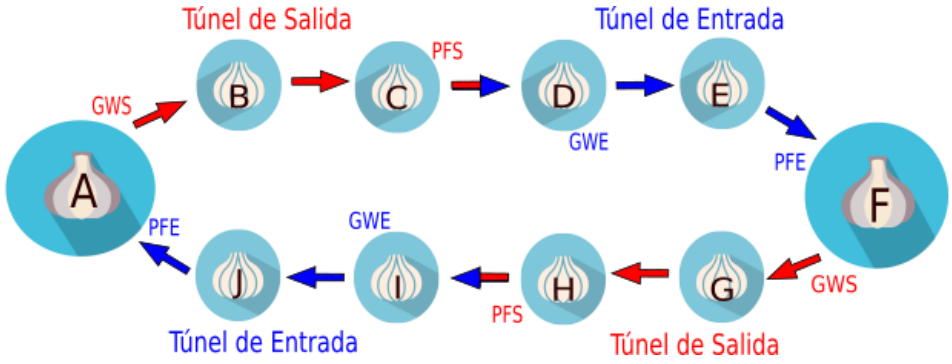


Figura 12. Comunicación I2P

GWS: Gateway de Salida
PFS: Punto Final de Salida
GWE: Gateway de Entrada
PFE: Punto Final de Entrada

3.4 Direcciones I2P

El servicio de nombres de servicios en la red I2P es provisto mediante libretas de direcciones locales, esto quiere decir que puede que las direcciones no correspondan a los mismos servicios de manera globalizada, pero se asegura que será el servicio que se requiere, siempre.

El servicio de nombres I2P es totalmente distribuido ya que es almacenado de manera local en los *routers*. Las direcciones son formadas por 512 bytes binarios convertidos a partir de la llave pública de 256 bytes más un certificado de 128 bytes para la firma, en versiones anteriores se usaba almacenado en BASE64 pero desde la v 0.8.8 fue cambiado a binario.

Cuando un cliente desea acceder a un servicio a partir de un nombre del tipo .i2p, el *router* busca en el archivo de base de datos donde se almacenan las libretas de direcciones. El archivo llamado *hostsdb.blockfile* realiza la búsqueda y devuelve la dirección .i2p legible para los humanos.

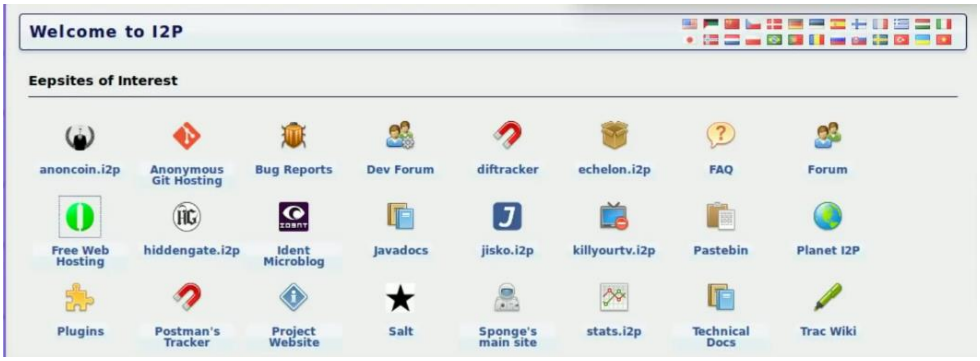


Figura 13. Nombres servicios I2P

En la figura 13 se puede observar un listado de algunos servicios accesibles a partir de su nombre .i2p, como se comentó al inicio del capítulo, si bien I2P puede ser adaptada para navegar la *surfaceweb* su desarrollo está volcado y optimizado para los servicios internos de la red.

3.5 Ventajas de la red I2P

- Está optimizada para sus servicios ocultos por lo que la velocidad de respuesta de los mismos es mejor que en la red Tor.
- Su topología descentraliza evita ser blanco de muchos ataques sobre su infraestructura crítica.
- Los túneles unidireccionales permiten mayor seguridad en la comunicación al no exponer la totalidad del tráfico por un mismo canal.
- El tiempo de vida de un túnel es considerablemente menor a la de un circuito típico de Tor, por lo que ataques por análisis de tráfico se hacen más difíciles.
- Está concebida para tráfico TCP y UDP.
- Prácticamente todos los nodos participan como elementos enrutadores.
- Redundancia de túneles, por lo que la disponibilidad y balanceo de carga son mejores.
- Sistema integrado de actualizaciones automáticas.

3.6 Desventajas de la red I2P

- Su fuerte no es el anonimato de la navegación en la *surfaceweb*, por lo que este tipo de conexiones suele ser lenta e inestable debido a la poca cantidad de nodos de salida.
- Está escrito en Java lo que no ha permitido llegar al nivel de optimización de memoria de la red Tor.
- La documentación es limitada y no existen muchos artículos académicos respecto a esta red.
- Es significativamente más pequeña comparada con la red Tor, con un menor número de usuarios la escogencia de nodos puede resultar difícil.
- Existen menos desarrolladores por lo que los cambios y mejoras suelen tardar un poco más que en Tor.
- Aún presenta algunos problemas de expansión y escalabilidad.

4. Otras Redes de la DarkWeb

Como se mencionó en el capitulo 1 de este documento, la *DarkWeb* se encuentra formada por redes aisladas que utilizan tecnologías exclusivas para su acceso y uso. Una pequeña muestra de ellas son Tor e I2P, pero no son las únicas, a continuación, se describirán brevemente algunas de las redes más conocidas en la actualidad, entre un inmenso abanico de posibilidades que se ofrecen en la gran Internet.

4.1 FreeNet

FreeNet es una red anónima totalmente descentralizada con un funcionamiento similar a I2P, basado en pares o *peers*. De acuerdo a sus creadores, FreeNet puede considerarse como un gran dispositivo de almacenamiento[45].

Una vez el usuario almacena un archivo en la red, recibe una clave que puede usarse para recuperar el archivo. Listados descentralizados con páginas que almacenan gran cantidad de estos archivos nutren la red y a través de ellos los clientes obtienen las claves para poder extraer dichos ficheros. El espacio de almacenamiento está distribuido en todos los nodos de la red. Pequeños espacios de disco de los nodos son usados para almacenar fragmentos de archivos cifrados, posteriormente se unen con sus pares guardados en otros nodos y se entregan a la FreeNet en el momento que sea solicitado por un cliente.

La operación de borrado no está disponible en la FreeNet, en cambio, un algoritmo de popularidad y uso de los archivos se encarga de guardar o eliminar los archivos dependiendo de qué tan frecuentemente son solicitados y de la capacidad actual de la red para almacenarlos.

Para cada archivo de la red existe una clave específica, las claves están dispuestas en forma de hash y construidas haciendo analogía a una dirección URI[46] del tipo *freenet: =CHK@ejemplo.txt.* Al final son estas claves las direcciones de acceso a los contenidos.

El funcionamiento de las conexiones en la red FreeNet se puede observar en la figura 14[47].

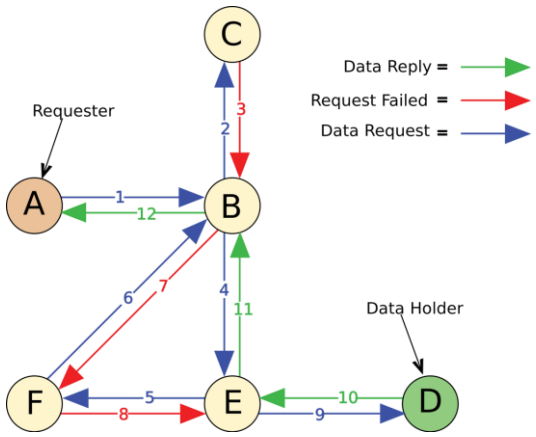


Figura 14. Topología conexión FreeNet

45 https://freenetproject.org/pages/documentation.html
46 https://tools.ietf.org/html/rfc3986
47 https://blog.deepwebbrasil.com/como-funciona-a-freenet/

Existen 4 tipos de claves asociadas a la FreeNet:

- CHK: Claves hash de contenido
- SSK : Claves de subespacio firmadas
- USK : Teclas de subespacio actualizables
- KSK : Claves firmadas con palabras clave

4.2 ZeroNet

La comunidad de ZeroNet la define como una red descentralizada, resistente a la censura, que fusiona la criptografía de la tecnología *blockchain* de Bitcoin y la red BitTorrent [48].

Básicamente ZeroNet es una red *Peer to Pe*er que utiliza la red y el protocolo p2p de BitTorrent para encontrar los nodos donde se alojan sus contenidos. No es en sí una red enfocada en el anonimato, pero confía que con una buena cantidad de nodos *torrent* la posibilidad de aislar a un cliente se dificulte, aun así puede trabajar sin problemas con redes anónimas como Tor.

Está escrita enteramente en lenguaje *python*. No usa ningún tipo de contraseña para autenticación, pero los contenidos están cifrados con el mismo sistema de generación de direcciones de monederos implementada en Bitcoin conocido como BIP32[49].

La red ZeroNet está enfocada en impedir que sus contenidos puedan ser borrados, es decir evitar la censura. Las conexiones son cifradas con protocolo TLS y usa el mecanismo *MessagePack*[50] para los mensajes de red. Esta red funciona en cualquier sistema operativo y no necesita un navegador en particular, su configuración es sencilla y la escalabilidad se presenta sin retrasos.

La mayoría de contenidos son sitios estáticos y dinámicos que comparten variedad de información, gran parte de estos son blogs. Una de las ventajas de ZeroNet es que permite desarrollo con bases de datos SQL lo que agiliza y dinamiza sus ofertas.

La red soporta direcciones del tipo .bit provista por *namecoins*[51] e.j http://127.0.0.1:xxxx/zerosecurity.bit (blog zero security) o http://127.0.0.1:xxxxx/search.kaffie.bit/ (buscador de sitios ZeroNet). También permite direcciones del tipo hash e.j http://127.0.0.1:xxxx1MagneTSMMkNmJTMsTGJtfuNsqGjzzXgQv (blog de series de televisión).

4.3 Morphis

Morphis es una de las redes más interesantes en la Deep Web ya que además del componente tecnológico, exalta un concepto ideológico profundo de una conciencia de orden mundial materializada en este tipo de redes que denominan como "Cerebro Mundial". Esta idea pretende lograr conocimiento libre, abierto y distribuido sin interferencias de gobiernos para y por toda la humanidad.

48 https://zeronet.readthedocs.io/en/latest/
49 https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
50 https://msgpack.org/
51 https://bit.namecoin.org/

Según su creador Sam Maloney, recientemente fallecido, Morphis es un almacén de datos encriptado distribuido a nivel global y destinado a reemplazar la nube de la Internet conocida, todo esto, sin censura y totalmente gratuito[52].

Morphis a igual que I2P incorpora el protocolo Kademlia DHT con ciertas modificaciones a través de conexiones tipo SSH, también modificadas por su creador. Es totalmente compatible con Tor y funciona orientada a conexión por medio de TCP. Siguiendo su filosofía de estar disponible para todos, su método de conexión es bastante simple: descargar y ejecutar. La parte de conexiones y firewalls es optimizada por el código propio y está disponible para la mayoría de sistemas operativos.

Aunque el creador desapareció muy joven y según él perseguido por su gobierno, alcanzó a incluir en la red su propio sistema de *email* llamado Dmail/Dpush[53] totalmente distribuido, resilente al *spam* y a la censura. Cuenta con tres interfaces: por navegador web, SSH, o una consola implementada y desarrollada para Morphis llamada mmu.

4.4 GnuNet

Por la misma línea de los proyectos con un amplio contenido social y político, se encuentra GnuNet, definida como una estructura de red para construir aplicaciones P2P seguras, descentralizadas, y que preserven la confidencialidad de los pares participantes. El objetivo de GnuNet es reemplazar la pila de protocolos inseguros utilizados en la *surfaceweb*[54].

Para su creación se investigaron otras redes como FreeNet, pero después de analizar el objetivo de GnuNet se optó por la creación de una red construida desde cero.

Los desarrolladores implementaron nuevos protocolos de cifrado de los contenidos como *Encoding for Censorship-Resistant Sharing* (ECRS)[55]. Básicamente es una estructura de cifrado de contenido resistente a la censura, este esquema optimiza la distribución de contenidos cifrados y la consulta de los mismos a través de redes descentralizadas.

Las consultas en esta red también son cifradas y los nodos intermediarios tienen la capacidad de conocer si una petición corresponde a algún contenido específico, pero nunca pueden conocer el contenido como tal, lo que deriva en un alto grado de privacidad. Para lograr esto, ECRS incorpora un sistema de descripciones cifradas de los archivos, así, el nodo compara si la consulta corresponde a la descripción y de esta manera solo la llave privada puede proveer el contenido.

El protocolo de enrutamiento utilizado por GnuNet es el GAP *Practical Anonymous networkinG*[56]. Desarrollado para esta red, se presenta como una alternativa para compartir ficheros de forma anónima a Tor (*onion*) e I2P (*garlic*). Esta técnica permite distribuir el contenido de forma segura haciendo que los ficheros vayan migrando de nodo en nodo, con esto, se evita la identificación del editor del archivo.

GAP no se esfuerza en impedir que se identifiquen los nodos intermedios participantes en la comunicación, si no en evitar que se identifique el iniciador y el destinatario de cualquier mensaje. En protocolos como Tor o I2P la fuente es reemplazada en cada salto por el nodo de turno y el remitente

52 https://morph.is/v0.8/
53 https://morph.is/v0.8/dpush-whitepaper.odt
54 https://gnunet.org/
55 https://gnunet.org/sites/default/files/ecrs.pdf
56 https://gnunet.org/sites/default/files/aff.pdf

original permanece oculto; en GAP no se busca impedir la conexión directa entre emisor y receptor, sino simplemente desacoplar las peticiones y respuestas de tal forma que un análisis de tráfico no pueda intuir cual es la fuente de la comunicación.

En este protocolo no se requiere que una petición o una respuesta sigan siempre un camino predeterminado como en los circuitos o los túneles, GAP optimiza los caminos distribuidos buscando no incurrir en problemas de latencia ni redundancia de rutas.

4.5 Resilio

Esta red, aunque conocida y con una cantidad considerable de nodos, tiene la particularidad diferencial con las anteriormente estudiadas de que su código fuente es privativo y se venden planes para su acceso "*premium"*. Si bien tiene una gran cantidad de ejecutables, incluidas distribuciones de *android* y algunos *firmwares* de *routers*, este es un punto que no aporta mucha confianza en el estricto sentido del anonimato.

El punto fuente de la red es la distribución anónima de ficheros mediante una red P2P, una característica interesante es que esta red permite implementarse dentro de entornos corporativos, es decir dentro de una red local sin necesidad de salir a Internet.

Resilio es una solución de la Deep Web enfocada para ambientes profesionales y empresariales. Utiliza el protocolo µTP2[57] para el enrutamiento, este protocolo está enfocado en el aprovechamiento de la velocidad llegando a lograr transferencias en torno de los Gbps y mínima perdida de paquetes. MTP2 es una evolución del protocolo de comunicaciones de BitTorrent diseñado para trabajar en redes mixtas y en balancear el tráfico dinámicamente y ajustarse al entorno de ancho de banda por el que transite.

Aunque Resilio se presenta como una red descentralizada, ofrece una interfaz de gerenciamiento y monitoreo central en la que se pueden realizar pequeñas tareas de administración. En la figura 15[58] se puede observar una solución de conexión típica.

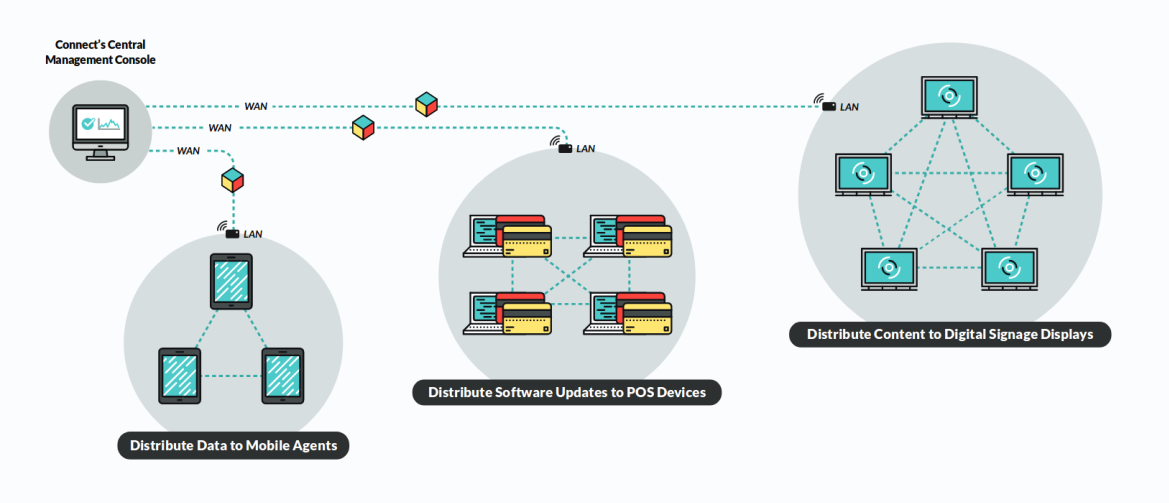


Figura 15. Topología red Resilio

57 https://www.resilio.com/blog/%CE%BCtp2-the-evolution-of-an-enterprise-grade-protocol
58 https://www.resilio.com/docs/Resilio%20Connect-%20Product%20Overview.pdf

5. Distribuciones más conocidas para navegar en la Deep Web

5.1 Tails (The Amnesic Incognito Live Sytem)

Tails es una distribución Linux diseñada para ser ejecutada en modo *live*, su objetivo es brindar el mayor nivel de anonimización posible para el usuario además de conservar su privacidad.

Está elaborado a partir de un kernel monolítico Debian Linux y es compatible con arquitecturas x86. Apareció por primera vez el 20 junio del 2009[59], como evolución de una distribución anterior basada en Gentoo llamada Incógnito. Tails pasó a recibir aportes de concepto por parte del proyecto Tor y de *Mozilla Fundation*. En los últimos años se convirtió en una de las distribuciones pensadas en anonimato, más usadas en el mundo.

Tiene como premisa conseguir que se cumplan dos de los conceptos contenidos en su nombre, amnesia (pérdida de memoria a largo plazo), e incógnito (ocultar su verdadera identidad). Pensando en esto sus creadores pre-configuraron una imagen que se pudiera ejecutar via USB live o DVD, con una serie de programas optimizados para salvaguardar la identidad de usuario mientras navega.

Para obtener la última versión de Tails simplemente se debe ingresar en el sitio web oficial:

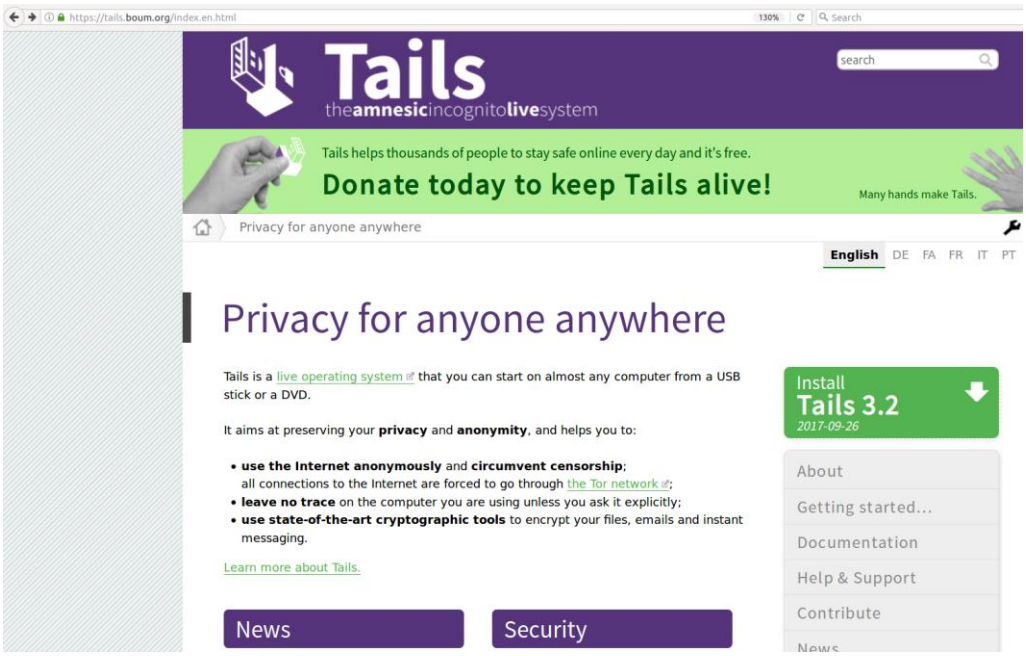


Figura 16. Sitio oficial distribución Tails

https://tails.boum.org/ , en la columna derecha los creadores presentan la opción de instalar o descargar la distribución. Como Tails está diseñada para correr en modo *live*, buscando aislarse y almacenar la menor cantidad de datos posible, escogemos la opción descargar y verificar vía *openpgp*[60] https://tails.boum.org/install/download/openpgp/index.en.html.

59 https://labs.riseup.net/code/projects/tails/repository/revisions?page=3982
60 https://www.openpgp.org/

En el Anexo A se muestra con detalle el procedimiento de instalación y ejecución del entorno Tails.

5.2 Whonix

Whonix se define como una distribución que tiene como objetivo preservar la privacidad y el anonimato ayudando a los usuarios a ejecutar aplicaciones de forma anónima[61].

Enfocado al uso de cualquier usuario sin mayores conocimientos técnicos, tiene una interfaz de instalación y ejecución muy sencilla e intuitiva. Está basado en la distribución Debian y optimizado para el uso con la Tor.

El diseño es especial, está formado por dos máquinas virtuales *Whonix-Gateway* y *Whonix-Workstation*, por lo general el sistema operativo anfitrión es el mismo del cliente, pero puede trabajar sin problemas en sistemas tipo *live*.

El sistema tiene como premisa la seguridad mediante el aislamiento, cada uno de las máquinas virtuales tiene funciones específicas *Whonix-Gateway* maneja la salida a la red Tor y *Whonix-Workstation* el trabajo con otras aplicaciones. Presenta un servicio *IRC*, *Thunderbird*, *Enigmail*, *Birdy*, todas las aplicaciones de la interfaz *Workstation* que requieran conexión a Internet están optimizadas para su trabajo bajo Tor.

La descarga se realiza a través de los repositorios oficiales en el sitio https://www.whonix.org/ después de dar click en la opción *Download* se selecciona el sistema operativo.

Figura 17. Sitio oficial distribución Whonix

En el Anexo A se muestra con detalle el procedimiento de instalación y ejecución del entorno Whonix.

61 https://www.whonix.org/wiki/About

5.3 Qubes OS

Sus desarrolladores describen Qubes OS como un sistema "razonablemente seguro"[62], partiendo de la premisa de que no existe ningún sistema totalmente seguro. Qubes utiliza pequeñas máquinas virtuales ligeras que denomina "*qubes*" para ejecutar los procesos de manera aislada al sistema anfitrión.

Este sistema operativo toma en serio la labor de aislamiento de los procesos y todo su desarrollo es basado en esta premisa, además, provee información de los entornos de ejecución discriminando por colores los niveles de seguridad al acceso de las aplicaciones a diferentes tipos de dominio.

Qubes también incorpora el concepto de *microkernel*[63]. Según los desarrolladores, cuanto más pequeño es el código a ejecutar la probabilidad de insertar líneas inseguras disminuye, el control del sistema operativo puede hacerse con mayor facilidad y el conocimiento del funcionamiento en general aumenta, esto redunda en robustez y seguridad.

Usa como núcleo un Fedora y está optimizado para funcionar de manera nativa en la máquina cliente. La descarga es sencilla desde el sitio oficial https://www.qubes-os.org/

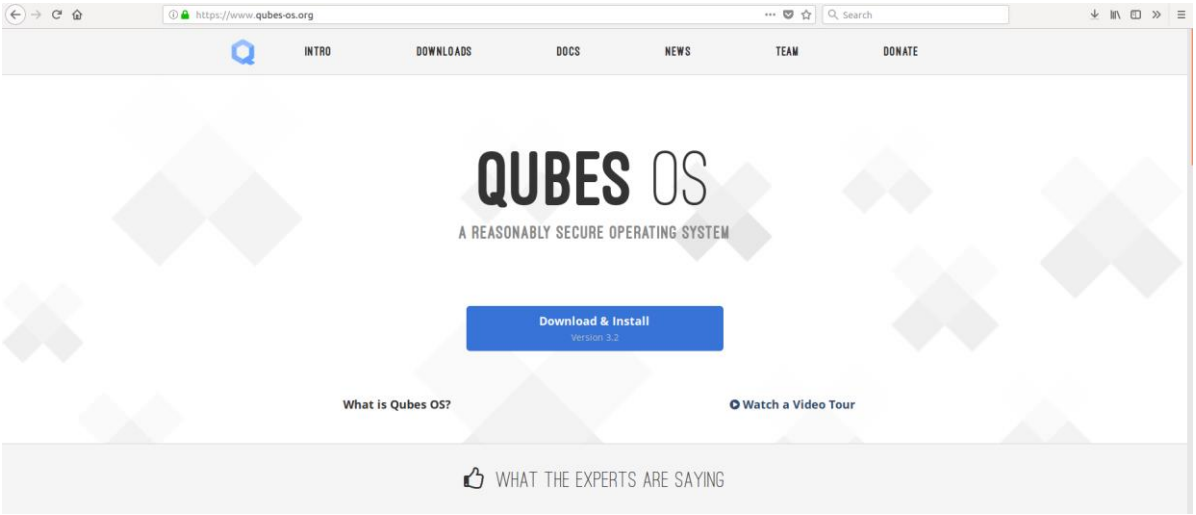


Figura 18. Sitio oficial distribución Qubes OS

6. Anexo A

Material Audiovisual, práctica distribuciones y redes Tor e I2P

62 https://www.qubes-os.org/
63 http://www.minix3.org/theses/herder-true-microkernel.pdf

7. Conclusiones

En el transcurso del documento se realizó una aproximación técnica a los dos tipos más comunes y extendidos de topologías de redes anónimas. Se mostraron a detalle las dos redes más usadas y difundidas a nivel global y las distribuciones de mayor crecimiento y respaldo por la comunidad de entusiastas en la *Deep Web*.

Con el conocimiento de las tecnologías y los servicios envueltos en el desarrollo de las diferentes redes protagonistas en la *Deep Web,* se brindaron herramientas para que el usuario consiga entender de una forma un poco más clara el funcionamiento y las posibilidades que pueden ofrecer este tipo de arquitecturas, así como crear una visión más acertada de los alcances de las acciones desarrolladas en ellas.

Más allá de las noticias sobre el cibercrimen, se pudo evidenciar la enorme cantidad de comunidades de entusiastas y académicos desarrollando nuevas aplicaciones, tecnologías y hasta filosofías para sustentar estas nuevas formas interconectar el mundo.

Observando la historia y el recorrido de los diferentes desarrollos, se contextualiza el esfuerzo de miles de personas que diariamente aportan parte de su tiempo y conocimiento en pro de la optimización y la mejora de los sistemas estudiados. Esto contribuye a cambiar drásticamente la percepción de las personas envueltas en este tipo de iniciativas.

Si bien para un usuario sin muchos conocimientos técnicos, el entendimiento de este nuevo concepto de redes es algo complejo, se buscó llevar en términos claros y conceptos específicos una idea básica para que los nuevos usuarios continúen el proceso de búsqueda de conocimiento y aprendizaje, indagando sobre las tecnologías aquí expuestas.

Es importante apuntar que se marcó la separación de algunos términos que, aunque importantes, suelen ser muy cercanos y pueden llevar al usuario a caer en errores de concepto. La *DarkWeb* está formada por *DarkNets* y aunque restringida a sus tecnologías hace parte de la *DeebWeb,* esta a su vez es parte de la gran Internet. Dependiendo la arquitectura de las *DarkNets* se pueden usar para servir de capa de anonimato hacia la *SurfaceWeb* y es una de las grandes ventajas que ofrece Tor comparada con otras *DarkNets*.

Al unir conceptos teóricos claros con la facilidad de percepción de los sistemas audiovisuales en las practicas, se buscó acercar al usuario de un nivel básico de conocimiento en tecnologías de la información, al uso responsable y consciente de las redes estudiadas.

Las opciones de mejora, aplicación y desarrollo para estas tecnologías son inagotables y las posibilidades para solucionar los problemas neurálgicos de nuestras sociedades colocan a la *DeepWeb* más que como un obstáculo, en una esperanza global. Si no, por lo menos se convierte una iniciativa loable a la que se le debe dar apoyo y crear una conciencia responsable alrededor de su uso.

Existen muchos temas de mayor profundidad técnica y experticia que quedaron sin abordar en este documento, como siempre, se hizo el mejor esfuerzo para exponer lo que se consideró más relevante para llevar más claramente los conceptos necesarios para el entendimiento de los temas tratados.

Referencias

- Carreu, Joel. Radical Evolution: The Promise and Peril of Enhancing Our Minds, Our Bodies -- and What It Means to Be Human, Broadway Books, 2006. ISBN 0767915038
- http://www.umich.edu/%7Earchive/linguistics/bigdummysguidetotheinternet
- Woodford, Chris. Internet: A Historical Encyclopedia, Volume 2, ABC-Clio, 2005. ISBN 185106590
- Berners-lee, Tim. Weaving the Web, The original design and ultima destiny of the world wide web, Harper Collins, 2000.
- http://archie.icm.edu.pl/archie-adv_eng.html
- http://infolab.stanford.edu/~backrub/google.html
- https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/
- http://msl1.mit.edu/ESD10/docs/darknet5.pdf
- https://www.onion-router.net/Publications/IH-1996.pdf
- Barlett Jamie, The Dark Net, Random House, 2014. ISBN 1473506034
- https://bitcoin.org/bitcoin.pdf
- https://metrics.torproject.org/userstats-censorship-events.html
- Network Design, Second Edition: Management and Technical Perspectives, Teresa C. Piliouras, Taylor & Francis, 2004.
- David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding Routing Information," Workshop on Information Hiding, Cambridge, UK, May, 1996.
- Network Design, Second Edition: Management and Technical Perspectives, Teresa C. Piliouras, Taylor & Francis, 2004 p 345
- https://atlas.torproject.org/#search/flag:Authority
- https://metrics.torproject.org/glossary.html#onion-service
- https://consensus-health.torproject.org/consensus-health.html#consensusparams
- https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt
- https://www.freehaven.net/anonbib/cache/wright03.pdf
- https://svn.torproject.org/svn/projects/design-paper/tor-design.html
- http://www.rfc-base.org/rfc-5246.html
- https://metrics.torproject.org/glossary.html#onion-service
- https://trac.torproject.org/projects/tor/wiki/doc/HiddenServiceNames#Howare.onionnamescreated
- https://tools.ietf.org/html/rfc3447
- https://www.ietf.org/rfc/rfc3174.txt
- https://tools.ietf.org/rfc/rfc4648.txt
- https://www.torproject.org/docs/onion-services.html.en
- Network Design, Second Edition: Management and Technical Perspectives, Teresa C. Piliouras, Taylor & Francis, 2004,p 144
- https://geti2p.net/_static/pdf/i2p_philosophy.pdf
- https://geti2p.net/es/docs
- https://petsymposium.org/2017/papers/hotpets/i2p-looking-for-group.pdf
- https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf
- https://geti2p.net/es/docs/tunnels/implementation
- https://geti2p.net/en/docs/how/tunnel-routing

- https://geti2p.net/es/docs/protocol/i2cp#options
- https://www.freehaven.net/doc/freehaven.pdf
- https://freenetproject.org/pages/documentation.html
- https://tools.ietf.org/html/rfc3986
- https://blog.deepwebbrasil.com/como-funciona-a-freenet/
- https://zeronet.readthedocs.io/en/latest/
- https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
- https://msgpack.org/
- https://bit.namecoin.org/
- https://morph.is/v0.8/
- https://morph.is/v0.8/dpush-whitepaper.odt
- https://gnunet.org/
- https://gnunet.org/sites/default/files/ecrs.pdf
- https://gnunet.org/sites/default/files/aff.pdf
- https://www.resilio.com/blog/%CE%BCtp2-the-evolution-of-an-enterprise-grade-protocol
- https://www.resilio.com/docs/Resilio%20Connect-%20Product%20Overview.pdf
- https://labs.riseup.net/code/projects/tails/repository/revisions?page=3982
- https://www.openpgp.org/
- https://www.whonix.org/wiki/About
- https://www.qubes-os.org/
- http://www.minix3.org/theses/herder-true-microkernel.pdf