Red de anonimización TOR y cibermercados negros

Trabajo Fin de Máster

Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones

Estudiante Alma María Aguilar Pérez

Consultor/a Enric Hernández Jiménez

Profesor/a responsable de la asignatura Victor García Font

Fecha Entrega 31 diciembre 2018

Índice

1 Introducción

El uso del anonimato online está viendo incrementada su popularidad en la sociedad actual, de manera que diferentes grupos de gente pueden expresar sus puntos de vista en temas controvertidos sin verse afectados por repercusiones políticas o legales. El anonimato les provee de libertad de expresión y protege su intimidad, pero además también evita que individuos con intenciones maliciosas se inmiscuyan en las vidas privadas de la gente, ya sea con objetivos comerciales, políticos o de otro tipo. Estas son algunas de las razones por la que más y más gente emplea herramientas como TOR (The Onion Router), siendo la red más popular de anonimato, utilizada por más de dos millones de usuarios diarios.

Actualmente la red Tor es usada por militares, periodistas, activistas, e incluso individuos que desean denunciar casos de de blanqueo de dinero o expresarse sin tapujos en sociedades donde existe censura.

Sin embargo, desafortunadamente, también es ampliamente utilizada para la realización de actividades delictivas debido a que precisamente brinda la posibilidad al usuario de preservar su identidad en internet. Un ejemplo es el caso de Silk Road, un mercado negro que se empleó para compra-venta de armas, licencias falsas y estupefacientes. Por tanto, resulta de especial interés, conocer las técnicas de desanonimización de usuarios y servicios, para destapar el cibercrimen cometido en esta red.

Con este TFM se pretende experimentar en un entorno virtualizado para ejemplificar el funcionamiento de Tor, así como estudiar algunas de las técnicas de desanonimización de usuarios y servicios más actuales.

1.1 Objetivos generales

A continuación se enumeran los objetivos generales que se pretenden alcanzar en la realización de este TFM:

- Demostrar comprensión detallada en el ámbito relacionado con la red TOR.
- Saber analizar diferentes alternativas y elegir la más adecuada, justificando la elección.
- Elaborar y defender un documento que sintetice un trabajo original en el ámbito de la seguridad de la información.
- Saber transmitir de forma eficiente y eficaz las partes más importantes de un contenido voluminoso a diferentes audiencias.

1.2 Objetivos específicos y tareas

- Comprender la arquitectura y métodos de operación de la red Tor.
- Conocer los servicios que proporciona esta red.
- Adquirir conocimientos sobre las técnicas de anonimato empleadas por los usuarios de esta red.
- Aprender sobre las técnicas para desanonimización de usuarios y servicios.
- Estudiar softwares que simulan la mecánica de funcionamiento de Tor: Shadow y experimenTOR.
- Ser capaz de presentar un caso práctico a pequeña escala (Shadow)
- Planteamiento de conclusiones

1.3 Enfoque y método seguido

La metodología a seguir para cumplir con los objetivos propuestos se define en los siguientes puntos:

- Planificación: se elabora un calendario para la consecución de cada una de las tareas a realizar con el objetivo de llevar a buen termino la conclusión del trabajo dentro de las fechas marcadas.
- Recopilación de información: esta fase corresponde con la etapa de documentación, donde se obtienen datos de diferentes fuentes.
- Análisis y selección de la información: se estudia la documentación recolectada y se clasifica según su valor, descartando las publicaciones redundantes o irrelevantes.
- Elaboración del contenido: se procede a la redacción de cada uno de los capítulos del trabajo.
- Extracción de conclusiones: se analiza el alcande del logro de los objetivos planteados inicialmente y se exponen las conclusiones alcanzadas tras la ponderación del trabajo de investigación realizado.

1.4 Planificación del Trabajo

En este apartado se muestra un diagrama de Gantt con las distintas tareas que se deben realizar para cumplir los objetivos presentados anteriormente. La fecha de inicio del trabajo es el 19/09/2018, y la fecha de finalización de éste es el 31/12/2018, fecha en que se debe entregar la memoria final.

Nombre	Inicio	Fin	Duración
Planificación TFM	sep 19	oct 5	13d
Arquitectura y funcionamiento TOR	**oct 5**	**oct 23**	**13d**
Recopilación información	oct 5	oct 9	3d
Análisis y selección de la información	oct 9	oct 15	5d
Elaboración del contenido	oct 15	oct 23	7d
Servicios TOR	**oct 23**	**nov 6**	**11d**
Recopilación información	oct 23	oct 25	3d
Análisis y selección de la información	oct 26	oct 31	4d
Elaboración del contenido	oct 31	nov 6	5d
Tecnicas de anonimato	**nov 7**	**nov 22**	**12d**
Recopilación información	nov 7	nov 9	3d
Análisis y selección de la información	nov 9	nov 15	5d
Elaboración del contenido	nov 15	nov 22	6d
Tecnicas de desanonimizacion	**nov 23**	**dic 12**	**14d**
Recopilación información	nov 23	nov 27	3d
Análisis y selección de la información	nov 27	nov 30	4d
Elaboración del contenido	nov 30	dic 12	9d
Shadow y ExperimenTOR	**dic 12**	**dic 21**	**8d**
Recopilación información	dic 12	dic 13	2d
Análisis y selección de la información	dic 14	dic 17	2d
Elaboración del contenido	dic 17	dic 21	5d
Caso práctico	dic 21	dic 28	6d
Conclusiones	dic 28	ene 2	4d

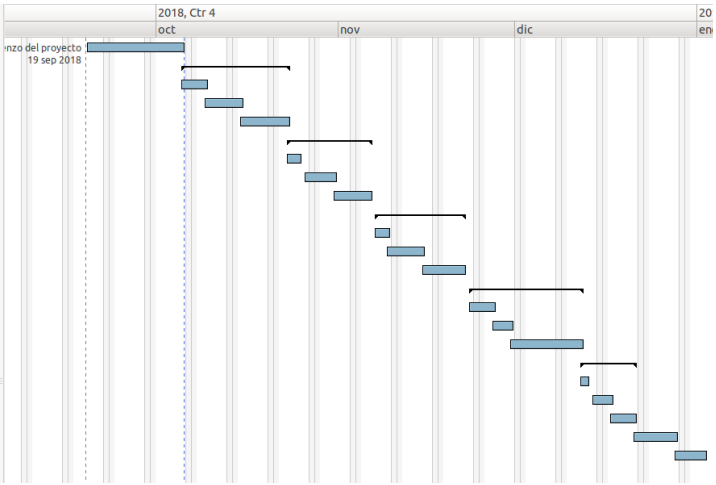


Figura 1: Planificación del trabajo

1.5 Estado del arte de la red Tor

La primera versión alfa de software libre para navegar por la red TOR se anunció el 20 de septiembre de 2002. De ahí surgió TOR como la evolución del proyecto Onion Routing del Laboratorio de Investigación Naval de los Estados Unidos. Actualmente TOR pertenece a The Tor Project, una organización sin ánimo de lucro afincada en Massachusetts y liderada por Roger Dingledine .

Como curiosidad y con el objetivo de conocer el número de usuarios de TOR en el mundo, la universidad de Oxford publicó un cartograma que muestra esta información entre agosto 2012 y julio 2013, según los datos de Tor Metrics, que son de dominio público.

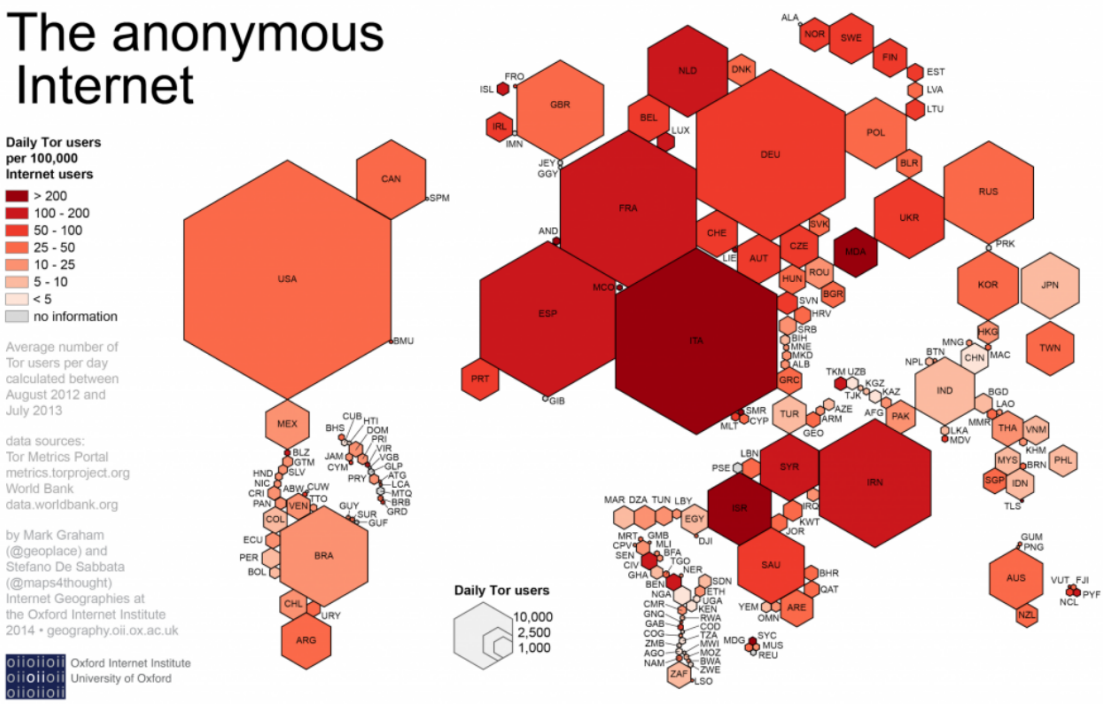


Figura 2: Número de usuarios de Tor en el mundo

Según la figura 2, se puede ver que más de la mitad de los usuarios de Tor se encuentran en Europa, que es también la región con la mayor penetración, ya que el servicio es utilizado por un promedio de 80 por cada 100.000 usuarios europeos de Internet.

Entre 2012 y 2013 Italia ocupa el segundo lugar después de los Estados Unidos en términos de número promedio de usuarios, ya que más de 126,000 personas acceden a Internet a través de Tor cada día desde los Estados Unidos. El servicio es popular en toda la región europea, con una alta penetración en Moldavia en particular.

Al observar el número de usuarios de Tor, como porcentaje de la población más grande de Internet, Medio Oriente y África del Norte tiene la segunda tasa más alta de uso, con un promedio de más de 60 por cada 100,000 usuarios de Internet que utilizan el servicio. Tor es particularmente popular en Israel, que cuenta con más usuarios de Tor que India, mientras que tiene menos del 4% de sus usuarios de Internet. El servicio también es muy popular en Irán, que representa el mayor número de usuarios de Tor fuera de Europa y los Estados Unidos, y cuenta con un 50% más de usuarios que Reino Unido, a pesar de tener solo un tercio de su población de Internet.

Sin embargo, si analizamos el último año, entre octubre 2017 y octubre 2018, según Tor Metrics los países que más usuarios usaron la red Tor vienen encabezados por Alemania y EEUU, seguidos de los Emiratos Arabes, Rusia, Ucrania, Francia, Indonesia, Paises Bajos, Reino Unido e India.

Country	Mean daily users
Germany	536434 (19.96 %)
United States	418092 (15.56 %)
United Arab Emirates	317226 (11.80 %)
Russia	253708 (9.44 %)
Ukraine	119128 (4.43 %)
France	117911 (4.39 %)
Indonesia	78745 (2.93 %)
Netherlands	75233 (2.80 %)
United Kingdom	66327 (2.47 %)
India	41789 (1.55 %)

Figura 3: Países con más usuarios conectados a Tor

En el último año podemos ver en la figura 4 que se muestra a continuación que se han conectado a la red Tor entre dos y casi cuatro millones y medio de usuarios de manera diaria a nivel mundial, siendo la segunda quincena de diciembre 2017 y enero 2018 los meses más significativos.

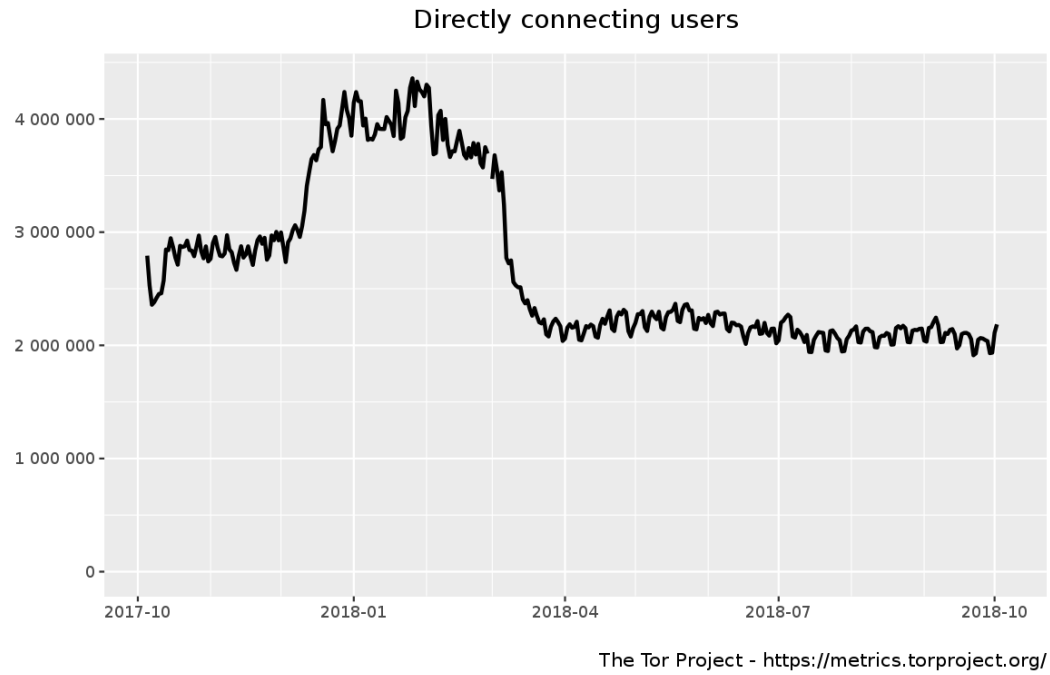


Figura 4: Usuarios conectados a Tor a nivel mundial

En España concretamente se puede ver que en el ultimo año se han conectado entre 20.000 y 40.000 usuarios siendo los meses más destacados los de octubre 2017 y abril 2018. Podría investigarse si en esos meses se produjeron algunos eventos relevantes en el país.

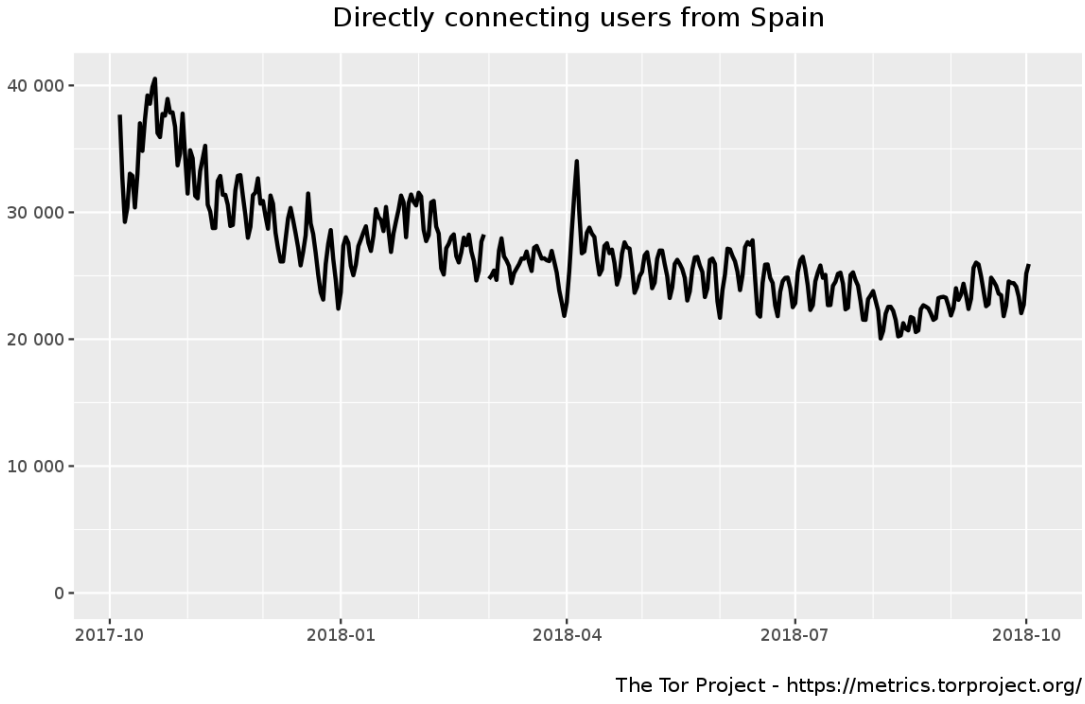


Figura 5: Usuarios conectados a Tor en España

2 Arquitectura y funcionamiento de Tor

2.1 Componentes Tor

La arquitectura básica de la red Tor está constituída por los siguientes componentes:

- **Clientes Tor (OP o Onion Proxy):** Un cliente Tor instala un software local considerado como un proxy onion, que empaqueta los datos de la aplicación en celdas del mismo tamaño (512 bytes) que envía a la red Tor. Una celda es la unidad básica de transmisión de Tor.

- **Nodo onion (OR o Onion Router):** Los nodos onion transmiten las celdas procedentes del cliente y del servidor Tor. Hay tres tipos de nodos onion: nodos de entrada, nodos intermedios y nodos de salida.

- **Servidores de directorio:** Los servidores de directorio almacenan la información de los routers onion y los servidores onion (hidden services), como por ejemplo, sus claves públicas.

- **Servidores onion (hidden servers):** Soportan las aplicaciones TCP como un servicio web o un servicio IRC.

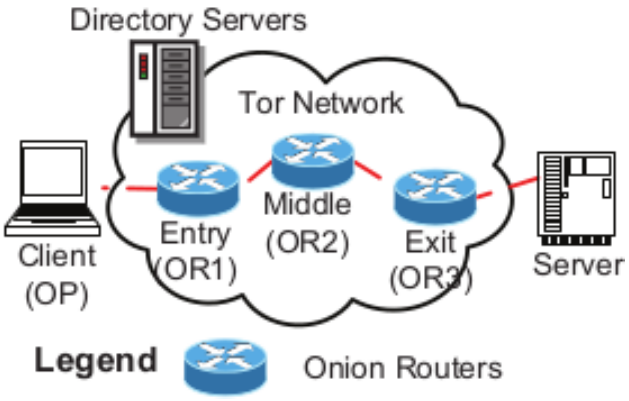


Figura 6: Arquitectura básica de la red Tor

2.2 Formato de una celda de Tor

A continuación se explica el formato de una celda de Tor. Los tres primeros bytes de la celda no están cifrados, de esta manera el router de Tor pueda leer esta cabecera. Los dos primeros bytes corresponden con el identificador del circuíto y el tercer byte se usa para indicar el comando específico de esta celda.

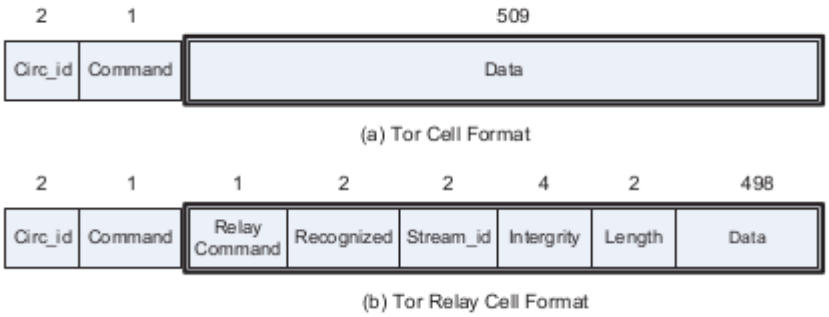


Figura 7: Formato de una celda de Tor

Hay dos tipos de celdas Tor como se puede ver en la figura: la celda de control (formato a) y la celda relay (formato b). La función de una celda de control es la de establecer un circuito nuevo o la de derribarlo, mientras que la función de una celda relay es la de transmitir los datos de la aplicación.

2.3 Protocolos

2.3.1 Protocolo de introducción: selección del circuito y su creación

La figura 8 ilustra el procedimiento para que un cliente construya un circuito. Como se muestra en la figura, el cliente primero establece una conexión TLS con el nodo de entrada utilizando el protocolo TLS. Luego el cliente envía una celda CELL CREATE a través de la conexión TLS y utiliza el protocolo de handshake Diffie-Hellman (DH) para negociar una clave base $K_1 = g^{xy}$ con el nodo de entrada, que responde con una celda CELL CREATED.

$H(K_1)$ es el valor hash de K_1 en la figura 8. Desde esta clave base, se generan una clave simétrica forward kf_1 y una clave simétrica backward kb_1. De esta manera, se crea el primer salto de este circuito, denotado como C1. Del mismo modo, el cliente extiende el circuito para incluir el segundo salto (C2) y el tercer salto (C3) del circuito.

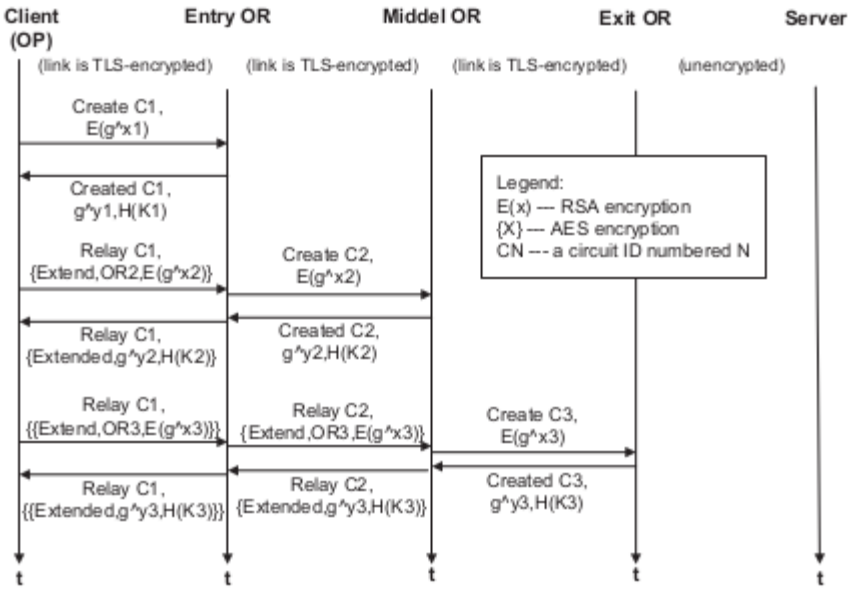


Figura 8: Proceso de creación de un circuito

Si describimos la tercera conexión para crear el tercer salto del circuíto, podemos observar que el cliente primero envía una celda relay al primer nodo con doble capa de incriptado AES con

el contenido de la clave g^{x3} encriptada con cifrado RSA. A continuación el nodo de entrada desencripta la primera capa y envía la celda relay al segundo nodo del circuíto sólo con una capa de cifrado AES, de manera que el segundo nodo desencripta la capa de cifrado que queda y envía la celda de control al tercer nodo, con la clave g^{x3} encriptada con cifrado RSA.

2.3.2 Protocolo de transmisión de datos

La figura 9 muestra el procedimiento de la transmisión de datos a través del circuito. Una vez establecido el circuito, el cliente envía una celda RELAY COMMAND BEGIN al nodo de salida encriptándolo de la siguiente manera: $\{\{\{Begin < IP, Port >\}_{kf_1}\}_{kf_2}\}_{kf_3}$

kf_1 indica la clave usada para encriptar la primera capa, kf_2 corresponde con la segunda capa y kf_3 con la tercera capa.

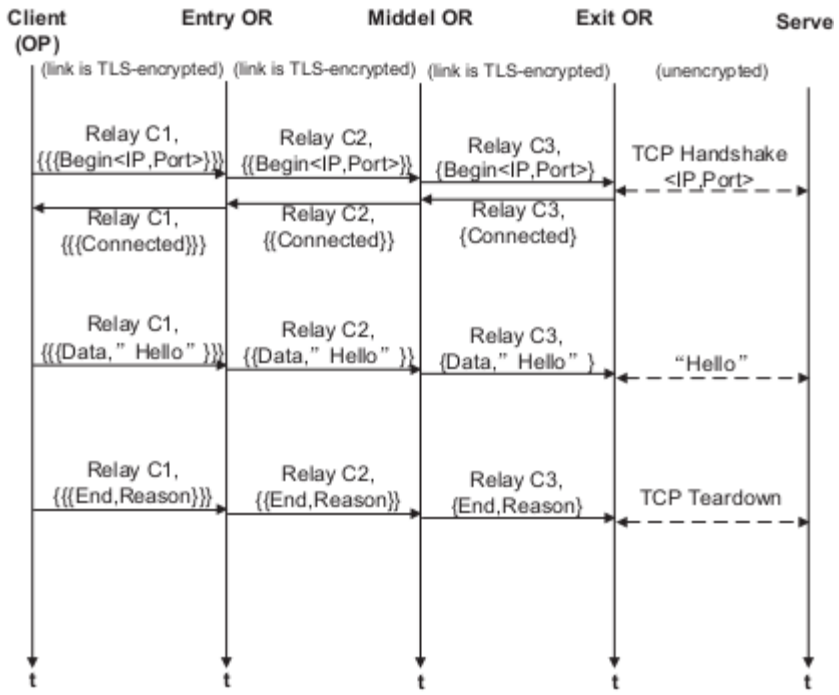


Figura 9: Proceso de transmisión de datos

Las tres capas de piel cebolla se van eliminando una a una cada vez que la celda pasa por un nodo onion a través del circuíto, de la siguiente manera: primero se elimina la capa exterior al ser desencriptada por el nodo de entrada (guard relay o entry OR), después se elimina la capa intermedia en el nodo intermedio (middle relay o middle OR) y la tercera capa, la capa interior, se elimina en el nodo de salida (exit relay o exit OR).

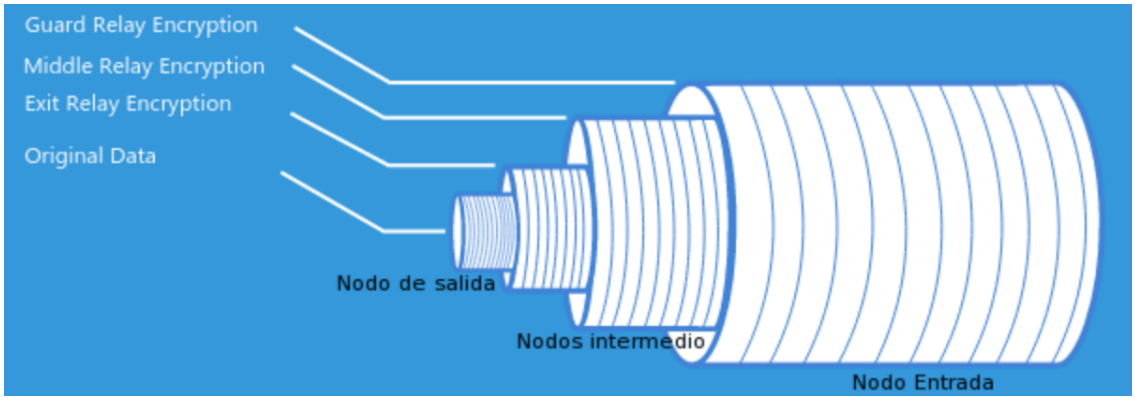


Figura 10: Capas de cebolla

Como se ilustra en la figura 9 cuando el nodo de salida elimina a través del descifrado la última capa, identifica que la celda es una petición que solicita abrir una transmisión TCP a un puerto en la IP destino que apunta al servidor remoto. Por lo tanto, el nodo de salida actúa como un proxy, construyendo una conexión TCP con el servidor y enviando al cliente una celda RELAY COMMAND CONNECTED.

2.3.3 Protocolo rendezvous

El protocolo rendezvous de Tor o protocolo de encuentro se actualizó a la version 3 en octubre 2017, y los servicios onion pasan de ser de 16 caracteres a 56 caracteres, con el aspecto siguiente:

dgrhgjyitohltiguyifi58396058495867ghtuyjbmcvdgr65he389ol.onion

Elementos del protocolo rendezvous:

- **Puntos de introducción (introduction points o IPO)**: son nodos que son seleccionados por el servicio onion y se publican en el servidor de directorios a través del descriptor creado por el servidor onion.

- **Puntos de encuentro (rendezvous points o RPO)**: son nodos elegidos por el cliente Tor. Tanto el cliente como el servicio onion establecerán un circuito de tres saltos al RPO, quien actuará como nodo que transmitirá los mensajes por el circuito correspondiente al servicio onion por un lado y por el otro, por el circuito correspondiente al cliente Tor. En el lado del cliente, en el circuíto de tres nodos, el tercer nodo corresponde con el RPO.

- **HSDir o Directorios de servicios onion (Hidden Service Directories):** son nodos Tor que almacenan extractos firmados por los servidores de servicios onions para que los usuarios puedan contactar con ellos.

- **Servidor onion (hidden server):** un servidor onion sirve varias aplicaciones TCP como por ejemplo, como servidor web o servidor IRC.

- **Cliente** – el software Tor que corre en el ordenador del usuario.

Tor hace posible que sus usuarios oculten sus ubicaciones al tiempo que ofrecen diversos tipos de servicios, como la publicación de una página web o un servidor de mensajería instantánea. Usando los "puntos de encuentro" (rendezvous points) de Tor, otros usuarios de Tor pueden conectarse a estos servicios onion, antes conocidos como servicios ocultos (hidden services), de forma que ni el cliente ni el servidor del servicio onion pueda conocer la identidad de red del otro.

A continuación se describe los detalles técnicos de cómo funciona este protocolo de encuentro.

Paso 1: Un servicio onion necesita anunciar su existencia en la red Tor antes de que los clientes puedan contactarlo. Por lo tanto, el servicio elige aleatoriamente algunos nodos, les construye circuitos y les pide que actúen como puntos de introducción al decirles su clave

pública. Para construir un circuito a los puntos de introducción, el servidor onion les enviará una celda RELAY COMMAND ESTABLISH INTRO y los puntos de introducción responderán con una celda RELAY COMMAND INTRO ESTABLISHED para informar al servidor onion que el circuito se ha establecido.

Al utilizar un circuito Tor completo, es difícil para cualquiera asociar un punto de introducción con la dirección IP del servidor onion. Si bien a los puntos de introducción y a otros puntos se les dice la identidad del servicio onion (clave pública), no se desea que aprendan sobre la ubicación del servidor onion (dirección IP).

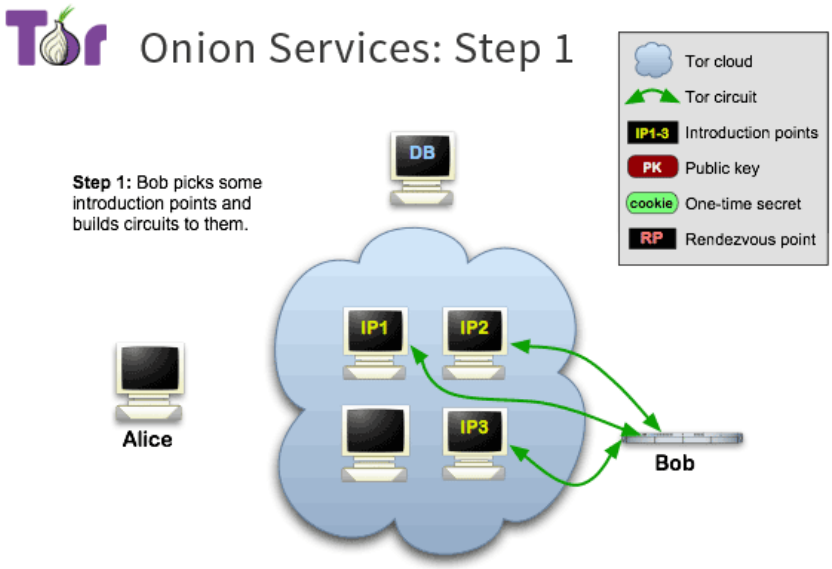


Figura 11: Paso 1 Protocolo rendezvous

Paso dos: el servicio onion crea un descriptor de servicio onion, que contiene su clave pública y la lista de puntos de introducción, y firma este descriptor con su clave privada. A continuación el servidor onion sube ese descriptor a una tabla hash que distribuye a un conjunto de nodos HSDir, con los que se conecta a través de otro circuito anónimo con tres saltos como mínimo. Una vez que ya se ha subido el descriptor a un servidor de directorios, el propietario del servicio onion puede ya publicitar la dirección onion para atraer usuarios a su servicio.

Los clientes solicitarán entonces la dirección onion XYZ.onion donde XYZ es un nombre de 16 o 56 caracteres derivado de la clave pública del servicio onion. Aunque puede parecer poco práctico utilizar un nombre de servicio generado automáticamente, tiene un objetivo importante: todos, incluidos los puntos de introducción, el directorio de tablas hash distribuidas y, por supuesto, los clientes, pueden verificar que están hablando con el servicio onion correcto.

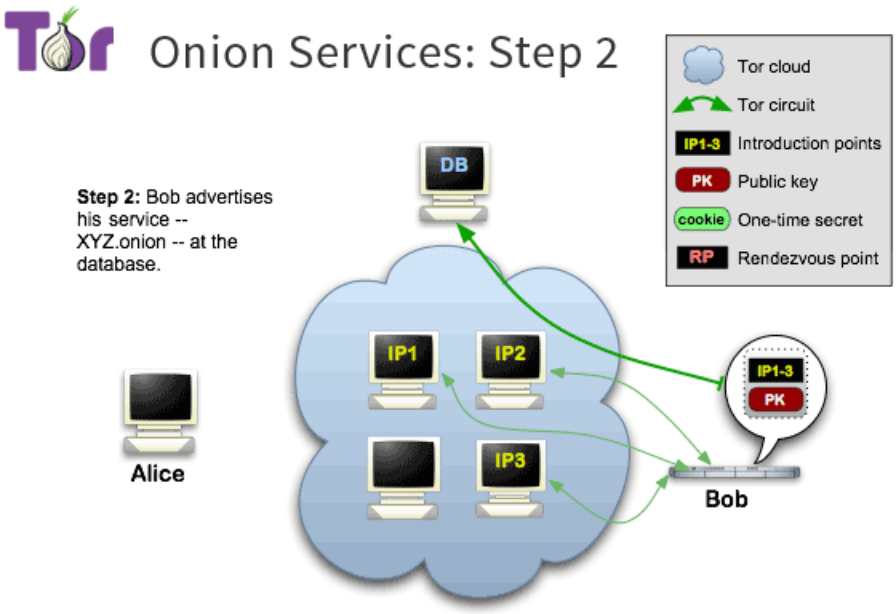


Figura 12: Paso 2 Protocolo rendezvous

Paso tres: Una vez que el cliente conoce la dirección onion del servicio al que se quiere conectar, habrá de descargarse el descriptor de la tabla hash distribuida del HSDir, para poder iniciar el establecimiento de la conexión con el servidor onion. Para ello el cliente necesita crear primero un circuito al servidor de directorio y si hay un descriptor para XYZ.onion (el servicio onion también podría estar desconectado o haberse dejado de servir hace ya tiempo, o podría haber un error tipográfico en la dirección onion), el cliente podrá conocer el conjunto de puntos de introducción y la clave pública correcta que necesitará utilizar.

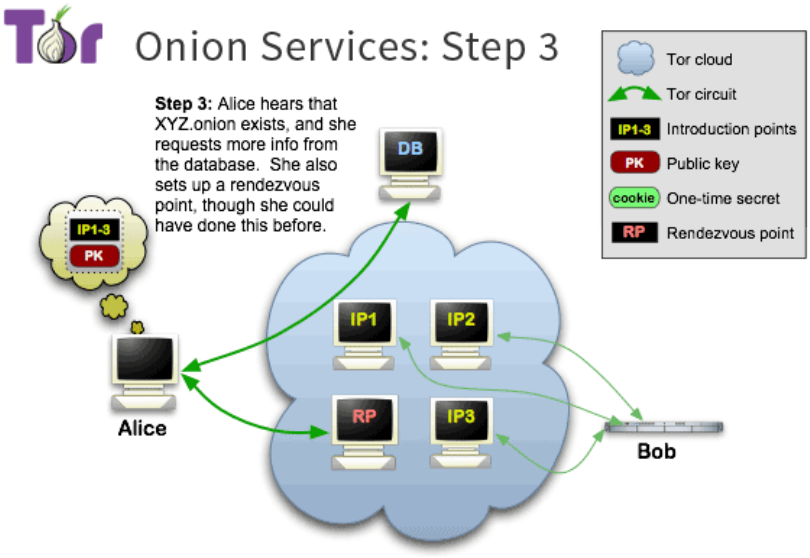


Figura 13: Paso 3 Protocolo rendezvous

Por otro lado, el cliente también crea un circuito anónimo a otro nodo escogido al azar y le pide que actúe como punto de encuentro (rendezvous point) al enviarle la celda RELAY COMMAND ESTABLISH RENDEZVOUS, la cual contiene una cookie rendezvous, que será empleada como una clave de un solo uso (one-time secret).

El punto de encuentro contestará con una celda RELAY COMMAND RENDEZVOUS ESTABLISHED para indicar que el establecimiento del circuito fue exitoso.

Paso cuatro: cuando el descriptor está presente y el punto de encuentro está listo, el cliente ensambla un mensaje de introducción cifrado con la clave pública del servicio onion. Este mensaje es una celda RELAY COMMAND INTRODUCE1 que incluye la dirección del punto de encuentro, la cookie rendezvous y la primera parte de un handshake criptográfico, una clave Diffie-Hellman g^x generada por el cliente Tor. El cliente entonces envía este mensaje a uno de los puntos de introducción solicitando que se entregue al servicio onion. Nuevamente, la comunicación se realiza a través de un circuito Tor de tres nodos: nadie puede relacionar el envío del mensaje de introducción a la dirección IP del cliente, por lo que el cliente permanece en el anonimato.

En el momento en que el punto de introducción recibe la celda RELAY COMMAND INTRODUCE1, contesta al cliente con una celda RELAY COMMAND INTRODUCE ACK. Cuando el cliente recibe la celda ACK, derriba este circuito hacia el punto de introducción. Entonces, el punto de introducción empaqueta la celda RELAY COMMAND INTRODUCE1 en una celda RELAY COMMAND INTRODUCE2 y envía la celda RELAY COMMAND INTRODUCE2 al servidor onion.

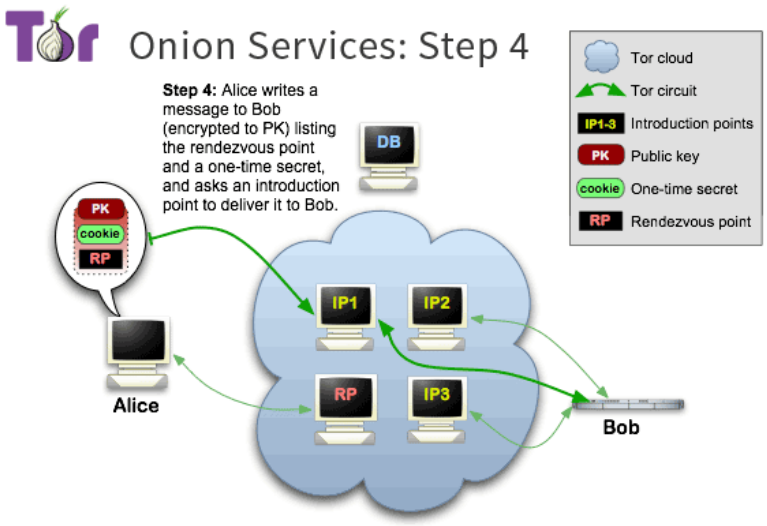


Figura 14: Paso 4 Protocolo rendezvous

Paso cinco: Cuando el servicio onion recibe la celda RELAY COMMAND INTRODUCE2, descifra el mensaje de introducción del cliente donde encuentra la dirección del punto de encuentro, la cookie rendezvous y el dato Diffie-Hellman g^x. Ahora es cuando el servidor onion genera el dato Diffie-Hellman g^y y deriva la clave $K=g^{xy}$. Esta es la clave que se usa para el cifrado punto a punto entre el cliente y el servidor. Después, el servidor onion crea un circuito anónimo al punto de encuentro y le envía un mensaje de encuentro en una celda RELAY COMMAND RENDEZVOUS1 que incluye la clave g^y , el valor hash de la clave K, es decir H(K) y la cookie rendezvous.

En este punto, es de especial importancia que el servicio onion se adhiera al mismo conjunto de nodos de entrada al crear nuevos circuitos. De lo contrario, un atacante podría ejecutar su propio nodo y forzar a un servicio onion a crear un número arbitrario de circuitos con la esperanza de que el nodo corrupto sea elegido como nodo de entrada y conozcan la dirección

IP del servidor onion a través del análisis de temporización. Este ataque fue descrito por Øverlier y Syverson en su artículo titulado Locating Hidden Servers.

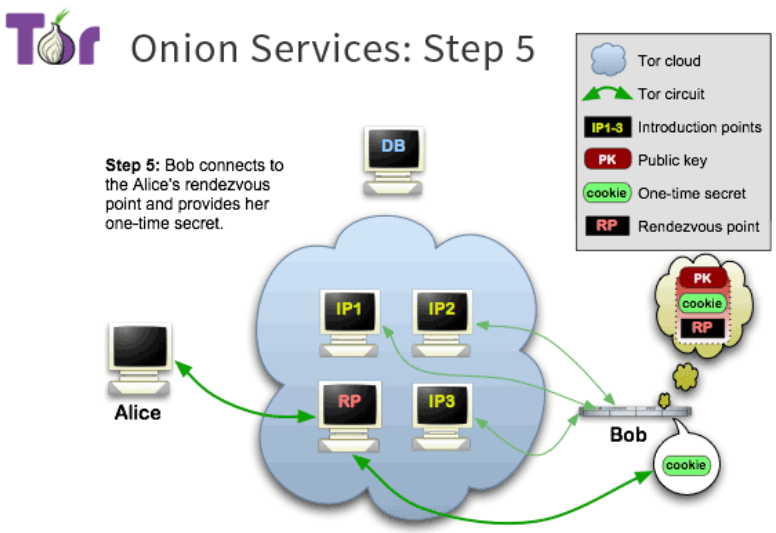


Figura 15: Paso 5 Protocolo rendezvous

En el último paso, el punto de encuentro obtiene la celda RELAY COMMAND RENDEZVOUS1 enviada por el servidor onion y compara la cookie rendezvous de esta celda con la que tiene procedente del cliente Tor de la celda que recibió en el paso tres, la celda RELAY COMMAND ESTABLISH RENDEZVOUS. Si las cookies coinciden, el punto de encuentro elimina la cookie de la celda RELAY COMMAND RENDEZVOUS1 y empaqueta el resto de los datos en una celda RELAY COMMAND RENDEZVOUS2 que envía al cliente.

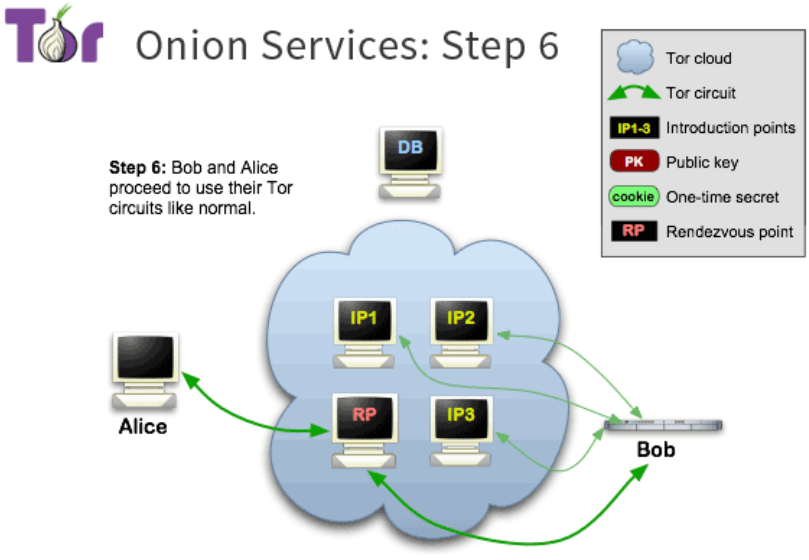


Figura 16: Paso 6 Protocolo rendezvous

Cuando el cliente recibe la celda RELAY COMMAND RENDEZVOUS2, desempaqueta la celda para obtener la celda RELAY COMMAND RENDEZVOUS1 que aún contiene la clave g^y y el hash H(K) del paso cinco, que fueron generados por el servidor onion. Entonces puede generar la clave $K=g^{xy}$ usando g^y y verificar que es correcto basándose en el hash H(K). Es de esta manera que el cliente y el servidor onion completan el handshake. Después de eso, tanto

el cliente como el servicio onion pueden usar sus circuitos hasta el punto de encuentro para comunicarse entre sí, utilizando esta clave K compartida entre el cliente y el servicio onion, El punto de encuentro simplemente retransmite los mensajes (cifrados de extremo a extremo con K) del cliente al servicio y viceversa.

Para empezar entonces con el comienzo de la comunicación entre el cliente y el servidor, el cliente envía una celda RELAY COMMAND BEGIN al servidor a través del circuito de seis saltos.

Por tanto, la conexión completa entre el cliente y el servicio onion consta de 6 nodos: 3 de ellos fueron seleccionados por el cliente, el tercero es el punto de encuentro y los otros 3 fueron seleccionados por el servicio onion.

En la figura siguiente se puede observar todo este proceso de manera más esquemática y concretamente el número de nodos implicados en cada circuito.

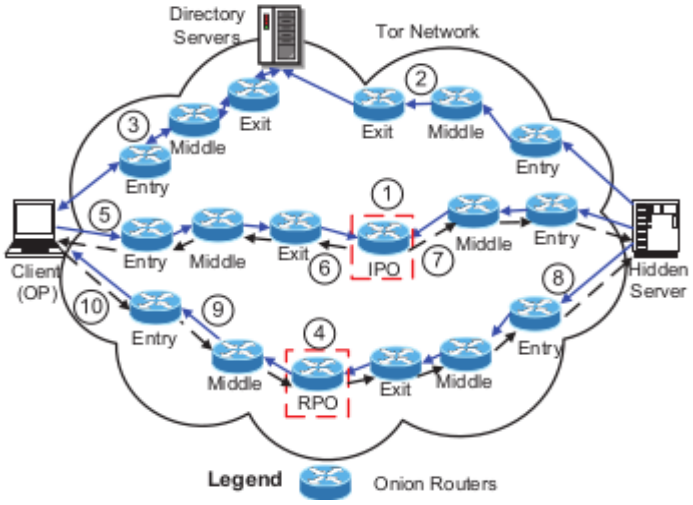


Figura 17: Proceso completo de las conexiones en Tor

Circuito 1: conexión entre el servidor onion y el punto de introducción IPO.

Circuito 2: conexión entre el servidor onion y el servidor de directorios.

Circuito 3: conexión entre el cliente y el servidor de directorios y viceversa.

Circuito 4: conexión entre el cliente y el punto de encuentro RPO.

Circuito 5: conexión entre el cliente y el punto de introducción IPO.

Circuito 6: conexión entre el punto de introducción IPO y el cliente.

Circuito 7: conexión entre el punto de introducción IPO y el servidor onion.

Circuito 8: conexión entre el servidor onion y el punto de encuentro RPO.

Circuito 9: conexión entre el punto de encuentro RPO y el cliente.

Circuito 10: conexión entre el cliente y el punto de encuentro RPO.

3 Instalación de Tor

En octubre 2018 la última versión de Tor Browser a instalar es Tor Browser 8.5a4.

La Instalación del navegador de Tor se realiza desde https://www.torproject.org/ pero en algunos países la website de Tor Project está bloqueada o censurada y no es posible descargarse Tor directamente, por lo que el Tor Project almacena un mirror del Tor Browser en GitHub: https://github.com/TheTorProject/gettorbrowser

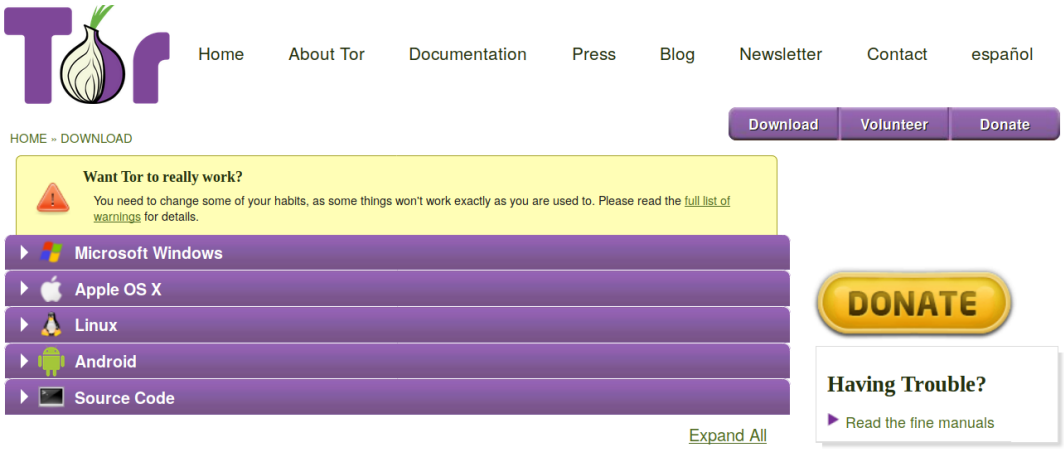


Figura 18: Descarga de Tor Browser

Tor es gratis y de código abierto para Windows, Mac, Linux/Unix y Android. En dispositivos Android, Tor es mantenido por Guardian Project. Actualmente, no hay una forma compatible de usar Tor en iOS, pero Guardian Project está trabajando para hacer que esto sea una realidad en el futuro.

No sólo se puede acceder a la red Tor a través del Tor Browser sino también a través de proxy-web como en el caso de Tor2web, accediendo desde un navegador estándar de Internet añadiendo a las direcciones .onion terminaciones como .to, .city, .cab y .direct pero de esta manera se pierden las características de privacidad y anonimato del usuario que navega. No es el caso de los servicios ocultos, que siguen permaneciendo ocultos, ya que Tor2web al funcionar como un proxy reverso, se encarga de enrutar todas las peticiones entrantes desde la "clear web" a la "deep web" de Tor.

Otra manera de usar Tor es a partir de Tails, una distribución Linux centrada en la privacidad y el anonimato que se ejecuta desde un pendrive USB, un DVD o una tarjeta SD, y no deja rastro alguno en el ordenador en que se utiliza ya que no hace uso del disco duro del host. Todos los datos y ficheros que maneja están encriptados, y todas las conexiones a Internet se encriptan y se transmiten a través de Tor.

Tails tiene la habilidad de hacerse pasar por una interfaz gráfica como Windows XP por lo que no levanta sospechas en zonas públicas y viene con un software pre-instalado: HTTPS, OpenPGP, Pidgin OTR, Truecrypt y KeePassX.

Para instalar Tor se descarga el fichero apropiado según el sistema operativo desde https://www.torproject.org/ Después se ejecuta uno de los dos comandos siguientes para extraer el paquete:

#tar -xvJf tor-browser-linux32-8.0.3_LANG.tar.xz

o (para la versión de 64 bits):

#tar -xvJf tor-browser-linux64-8.0.3_LANG.tar.xz

(donde LANG es el idioma en el que está el fichero).

Una vez hecho, nos emplazamos en el directorio del Tor browser con:
#cd tor-browser_LANG

Para ejecutar el Tor Browser, sólo hay que hacer click en el Tor Browser o el icono de setup del Tor Browser o ejecutar el fichero start-tor-browser.desktop en la terminal:

#./start-tor-browser.desktop

Esto lanzará el Tor Launcher y una vez que se conecta a Tor, lanzará Firefox. No se debe de desempaquetar o ejecutar TBB (Tor Browser Bundle) como root.

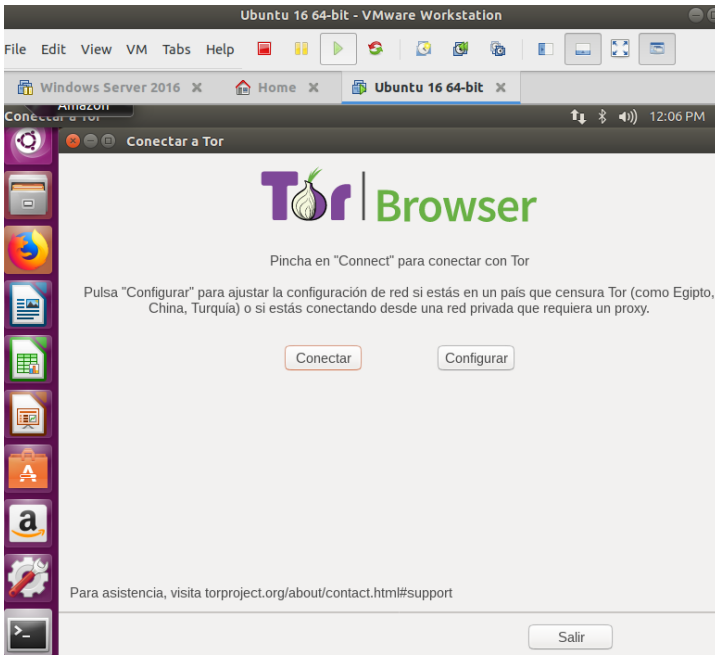


Figura 19: Instalación de Tor Browser 1

Figura 20: Instalación de Tor Browser 2

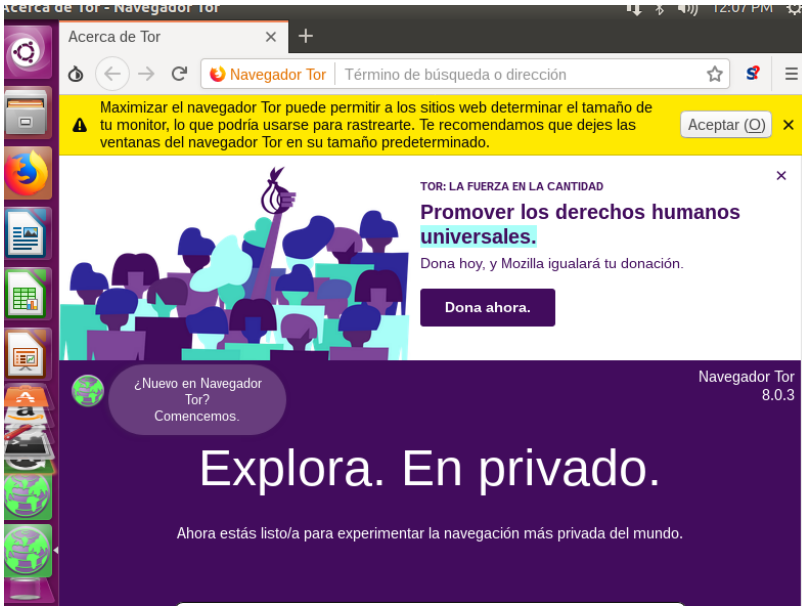


Figura 21: Navegador de Tor desde Firefox

3.1 Puertos abiertos para uso de Tor como cliente

Tor puede intentar conectarse a cualquier puerto que se anuncie en el directorio como un ORPort (para hacer conexiones Tor) o un DirPort (para obtener actualizaciones del directorio). Hay una variedad de estos puertos: muchos de ellos se ejecutan en 80, 443, 9001 y 9030, pero también se usan otros puertos.

Cuando se usa Tor como cliente, probablemente se podría abrir solo esos cuatro puertos, pero para obtener la mayor diversidad en sus nodos de entrada, y por lo tanto la mayor seguridad, así como la mayor robustez en su conectividad, lo recomendable es permitir que se conecte con todos ellos.

3.2 Requisitos para actuar como nodo Tor

Los requisitos para actuar como nodo Tor dependen del tipo de nodo (de salida o no salida) y del tipo de banda que tienen. En resumen, se ha de disponer de un ancho de banda aceptable, una dirección Ip fija y enrutable publicamente, 200 Mb de almacenamiento de disco, una CPU moderna y 1 GB de RAM para un nodo de no salida.

3.2.1 Ancho de banda y conexiones

- Un nodo de no salida debe poder manejar al menos 7000 conexiones simultáneas. Esto puede abrumar a los routers de nivel de consumidor pero en el caso de que se ejecute el nodo Tor desde un servidor (virtual o dedicado) en un centro de datos no habrá problemas. Los nodos de salida rápidos (> = 100 Mbit / s) generalmente tienen que manejar muchas más conexiones simultáneas (> 100k).

- Se recomienda que un nodo tenga disponible para Tor un ancho de banda de carga de al menos 16 Mbit/s (Mbps) y el mismo valor para la descarga. Los requisitos mínimos

para un nodo son 10 Mbit/s (Mbps). Si tiene menos de 10 Mbit/s (Mbps) pero al menos 1 Mbit/s, se recomienda ejecutar un puente con soporte obfs4.

3.2.2 Dirección IPv4 pública

Con respecto a este punto, cada nodo necesita una dirección IPv4 pública, ya sea directamente en el host (preferido) o mediante NAT y reenvío de puertos. Por otro lado, no se requiere que la dirección IPv4 sea estática, pero se prefieren las direcciones IP estáticas. Además, se necesita que la dirección IPv4 permanezca sin cambios durante al menos 3 horas (ya que lleva tiempo distribuir la nueva lista de IPs de los nodos a los clientes, lo que sucede una vez cada hora).

La conectividad IPv6 adicional es excelente y se recomienda, pero no es un requisito. Es de interés saber que solo se pueden ejecutar dos nodos Tor por dirección IPv4 pública.

3.2.3 Requisitos de Memoria

- Un nodo de no salida de <40 Mbit/s debe tener al menos 512 MB de RAM disponible.
- Un nodo de no salida más rápido que 40 Mbit/s debe tener al menos 1 GB de RAM.
- En un nodo de salida, se recomienda al menos 1.5 GB de RAM por instancia de Tor.

3.2.4 Almacenamiento de disco

Tor no necesita mucho almacenamiento en disco. Un nodo Tor típico necesita menos de 200 MB para datos relacionados con Tor.

3.2.5 CPU

- Cualquier CPU moderna es aceptable.
- Se recomienda utilizar CPU con soporte AESNI (que mejorará el rendimiento y permitirá hasta unos 400-450 Mbps en cada dirección en una sola instancia en las CPU modernas). Si el archivo /proc/cpuinfo contiene la palabra aes, indica que la CPU tiene soporte para AESNI.

3.2.6 Tiempo de actividad

Tor no tiene requisitos de tiempo de actividad, pero si el nodo no se ejecuta durante más de 2 horas al día, su utilidad es limitada. Lo ideal es que el nodo se ejecute en un servidor que funcione 24/7.

3.3 Ejemplo de circuíto Tor

Para ver un ejemplo de circuito Tor con los seis nodos intermedios entre el cliente y el servidor, se selecciona un dominio onion, por ejemplo xmh57jrzrnw6insl.onion, que corresponde con un buscador en Tor llamador Torch a partir del cual se pueden encontrar sitios conocidos de Tor.

Como se puede ver en la figura siguiente, las direcciones IP y país que ha visitado el navegador Tor Browser son:

1.- Francia 195.154.164.243
2.- Alemania 5.9.44.29
3.- Alemania 87.118.122.120

Los tres siguientes nodos no se sabe su IP porque corresponde con el circuíto rendezvous que crea el servidor onion. El último host del circuíto es el servidor onion.

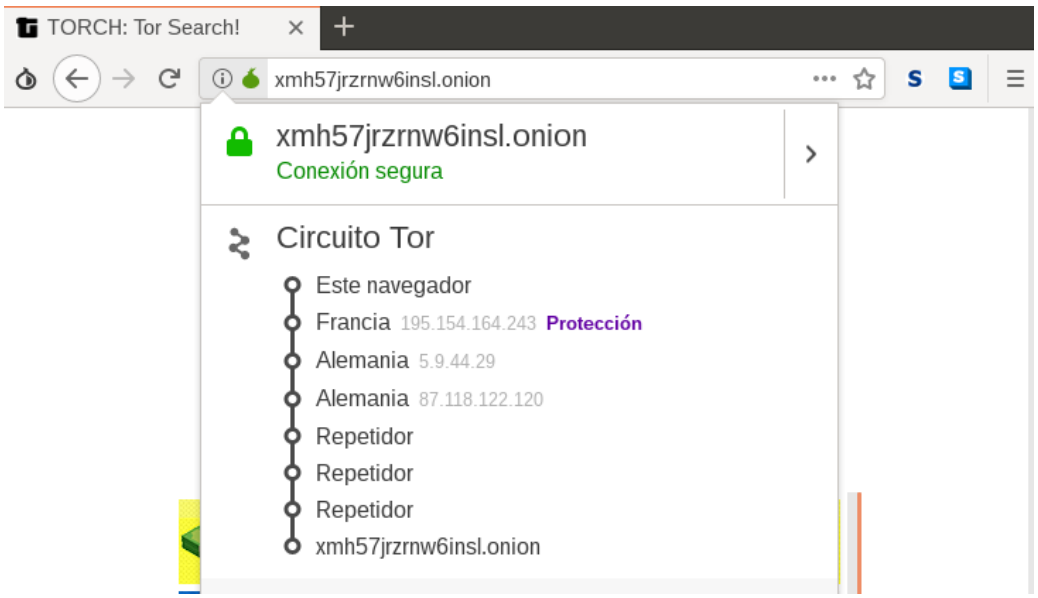


Figura 22: Ejemplo de circuito Tor

A través de la herramienta Wireshark podemos ver también el protocolo de handshake entre nuestro host (192.168.1.50) y el primer host del circuíto (195.154.164.243).

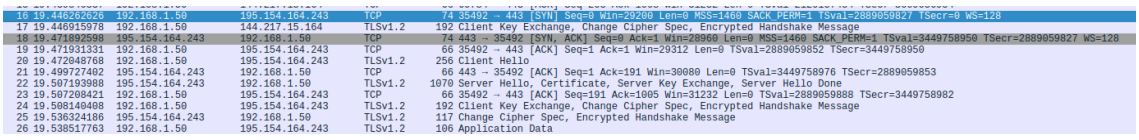


Figura 23: protocolo de handshake en Wireshark con el primer nodo

Además de conexiones con servicios onion, el cliente puede utilizar Tor para acceder a sitios web de la clearnet, cuyos servidores no están dentro de la red Tor.

4 Servicios en Tor

Existe un buscador en Tor llamador Torch a partir del cual se pueden encontrar sitios conocidos de Tor. Direccion Torch: xmh57jrzrnw6insl.onion

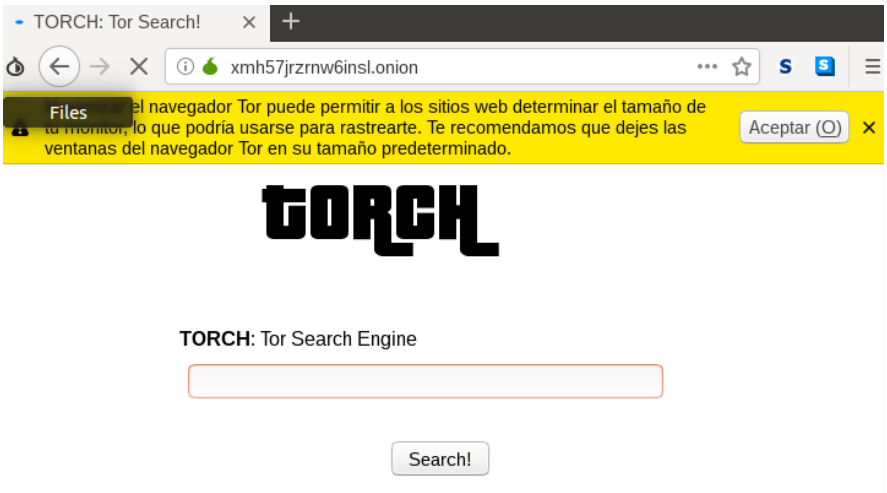


Figura 24: Buscador Torch

En las imágenes siguientes se puede ver algunos de los servicios que se anuncian en este buscador.

Figura 25: Servicios ocultos anunciados en Torch

Algunos de estos servicios son:

- **Black Market Guns:** un mercado negro donde venden y compran armas de fuego
- **Counterfeiting center:** un centro para obtener dinero falso, documentos de identidad falsos, pasaportes falsos, etc

- **Silk Market 3:** un mercado negro donde se venden drogas, pastillas, setas, semillas, etc

- **Krush Market:** otro mercado negro donde se sirven drogas, porno, falsificaciones, números CCV de tarjetas de crédito…

- **Credit card dumps:** volcados de tarjeta de crédito. Esto es una copia digital no autorizada de toda la información contenida en la banda magnética de una tarjeta de crédito activa, creada con la intención de hacer ilegalmente una tarjeta de crédito falsa que los ciberdelincuentes pueden utilizar para realizar compras.

- **Legal Teen Porn:** portal de pornografía de adolescentes

- **Most Popular Tor Sites:** los sitios más populares de Tor

Como se puede ver en la figura siguiente, Tor, además de poder servir de vehículo de expresión para ciudadanos que viven en sociedades opresivas y con censura o aquellos que desean expresarse libremente de manera anónima, se emplea para varios tipos de delitos: blanqueo de dinero, abuso sexual infantil, venta de armas, malware, falsificación documental, falsificación de monedas, venta de drogas, pagos de rescates, carding, venta de productos robados, venta de productos de farmacia, propaganda yihadista, doxing, sicarios...

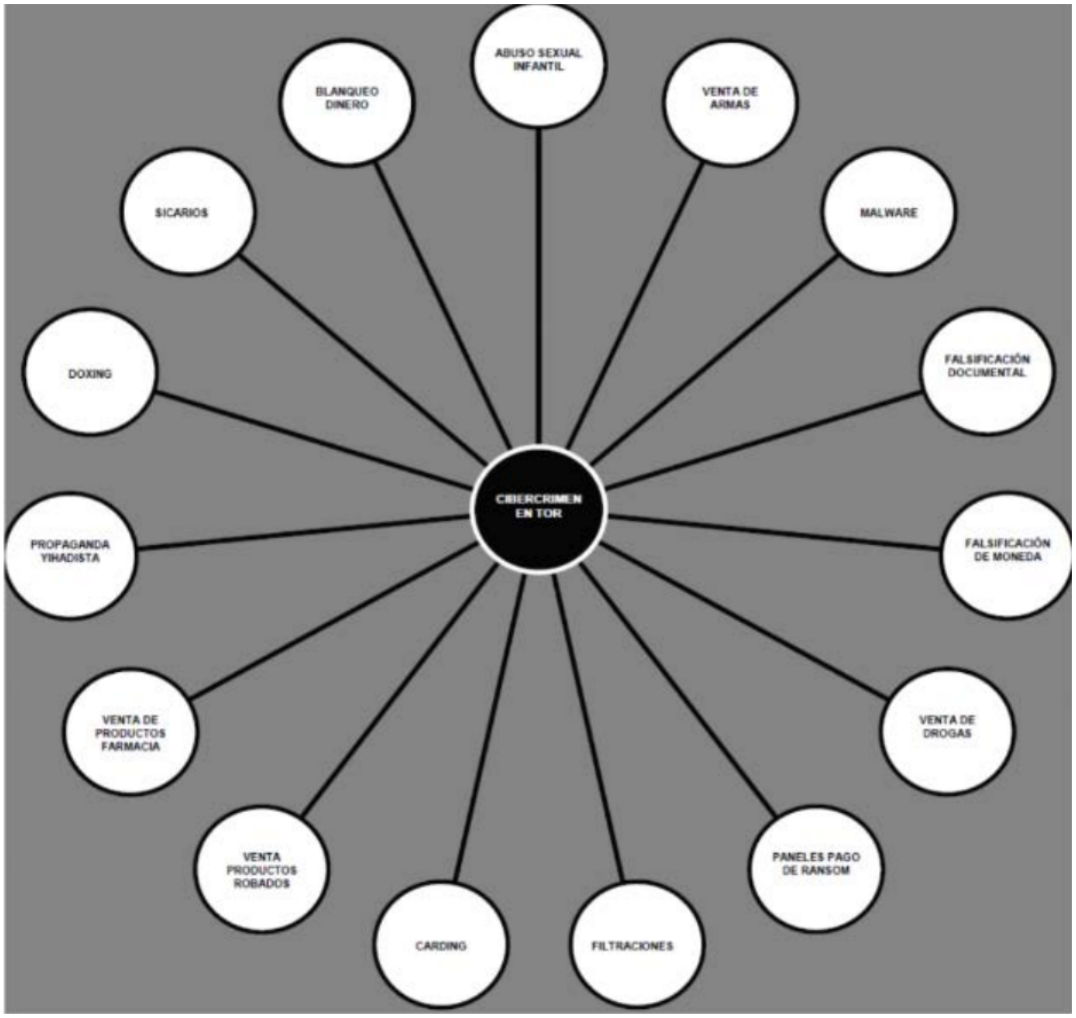


Figura 26: Delitos en Tor

4.1 Blanqueo de dinero

Los blanqueadores de dinero utilizan el bitcoin para blanquear dinero obtenido con otros delitos, puesto que no existe ninguna autoridad u organismo que supervise esta criptomoneda y aunque las transacciones quedan registradas y se pueden rastrear, en principio no permiten identificar al titular real del dinero.

Para convertir los bitcoins en dinero real, los blanqueadores tienen varias opciones, como por ejemplo el uso de web exchangers, que permiten el cambio de moneda virtual por otras de curso legal. Otro método es el uso de servicios de trading o de intercambio, que permiten la compra y venta de bitcoins como si fueran acciones.

Un ejemplo de sitio onion dedicado al negocio de blanqueo de dinero fue Liberty Reserve, que fué clausurado por las autoridades en mayo 2013.

4.2 Abuso sexual infantil

La red Tor debido al carácter anónimo que presenta, se convierte en el foro ideal para la pederastia, en el que criminales de todo el mundo pueden compartir su material bajo diferentes sobrenombres.

Algunos sitios onion dedicados a la pornografía infantil fueron: PedoBook (cerrado en diciembre 2012), Freedom Hosting (desarticulado en agosto 2013), The Love Zone (incautado en diciembre 2014), PlayPen (clausurado en febrero 2015), Giftbox Exchange (desarticulado en noviembre 2016), Elysium (desmantelado en junio 2017) y Child's Play (intervenido en septiembre 2017).

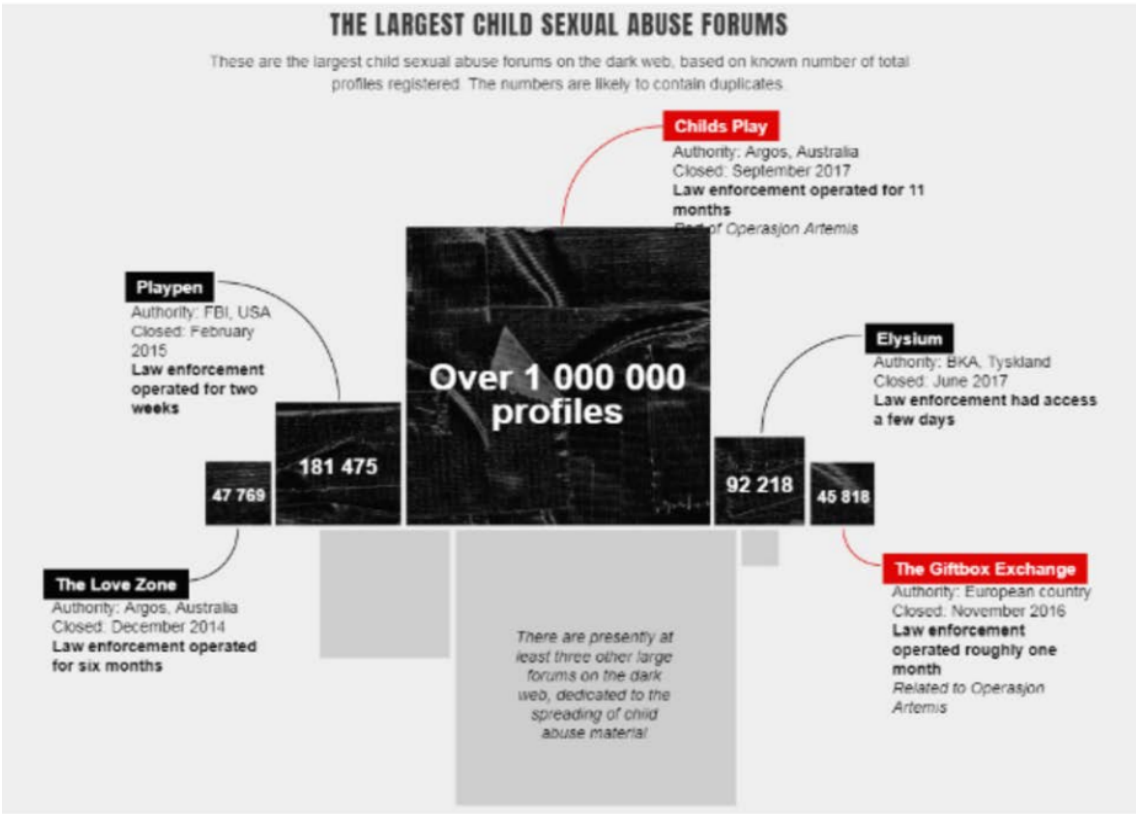


Figura 27: Sitios onion de pornografía infantil clausurados en los últimos años

4.3 Venta ilegal de armas

Enviar a los clientes archivos digitales con contenido ilegal es relativamente sencillo, sobre todo si estos van cifrados. El problema es enviar material físico ilegal, sobre todo si éste tiene que cruzar el charco. Como es de esperar, la gran mayoría de armas ilegales que se venden en Tor vienen de Estados Unidos, y algunas de ellas se envían a Europa.

Los vendedores utilizan diversos métodos para enviarlas, pero el más común y seguro es desarmarlas en muchas partes diferentes y enviarlas en paquetes separados, escondiéndolas en dispositivos como minicadenas, altavoces o impresoras para no levantar sospechas.

Figura 28: Ejemplo de arma de fuego desmontada

El porcentaje de armas vendidas en Tor es ligeramente inferior al 1% de todo lo que se vende en esta red. Además de armas físicas, también hay todo tipo de manuales sobre cómo hacer bombas y explosivos caseros.

Del total de productos vendidos en Tor referentes a armas, el 40% son pistolas, y el 30% es material digital como manuales para hacer explosivos o planos 3D para hacer pistolas con impresoras 3D.

4.4 Malware

Los cibercriminales han encontrado en Tor un lugar donde alojar *malware*, aprovechando justamente las posibilidades de anonimato que ofrece. De este modo, cualquier tráfico generado por el *malware* que intente ser analizado y capturado por autoridades o investigadores, resultará complicado de rastrear para capturar a los ciberdelincuentes.

Así, se pudo ver cómo el troyano Agent.PBI utilizaba el protocolo HTTP mediante la red de Tor para realizar acciones como la descarga de códigos maliciosos adicionales desde Internet u otras ubicaciones remotas, ejecutar archivos y actualizarse a versiones más recientes. También se pudo observar cómo crecían las *botnets* basadas en Tor y los usuarios dentro de ellas.

Otros ejemplos de malware fueron MACSPY, MacRamson, Karmen Ramsonware Raas, Ramsonware-as-a-Service.

En Tor se venden desde troyanos, botnets, exploits, exploit kits, keyloggers, scripts para realizar DDoS y software para hackear móviles para controlar y acceder a información como el historial de llamadas, mensajes de texto y más.

4.5 Falsificación documental y de dinero

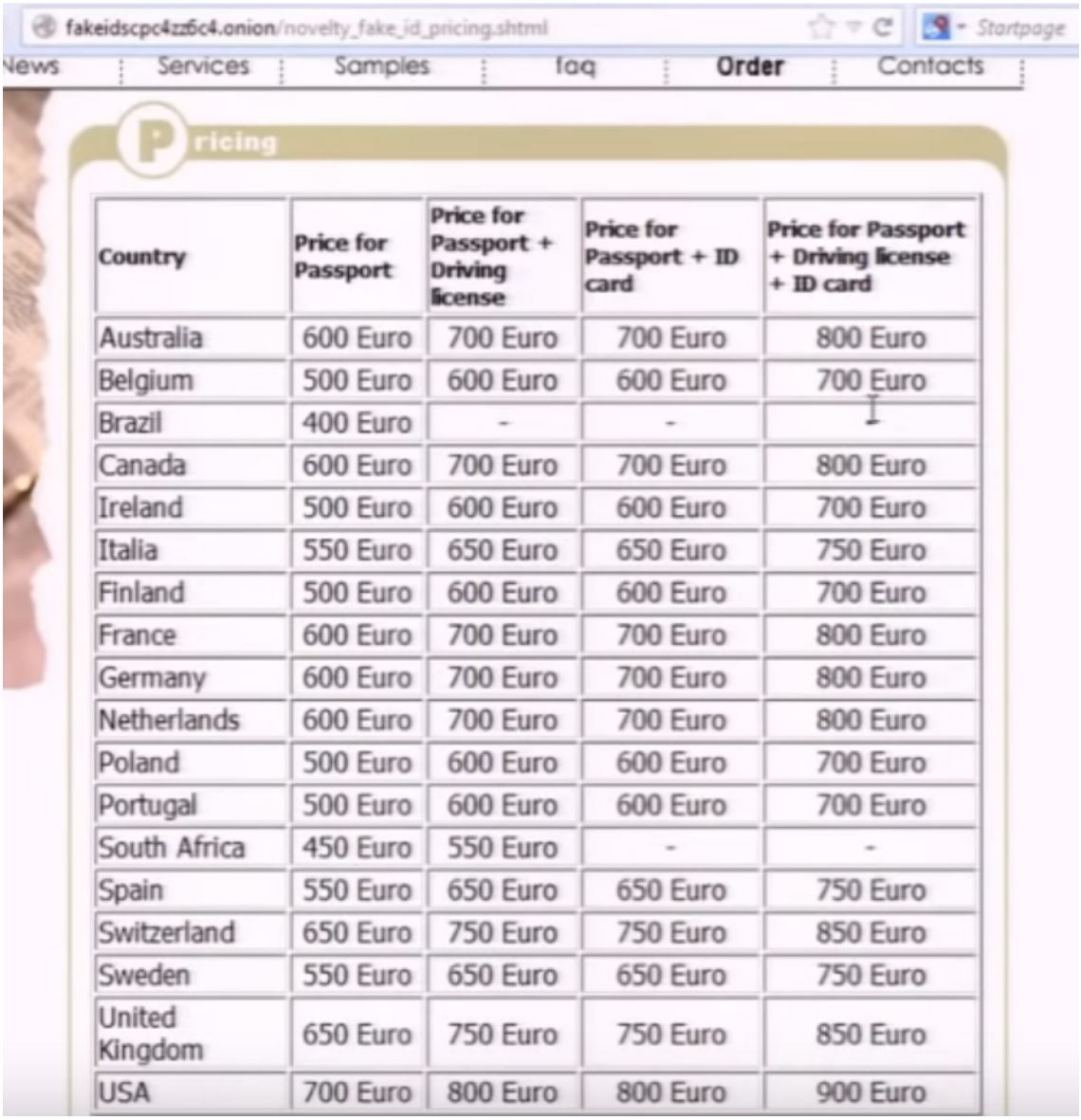
Conseguir un pasaporte falso en Tor no es difícil, pero tampoco es barato. Los precios varían, dependiendo del país del que se quiera el pasaporte. Muchas veces depende también de cómo de buena tenga que ser la falsificación, en función de los detectores que tenga cada país para su detección, para que parezca original.

En las siguientes figuras se puede ver ejemplos de falsificación de pasaportes y su precio. Así, por ejemplo, un pasaporte falso de EEUU cuesta 1000 dolares, mientras que uno de Japón son 700 dolares.

Figura 29: Oferta de pasaportes falsos

Además de los pasaportes también se ofrecen falsificaciones como pueden ser carnés de conducir o incluso dinero.

En la siguiente figura se puede ver que se venden pasaportes falsos de España por 550 euros y que incluso se ofrece un paquete que incluye pasaporte, DNI y carnet de conducir por 750 euros.



Country	Price for Passport	Price for Passport + Driving license	Price for Passport + ID card	Price for Passport + Driving license + ID card
Australia	600 Euro	700 Euro	700 Euro	800 Euro
Belgium	500 Euro	600 Euro	600 Euro	700 Euro
Brazil	400 Euro	-	-	
Canada	600 Euro	700 Euro	700 Euro	800 Euro
Ireland	500 Euro	600 Euro	600 Euro	700 Euro
Italia	550 Euro	650 Euro	650 Euro	750 Euro
Finland	500 Euro	600 Euro	600 Euro	700 Euro
France	600 Euro	700 Euro	700 Euro	800 Euro
Germany	600 Euro	700 Euro	700 Euro	800 Euro
Netherlands	600 Euro	700 Euro	700 Euro	800 Euro
Poland	500 Euro	600 Euro	600 Euro	700 Euro
Portugal	500 Euro	600 Euro	600 Euro	700 Euro
South Africa	450 Euro	550 Euro	-	-
Spain	550 Euro	650 Euro	650 Euro	750 Euro
Switzerland	650 Euro	750 Euro	750 Euro	850 Euro
Sweden	550 Euro	650 Euro	650 Euro	750 Euro
United Kingdom	650 Euro	750 Euro	750 Euro	850 Euro
USA	700 Euro	800 Euro	800 Euro	900 Euro

Figura 30: Empresa que ofrece pasaportes, carnets de conducir y documentos de identidad falsos

4.6 Venta de drogas

Al más puro estilo de plataformas como Ebay, diferentes mercados negros ofrecen una gran variedad de sustancias y estupefacientes ilícitos, divididas por categorías: esteroides, disociativos, estimulantes, etc. Drogas como cocaína, burundanga o hachís figuran entre los muchos productos disponibles. Aunque también medicación como oxidocona o xanax se ofertan en este tipo de portales. Y no sólo sustancias, también se venden materias primas y utensilios para el abastecimiento de los productores de drogas.

Uno de los mercados más populares fue Silk Road, que aunque se desmanteló su versión 2.0 en 2013 con la detención de su creador, otros mercados han seguido su estela y se han hecho con sus clientes y vendedores. Cabe decir que Silk Road resurgió a principios del 2017 con su versión 3.0.

Si bien es cierto que este tipo de sitios web cambian y desaparecen a menudo, existen páginas que recogen y actualizan los diferentes mercados negros disponibles, actualizándolas con

bastante frecuencia. Además, muchos de estos sitios web han establecido un sistema por invitación, de manera que necesitas una para poder acceder a los servicios que ofrecen y realizar transacciones en dicho sitio, de esta manera pueden llevar cierto control y publicitarse solo en aquellos sitios que consideren seguros para sí mismos.

4.7 Pagos de rescates

Un tipo de malware bastante utilizado en los últimos años es el conocido como ransomware. Este software malicioso infecta un determinado PC elegido por el ciberdelincuente y le da la capacidad a éste de bloquearlo desde una ubicación remota e incluso encriptar los archivos, quitándole el control de toda la información y datos almacenados al usuario, al que normalmente el delincuente devolverá el control e información requisada siempre a cambio de una cantidad monetaria estimada por el ciberdelincuente. La comunicación entre la máquina infectada y el servidor de mando y control es a través de Tor, lo que le hace muy difícil de analizar o desmontar.

Por lo general, los afectados deben pagar un rescate equivalente a algunos cientos de euros en Bitcoin u otra criptomoneda para poder liberar sus archivos. A menudo el coste aumenta con el tiempo hasta llegar a una fecha límite en la que, se supone, se destruyen los archivos.

Un ejemplo de ramsonware es CryptoWall, una familia de ransomware diseñada para usar un algoritmo de cifrado sofisticado para hacer que los archivos sean inaccesibles en las computadoras seleccionadas. Los investigadores de malware detectaron la primera versión de este ransomware en 2013. Desde entonces, el cripto virus fue actualizado varias veces. Mientras que algunas versiones se pueden descifrar de forma gratuita; otras son aún irrompibles en 2018.

Los investigadores asumen que el virus CryptoWall Locker ha sido desarrollado por el mismo grupo de estafadores que podrían ser acusados de la creación de CryptoDefense, Cryptolocker, BitCrypt, Critroni y Cryptorbit.

A continuación se puede ver una tabla del valor en dólares y euros que generaron los siguientes ransomwares:

Ransomware	**Rescate recibido en Bitcoin**	**Valor en dólares**	**Valor en euros**
CryptoWall	5.351,2329	2.220.909,12	1.804.055,58
CryptoLocker	1403,7548	449.274,97	364.948,30
DMA Locker	339,4591	178.162,77	144.722,51
WannaCry	47,1743	86.076,76	69.920,58
CryptoDefense	126,6960	63.859,49	51.873,38
NotPetya	4,0576	9.835,86	7.989,72
KeRanger	9,9990	4.173,12	3.389,85

Figura 31: Beneficios obtenidos por diferentes ransomwares

4.8 Filtraciones

El robo y posterior revelación de información confidencial y secreta a diferentes gobiernos y empresas por parte de varios grupos de hackers ha sido uno de los motivos de la creciente popularidad de estos medios, especialmente de la darknet.

Ejemplos de esto son casos, como por ejemplo el de Wikileaks o el caso Snowden en el año 2013, en el que un antiguo consultor e informante de la CIA y NSA, desveló información secreta a través de diferentes periódicos de documentos clasificados como alto secreto sobre varios programas de las organizaciones con las que trabajó y colaboró.

Otros caso es el robo de datos de la empresa Sony en abril 2011, en el que muchos usuarios vieron cómo los datos de sus cuentas de PlayStation eran publicadas en Internet.

4.9 Carding

Carding se refiere al robo de datos de tarjetas de crédito para luego venderlos. Estos datos pueden ser obtenidos mediante hacking, phishing o ingeniería social. Hay dos tipos de carding: carding físico y carding virtual:

Carding Físico: Consiste en conseguir la información de las bandas magnéticas (dumps) de las tarjetas de crédito o débito. Mediante el uso de un falso lector, se roba la información para luego montar esos dumps en plásticos en blanco, logrando así tener una copia idéntica a la tarjeta propia del titular. De esta manera se podrá realizar compras en centros comerciales o retirar directamente dinero de los cajeros de los bancos.

Carding Virtual: Consiste en usar los datos de tarjetas bancarias, para realizar compras de bienes o servicios en tiendas online a lo largo de la red. Para esto es necesario tener una configuración con la cual se pueda emular al titular de la tarjeta bancaria, pero sobre todo que garantice el anonimato para evitar ser rastreado.

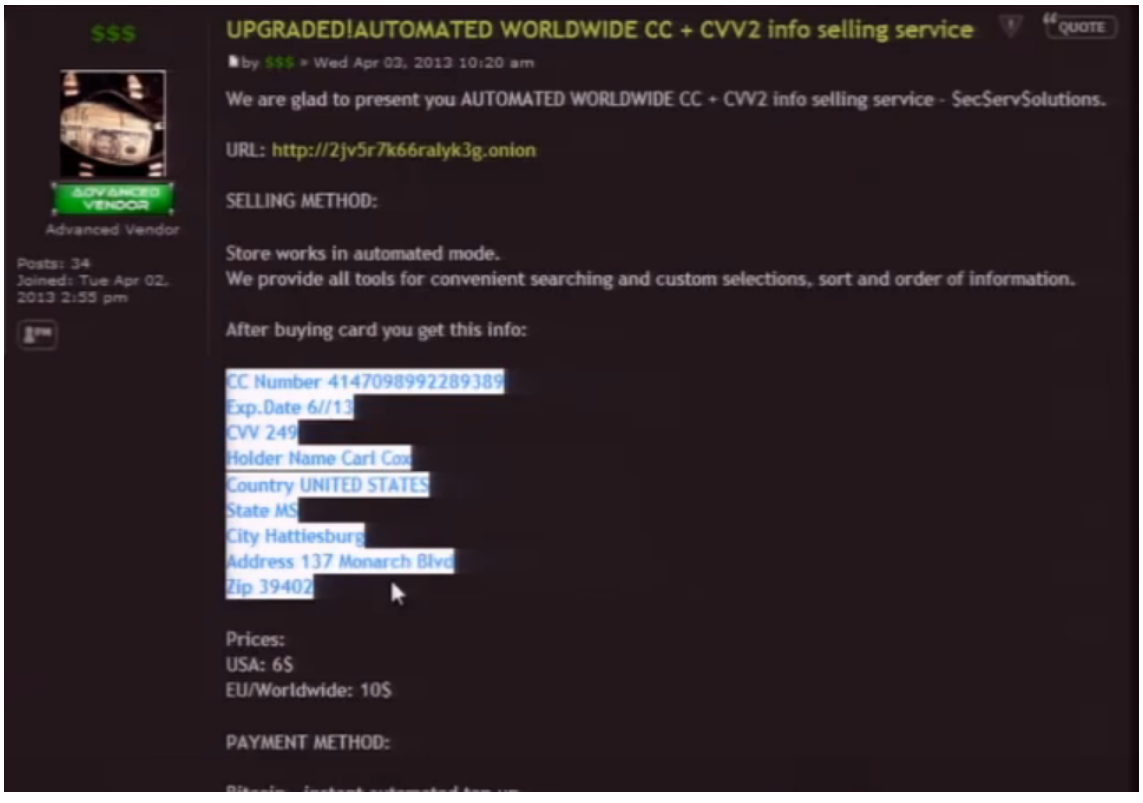


Figura 32: Carding virtual

Algunos sitios de carding son: Uncle Sam, Scrooge McDuck, Mr. Bin, Try2Swipe, Popeye, y Royaldumps.

En la mayoría de los sitios de carding, la compra de las tarjetas requiere primero ponerse en contacto con el propietario de las tiendas directamente a través de un mensaje instantáneo.

En el año 2016 se cerró uno de los sitios más grandes de carding del mundo, CarderPlanet.

4.10 Venta de productos robados

En Tor se pueden comprar distintos tipos de tecnología robada como productos Apple, televisores, ordenadores, Xbox, PS4 y otros artículos de electrónica. En esencia se vende cualquier producto cuyo valor económico de primera mano sea elevado, artículos en los que se requiera anonimato para su compra, de dudosa reputación o cuya demanda sea alta.

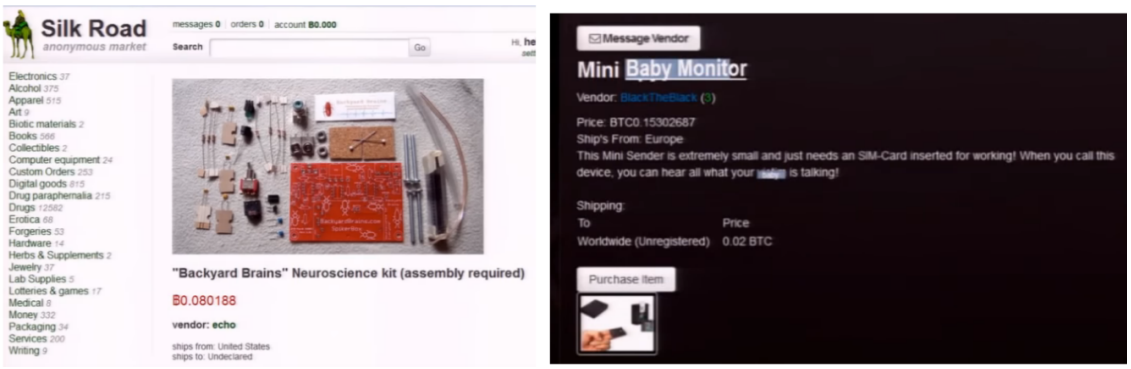


Figura 33: Ejemplos de compras en Tor (no necesariamente productos robados)

Uno de los productos más populares son contraseñas de cuentas robadas, por ejemplo de cuentas de Netflix, Spotify, Amazon, Paypal, cuentas de páginas pornográficas como Brazzers, etc. Los precios pueden variar desde menos de un euro hasta alrededor de los cien o doscientos euros.

Para evitar los fraudes en estas ventas los mercadillos online implementan un sistema de votos para calificar a los vendedores, por lo que incluso se pueden medir los riesgos a la hora de comprar.

Se venden incluso guías para robar cuentas de Paypal o servicios para que un hacker robe una cuenta en particular.

4.11 Venta de productos de farmacia

En febrero 2018 se desarticuló la mayor red de venta ilegal de medicamentos de disfunción eréctil y tratamiento adelgazante a través de internet que funcionaba en España. El grupo criminal utilizaba criptomonedas bitcoins para el pago a sus proveedores y así ocultar a la acción policial y judicial las transacciones monetarias. Los puntos de ventas principales eran prostíbulos y lugares de ocio.

El fármaco se enviaba de forma fraccionada, separando pastillas y cajas, para evitar el control aduanero y, además, no llegaban a España directamente desde la India (país donde se fabricaban) sino pasando por Suiza y Alemania.

Otra página que oferta algunos medicamentos para los cuales es necesaria una receta es la página Bitpharma. Algunos son extremadamente adictivos (como el Diazepam) y su consumo debe ser revisado constantemente por un especialista.

Otro ejemplo es la página Lion Pharma, que ofrece esteroides, isotretinoína, anastrozol, tadalafil…

Por otro lado, la página Peoples Drug Store oferta desde metacualona (un depresivo general del sistema nervioso central) hasta Suboxone, un medicamento orientado a la terapia de sustitución de los opiáceos para evitar el síndrome de abstinencia.

Los riesgos de comprar medicamentos en estos lugares son extremos, la mayoría ilegales y falsificados. El cliente se expone a sustancias que no están controladas por ningún organismo por lo que seguramente se recibe una falsificación que pueda producir graves daños en el organismo.

4.12 Propaganda yihadista y terrorismo

Grupos terroristas, especialmente yihadistas, utilizan Tor para congregar y comunicarse con sus militantes.

Sitios dedicados a la divulgación de la ideología yihadista, o incitando a alistarse al ejército para lo que ellos mismos denominan como la guerra santa, se alojan en Tor, incluso recaudan fondos mediante el uso de bitcoins. Existen sitios web que piden donaciones para su causa, y además enseña a los militantes y simpatizantes a comprar armas para la yihad en Tor.

Estas organizaciones terroristas, como por ejemplo Al Qaeda o el ISIS, suelen tener una web a modo de señuelo, alojada normalmente en la web superficial, y que contienen material sin importancia de divulgación y llamamiento a la ideología que defienden. Esta táctica es utilizada para que las fuerzas de seguridad crean que esas son sus páginas de difusión reales, mientras que las organizaciones operan en otras situadas en Tor, evitando o al menos intentando evitar que las autoridades lleguen hasta sus sitios web de importancia.

4.13 Doxing

En Tor se venden también servicios de doxing. Este servicio consiste en conseguir información pública y privada acerca de una persona, empresa, etc., siempre a través de Internet, con intenciones maliciosas. Esta información puede ser nombre, edad, sexo, nombre de los hermanos, padres, hijos, amigos o algún dato similar de la pareja.

También se puede dar el caso de que se consigan otros datos como número de la Seguridad Social, DNI, empleo, dirección, claves, números de teléfono, cuentas bancarias, si se tienen créditos hipotecarios, etc.

Al tener esta información en manos equivocadas puede tener efectos negativos sobre el dueño de esta información, por ejemplo:

- Robo de identidad
- Creación de documentos falsos (pasaportes, actas de nacimiento)
- Extorsión
- Robo
- Vender información
- Fraudes

4.14 Sicarios

Los asesinos a sueldo encuentran en la darknet un medio perfecto donde publicitarse y ofrecer sus servicios.

Los precios de este tipo de servicios varían en función de diferentes variables, como son el país de residencia (tanto del sicario como del objetivo), costes de desplazamiento, la importancia del objetivo o futura víctima (figuras públicas, políticos o empresarios importantes), el tiempo para la realización del trabajo, la edad del objetivo, etc.

Algunos ofrecen servicios extras por un extra de dinero, como por ejemplo el envío de una foto del cadáver al contratante del servicio. Otros incluso tienen algunas excepciones como no matar a menores de 16 años o políticos de altos cargos.

4.15 Otros delitos

A parte de los delitos nombrados, existen otros como por ejemplo las películas denominadas “snuff”, en las cuales se filman violaciones, torturas, suicidios o asesinatos, entre otros. También, aunque en menor medida, circula otro tipo de material que podría calificarse como poco de excéntrico, como vídeos de carácter escatológico o incluso “crush fetish”, en las cuales se puede ver el maltrato y asesinato de animales aplastándolos.

También se puede encontrar delitos de piratería, estafa, ciberacoso, extorsión, hacktivismo, etc.

4.16 Sitios populares en Tor

En la figura siguiente se puede ver algunas direcciones onion de los sitios más populares en Tor según el buscador Torch. A la derecha de la imagen se observa un cuadro con una clasificación de tipos de sitios para realizar búsquedas según la categoría: motores de búsqueda, wikis, servicios financieros, tiendas o servicios comerciales, drogas, redes sociales, foros, pornografía, mensajería, hacking, virus o cracks, carding, hosting, blogs, temas políticos, seguridad, apuestas, libros, warez, escrow o fideicomiso y contactos.

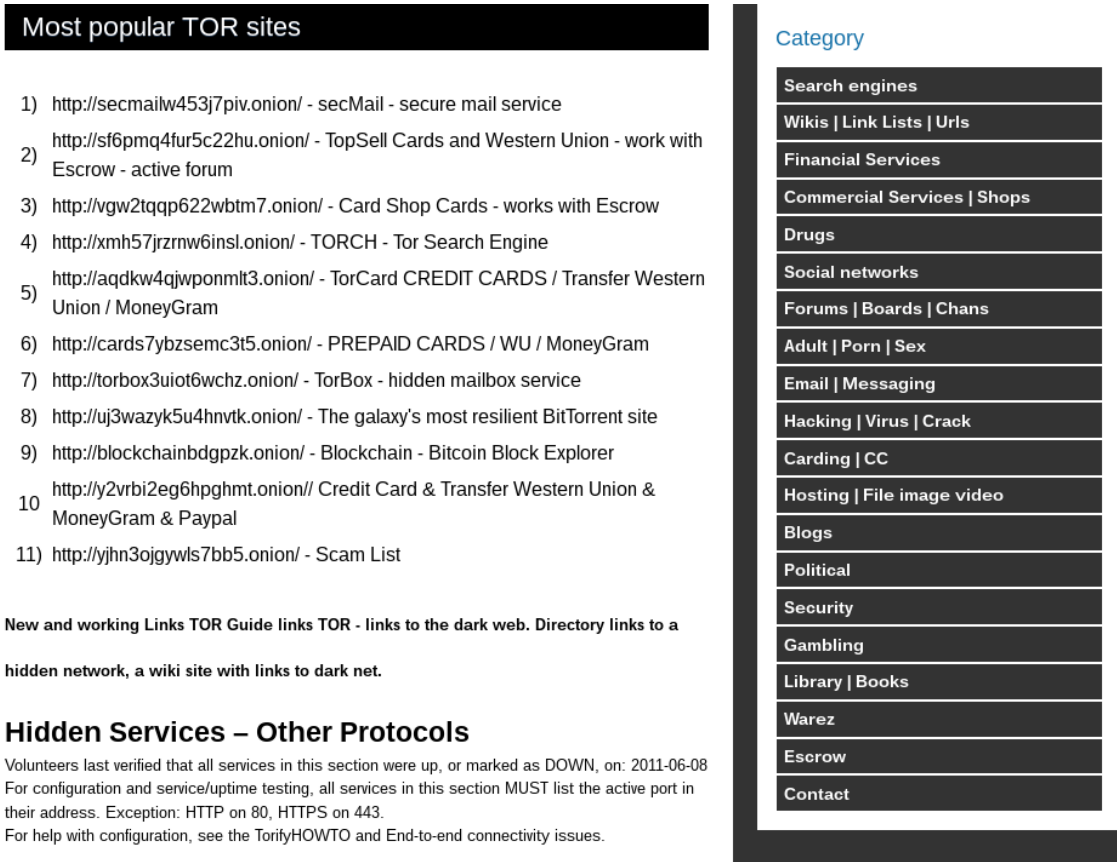


Figura 34: Sitios más populares de Tor según Torch

A continuación se pueden ver ejemplos de algunas de estas categorías en las siguientes figuras:

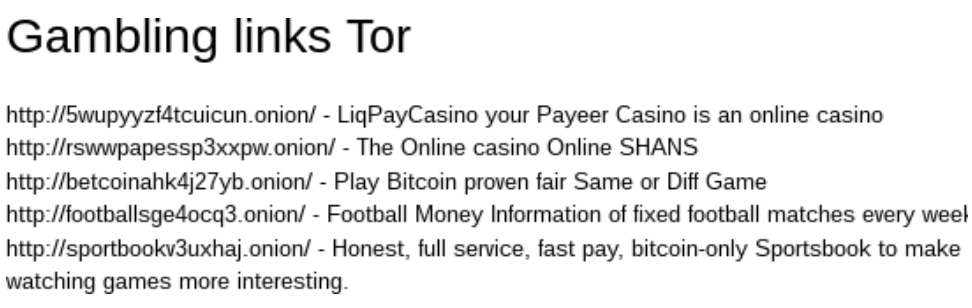


Figura 35: Enlaces sobre apuestas

Figura 36: Enlaces sobre temas de seguridad

Aquí vemos por ejemplo el dominio onion de Whonix, una distribución de seguridad basada en Debian.

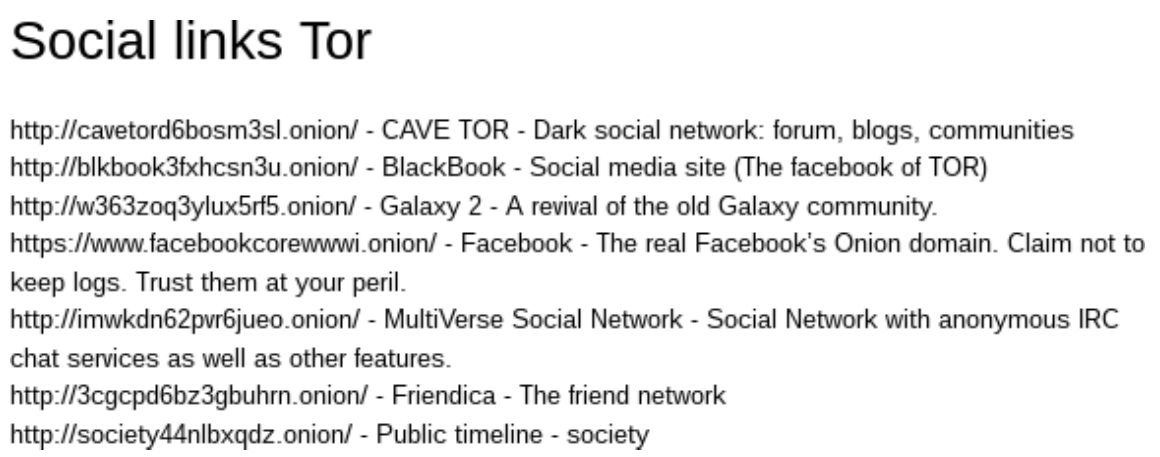


Figura 37: Enlaces a de redes sociales

Podemos ver cómo por ejemplo Facebook tiene su sitio en Tor, con una dirección onion reconocible y fácil de recordar: https://www.facebookcorewwwi.onion

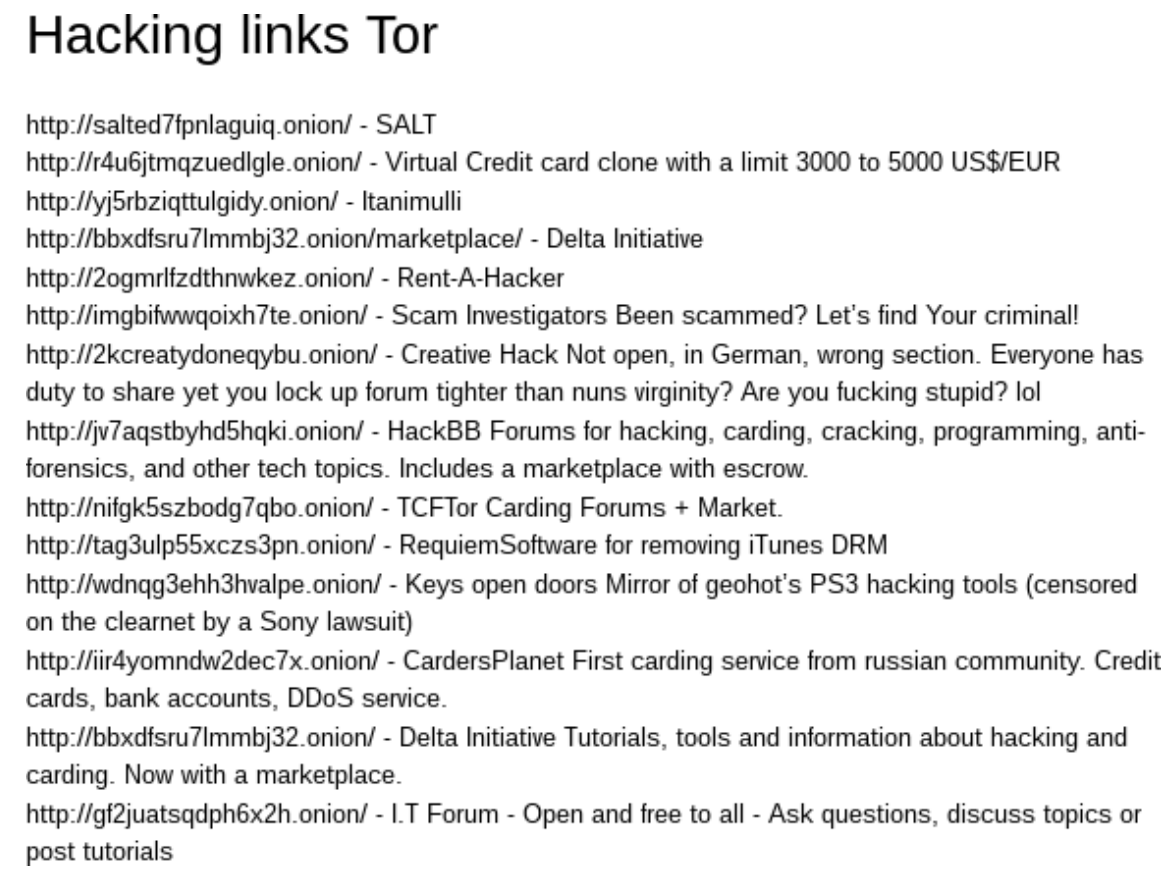


Figura 38: Enlaces sobre temas de hacking

A continuación se pueden ver enlaces a foros y medios de comunicación social. Algunos de estos enlaces no están disponibles, como se indica con la palabra "down". Hay foros para hablar de tarjetas de crédito clonadas, la red social de Tor, foros sobre hacktivismo ético, foros para programar, foros sobre temas sexuales, foros de mercados negros como Agora o Pandora, etc.

Forums Boards Chans links Tor

http://thehub7dnl9nmcz5.onion/ – The Hub Forum.
http://cavetord6bosm3sl.onion/ - CAVE TOR - Dark social network: forum, blogs, communities
http://ccshophv5gxsge6o.onion/ – Forum for Cloned credit cards, Paypal or anything related to the financial matters.
http://i25c62nvu4cleqyz.onion/ Evolution Community Forums. Redirect link evolution.to
http://zqktlwi4fecvo6ri.onion/ – The Tor Social Network, get in Contact with other
http://vm3rhgs2uhwas5rt.onion/ – Multilingual forum. Down | 2014-07-29
http://kf66ipjq43tjmonh.onion/ – Forums on all topics.
http://pyl7a4ccwgqxm6rd.onion/ – Ethical hacktivism for a better world Down | 2014-07-29
http://t4is3dhpd2jd4yhw.onion/ New Onionforum replacement since 1.0 and 2.0 are down now. Down | 2014-07-29
http://torbankdnnnvap4f.onion/ Tor's only banking dedicated forum and marketplace. Down | 2014-07-29
http://nifgk5szbodt7qbo.onion/ – Tor Carding Forums + Market.
http://s6cco2jylmxqcdeh.onion/ – Cebolla Chan v.3.0
http://i4rx33ibpyitqayh.onion/ - Agora Marketplace official forum.
http://bl3j73talvhwydx5.onion/ - Pandora Marketplace official forum.
http://fcoinjqbc6en3pe3.onion/ - Project contribution and marketplace.
http://z2hjm7uhwisw5jm5.onion/ – Paypal accounts, credit cards, we have everything!!
http://zw3crggtadila2sg.onion/ – /b/, /i/, programming, revolution, tons of other boards.
http://rrcc5uuudhh4oz3c.onion/ – Know or need to know something? Ask and share at this underground intelligence gathering network. (New Board. Much Better) (2013-08-11 UTC Appears that admins have ditched?)
http://3mrdrr2gas45q6hp.onion:2000/ - Torduckin0 #1st Citadel BBS with chat and IM to support Torduckin.
http://qm3monarchzifkwa.onion/ Anonymous BBS gopher interface, telnet interface – Another variation of the talks style of board.
http://jv7aqstbyhq5hqki.onion/ – Forums for hacking, carding, cracking, programming, anti-forensics, and other tech topics. Also a marketplace with escrow.
http://doxbinicvjqqmohl.onion/ – DOX go here. A pastebin for personally identifiable information. All information is allowed.
http://chippyits5cqbd7p.onion/ – Archive of Hack The Planet's past work.
http://exposed36mq3ns23.onion/ – Tor mirror of exposed.su/exposed.re.
http://djmrwe2r45qf5bjj.onion/ – Thunder's Place Penis Enlargement and Male Sexual Health Forum.
http://76qugh5bey5gum7l.onion/ – Up for several months now. Quite a variety of music 24/7.
http://u4uoz3aphqbdc754.onion/ – The anti-social network lel. Down | 2014-07-29
http://e266al32vpuorbyg.onion/ – Deep chat Down | 2014-07-29
http://hbjw7wjeoltskdol.onion/ – Very active Deep chat Down | 2014-07-29
http://pbuleijc2kwnkxsz.onion/ This TriPh0rce's personal page (former admin of TriChan). Several interesting links up. Down | 2014-07-29
http://odduuqnxvzqftyfh.onion/ — (TORified Version 2.0. of yiff/clop/etc user-contrib short stories)

Figura 39: Enlaces a foros y medios de comunicación social

Chat centric services

Some people and their usual server hangouts may be found in the Contact Directory.

IRC

Below is a list of *DEAD* irc servers from Anonet:

- AnoNet – Each server is on its own network and connects to a chat cloud
  - irc1.srn.ano, clearnet
  - elef7kcrczguvamt.onion:15783 – Direct access to the AnoNet chat cloud. Use an IRC server to connect.
  - irc3.srn.ano
  - irc2.srn.ano, clearnet – Still connects to the old AnoNet chat cloud; that will soon change.
  - irc4.srn.ano
  - irc.cananon.ano Web Chat Version join #Anonet

- OFTC IRC – OFTC – IRC server

running on: (various).oftc.net, ports:: plaintext: 6667 ssl: 6697

- Federation: OnionNet – IRC network comprised of:
  - Circle IRC – Circle IRC server.
  - FTW IRC – FTW IRC server.
  - Nissehult IRC – Nissehult IRC server.
  - Renko IRC – Renko IRC Server.
- OpenSource, info – Drug chat
- Dark Tunnel Irc2p gateway – Gateway to the Irc2p IRC network on I2P.

running on: unknown, ports:: plaintext: 6668, ssl: none

- Chi's Tunnel to Irc2p – New Gateway to the Irc2p IRC network (old one was down)
- WANNABE: Federation OnionNet :
- New Ngircd – Yep this is a new ircd in OnionLands looking for peering in OnionNet
- freenode IRC – freenode IRC server

running on: (various).freenode.net, ports:: plaintext: 6667 ssl: 6697/7070

- NeoturbineNET IRC – NeoturbineNET IRC server

running on: kropotkin.computersforpeace.net, ports:: plaintext: none ssl: 6697

- FREEFOR – hackint – hackint is a communication network for the hacker community.

running on: lechuck.darmstadt.ccc.de, ports:: plaintext: none ssl: 6697

- Agora Anonymous – Agorist IRC server
- HeavyCrypto – HeavyCrypto IRC

running on: unknown, ports:: ssl: 6697

- Anonimowy IRC – Anonimowy IRC (Polish anonymous IRC server)

running on: unknown, ports:: plaintext: 6667, ssl: 6697

- prooops.eu IRC – prooops.eu IRC (Clearnet http://irc.prooops.eu/)

running on: unknown, ports:: plaintext: 6667, ssl: 6697

- Team Mondial IRC – Port: 6667 SSL: 6697 == New onion anonymous webmail service (URSSMail) / Escrow Expertz
- KeratNet – Kerat – Ports 6667, ssl:6697
-

Figura 40: Enlaces a servicios de chat

SILC

- fxb4654tpptq255w.onion:706 – SILCroad, public server. [discuss/support]
- kissonmbczqxgebw.onion:10000 – KISS.onion – Keep It Simple and Safe – ditch the web browser, use SILC to communicate securely (using Pidgin with OTR)
-

XMPP (formerly Jabber)

- xmpp:tortureregex47xf.onion:5222 – Public XMPP with MUC (multi user chat) enabled. No Child Porn and racism here, any breach will result in a ban. Maintained by Creative Hack
- xmpp:okj7xc6j2szr2y75.onion:5222 – xmpp:jabber.ccc.de:5222 as a hidden service
- xmpp:3vnjj7h6c6ww2yh5.onion:5222 – instant messager for Liberty's Hackers – chat room fo Liberty's Hackers and french users – chat room for international sharing – **No CP and no Racism please**.
- xmpp:cyjabr4pfzupo7pg.onion:5222 – CYRUSERV Community Jabber, a public server ran by CYRUSERV.

Figura 41: Enlaces a servicios de chat (continuación)

Wikis lists links Tor

http://linkdirdgrhkr2zm.onion/ – LINK DIR ONION - directory sites, comments, add site
http://zqktlwi4fecvo6ri.onion/ – Uncensored Hidden Wiki
http://hdwikicorldcisiy.onion/ – HD Wiki
http://276okalwqoour5vc.onion/ – List of random links
http://wiki5kauuihowqi5.onion/ – Onion Wiki – 650+ working 05.2017 deep web links
http://torlinkbgs6aabns.onion/ – TorLinks
http://jh32yv5zgayyyyts3.onion/ – Hidden Wiki .Onion Urls
http://wikitjerrta4qgz4.onion/ – Hidden Wiki – Tor Wiki
http://cregan3gnq6spjeb.onion/ – Cregan's List

Figura 42: Enlaces a listas de wikis

Political links Tor

http://6sgjmi53igmg7fm7.onion/index.php?title=Main_Page – Bugged Planet
http://faerieuaahqvzgby.onion/ – Fairie Underground
http://2r2tz6wzqh7gaji7.onion/ – Kavkaz Center
http://tnysbtbxsf356hiy.onion/ – The New Yorker Strongbox
http://duskgytldkxiuqc6.onion/ – Example rendezvous points page
http://rrcc5uuudhh4oz3c.onion/ – The Intel Exchange Forum :: Information and discussion on various topics, ranging from Illegal Activities and Alternative Energy, to Conspiracy Theories and Hacking. Same people from SnapBBS on a fully secure, moderated and categorized forum.
http://opnju4nyz7wbypme.onion/weblog/index.html – A7B blog :: a blog dedicated to the restoration of a limited constitutional republic in the USA
http://assmkedzgorodn7o.onion/ – Anonymous, safe, secure, crowdfunded assassinations.
http://duskgytldkxiuqc6.onion/comsense.html – Commo Sense by Thomas Paine
http://nwycvryrozllb42g.onion/ – Destination Unknown
http://zbnnr7qzaxlk5tms.onion/ – Wiki Leaks

Figura 43: Enlaces sobre temas políticos

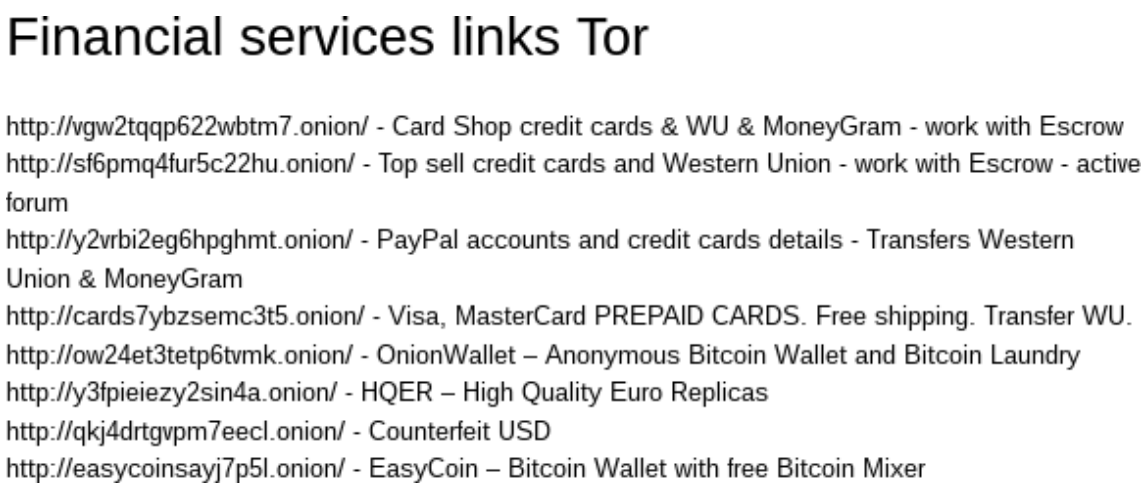


Figura 44: Enlaces sobre servicios financieros

5 Técnicas de desanonimización de usuarios y servicios ocultos

Un adversario puede escuchar fácilmente entre un nodo de salida y un servidor oculto y observar la comunicación. Por lo tanto, depende de cada usuario asegurarse de que el contenido que está enviando al servidor esté encriptado y que no pueda servir de pista para revelar la identidad del usuario.

Las técnicas de desanonimización tanto de usuarios como de hidden services implican la explotación de errores humanos, además de métodos complicados que pueden aprovechar fallos del software. También hay que tener en cuenta los fallos de seguridad operacional (OPSEC), que generalmente están relacionados con los errores cometidos por usuarios o los administradores de los servicios ocultos.

En este apartado también se describirán las técnicas de correlación de flujo, ataques dirigidos a los sistemas afiliados a la red Tor, el uso de direcciones bitcoins para desanonimizar a usuarios, la técnica de Website Fingerprinting, el ataque Raptor, el control de los HSDir y otros ataques para desanonimizar a servicios ocultos como el empleo inadecuado de los servicios de SSH, certificados SSL y el módulo de estado de los servidores Apache.

5.1 Fallo humano

Se podría considerar que la vulnerabilidad más seria que hace peligrar el anonimato de Tor se basa en el fallo humano y comprende desde el desconocimiento de los administradores del hidden service de cómo desplegar un servicio en Tor al usar tutoriales inadecuados en Internet, errores en la configuración, que se ofrezcan los mismos o similares servicios tanto en la surface web como en la dark web, el uso de ingeniería social para ganar la confianza de los usuarios o usar técnicas para sacar al usuario de Tor a partir del uso de determinados ficheros o malware...

Un ejemplo de error humano, fue el caso de la primera versión de Silk Road, la cual llegó a difundir mensajes de error que revelaban la IP real del servidor.

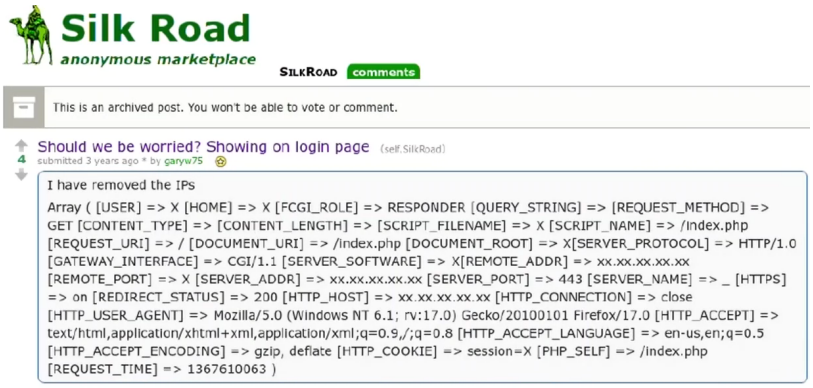


Figura 45: Mensaje de error de Silk Road

5.1.1 Ingeniería social

Un caso de ingeniería social fue el caso Child's Play en el que la policía australiana compartió pornografía infantil con pedófilos durante un año después de tomar el control del hidden service en septiembre del 2016, con el objetivo de atrapar a los delincuentes.

Figura 46: Sitio onion Childs Play

Otro ejemplo de error humano fue el caso Playpen en febrero del 2015, donde debido a una configuración incorrecta del servidor que alojaba el hidden service Playpen, el sitio web estaba disponible para el acceso en Internet normal a los usuarios que conocían la verdadera dirección IP del servidor. Esa dirección apuntaba a un servidor en Carolina del Norte, alojado por una compañía llamada CentriLogic. El FBI tomó control del sitio web y continuó sirviendo contenido durante dos semanas para arrestar a sus usuarios, empleando también por tanto, ingeniería social.

5.1.2. Fallos en la configuración de servicios hidden services en Tor

En los tutoriales que explican como desplegar un sitio web en Tor, se producen dos tipos de fallos:

i) No se menciona la cuestión de cambiar al usuario y temas de privilegios. Se deberá de crear un usuario específico para lanzar el hidden service de Tor con unos permisos restringidos. Este usuario es un usuario distinto al usuario con el que se accede al servidor de manera habitual, por lo que deberá de ser un usuario no privilegiado.

ii) Las páginas en Tor se valen de un fichero que se llama “private key” que es el que le da la titularidad al dominio. Si alguien roba ese fichero, tiene la posibilidad de robar el dominio. Por tanto ese fichero debe estar guardado en una parte restringida o que sea inaccesible excepto para el servicio.

Hay muchas páginas en Tor que están mal configuradas al seguir tutoriales erróneos que aconsejan guardar la private key en la misma ruta que el servidor web. Como la carpeta se ha de llamar ‘hidden_service’, si escribimos después de la ruta de la página web: web.onion/hidden_service, se accede a la carpeta /hidden_service de modo que se puede tener acceso a la clave privada.

5.1.3 Uso de phising a través de ficheros

Una vez que se tiene acceso a la clave privada, se puede incautar la web del hidden service con el objetivo de que los usuarios no puedan hacer uso de la página. Para ello se levantaría un servidor apache con la clave privada del hidden service, después se levanta la instancia de Tor y se hace el cambiazo de la página, de modo que todos los usuarios que quieran visitar el dominio onion, accederán a la nueva página. Esta nueva página puede mostrar desde un “Acceso no permitido” a una página que simule la página del hidden service para funcionar como phising. En este último caso, se puede conseguir obtener la IP de los clientes del hidden service, por ejemplo de la siguiente manera:

1. Se sube un fichero del tipo svg al servidor que simula ser el hidden service, de forma que al fichero svg se le inyecta una línea que es un enlace https que apunta a un servidor que controla el atacante.

2. El cliente del hidden service puede descargar el fichero svg que aparenta ser un archivo pdf y al abrirlo, no se abre con el navegador Tor Browser, sino con Firefox (navegador que no enruta la comunicación por Tor)

3. El servidor que recibe la petición https registra la dirección IP pública de la persona que se decargó el fichero a través de Tor, el navegador y el sistema operativo que usa. Además de conocer la IP y su geolocalización, se puede conocer si la persona navega con un móvil, un portátil, una tablet…

Se recomienda por tanto ser precavido con respecto a los ficheros que se descargan desde Tor, puesto que por ejemplo, una simple carpeta de ficheros mp3 puede contener un archivo m3u que permite a la mayoría de los reproductores de música buscar la imagen del álbum en la fuente online a la que apunta el fichero m3u. De esta manera el atacante puede compartir este álbum de música con sus víctimas y ver las conexiones que le llegan y sus direcciones IP reales.

Precisamente con respecto a los reproductores gratuitos, la mayoría de ellos, se han visto afectados por fugas de privacidad no intencionadas. Un ejemplo fue el del bug del reproductor VLC en mayo del 2017. El fallo consistía en que el reproductor ignoraba las configuraciones de privacidad, de forma que los ficheros de subtítulos de series o películas eran manipulados con la finalidad de que los atacantes pudieran tomar control de una amplia gama de dispositivos, tales como ordenadores, smartphones, tablets o smart Tvs.

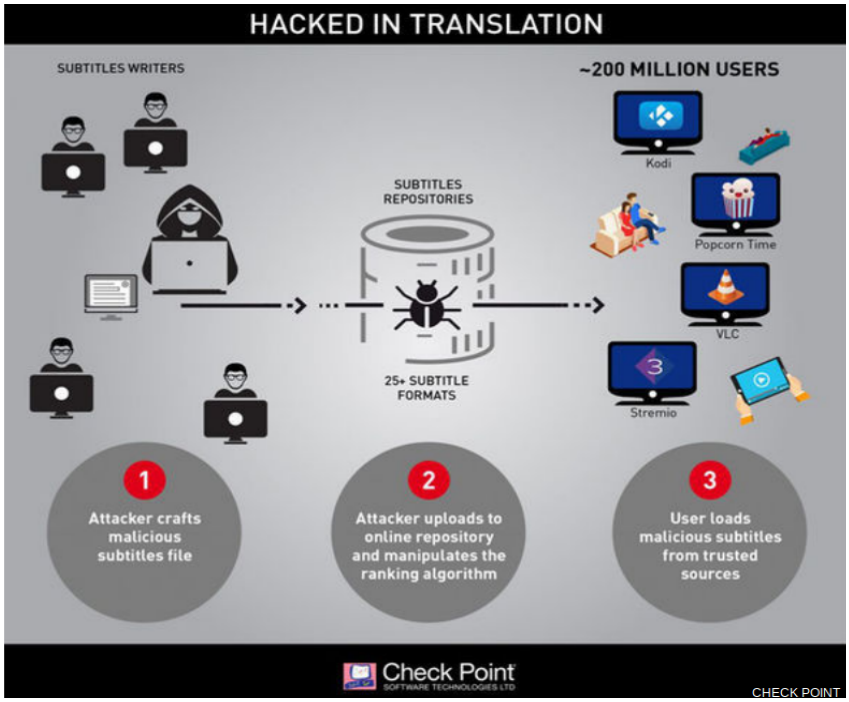


Figura 47: Vulnerabilidad del reproductor VLC

5.2 Fallos de Seguridad Operacional (OPSEC)

OPSEC (Operational Security) es el proceso de proteger piezas individuales de información que agrupadas juntas pueden dar un perfil completo (llamado agregación). Algunas medidas para proteger la información son el uso de software de cifrado de emails, tomar precauciones ante escuchas subrepticias, prestar mucha atención a fotografías tomadas (como elementos en el fondo), o no hablar abiertamente en las redes sociales…

Esencialmente la idea es no compartir nada, porque cualquier información puede ser crítica ya sea para el presente como para el futuro. En este último caso, aunque pueda parecer que los datos por separado puedan no tener relevancia, juntos pueden tener mucho valor.

5.2.1 Caso Ross Ulbricht

Uno de los ejemplos más famosos de este enfoque es la manera en la que se reveló la identidad de Ross Ulbricht, el cerebro detrás de Silk Road, en octubre del 2013. Los organismos de seguridad realizaron varias observaciones del comportamiento online de Ross y los correlacionó para revelar su identidad y acusarlo de dirigir el mercado darknet "Silk Road". Al recopilar información acumulativa crítica fueron capaces de desanonimizarlo.

Por poner un ejemplo, uno de sus errores de seguridad fue el uso de varios documentos de identificación falsos que incluían su imagen real, pero con nombres diferentes. Ross usó múltiples alias, entre ellos "frosty", "Dread Pirate Roberts" (DPR) y "altoid" en Silk Road y foros en línea a través de los cuales se comunicó con sus clientes.

5.2.2 Caso Cazes

Otro ejemplo más actual de errores de OPSEC sucedió en julio del 2017, cuando se produce el desmantelamiento del mercado negro Alphabay, (mercado que había tomado el testigo tras la caída de Silk Road y su segunda versión), por fallos de seguridad operativa por parte del fundador, Alexander Cazes.

Una de las claves del arresto de Cazes fue su error a la hora de utilizar su dirección de correo electrónico. En diciembre de 2016 las autoridades descubrieron que el correo electrónico personal de Cazes ("Pimp*Alex*91@hotmail.com") se incluía en la cabecera del mensaje de bienvenida de AlphaBay a sus nuevos usuarios en diciembre de 2014. El número “91” de la dirección de correo electrónico, coincidía con su fecha de nacimiento (19 de octubre de 1991).

La dirección había sido además reutilizada en un buen número de sitios web (incluido LinkedIn) y en otro error fatal Cazes había reutilizado contraseñas para varios servicios en lugar de diferenciarlas y gestionarlas con aplicaciones y servicios tipo LastPass, KeePass o 1Password.

Cazes también había usado esa dirección con una cuenta de PayPal que tenía verificación en dos pasos, pero que mostraba en ese proceso los cuatro últimos dígitos de su número de teléfono.

Además, durante el arresto las agencias de seguridad descubrieron el portátil de Cazes abierto y desbloqueado. Estaba de hecho conectado al servidor en el que estaba hospedado el sitio web de AlphaBay, en el que Cazes tenía como nombre de usuario "Admin".

5.2.3 Consejos de Seguridad Operacional

Algunos consejos OPSEC para usuarios de Tor, son tan sencillos como los siguientes:

- Desconectar los micrófonos del ordenador.
- Cubrir la webcam del ordenador con una cinta.
- No usar Tor en sitios públicos o dejar el ordenador sin vigilar en un sitio público.
- No usar el mismo email y/o usuario en la clearnet y en Tor.
- No tener el Javascript habilitado todo el tiempo.
- No tener la ventana en pantalla completa (maximizar el navegador Tor puede permitir a los sitios web determinar el tamaño del monitor del ordenador, lo que podría usarse para rastrear al usuario).

5.3 Ataques de Correlación

Este tipo de ataque es posible para cualquier tipo de sistema de anonimato que trabaje a tiempo real y funcione sobre TCP, de modo que si el atacante puede ver el tráfico que entra y que sale en la red y compara los patrones de tráfico, puede descubrir la IP real del usuario.

Un atacante que observa el tráfico que llega al primer nodo de retransmisión (guardia de entrada), así como el tráfico que llega al destino final (hidden service, nodo de retransmisión de salida ... etc.) puede utilizar el análisis estadístico para determinar que pertenecen al mismo circuito. Es interesante mencionar que el atacante no necesita tener control total sobre el primer y último nodo a lo largo de un circuito Tor para poder correlacionar los flujos de tráfico supervisados en esos nodos de retransmisión. El atacante solo necesita poder controlar el tráfico.

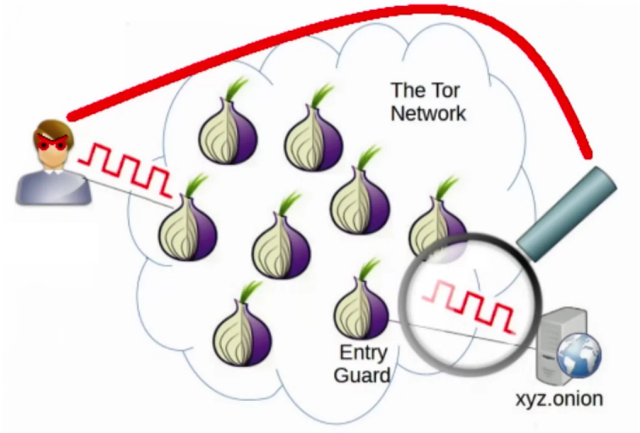


Figura 48: Ataque de correlación

Estos ataques son de alguna manera fáciles de ejecutar cuando el conjunto de anonimato (número de clientes que usan Tor) es relativamente pequeño. En otras palabras, si hay un pequeño número de personas que usan Tor, dentro del contexto de una red específica, entonces es relativamente más fácil desanonimizarlas. Las formas más complejas de ataques requieren técnicas más complicadas de análisis estadístico del tráfico y de la sincronización.

5.3.1 Caso del estudiante de Harvard y Guerilla Mail

Un ejemplo de desanonimización a través de correlación, que no necesitó realizar formas sofisticadas de análisis estadístico, sucedió en diciembre del 2013 cuando un estudiante de la Universidad de Harvard fue arrestado por enviar amenazas de bombas falsas, a través de Tor, para retrasar la fecha de un examen.

Según los datos del FBI, los correos electrónicos se enviaron desde la wifi del campus desde un correo electrónico proporcionado por Guerilla Mail, un proveedor de correo electrónico que permite a los usuarios crear correos electrónicos temporales. Guerilla incluye la dirección IP del remitente en todos los correos electrónicos salientes, y en este caso particular, además indicaba uso de Tor. La correlación ayudó al FBI a identificar al estudiante, quien confesó durante el interrogatorio.

5.4 Ataques dirigidos a los sistemas afiliados a la red Tor

Tor no es más que un servicio que puede estar ejecutando un servidor o un usuario. Como tales, los sistemas afiliados a la red de Tor todavía son vulnerables a los ciberataques tradicionales. Dependiendo de la exposición y las configuraciones especiales del sistema, se podrían utilizar varias técnicas para descubrir la identidad real de un usuario web o un servicio oculto dentro de la red Tor.

Los ataques específicos al nivel de la aplicación incluyen el manejo de sesiones, la validación de entradas y el control de acceso, mientras que al nivel del sistema operativo, los ataques generalmente se dirigen a una configuración incorrecta. Además, el rendimiento del sistema puede verse afectado por los ataques DDoS de denegación de servicio, lo que puede precipitar un fallo del sistema.

Normalmente, los ataques de validación de entrada dependen de la inyección y, por lo general, implican buffer overflows, cross site scripting (XSS) y carga de archivos maliciosos. Los ataques de manejo de sesión (session handling) se basan en tokens de orientación intercambiados a lo largo de la comunicación para garantizar un estado correcto en los dos puntos finales de la comunicación e incluyen adivinación de valor de token, intercepción de token y fijación de sesión. Los ataques de control de acceso se centran en la escalada de privilegios, es decir, un usuario normal será promocionado a un usuario con privilegios de administrador.

5.4.1 Vulnerabilidades 0 Day para Tor Browser

Las vulnerabilidades en Tor se pagan muy ampliamente, de hecho existen empresas como Zerodium, que ofrecen recompensas de un millón de dolares por exploits 0-Day, que después venderán a gobiernos u otros postores.

Los exploits que compra Zerodium deben aprovechar la vulnerabilidad de ejecución remota de código, el vector de ataque inicial debe ser una página web y debe funcionar contra la última versión de Tor Browser. Además, el exploit 0-Day de Tor debe funcionar sin requerir la interacción del usuario, excepto que las víctimas visiten una página web.

Otros exploits, como la entrega a través de un documento malicioso, no pueden acceder a la recompensa mencionada, pero Zerodium puede, a su entera discreción, realizar una oferta distinta para adquirir tales exploits.

Figura 49: Recompensa de Zerodium

Como el navegador de Tor está basado en Firefox, si se encuentran vulnerabilidades 0-Day para Firefox, estas también son vulnerabilidades para el navegador de Tor.

5.4.1.1 Vulnerabilidad en Freedom Hosting

En agosto del 2013 se pudo desarticular la mayor red de pedofilia en Internet de aquella época, Freedom Hosting, gracias a una vulnerabilidad 0-Day en Firefox 17, software que utilizaba Tor Browser Bundle en la versión 2.3.25-5, donde se podía inyectar un javascript malicioso que permitía identificar al usuario.

El javascript malicioso era un pequeño ejecutable de Windows oculto en una variable llamada "Magneto", pero el código de Magneto no descargaba nada, sino que buscaba la dirección MAC de la víctima (a través de gethostname ()) y el nombre de host del Windows de la víctima (a través de SendARP en gethostbyname () -> h_addr_list).

Luego envíaba esta información (codificada como una solicitud web HTTP estándar) a un servidor fuera de Tor con IP 65.222.202.54, y añadía a la solicitud un identificador único que vinculaba a la víctima con su visita al sitio web pirateado de Freedom Hosting, con el objetivo de exponer la dirección IP real del usuario.

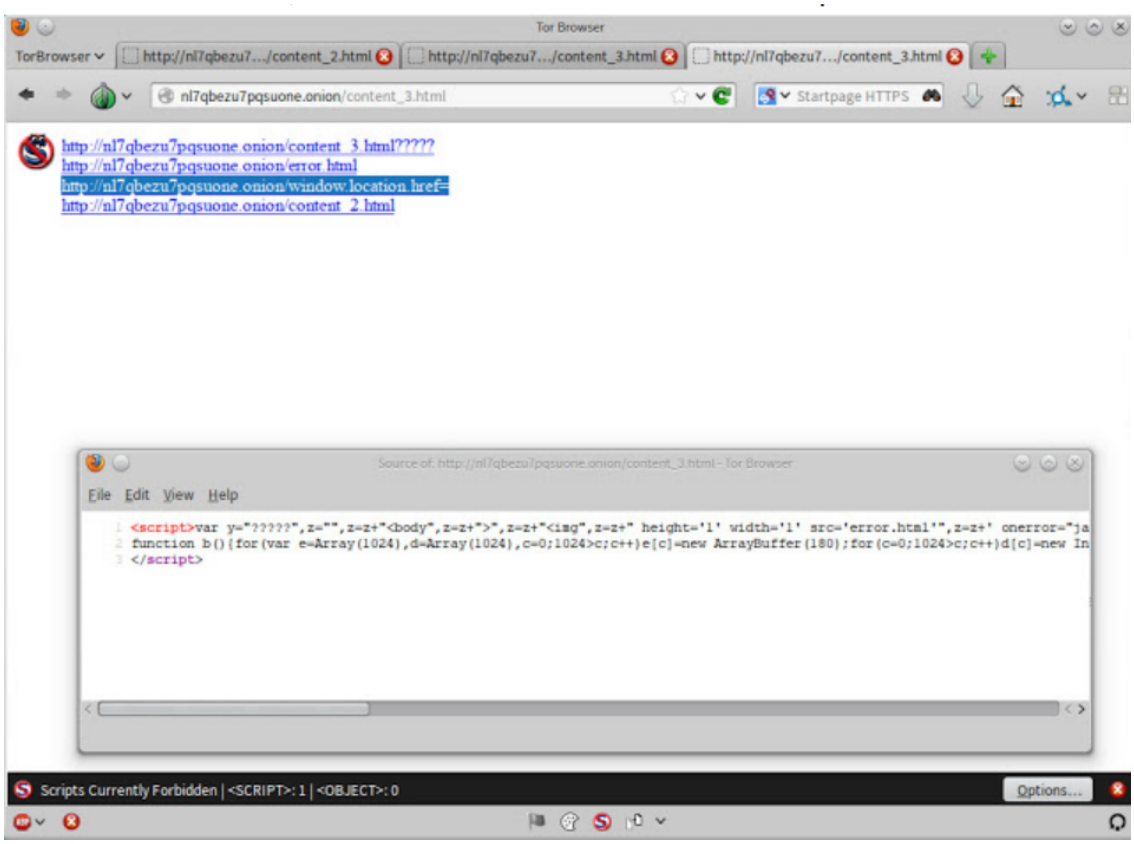


Figura 50: Vulnerabilidad en Freedom Hosting

5.4.1.2 Vulnerabilidad en GiftBox

En noviembre 2016 se utilizó un exploit 0-Day para Tor Browser para exponer las IPs de los usuarios de GiftBox, otro website de pornografía infantil. Esto se realizaba a través de un código javascript malicioso, el cual consistía en un fichero HTML y un fichero CSS que permitían tener acceso a “VirtualAlloc” en “kernel32.dll”, de modo que el objetivo era obtener las direcciones MAC de las interfaces de red y enviarlas a un servidor con IP 5.39.27.226, localizado en Francia.

[tor-talk] Javascript exploit

firstwatch at sigaint.org firstwatch at sigaint.org
Tue Nov 29 21:55:23 UTC 2016

```
This is an Javascript exploit actively used against TorBrowser NOW. It
consists of one HTML and one CSS file, both pasted below and also
de-obscured. The exact functionality is unknown but it's getting access to
"VirtualAlloc" in "kernel32.dll" and goes from there. Please fix ASAP. I
had to break the "thecode" line in two in order to post, remove ' + ' in
the middle to restore it.
```

Figura 51: Exploit 0-Day que expuso a los usuarios de GiftBox

Este exploit permitía la ejecución de código remoto en sistemas de Windows y se cargaba en la página de confirmación que aparecía tras haberse logueado el usuario en la website.

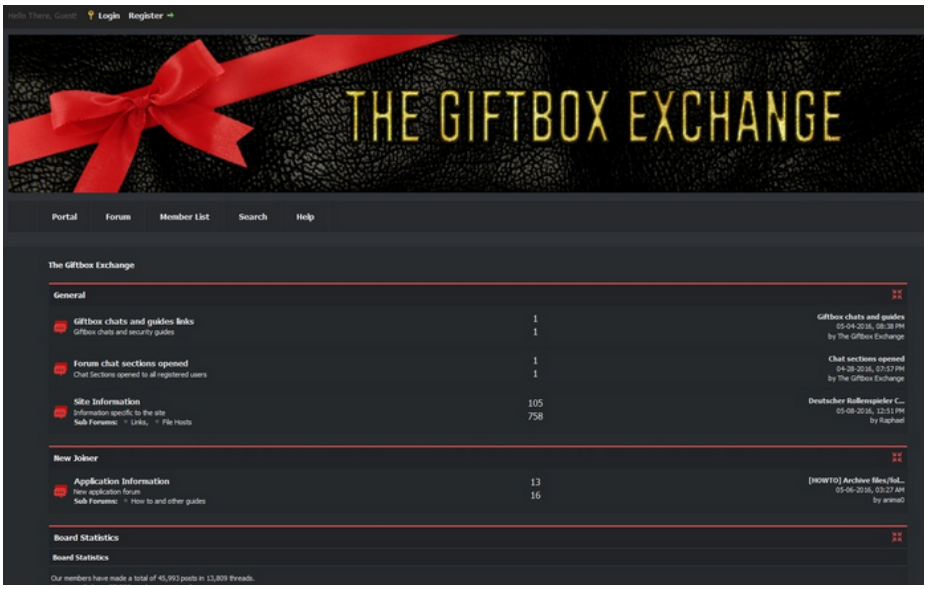


Figura 52: Sitio onion The GiftBox Exchange

5.4.1.3 Vulnerabilidad en Freedom Hosting II

En febrero 2017 un hacker anónimo comenzó a descargar una gran base de datos robada de Freedom Hosting II, exponiendo potencialmente a sus usuarios, usando un exploit ya conocido. El hack que destapó a Freedom Hosting II consistió esencialmente en iniciar un nuevo sitio en el servidor y crear un enlace symlink para obtener acceso al directorio raíz del servicio. Esto permitió al hacker navegar por todo el servidor ya que obtuvo permisos de lectura a todo el servidor. El enlace se creaba ejecutando un comando simple de PHP: symlink("/", "./symroot");

El método que siguió el hacker fue el siguiente:

```
1.- Creó un sitio nuevo o login en un sitio antiguo

2.- Se logueó y estableció la contraseña sftp

3.- Se logueó via sftp y creó un enlace symlink en el directorio raiz
```

```
4.- Desabilitó DirectoryIndex en .htaccess

5.- Abilitó mod_autoindex en .htaccess

6.- Desabilitó el motor php en .htaccess

7.- Añadió el tipo de texto plano para los ficheros .php en .htaccess

8.- Navegó por los ficheros

9.- Encontró /home/fhosting

10.- Vió el contenido del fichero index.php en /home/fhosting/www/

11.- Encontró la configuración en /home/fhosting/www/_lbs/config.php

12.- Copió y pegó los detalles de conexión de la base de datos al login de
phpmyadmin

13.- Encontró los usuarios activos con acceso a la consola en /etc/passwd

14.- Buscó en los scripts y comprendió cómo funcionaba el reseteo de las
contraseñas

15.-Lanzó de manera manual un reseteo de la contraseña sftp para el usuario
‘user’

16.- Se conectó via ssh

17.- Ejecutó ‘sudo -i’

18.- Editó ssh config en /etc/ssh/sshd_config para permitir realizar login
con ‘root’

19.- Ejecutó ‘passwd’ para establecer la contraseña de root

20.- Se reconectó via ssh como root
```

5.4.1.4 Vulnerabilidad Tormoil

En noviembre 2017 aparece una vulnerabilidad de Firefox, llamada Tormoil, cuyo fallo reside en la forma en la que Firefox gestiona los enlaces “file://”, de modo que cuando un usuario de la red Tor se conecta a una web que contiene un enlace de este tipo, el navegador como tal establece una conexión directa con el servidor, ignorando Tor Browser. Así, al establecerse una conexión directa con el servidor en lugar de hacerlo a través de los proxies y nodos de Tor Browser, la IP real del usuario queda registrada en el servidor, por lo que las IPs de los usuarios quedaban expuestas. El fallo afectó a las versiones 7.0.8 y anteriores de Tor Browser en sistemas MacOS y Linux.

Con la actualización 7.0.9 del Tor Browser, se implementó el parche para solucionar la vulnerabilidad TorMoil.

5.5 Algunos ataques específicos a Servicios Ocultos

A parte de las técnicas mencionadas anteriormente para desanonimizar servicios ocultos, se verá a continuación otras formas de ataques que explotan fallos y errores que pueden revelar información crítica sobre un sitio web de Tor.

5.5.1 Servicios SSH tanto en Tor como en la clearnet

Los servicios SSH se utilizan normalmente para proporcionar un inicio de sesión remoto a las máquinas Linux para una dirección onion. Si se ofrece el mismo servicio SSH en una dirección IP pública, así como a través de una dirección onion, esto conducirá al descubrimiento de la dirección IP del servicio oculto de Tor.

Por tanto, si el atacante se conecta a una dirección onion a través de SSH, verá el fingerprint que tiene asociado y que será el mismo si se conecta a través de SSH directamente al servidor, identificándolo de manera inequívoca. Se puede ver un ejemplo en la figura siguiente:

```
# torsocks ssh -p 22022 root@msydqstlz2kzerdg.onion
RSA key fingerprint is a7:93:84:a6:97:fa:25:65:77:c9:58:bb:fe:8e:e2:2f
# ssh root@ahmia.fi
RSA key fingerprint is a7:93:84:a6:97:fa:25:65:77:c9:58:bb:fe:8e:e2:2f
→ ahmia.fi == msydqstlz2kzerdg.onion
```

Figura 53: Ejemplo de uso de SSH en Tor y en la clearnet para un mismo servicio

5.5.2 Certificados SSL

Yonathan Klijnsma, investigador de RiskIQ, revela en agosto 2018 que algunos servidores ocultos de Tor mal configurados y que utilizan certificados SSL son los principales responsables de que se filtren las direcciones IP de los usuarios.

Un servidor configurado de manera correcta alojado en Tor solamente necesita escuchar en el localhost 127.0.0.1. Sin embargo aquellos mal configurados tienen su servidor local Apache o Nginx con puertos de escucha hacia otra dirección o 0.0.0.0. Cuando no se usa un firewall, es obligatorio que los servidores escuchen en 127.0.0.1.

Pero no sólo se exponen las IPs de los usuarios, sino que se pueden identificar los servicios Tor mal configurados y sus direcciones IP correspondientes, al vincular los certificados SSL a sus direcciones IP alojadas.

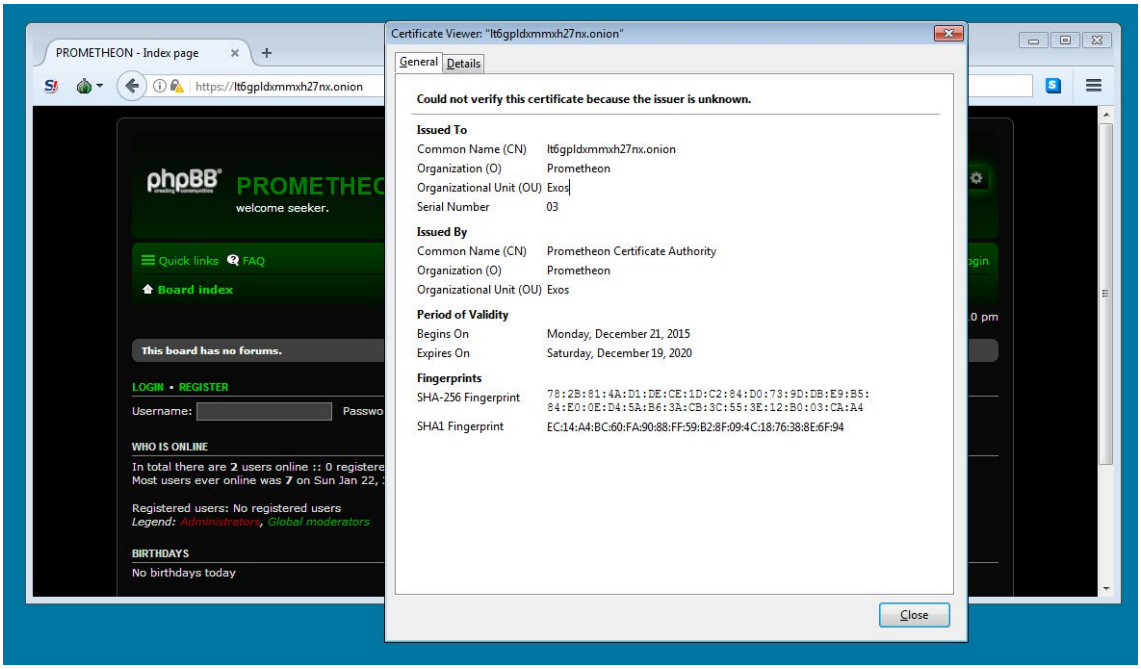


Figura 54: Sitio Tor con certificado SSL

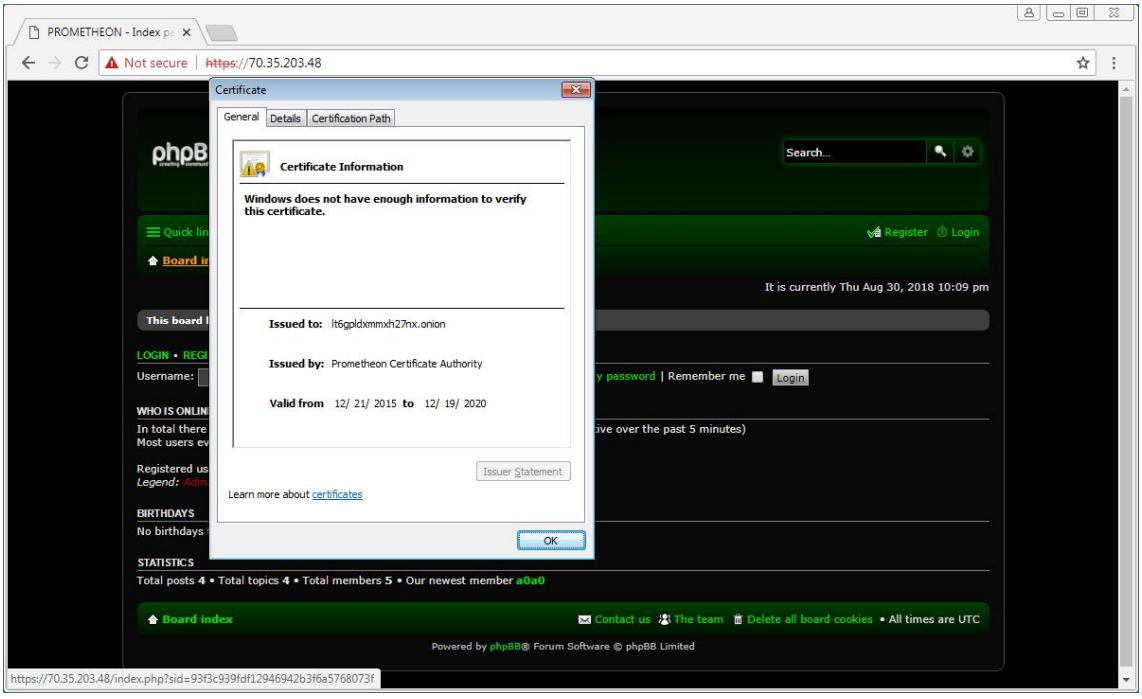


Figura 55: Certificado Tor expuesto en una dirección IP pública

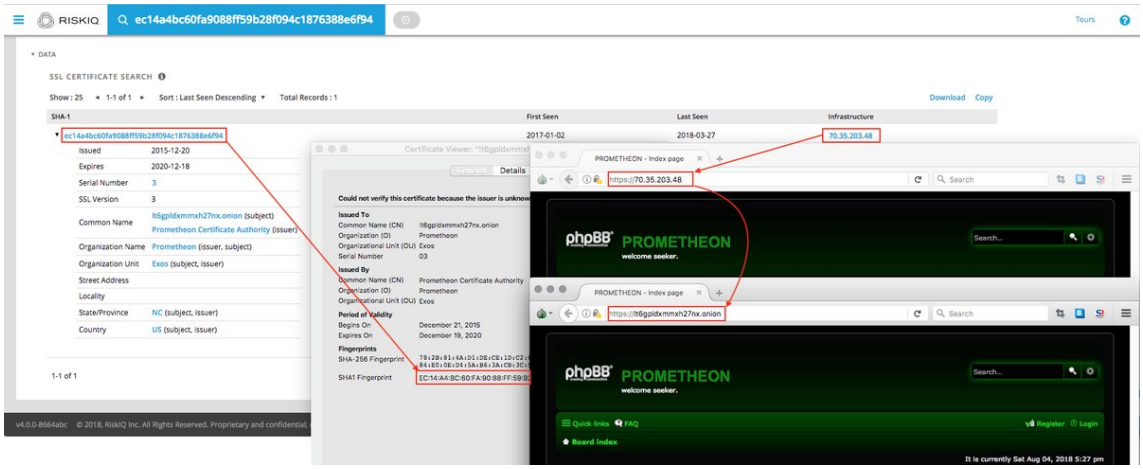


Figura 56: Exposición de un sitio onion debido a una incorrecta configuración

Una vez que un administrador de un hidden service agrega un certificado SSL a un sitio web, el dominio .onion se agrega al certificado después de que el campo Nombre común del certificado informe a la dirección .onion del servicio anónimo. Un servidor mal configurado escuchando en una dirección IP pública, tiene el certificado SSL asociado tanto con el sitio web como con la dirección onion, comprometiendo así el servicio onion.

5.5.3 Modulo de estado de los servidores Apache: 127.0.0.1/server-status

No modificar la configuración por defecto de los servidores web Apache de los hidden services, podría revelar detalles acerca del tráfico de Tor que se maneja a través de estos servidores. En concreto, la configuración del servidor Apache que causa este problema es el módulo de estado del servidor que viene activado siempre por defecto.

El módulo de estado de Apache ayuda a los administradores de los servidores a monitorizar la salud del servidor web con una interfaz HTML. La salida de este módulo es accesible desde conexiones a localhost http://127.0.0.1/server-status/, lo que habitualmente en la Internet pública supone una configuración segura, de modo que sólo los usuarios que tienen las credenciales adecuadas pueden acceder a la página del estado del servidor. Sin embargo, como en el caso de la red Tor, las conexiones de Tor Browser a los dominios onion se enrutan desde localhost, el servidor Apache mostrará al público: http://esehiddenservice.onion/server-status

Esta página muestra los datos de la configuración de un servidor, el tiempo de actividad, el uso de recursos, el tráfico total y las peticiones HTTP activas. Precisamente estos son los detalles que ayudan a detectar la zona horaria del servidor, su posición geográfica relativa, la configuración del idioma o incluso su dirección IP. Por lo tanto el agujero de seguridad es claro, ya que es precisamente toda esa información la que los usuarios de Tor desean que no se conozca.

La solución consiste en deshabilitar el módulo de estado en el servidor Apache. Para ello el código a ejecutar en la consola es: *#sudo ap2dismod status*

Donde:

```
"ap2" significa Apache 2.x
"dis" significa inhabilitar
"mod" significa módulo
```

5.6 Análisis de direcciones Bitcoins

Servicios como Bitcoin se introdujeron para proporcionar el anonimato de las transacciones en línea y la navegación web. Debido a su modelo pseudoanónimo, Bitcoin carece de seguridad operativa retroactiva, lo que significa que se podría utilizar información histórica para identificar a un determinado usuario de un hidden service.

Un método para revelar a los usuarios de un hidden service es explotar la información pública filtrada por redes sociales online, por el Blockchain de Bitcoin y por las websites onion. Esto, por ejemplo, permite a un adversario vincular a un usuario con una dirección de Twitter a un hidden service de Tor, encontrando al menos una transacción pasada en el Blockchain que muestre sus direcciones de Bitcoin declaradas públicamente.

Por tanto, para relacionar un usuario en Bitcoin con un hidden service, se necesita buscar una transacción en el Blockchain cuyo input sea cualquiera de las direcciones del wallet del usuario de Bitcoin y cuyo output sea una dirección Bitcoin de un hidden service. Para ello se ha de descargar todo el Blockchain con un software (por ejemplo Bitcoin Core), pudiendo necesitarse varios días para la descarga, debido al gran tamaño que puede ocupar el Blockchain.

Para realizar el proceso de desanonimización los pasos son los siguientes:

1.- Recogida de datos:

i) Se obtiene una lista de usuarios de Twitter con direcciones de Bitcoin a partir de una herramienta como Twitter Decahose, la cual ofrece un 10% de muestreo aleatorio en tiempo real de todos los tweets públicos a través de una conexión streaming.

ii) También se recoge información de usuarios del foro BitcoinTalk para extraer direcciones públicas de usuarios de Bitcoin.

iii) Por otro lado, se reúne también una colección de datos que incluye direcciones Bitcoin publicadas en los hidden services, utilizadas para recibir pagos.

2.- Para cada dirección Bitcoin publicada en los hidden services, se ejecuta una consulta del historial de transacciones del Blockchain con esa dirección.

3.- Esta consulta devuelve una serie de transacciones en las que la dirección aparece tanto como input como output.

4.- Después se ejecuta la misma consulta para una dirección Bitcoin de un usuario de Bitcoin, en la colección de datos obtenidos a partir de los usuarios de Twitter con direcciones Bitcoin y de los usuarios del foro BitcoinTalk.

5.- De los dos resultados de las consultas entre usuarios y hidden services, se lleva a cabo una comparación para encontrar emparejamientos.

6.- Si alguna dirección de un usuario se encuentra como input de una transacción donde una dirección de un hidden service aparece como output, entonces el usuario tiene una relación con ese servicio y se puede establecer un enlace.

7.- Si el usuario al que se le ha identificado con un hidden service tiene asociadas identidades online, se le puede desanonimizar con diferentes niveles de certeza, dependiendo de cuánta información personal identificable haya compartido en sus perfiles de usuario de redes sociales. Se puede obtener la edad, localización, género y su dirección de correo electrónico.

5.7 Website Fingerprinting

Este es un ataque, donde un adversario observa el patrón de tráfico entre el cliente Tor y el nodo de entrada y en base a los resultados realiza hipótesis sobre qué sitio web está visitando el usuario.

El procedimiento general del website fingerprinting es el siguiente:

1.- El atacante primero escucha y registra los rastros de paquetes de varios sitios web que pueden ser sitios web posibles que el usuario visita. El atacante utiliza una herramienta de análisis de tráfico (por ejemplo, tcpdump) para capturar los rastreos de paquetes de capa IP. Estos paquetes son conocidos como instancias de entrenamiento (training instances).

2.- Utilizando información capturada, como la longitud del paquete, el tiempo enviado y recibido, etc., el atacante crea un perfil del sitio web, también conocido como fingerprint o huella digital.

3.- A continuación, el atacante escucha en la red del usuario víctima y captura de manera similar el paquete que entra y sale del usuario. Estos datos no serán idénticos a la huella digital que el atacante crea en el paso 2, debido a una diferencia en el usuario, la posible fragmentación de paquetes y las actualizaciones del sitio web. Estos paquetes se conocen como instancias de prueba (testing instances).

4.- El atacante intenta comprometer al usuario víctima comparando el fingerprint del sitio web y los datos del usuario observándolos a simple vista, o utiliza métodos estadísticos para llegar a una conclusión probabilística.

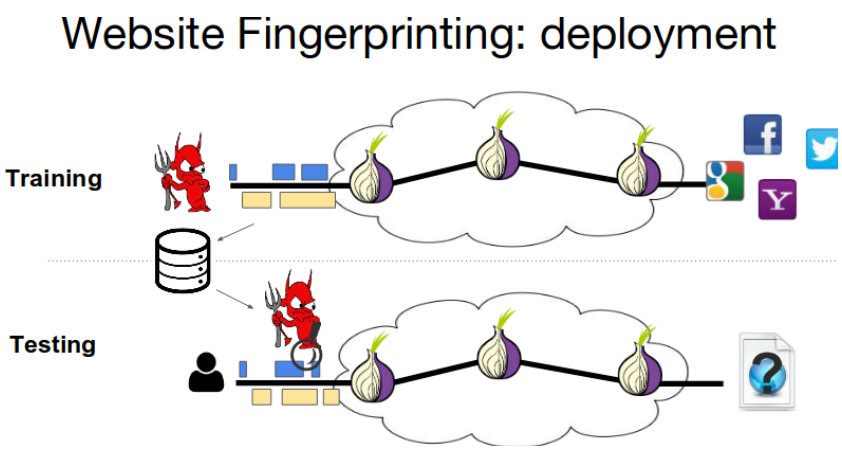


Figura 57: Desarrollo de la técnica Website Fingerprinting

5.8 Ataque Raptor

Ataque basado en que el atacante controle un *Sistema Autónomo*, lo que le da una posición relevante sobre el tráfico de la red. Un sistema autónomo es una red de gran tamaño que se comporta de forma autónoma en cuanto a su enrutamiento interno, y controla el encaminamiento de tráfico hacia y desde otros sistemas autónomos por medio de protocolos de enrutamiento de frontera llamados BGP (Border Gateway Protocols).

Controlando un Sistema Autónomo, se puede reunir suficiente información para desanonimizar a los usuarios de servicios ocultos en Tor, ya sea porque el sistema autónomo controla los nodos de entrada y de salida en Tor, porque controle un nodo de entrada y un nodo intermedio o un nodo intermedio y un nodo de salida. Se puede llegar a obtener las direcciones IP tanto del usuario como del servidor onion, así como el contenido de los paquetes enviados.

5.9 Control de los HSDir

Un atacante que tuviera el control de todos los HSDirs responsables de un servicio oculto podría saber cuándo un cliente pretende conectarse, y mediante un ataque de correlación, podría desvelar su identidad al reconocer en el nodo de entrada (controlado por el atacante) un patrón de tráfico específico enviado desde el HSDir que recibió la petición del cliente.

Sin embargo, este ataque tiene la ventaja de no requerir un nodo de entrada para llevarlo a cabo, ya que con monitorizar la entrada a la red (ya sea a través de un ISP, un punto de acceso malicioso, etc.) sería suficiente para realizar un ataque de correlación y desanonimizar un usuario.

6 Shadow: Definición y arquitectura

Shadow es un simulador de red de eventos discretos único que ejecuta aplicaciones reales como Tor y sistemas distribuidos de miles de nodos en una sola máquina. Una de las finalidades de Shadow es realizar investigaciones sobre Tor, para mejorar su seguridad y rendimiento, sin comprometer el anonimato de los usuarios de Tor.

Shadow se diferencia de otros simuladores de red en que ejecuta de manera nativa aplicaciones reales, como Tor. Es capaz de simular tiempo, operaciones de red y operaciones criptográficas e incluye un modelo preciso de retraso de CPU, de modo que al tomar medidas de ejecución de código puede caracterizar de manera precisa los retrasos de procesamiento de la aplicación. Shadow también modeliza latencia y ancho de banda por medio de medidas que toma del Internet real y se ejecuta en una máquina Linux sin privilegios de root, de manera que cualquiera puede realizar experimentos empleando este simulador.

6.1 Plano de simulación

El primer paso para usar Shadow es crear un plano de un experimento. El formato del plano es XML estándar. El archivo XML le dice a Shadow cuándo debe crear cada host virtual o nodos y qué software debe ejecutar cada uno de ellos. También especifica la estructura de la topología de la red y las propiedades de la red, como la latencia del enlace, la fluctuación de fase y las tasas de pérdida de paquetes.

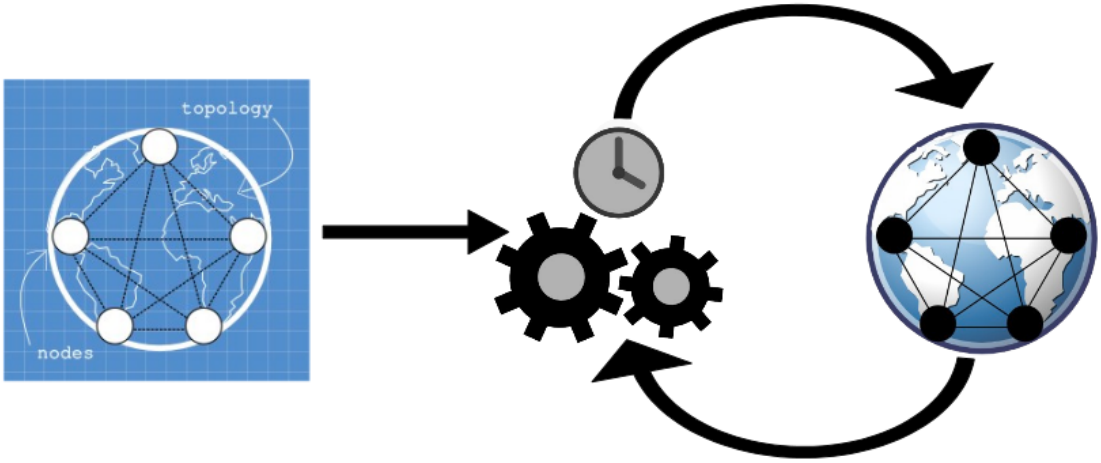


Figura 58: Funcionamiento de Shadow

El motor de eventos de Shadow inicializa los hosts utilizando el plano de simulación y ejecuta eventos en su nombre hasta que se completa la simulación.

6.2 Funcionamiento de Shadow con Tor

Como se puede ver en la figura 18, el espacio de memoria está separado en dos partes:

- En la parte de arriba está el espacio de memoria del motor de Shadow que incluye el motor de eventos, el manejo de los nodos, etc.

- En la parte de abajo del diagrama está el espacio de memoria de Tor, que es ejecutado por Shadow como un plugin.

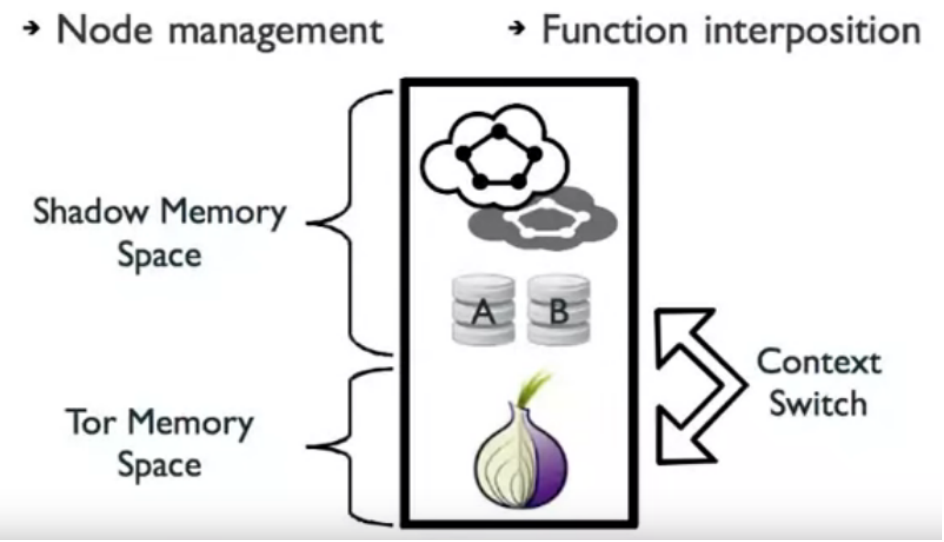


Figura 59: Espacio de memoria del proceso general de Shadow con Tor

Por tanto, para poder manejar múltiples nodos en un sólo simulador usando una aplicación real, se necesita intercambiar los espacios de memoria donde Tor tiene un estado variable. Es decir, se obtienen las direcciones de todo lo que cambia en Tor y se copian en cada uno de los nodos del simulador. Después cuando se desea que un nodo en particular ejecute Tor, se toma la versión que ese nodo tiene de su estado (que se ha copiado en el espacio de memoria de Shadow) y se efectua un "cambio de contexto" del kernel (context switch), donde se copia esa memoria del almacenamiento de Shadow al espacio de memoria de Tor y a continuación se permite que ese nodo ejecute Tor.

Cuando Tor acaba usando ese nodo, se copiará su estado otra vez en el espacio de memoria de Shadow. De esta manera, se puede permitir que varios nodos se ejecuten en sólo un simulador con una copia de la aplicación de Tor. Al emplear los nodos de esta manera se minimiza la cantidad de memoria que usa el simulador. Lo único que se duplica entre nodos es el estado, que ciertamente cambia durante la ejecución de la aplicación.

Cuando Tor corre en un nodo del simulador, funcionará igual que cuando se ejecuta en un host: creará sockets, intentará enviar y recibir información de la red y hacer todo lo demás que le permite realizar el sistema. Para proceder así con Shadow, se emplea una técnica llamada "interposición de funciones", en la que se utiliza la variable de entorno `lD-preload` para interceptar las llamadas de funciones y redirigirlas hacia la red de simulación que se está utilizando.

6.2.1 Interposición de funciones

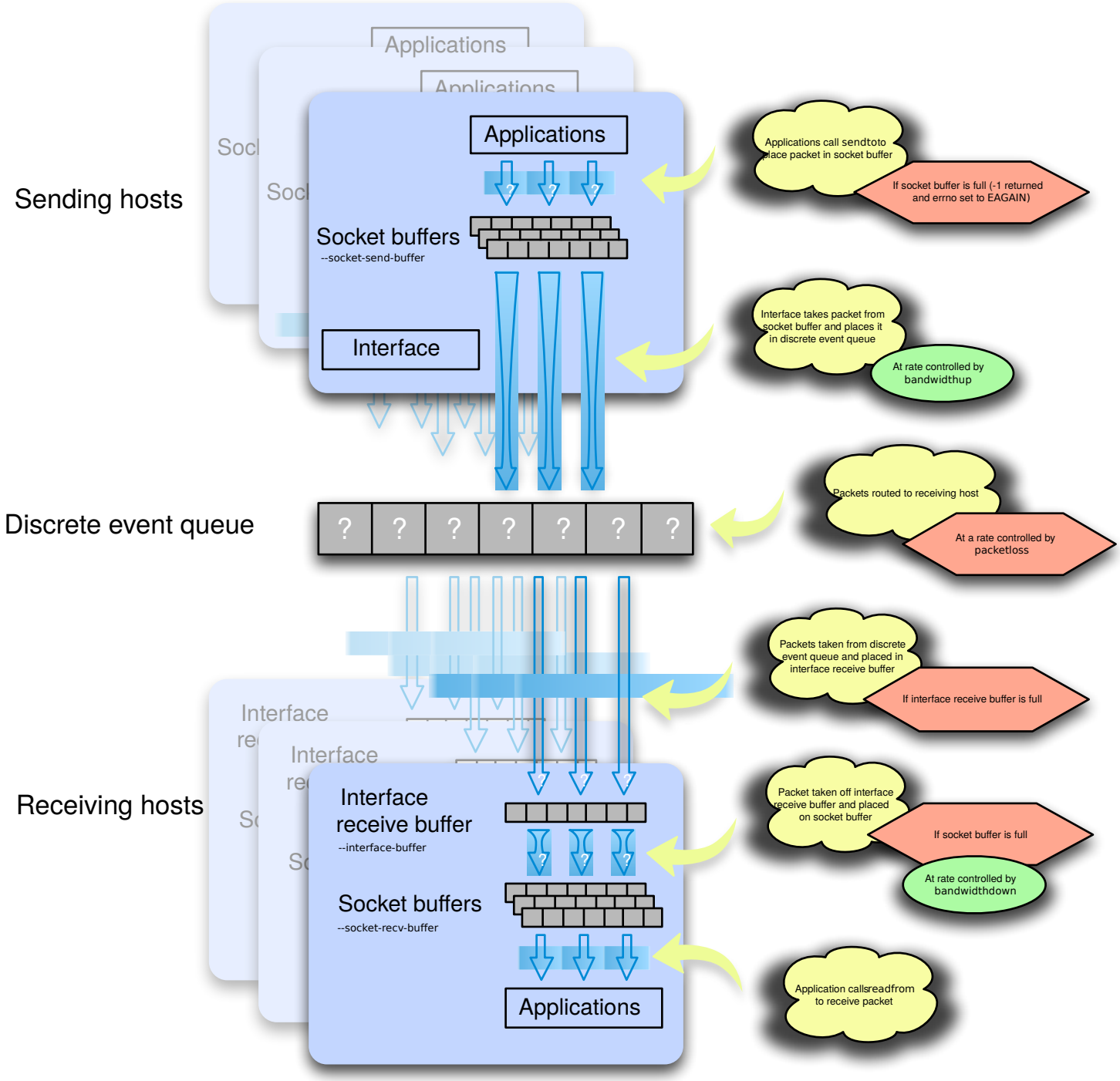
Shadow ejecuta aplicaciones reales que se ejecutan en sistemas normales de tipo UNIX. Estas aplicaciones esperan una amplia gama de bibliotecas de kernel disponibles para su uso. Por ejemplo, el envío y la recepción de datos a través de la red, la hora del sistema y el sondeo de dispositivos generalmente son manejados por el kernel en algún nivel. Estas funciones del sistema (y otras) se interceptan y se redirigen a través de versiones específicas de Shadow. De esta manera, Shadow brinda a los hosts la capacidad de comunicarse sin problemas entre ellos a través de la red virtual sin requerir ningún cambio en el código de la aplicación.

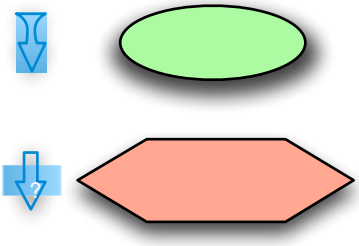
6.2.2 Eventos de tiempo discreto

Shadow crea varios eventos de arranque después de extraer la información del archivo de plano XML suministrado. Cada uno de estos eventos se ejecuta en un instante de tiempo discreto durante la simulación y hará que los hosts virtuales comiencen a ejecutar el software especificado (en nuestro caso Tor), que a su vez generará eventos adicionales para que Shadow los procese. Shadow realiza un seguimiento del tiempo que cada host virtual dedica al procesamiento dentro de la aplicación y retrasa los eventos según la velocidad de la CPU virtual configurada del host. Los eventos se ejecutan continuamente hasta la hora de finalización de la simulación.

A medida que las aplicaciones se envían datos entre sí, Shadow empaqueta esos datos en un tipo interno y transfiere el puntero entre varias colas. Este proceso implica el uso de la cola principal de eventos Shadow para transferir los eventos de paquetes entre hosts virtuales, y la limitación de velocidad para garantizar que cada host tenga la capacidad de ancho de banda deseada. La siguiente imagen, puede ayudar a visualizar este proceso:

Packet Flow in Shadow





v0.1 (2013-06-14)

Steven J. Murdoch

Los datos de la aplicación de extremo a extremo fluyen a través del socket Shadow y los buffers de interfaz, mientras que la cola de eventos discretos facilita la transferencia de datos entre hosts virtuales.

7 ExperimenTor

La emulación de red es otra técnica para experimentar con Tor. En contraste con la simulación, los emuladores realizan todas las operaciones en tiempo real en nodos virtuales, creando una representación precisa de la red. Puesto que todas las operaciones tienen que realizarse simultáneamente, esto resulta en extensos requerimientos de recursos. Hay dos emuladores de Tor dedicados: ExperimenTor y SNEAC. En este apartado se define la herramienta ExperimenTor y se compara con Shadow.

ExperimenTor es un emulador de red destinado a mejorar las investigaciones sobre Tor y requiere al menos dos máquinas físicas o virtuales: mientras que el núcleo del emulador es responsable de emular la red, Tor y otras aplicaciones como los clientes de BitTorrent se ejecutan en uno o más emuladores y uno o más bordes del emulador (emulator edges). ExperimenTor utiliza software de Tor no modificado y el tamaño de los experimentos solo están limitados por disponibilidad de recursos de hardware. ExperimenTor genera una red de escala reducida que representa a Tor en términos de distribución de ancho de banda.

En contraste con Shadow, los experimentos no incluyen los efectos de fondo de Internet como jitter no determinista o pérdida de paquetes. Varios proyectos de investigacion utilizaron ExperimenTor para la emulación de la red Tor. Como ejemplo, Wacek et al. utilizaron ExperimenTor para analizar qué técnica de selección de nodos proporciona las mejores propiedades de anonimato y rendimiento. Sin embargo, ExperimenTor se basa en una versión obsoleta de FreeBSD y por lo tanto ya no es mantenido. ExperimenTor ha sido reemplazado según los últimos trabajos en la Universidad de Waterloo por SNEAC, el emulador de red escalable para comunicación anónima.

8 Instalación de Shadow

Se procede, a continuación, a la instalación de Shadow, para realizar un ejemplo de simulación de la red Tor en un entorno virtualizado.

1) Primero necesitamos instalar las dependencias siguientes:

```
#sudo apt-get install -y gcc g++ python libglib2.0-0 libglib2.0-dev
libigraph0v5 libigraph0-dev cmake make xz-utils

#sudo apt-get install libc-dbg

#sudo apt-get install -y python-matplotlib python-numpy python-scipy python-
networkx python-lxml

#sudo apt-get install -y git dstat screen htop
```

2) `#git clone https://github.com/shadow/shadow.git`

```
3) #cd shadow
   #./setup build --clean --debug –test
```

```
4) #./setup install
```

```
5) #./setup test
```

6) Hay que añadir la ruta `/home/${USER}/.shadow/bin` a la variable de entorno PATH (por ejemplo en `~/.bashrc` o `~/.bash_profile`).

```
#echo "export PATH=${PATH}:/home/${USER}/.shadow/bin" >> ~/.bashrc && source ~/.bashrc
```

7) Comprobar que Shadow está instalado y funciona:

```
#shadow –version
```

```
#shadow –help
```

A continuación podemos ver cada uno de estos pasos en las siguientes figuras:

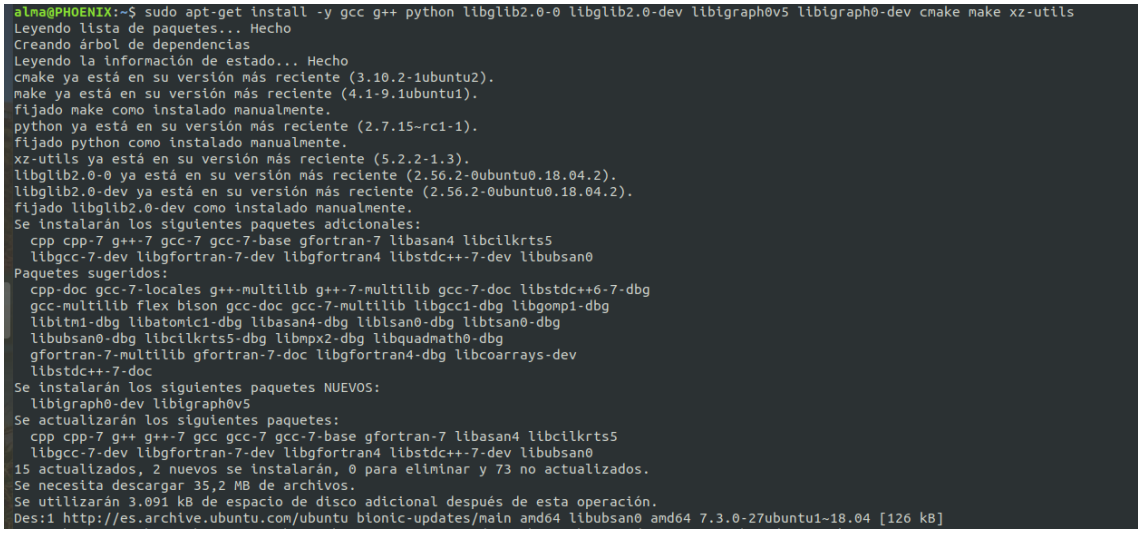


Figura 60: Paso 1: instalación dependencias

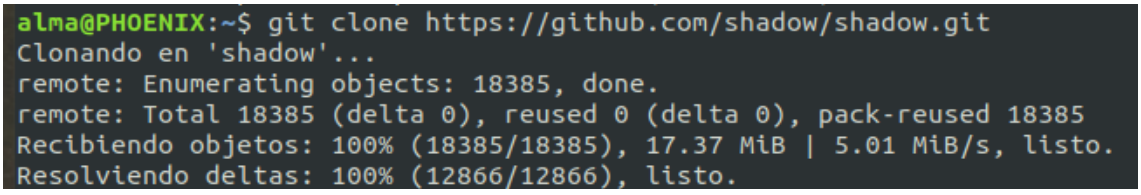


Figura 61: Paso 2: #git clone https://github.com/shadow/shadow.git

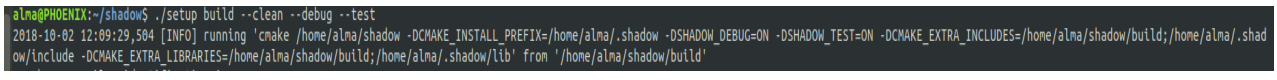


Figura 62: Paso 3: #./setup build --clean --debug –test

```
-- The C compiler identification is GNU 7.3.0
-- The CXX compiler identification is GNU 7.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- System name: Linux
-- System version: 4.15.0-34-generic
-- System processor: x86_64
-- CMAKE_MODULE_PATH = /home/alma/shadow/cmake/;/home/alma/.shadow/include;/home/alma/shadow/build
-- Check if the system is big endian
-- Searching 16 bit integer
-- Looking for sys/types.h
-- Looking for sys/types.h - found
-- Looking for stdint.h
-- Looking for stdint.h - found
```

Figura 63: Paso 4: #./setup install

```
[ 94%] Built target shadow-plugin-test-random
[ 94%] Built target test-signal
[ 95%] Built target shadow-plugin-test-signal
[ 95%] Built target shadow-plugin-test-sleep
[ 96%] Built target test-sleep
[ 97%] Built target test-sockbuf
[ 98%] Built target shadow-plugin-test-sockbuf
[ 98%] Built target test-tcp
[ 98%] Built target shadow-plugin-test-tcp
[ 99%] Built target shadow-plugin-test-timerfd
[100%] Built target test-timerfd
Install the project...
-- Install configuration: "Debug"
-- Installing: /root/.shadow/include/shd-config.h
-- Installing: /root/.shadow/share/cmake/Modules/FindDL.cmake
-- Installing: /root/.shadow/share/cmake/Modules/FindRPTH.cmake
-- Installing: /root/.shadow/share/cmake/Modules/FindM.cmake
-- Installing: /root/.shadow/share/cmake/Modules/FindGLIB.cmake
-- Installing: /root/.shadow/share/cmake/Modules/ShadowTools.cmake
-- Installing: /root/.shadow/share/cmake/Modules/FindRT.cmake
-- Installing: /root/.shadow/share/cmake/Modules/FindIGRAPH.cmake
-- Installing: /root/.shadow/share/topology.graphml.xml
-- Installing: /root/.shadow/bin/shadow
-- Set runtime path of "/root/.shadow/bin/shadow" to "/root/.shadow/lib"
-- Installing: /root/.shadow/lib/libshadow-interpose.so
-- Set runtime path of "/root/.shadow/lib/libshadow-interpose.so" to "/root/.shadow/lib"
-- Installing: /root/.shadow/lib/libshadow-interpose-helper.so
-- Installing: /root/.shadow/bin/tgen
-- Set runtime path of "/root/.shadow/bin/tgen" to "/root/.shadow/lib"
-- Installing: /root/.shadow/lib/libshadow-plugin-tgen.so
-- Set runtime path of "/root/.shadow/lib/libshadow-plugin-tgen.so" to "/root/.shadow/lib"
-- Installing: /root/.shadow/bin/elfedit
-- Installing: /root/.shadow/bin/display-relocs
-- Installing: /root/.shadow/lib/ldso
-- Set runtime path of "/root/.shadow/lib/ldso" to ""
-- Installing: /root/.shadow/lib/libvdl.so
-- Set runtime path of "/root/.shadow/lib/libvdl.so" to ""
```

Figura 64: Paso 5: parte 1 del log #./setup test

```
Test project /home/alma/shadow/build/src/external/elf-loader
      Start 33: elfloader-test29
      Start 32: elfloader-test28
      Start  1: elfloader-internals
      Start 30: elfloader-test25
 1/34 Test #30: elfloader-test25 .................   Passed    0.30 sec
      Start 13: elfloader-test8_5
 2/34 Test #13: elfloader-test8_5 ................   Passed    0.20 sec
      Start  8: elfloader-test4
 3/34 Test #32: elfloader-test28 .................   Passed    0.65 sec
 4/34 Test  #1: elfloader-internals ..............   Passed    0.65 sec
 5/34 Test  #8: elfloader-test4 ..................   Passed    0.03 sec
      Start  6: elfloader-test2
      Start  7: elfloader-test3
      Start 10: elfloader-test6
 6/34 Test  #6: elfloader-test2 ..................   Passed    0.10 sec
 7/34 Test  #7: elfloader-test3 ..................   Passed    0.10 sec
 8/34 Test #10: elfloader-test6 ..................   Passed    0.10 sec
      Start 34: elfloader-registers
      Start 14: elfloader-test9
      Start 16: elfloader-test11
```

Figura 65: Paso 5: parte 2 del log #./setup test

```
100% tests passed, 0 tests failed out of 34

Total Test time (real) =   2.10 sec
Test project /home/alma/shadow/build
      Start 25: determinism1-shadow
      Start 26: determinism2-shadow
      Start 30: epoll-writeable-shadow
      Start  4: dynlink-shadow
 1/66 Test #25: determinism1-shadow ......................   Passed    0.15 sec
      Start 33: phold-shadow
 2/66 Test #26: determinism2-shadow ......................   Passed    0.46 sec
      Start 34: phold-threaded-shadow
 3/66 Test  #4: dynlink-shadow ...........................   Passed    1.00 sec
      Start 46: sockbuf-shadow
```

Figura 66: Paso 5: parte 3 del log #./setup test

```
      Start 55: tcp-nonblocking-epoll-loopback
62/66 Test #55: tcp-nonblocking-epoll-loopback ...........   Passed    0.20 sec
      Start 47: tcp-blocking-loopback
63/66 Test #47: tcp-blocking-loopback ....................   Passed    0.20 sec
      Start 59: tcp-nonblocking-select-loopback
64/66 Test #59: tcp-nonblocking-select-loopback ..........   Passed    0.20 sec
      Start 63: tcp-iov
65/66 Test #63: tcp-iov ..................................   Passed    0.20 sec
      Start 51: tcp-nonblocking-poll-loopback
66/66 Test #51: tcp-nonblocking-poll-loopback ............   Passed    0.21 sec

100% tests passed, 0 tests failed out of 66

Total Test time (real) =   9.45 sec
```

Figura 67: Paso 5: parte 4 del log #./setup test

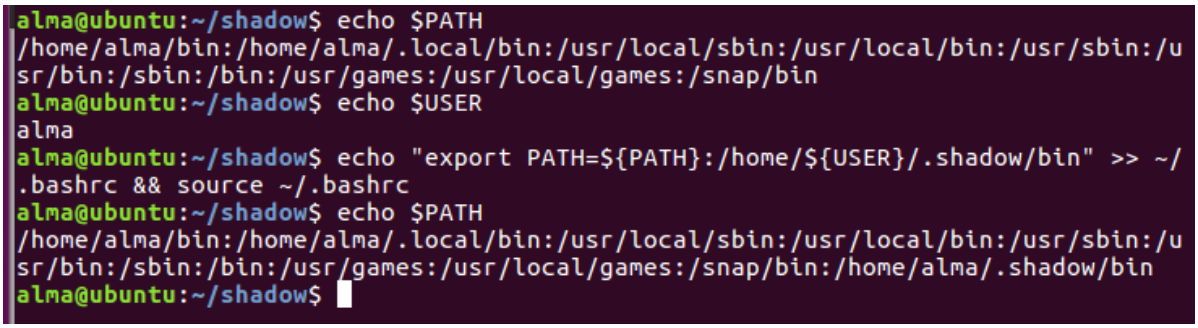


Ilustración 68: Paso 6

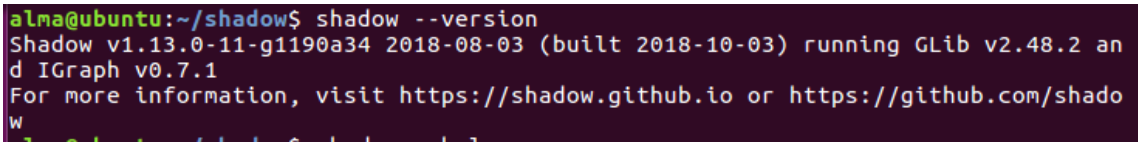


Figura 69: Paso 7 parte 1: #shadow --version

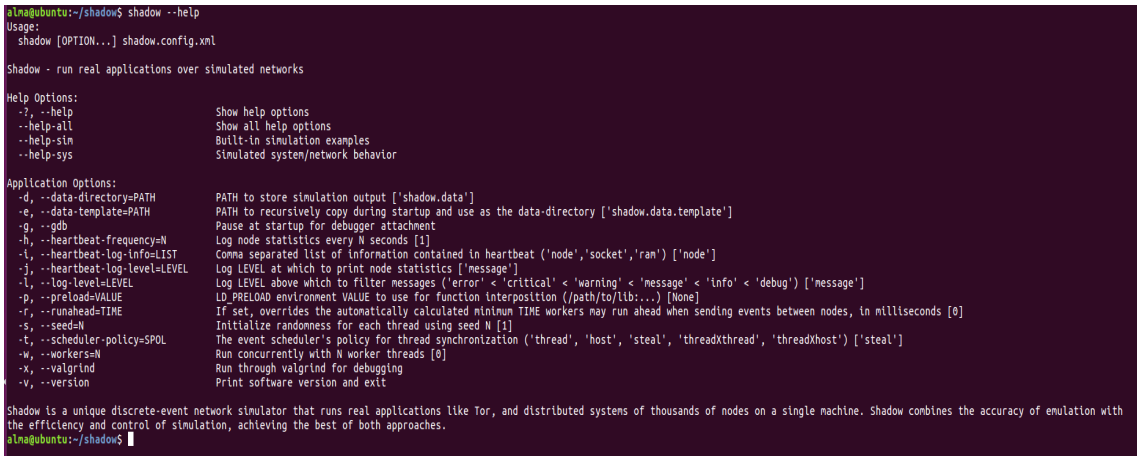


Figura 70: Paso 7 parte 2: #shadow --help

9 Ejemplo práctico: Tor en Shadow

9.1 Instalación del plugin shadow-plugin-tor

Para ejecutar Tor en Shadow primero es necesario instalar el plugin de Shadow “shadow-plugin-tor”.

Se puede obtener en: `#git clone` https://github.com/shadow/shadow-plugin-tor.git

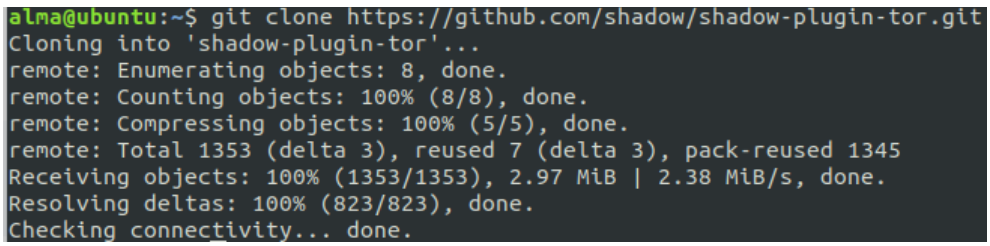


Figura 71: Instalación shadow-plugin-tor

Los comandos a lanzar a continuación son:

```
#cd shadow-plugin-tor
#./setup dependencies
#./setup build
#./setup install
```

Se pueden encontrar ejemplos de configuraciones en el directorio: `shadow-plugin-tor/resource`

El fichero `shadow.config.xml` debe especificar que el plugin de Tor y las bibliotecas de precarga de Tor deben estar cargados, y que debe ejecutar una instancia del plugin Tor como un proceso virtual.

```
<plugin id="tor" path="~/.shadow/lib/libshadow-plugin-tor.so" />
 <plugin id="tor-preload" path="~/.shadow/lib/libshadow-preload-tor.so" />
 <host id="relay">
   <process plugin="tor" preload="tor-preload" arguments="[...]"  [...] />
 </host>
```

9.2 Ejecución de la simulación

Para lanzar la simulación, se ha de ejecutar el siguiente comando:

```
#shadow shadow.config.xml > shadow.log
```

El directorio `shadow.data/hosts/` contiene un directorio de datos privados para cada host que se ejecuta en el experimento.

Los tipos de hosts que se crean para esta simulación son: bridge, torclient, torhiddenserver, nodos relay, nodos exit y 4authority. También hay posibilidad de crear hosts del tipo torhiddenclient y torbridgeclient.

Los archivos comunes para el torclient, 4authority, relay, exit, bridge y hiddenserver son: los logs torctl.log y tor.log, cached-certs, cached-microdesc-consensus, cached-microdescs.new, lock y state.

Además en el torclient y en el hiddenserver está el log tgen.log. Por otro lado en el hiddenserver también se crea un directorio llamado hs que contiene los archivos hostname y private_key.

En el relay, exit1 y el bridge están: fingerprint, cached-consensus y cached-descriptors.new. Y directorios con:

- keys (master_id_public_key, master_id_secret_key, signing_secret_key, secret_id_key, secret_onion_key y secret_onion_key_ntor)
- diff-cache

En el bridge además se crea el fichero hashed-fingerprint.

En el 4authority además se generan los archivos: cached-descriptors, cached-descriptors.new, cached-extrainfo.new, key-pinning-journal, router.stability, sr-state, v3-status-votes, fingerprint, cached-consensus y cached-descriptors.new. Y directorios con:

- keys (authority_certificate, authority_identity_key, master_id_public_key, master_id_secret_key, signing_cert, signing_secret_key, secret_id_key, secret_onion_key y secret_onion_key_ntor)
- diff-cache

Podemos ver estos datos en las siguientes figuras:

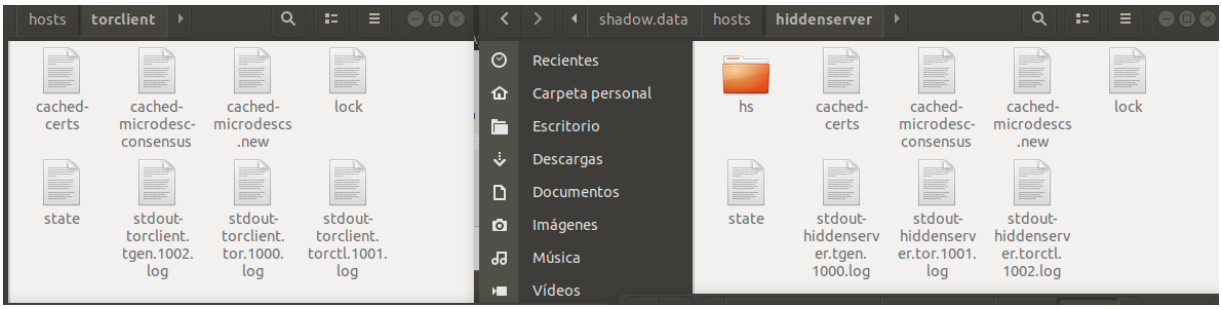


Figura 72: Contenido de los hosts de la simulación parte 1

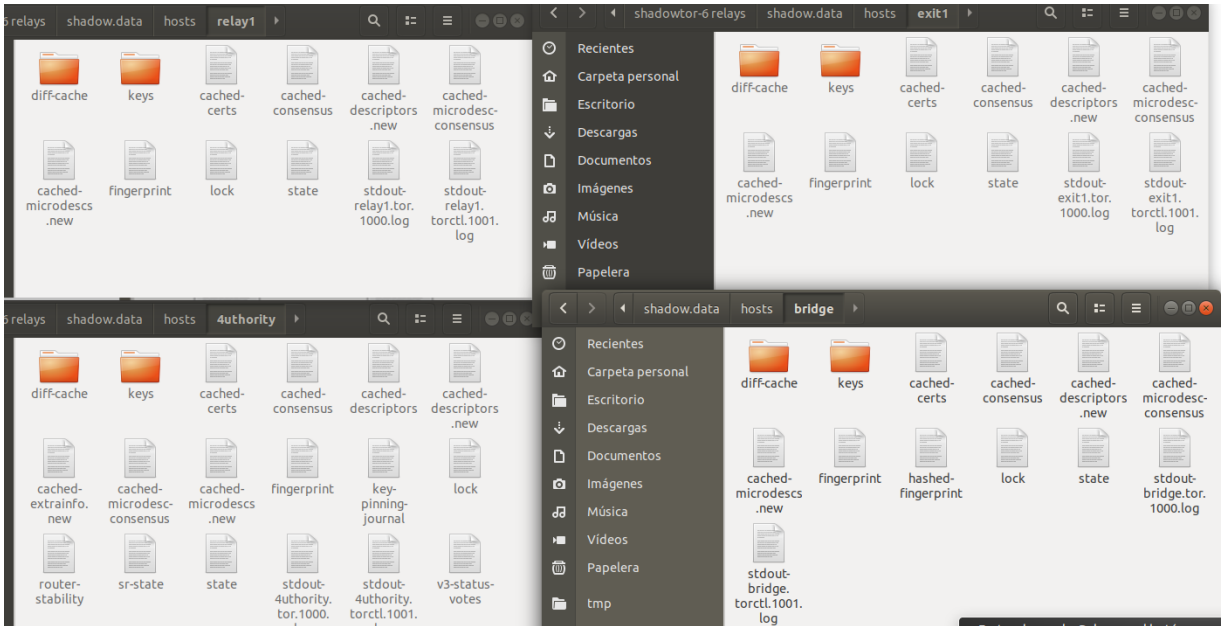


Figura 73: Contenido de los hosts de la simulación parte 2

9.2.1 Archivos destacables

shadow.config.xml: Este fichero especifica cuándo se crea cada nodo virtual y qué software corre en cada nodo virtual. También indica la proporción de nodos de entrada, de salida y nodos intermedios en la simulación.

Cuando se simula un nodo en una simulación de Tor en Shadow se le asigna una geo-localización (por ejemplo con un código del pais como "US") y cuando dos nodos se comunican entre ellos, su comunicación se ve afectada por la latencia, pérdida de paquetes y jitter que existe entre las dos localizaciones. Esto se puede ver en la figura siguiente:

```
    <topology>
    <![CDATA[<?xml version="1.0" encoding="utf-8"?><graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
      <key attr.name="packetloss" attr.type="double" for="edge" id="d9" />
      <key attr.name="jitter" attr.type="double" for="edge" id="d8" />
      <key attr.name="latency" attr.type="double" for="edge" id="d7" />
      <key attr.name="type" attr.type="string" for="node" id="d5" />
      <key attr.name="bandwidthup" attr.type="int" for="node" id="d4" />
      <key attr.name="bandwidthdown" attr.type="int" for="node" id="d3" />
      <key attr.name="countrycode" attr.type="string" for="node" id="d2" />
      <key attr.name="ip" attr.type="string" for="node" id="d1" />
      <key attr.name="packetloss" attr.type="double" for="node" id="d0" />
      <graph edgedefault="undirected">
        <node id="poi-1">
          <data key="d0">0.0</data>
          <data key="d1">0.0.0.0</data>
          <data key="d2">US</data>
          <data key="d3">10240</data>
          <data key="d4">10240</data>
          <data key="d5">net</data>
        </node>
        <edge source="poi-1" target="poi-1">
          <data key="d7">50.0</data>
          <data key="d8">0.0</data>
          <data key="d9">0.0</data>
        </edge>
      </graph>
    </graphml>]]>
    </topology>
```

Figura 74: Especificación de latencia, jitter, ip, etc en shadow.config.xml

tgen: Este archivo es un generador básico de tráfico que permite simular los comportamientos de tráfico de los usuarios, como son las descargas bulk y web surfing.

9.2.2 Análisis de los logs

Cuando se ejecuta Tor en Shadow, se crean varios logs: `client-transfers-complete.log`, `shadow.log`, los logs de cada host (`torctl.log` y `tor.log`) y el log `tgen.log` para el cliente y el hidden server.

El log `client-transfers-complete.log` es un archivo que recoge las transferencias completas ejecutadas entre los hosts.

```
2000-01-01 00:21:21 946686081.111927 [message] [shd-tgen-transfer.c:1519] [_tgentransfer_log] [transfer-complete] transport TCP,246,NULL:11.0.0.8:45795,NULL:0.0.0.0:0,fileserver:
11.0.0.1:80,state=SUCCESS,error=NONE transfer transfer,5,client,GET,1048576,fileserver,11,state=SUCCESS,error=NONE total-bytes-read=1048648 total-bytes-write=51 payload-bytes-read=1048576/1048576
(100.00%) usecs-to-socket-create=0 usecs-to-socket-connect=100000 usecs-to-proxy-init=-1 usecs-to-proxy-choice=-1 usecs-to-proxy-request=-1 usecs-to-proxy-response=-1 usecs-to-command=100000 usecs-to-
response=200013 usecs-to-first-byte=200013 usecs-to-last-byte=822386 usecs-to-checksum=822386
```

Figura 75: log client-transfers-complete.log

Se puede saber el número de transferencias realizadas simplemente realizando un búsqueda en el archivo por “transfer-complete” y viendo el número de resultados que se obtiene o bien ejecutando el siguiente comando en consola:

```
#for d in shadow.data/hosts/*client*; do grep "transfer-complete" ${d}/* ;
done | tee client-transfers-complete.log | wc -l
```

El log `shadow.log` imprime mensajes de todos los procesos de simulación de todos los hosts. Este registro se usa para verificar resultados de simulación y el rendimiento de las redes virtuales creadas.

El `log torctl.log` muestra el esquema de conexiones entre los nodos dentro de la simulación y el host concreto del log, mientras que el log `tor.log` muestra todos los procesos de la simulación de este host concreto, ya sean intentos de conexión, conexiones fallidas y exitosas, descarga de ficheros, gestión de autenticación, envío y recepción de celdas, etc.

9.2.2.1 Formato del log client-transfers-complete.log

El mensaje transfer-complete muestra cada conexión construída y los tiempos de recolección de datos de conexiones exitosas. Clientes conectados exitosamente a un servidor tienen varios tipos de datos. Son:

- *total-bytes-read:* cantidad de datos descargados
- *total-bytes-write:* cantidad de datos cargados
- *payload-bytes-read:* cantidad total de datos recopilados de la combinación de carga y descarga
- *msecs-to-command:* tiempo en que el comando de conexión se inicia (en milisegundos)
- *msecs-to-response:* tiempo de respuesta de la conexión (en milisegundos)
- *msecs-to-first-byte:* tiempo que se tarda en obtener el primer byte de datos (en milisegundos)
- *msecs-to-last-byte:* tiempo que se tarda en obtener el último byte de datos (en milisegundos)
- *msecs-to-checksum:* tiempo para validar los datos adquiridos (en milisegundos)

9.2.2.2 Formato del log shadow.log

El archivo shadow.log posee el siguiente formato:

```
real-time [thread-id] virtual-time [logdomain-loglevel] [hostname~ip] [function-name] MESSAGE
```

- *real-time*: el tiempo real desde el comienzo del experimento representado en `horas:minutos:segundos:microsegundos`
- *thread-id*: el identificador del hilo de ejecución que generó el mensaje.
- *Virtual-time*: el tiempo simulado desde el comienzo del experimento, representado en `horas:minutos:segundos:nanosegundos`
- *logdomain*: ya sea shadow o el nombre de uno de los plugins como se especifica en la etiqueta *id* del elemento *plugin* en el fichero XML (`tgen`, `tor`, `bitcoin`)
- *loglevel*: uno de los siguientes: `error` < `critical` < `warning` < `message` < `info` < `debug`, en ese orden dado.
- *Hostname*: el nombre del nodo como se especifica en la etiqueta *id* del elemento *node* en el fichero XML.
- *Ip*: la dirección IP del nodo como se especifica en la etiqueta *ip* del elemento *node* en el fichero XML, o una dirección IP aleatoria si no está especificado.
- *Function-name*: el nombre de la función logging del mensaje.
- *MESSAGE*: el mensaje concreto que se ha de registrar.

9.2.3 Creación de los hosts

Se procede a ejecutar la simulación que viene por defecto en el pluginde Shadow `shadow-plugin-tor` para una configuración mínima de Tor, que incluye dos nodos exit, dos nodos

relay, un bridge, un servidor de directorios 4authority, un torclient, un torhiddenclient, un torbridgeclient, un client, un fileserver y un hiddenserver.

En el log shadow.log se ve cómo se crean los hosts y se asignan IPs:

fileserver~11.0.0.1
hiddenserver~11.0.0.2
4uthority~11.0.0.3
exit1~11.0.0.4
exit2~11.0.0.5
relay1~11.0.0.6
relay2~11.0.0.7
client~11.0.0.8
torclient~11.0.0.9
torhiddenclient~11.0.0.10 (hxsttdz4esasch5x.onion:80)
torbridgeclient~11.0.0.11
bridge~100.0.0.1

```
00:00:00.016556 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '2' name 'fileserver'
00:00:00.016615 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '3' name 'hiddenserver'
00:00:00.016631 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '4' name '4uthority'
00:00:00.016645 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '5' name 'exit1'
00:00:00.016665 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '6' name 'exit2'
00:00:00.016679 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '7' name 'relay1'
00:00:00.016696 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '8' name 'relay2'
00:00:00.016731 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '9' name 'bridge'
00:00:00.016744 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '10' name 'client'
00:00:00.016754 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '11' name 'torclient'
00:00:00.016771 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '12' name 'torhiddenclient'
00:00:00.016803 [thread-0] n/a [message] [n/a] [shd-host.c:104] [host_new] Created host id '13' name 'torbridgeclient'
```

Figura 76: Creacion de los hosts

```
00:00:00.016981 [thread-0] 00:00:00.000000000 [message] [fileserver~11.0.0.1] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.1' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017057 [thread-0] 00:00:00.000000000 [message] [hiddenserver~11.0.0.2] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.2' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017123 [thread-0] 00:00:00.000000000 [message] [4uthority~11.0.0.3] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.3' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017189 [thread-0] 00:00:00.000000000 [message] [exit1~11.0.0.4] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.4' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017250 [thread-0] 00:00:00.000000000 [message] [exit2~11.0.0.5] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.5' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017309 [thread-0] 00:00:00.000000000 [message] [relay1~11.0.0.6] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.6' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017370 [thread-0] 00:00:00.000000000 [message] [relay2~11.0.0.7] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.7' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017431 [thread-0] 00:00:00.000000000 [message] [bridge~100.0.0.1] [shd-topology.c:2409] [topology_attach] attached address '100.0.0.1' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=100.0.0.1, citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017586 [thread-0] 00:00:00.000000000 [message] [client~11.0.0.8] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.8' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017672 [thread-0] 00:00:00.000000000 [message] [torclient~11.0.0.9] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.9' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017793 [thread-0] 00:00:00.000000000 [message] [torhiddenclient~11.0.0.10] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.10' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))

00:00:00.017877 [thread-0] 00:00:00.000000000 [message] [torbridgeclient~11.0.0.11] [shd-topology.c:2409] [topology_attach] attached address '11.0.0.11' to vertex 0 ('poi-1') with attributes (ip=0.0.0.0,
citycode=(null), countrycode=US, geocode=(null), type=net) using hints (ip=(null), citycode=(null), countrycode=(null), geocode=(null), type=(null))
```

Figura 77: Asignación de IPs

```
00:00:00.017012 [thread-0] 00:00:00.000000000 [message] [fileserver~11.0.0.1] [shd-host.c:319] [host_boot] Booted host id '2' name 'fileserver' with seed 1105148392, ip 11.0.0.1, 102400 bwUpKiBps, 102400
bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017050 [thread-0] 00:00:00.000000000 [warning] [hiddenserver~11.0.0.2] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio
of 1.0 to the simulator host

00:00:00.017081 [thread-0] 00:00:00.000000000 [message] [hiddenserver~11.0.0.2] [shd-host.c:319] [host_boot] Booted host id '3' name 'hiddenserver' with seed 2853875478, ip 11.0.0.2, 102400 bwUpKiBps,
102400 bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017117 [thread-0] 00:00:00.000000000 [warning] [4uthority~11.0.0.3] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio of
1.0 to the simulator host

00:00:00.017150 [thread-0] 00:00:00.000000000 [message] [4uthority~11.0.0.3] [shd-host.c:319] [host_boot] Booted host id '4' name '4uthority' with seed 645940202, ip 11.0.0.3, 10240 bwUpKiBps, 10240
bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017184 [thread-0] 00:00:00.000000000 [warning] [exit1~11.0.0.4] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio of 1.0
to the simulator host

00:00:00.017209 [thread-0] 00:00:00.000000000 [message] [exit1~11.0.0.4] [shd-host.c:319] [host_boot] Booted host id '5' name 'exit1' with seed 2059732742, ip 11.0.0.4, 10240 bwUpKiBps, 10240 bwDownKiBps,
131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017242 [thread-0] 00:00:00.000000000 [warning] [exit2~11.0.0.5] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio of 1.0
to the simulator host

00:00:00.017271 [thread-0] 00:00:00.000000000 [message] [exit2~11.0.0.5] [shd-host.c:319] [host_boot] Booted host id '6' name 'exit2' with seed 663466108, ip 11.0.0.5, 10240 bwUpKiBps, 10240 bwDownKiBps,
131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017303 [thread-0] 00:00:00.000000000 [warning] [relay1~11.0.0.6] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio of
1.0 to the simulator host

00:00:00.017332 [thread-0] 00:00:00.000000000 [message] [relay1~11.0.0.6] [shd-host.c:319] [host_boot] Booted host id '7' name 'relay1' with seed 2362317446, ip 11.0.0.6, 10240 bwUpKiBps, 10240
bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017365 [thread-0] 00:00:00.000000000 [warning] [relay2~11.0.0.7] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio of
1.0 to the simulator host

00:00:00.017392 [thread-0] 00:00:00.000000000 [message] [relay2~11.0.0.7] [shd-host.c:319] [host_boot] Booted host id '8' name 'relay2' with seed 2717698814, ip 11.0.0.7, 10240 bwUpKiBps, 10240
bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017425 [thread-0] 00:00:00.000000000 [warning] [bridge~100.0.0.1] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio of
1.0 to the simulator host
```

Figura 78: Arranque de los equipos parte 1

```
00:00:00.017451 [thread-0] 00:00:00.000000000 [message] [bridge~100.0.0.1] [shd-host.c:319] [host_boot] Booted host id '9' name 'bridge' with seed 742907270, ip 100.0.0.1, 10240 bwUpKiBps, 10240
bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017569 [thread-0] 00:00:00.000000000 [warning] [client~11.0.0.8] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio of
1.0 to the simulator host

00:00:00.017611 [thread-0] 00:00:00.000000000 [message] [client~11.0.0.8] [shd-host.c:319] [host_boot] Booted host id '10' name 'client' with seed 2430661882, ip 11.0.0.8, 10240 bwUpKiBps, 10240
bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017665 [thread-0] 00:00:00.000000000 [warning] [torclient~11.0.0.9] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay ratio of
1.0 to the simulator host

00:00:00.017695 [thread-0] 00:00:00.000000000 [message] [torclient~11.0.0.9] [shd-host.c:319] [host_boot] Booted host id '11' name 'torclient' with seed 286595330, ip 11.0.0.9, 10240 bwUpKiBps, 10240
bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017783 [thread-0] 00:00:00.000000000 [warning] [torhiddenclient~11.0.0.10] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay
ratio of 1.0 to the simulator host

00:00:00.017819 [thread-0] 00:00:00.000000000 [message] [torhiddenclient~11.0.0.10] [shd-host.c:319] [host_boot] Booted host id '12' name 'torhiddenclient' with seed 1640239920, ip 11.0.0.10, 10240
bwUpKiBps, 10240 bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision
00:00:00.017868 [thread-0] 00:00:00.000000000 [warning] [torbridgeclient~11.0.0.11] [shd-cpu.c:40] [cpu_new] unable to determine raw CPU frequency, setting 2500000 KHz as a raw estimate, and using delay
ratio of 1.0 to the simulator host

00:00:00.017900 [thread-0] 00:00:00.000000000 [message] [torbridgeclient~11.0.0.11] [shd-host.c:319] [host_boot] Booted host id '13' name 'torbridgeclient' with seed 2529334784, ip 11.0.0.11, 10240
bwUpKiBps, 10240 bwDownKiBps, 131072 initSockSendBufSize, 174760 initSockRecvBufSize, 2500000 cpuFrequency, 0 cpuThreshold, 200 cpuPrecision


00:00:00.017905 [thread-0] n/a [message] [n/a] [shd-scheduler.c:69] [_scheduler_startHosts] 12 hosts are booted
```

Figura 79: Arranque de los equipos parte 2

También se le asignan fingerprints a los siguientes hosts:

4uthority~ A52CA5B56C64D864F6AE43E56F29ACBD5706DDA1

exit1~ 0A9B1B207FD13A6F117F95CAFA358EEE2234F19A

exit2~ 4EBB385C80A2CA5D671E16F1C722FBFB5F176891

relay1~ 3FB0BD7827C760FE7F9DD810FCB10322D63AB4CF

relay2~ FF197204099FA0E507FA46D41FED97D3337B4BAA

bridge~F63C257B0819549FCD3E476FB534C08E550AC29D

El dominio del hidden server es hxsttdz4esasch5x.onion

9.2.4 Creación de un circuíto de tres nodos

A continuación se ve un ejemplo de creación de un circuíto de tres nodos, sobre el que se basa la propiedad de anonimato de la red Tor. Para ello se ha consultado el log tor.log del host hiddenserver. En este tipo de circuítos se espera encontrar las celdas CREATE y EXTEND.

Primero se observa que se crea el primer salto (first hop) a través de la celda CREATE al nodo relay2.

```
Jan 01 00:15:03.000 [info] circuit_send_first_onion_skin(): First hop: finished sending CREATE cell to '$FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2 at 11.0.0.7'
Jan 01 00:15:03.000 [info] channel_tls_process_netinfo_cell(): Got good NETINFO cell from 11.0.0.7:9111; OR connection is now open, using protocol version 5. Its ID digest
FF197204099FA0E507FA46D41FED97D3337B4BAA. Our address is apparently 11.0.0.2.
```

Figura 80: Creación del primer nodo - celda CREATE

A continuación se envía una celda EXTEND y se crea el segundo salto, nodo cuyo fingerprint es F63C257B0819549FCD3E476FB534C08E550AC29D y corresponde con el nodo bridge.

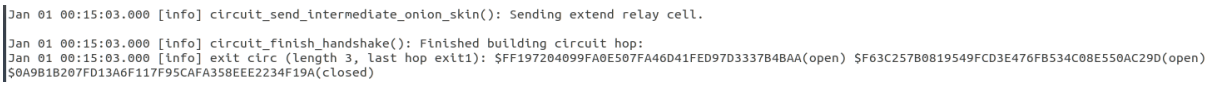


Figura 81: Creación del segundo nodo - celda EXTEND

Después se vuelve a enviar otra celda EXTEND y se elige el nodo exit1 para el último salto. Finalmente se da por construido el circuito (circuit built).

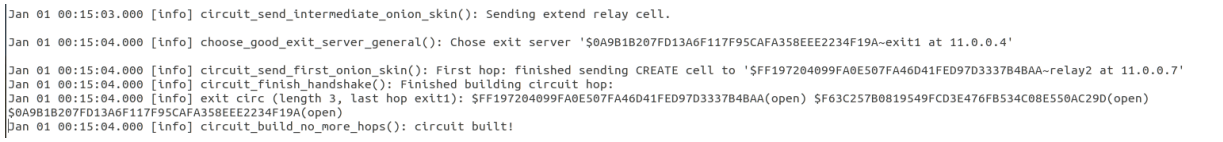


Figura 82: Creación del tercer nodo - celda EXTEND

9.2.5 Otros ejemplos de celdas encontradas

A continuación se mostrarán algunos ejemplos de celdas pertenecientes al protocolo rendezvous.

9.2.5.1 Celdas ESTABLISH INTRO y INTRO ESTABLISHED

En el primer paso del protocolo rendezvous, el hidden service crea circuítos para establecer los puntos de introducción (IPOs) y las celdas que se envían por estos circuitos son RELAY COMMAND ESTABLISH INTRO y RELAY COMMAND INTRO ESTABLISHED.

En el log del nodo relay2 se puede ver un ejemplo de celda ESTABLISH_INTRO para crear un punto de introducción IPO.

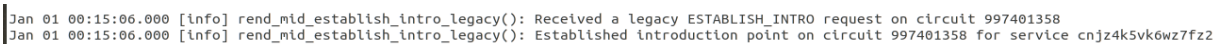


Figura 83: Ejemplo de celda ESTABLISH INTRO

En los logs del hiddenserver se puede ver que se crean tres circuítos para establecer tres puntos de introducción IPOs.

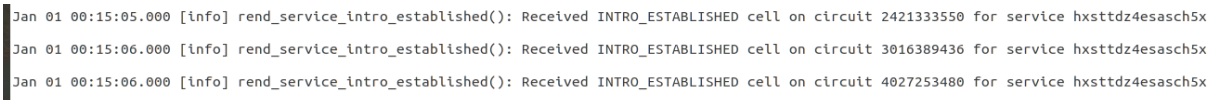


Figura 84: Ejemplo de celda INTRO ESTABLISHED

El circuito 2421333550 tiene ID 6 según se puede ver en el log torctl.log del hiddenserver.

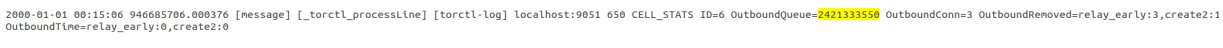


Figura 85: Identificador del circuito 2421333550

Para la creación del circuíto con ID 6 se puede ver en el log torctl.log del hiddenserver, que primero se lanza la ejecución:

Figura 86: Inicio de la creación del circuíto con ID 6 según el log torctl.log

Después se encuentra el primer nodo que corresponde con el relay2.

```
2000-01-01 00:15:05 946685705.119201 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 EXTENDED $FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2 BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME
PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING REND_QUERY=hxsttdz4esasch5x TIME_CREATED=2000-01-01T00:15:05.001195
```

Figura 87: Asignación del primer nodo según el log torctl.log

A continuación se extiende el circuíto añadiéndole el siguiente nodo que será el exit2.

```
2000-01-01 00:15:05 946685705.320219 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 EXTENDED
$FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2,$4EBB385C80A2CA5D671E16F1C722FBFB5F176891~exit2 BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING
REND_QUERY=hxsttdz4esasch5x TIME_CREATED=2000-01-01T00:15:05.001195
```

Figura 88: Asignación del segundo nodo según el log torctl.log

Seguidamente se conecta el tercer nodo, asignándose el nodo exit1.

```
2000-01-01 00:15:05 946685705.629251 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 EXTENDED
$FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2,$4EBB385C80A2CA5D671E16F1C722FBFB5F176891~exit2,$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A~exit1 BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME
PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING REND_QUERY=hxsttdz4esasch5x TIME_CREATED=2000-01-01T00:15:05.001195
```

Figura 89: Asignación del tercer nodo según el log torctl.log

Finalmente se da por concluída la construcción del circuito con los nodos relay2, exit2 y exit1.

```
2000-01-01 00:15:05 946685705.629251 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 BUILT
 $FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2,$4EBB385C80A2CA5D671E16F1C722FBFB5F176891~exit2,$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A~exit1 BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME
PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING REND_QUERY=hxsttdz4esasch5x TIME_CREATED=2000-01-01T00:15:05.001195
```

Figura 90: Circuito de tres nodos creado según el log torctl.log

En la figura siguiente se pueden ver juntos los pasos anteriores:

```
2000-01-01 00:15:05 946685705.001195 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 LAUNCHED BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING
TIME_CREATED=2000-01-01T00:15:05.001195

2000-01-01 00:15:05 946685705.119201 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 EXTENDED $FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2 BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME
PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING REND_QUERY=hxsttdz4esasch5x TIME_CREATED=2000-01-01T00:15:05.001195

2000-01-01 00:15:05 946685705.320219 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 EXTENDED
$FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2,$4EBB385C80A2CA5D671E16F1C722FBFB5F176891~exit2 BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING
REND_QUERY=hxsttdz4esasch5x TIME_CREATED=2000-01-01T00:15:05.001195

2000-01-01 00:15:05 946685705.629251 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 EXTENDED
$FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2,$4EBB385C80A2CA5D671E16F1C722FBFB5F176891~exit2,$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A~exit1 BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME
PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING REND_QUERY=hxsttdz4esasch5x TIME_CREATED=2000-01-01T00:15:05.001195

2000-01-01 00:15:05 946685705.629251 [message] [_torctl_processLine] [torctl-log] localhost:9051 650 CIRC 6 BUILT
 $FF197204099FA0E507FA46D41FED97D3337B4BAA~relay2,$4EBB385C80A2CA5D671E16F1C722FBFB5F176891~exit2,$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A~exit1 BUILD_FLAGS=IS_INTERNAL,NEED_UPTIME
PURPOSE=HS_SERVICE_INTRO HS_STATE=HSSI_CONNECTING REND_QUERY=hxsttdz4esasch5x TIME_CREATED=2000-01-01T00:15:05.001195
```

Figura 91: Creación de los tres nodos según el log torctl.log

En el log tor.log del hidden service para este circuito 2421333550, se puede ver la correspondencia de la creación del circuito para establecer el punto IPO para el servicio del hidden service (hxsttdz4esasch5x):

```
Jan 01 00:15:05.000 [info] rend_service_intro_has_opened(): Established circuit 2421333550 as introduction point for service hxsttdz4esasch5x
Jan 01 00:15:05.000 [info] internal (high-uptime) circ (length 3): $FF197204099FA0E507FA46D41FED97D3337B4BAA(open) $4EBB385C80A2CA5D671E16F1C722FBFB5F176891(open)
$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A(open)
```

Figura 92: Creación de los tres nodos según el log tor.log

Si se buscan los fingerprints de cada uno de los nodos, vemos que efectivamente coinciden con los nodos relay2, exit2 y exit1, como habíamos definido antes para el circuito 2421333550 con ID 6.

$FF197204099FA0E507FA46D41FED97D3337B4BAA(open)	–	relay2
$4EBB385C80A2CA5D671E16F1C722FBFB5F176891(open)	–	exit2
$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A(open)	–	exit1

Para el circuito 3016389436 (ID 7) los nodos son exit1, exit2 y relay2.

$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A(open) – exit1
$4EBB385C80A2CA5D671E16F1C722FBFB5F176891(open) – exit2
$FF197204099FA0E507FA46D41FED97D3337B4BAA(open) – relay2

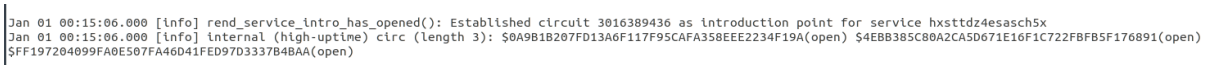


Figura 93: Creación del circuito 3016389436 (ID 7)

Para el circuito 4027253480 (ID 9) los nodos son exit1, exit2 y relay1.

$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A(open) – exit1
$4EBB385C80A2CA5D671E16F1C722FBFB5F176891(open) – exit2
$3FB0BD7827C760FE7F9DD810FCB10322D63AB4CF(open) – relay1

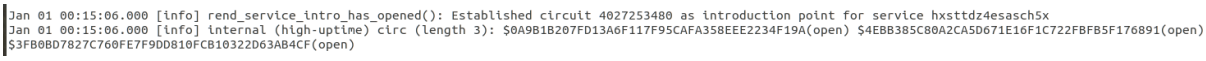


Figura 94: Creación del circuito 4027253480 (ID 9)

9.2.5.2 Celdas ESTABLISH RENDEZVOUS y RENDEZVOUS ESTABLISHED

En el tercer paso del protocolo rendezvous, el cliente crea un circuito para conectarse con el punto de encuentro (RP) y las celdas que se envían son RELAY COMMAND ESTABLISH RENDEZVOUS y RELAY COMMAND RENDEZVOUS ESTABLISHED.

Podemos ver un ejemplo en el nodo exit2.

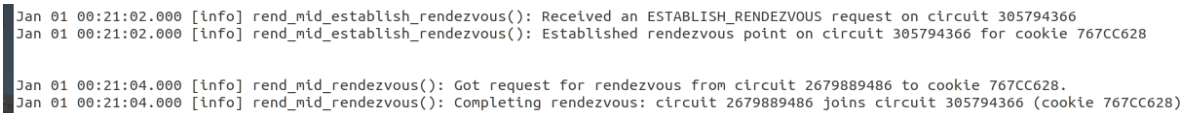


Figura 95: Ejemplo de celdas ESTABLISH RENDEZVOUS y RENDEZVOUS ESTABLISHED

Además podemos ver la cookie rendezvous, cuyo valor es 767CC628. Esta cookie se incluirá en el mensaje que el cliente enviará a uno de los puntos de introducción solicitando que se entregue al servicio onion.

9.2.5.3 Celdas INTRODUCE1, INTRODUCE ACK e INTRODUCE2

En el paso 4 las celdas que se envían son RELAY COMMAND INTRODUCE1, RELAY COMMAND INTRODUCE ACK y RELAY COMMAND INTRODUCE2.

En el nodo relay 1 se puede ver un ejemplo de celda INTRODUCE1, como se puede ver a continuación:

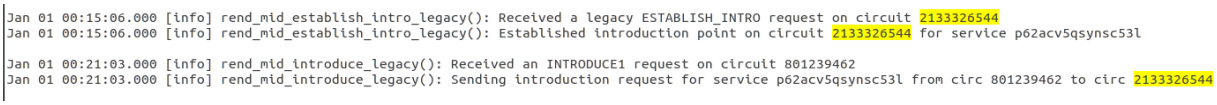


Figura 96: Ejemplo de celda INTRODUCE1 en el relay1

También podemos ver un ejemplo de INTRODUCE1 en el torhiddenclient y que además recibe un ack.

```
Jan 01 00:21:03.000 [info] connection_ap_handshake_attach_circuit(): Found open intro circ 3027468702 (id: 49). Rend circuit 3524028674 (id: 46); Sending introduction. (stream 1 sec old)
Jan 01 00:21:03.000 [info] rend_client_send_introduction(): Sending an INTRODUCE1 cell

Jan 01 00:21:03.000 [info] pathbias_count_use_attempt(): Used circuit 46 is already in path state use succeeded. Circuit is a Hidden service client: Pending rendezvous point currently open.


Jan 01 00:21:03.000 [info] rend_client_introduction_acked(): Received ack. Telling rend circ...


Jan 01 00:21:04.000 [info] connection_ap_handshake_attach_circuit(): pending-join circ 3524028674 (id: 46) already here, with intro ack. Stalling. (stream 2 sec old)
```

Figura 97: Ejemplo de celda INTRODUCE1 en el torhiddenclient y ack

En el hiddenserver podemos ver que aparecen celdas INTRODUCE2 y que de nuevo contienen la cookie rendezvous.

```
Jan 01 00:21:03.000 [info] rend_service_receive_introduction(): Received INTRODUCE2 cell for service "hxsttdz4esasch5x" on circ 4027253480.

Jan 01 00:21:03.000 [info] rend_service_receive_introduction(): Accepted intro; launching circuit to $4EBB385C80A2CA5D671E16F1C722FBFB5F176891~$4EBB385C80A2CA5D67 at 11.0.0.5 (cookie 767CC628) for service
hxsttdz4esasch5x.

Jan 01 00:21:04.000 [info] rend_service_rendezvous_has_opened(): Done building circuit 3076305450 to rendezvous with cookie 767CC628 for service hxsttdz4esasch5x
Jan 01 00:21:04.000 [info] internal circ (length 4): $0A9B1B207FD13A6F117F95CAFA358EEE2234F19A(open) $F63C257B0819549FCD3E476FB534C08E550AC29D(open) $A52CA5B56C64D864F6AE43E56F29ACBD5706DDA1(open)
$4EBB385C80A2CA5D671E16F1C722FBFB5F176891(open)

Jan 01 00:21:05.000 [info] connection_edge_finished_connecting(): Exit connection to "(rendezvous)":8080 (127.0.0.1) established.
```

Figura 98: Ejemplo de celda INTRODUCE2 y cookie rendezvous

9.2.5.4 Celdas RENDEZVOUS2 y BEGIN

En el sexto paso del protocolo rendezvous se envían celdas del tipo RELAY COMMAND RENDEZVOUS2 y RELAY COMMAND BEGIN.

En el torhiddenclient vemos ejemplos de estas celdas. Primero la celda de RENDEZVOUS2 enviada por el hidden service se recibe por el circuito 3524028674 (ID 46), que es el que se va a usar finalmente para la transmisión de datos, como se verá a continuación.

```
Jan 01 00:21:04.000 [info] hs_client_receive_rendezvous2(): Got RENDEZVOUS2 cell from hidden service on circuit 3524028674.
Jan 01 00:21:04.000 [info] connection_ap_handshake_attach_circuit(): rend joined circ 3524028674 (id: 46) already here. Attaching. (stream 2 sec old)
Jan 01 00:21:04.000 [info] pathbias_count_use_attempt(): Used circuit 46 is already in path state use succeeded. Circuit is a Hidden service client: Active rendezvous point currently open.


Jan 01 00:21:04.000 [info] rend_client_note_connection_attempt_ended(): Connection attempt for hxsttdz4esasch5x has ended; cleaning up temporary state.
Jan 01 00:21:04.000 [info] link_apconn_to_circ(): Looks like completed circuit to hidden service does allow optimistic data for connection to hxsttdz4esasch5x
```

Figura 99: Ejemplo de celda RENDEZVOUS2

Seguidamente se ve un ejemplo de la celda BEGIN, con una celda relay 0 que envía un stream con identificador 45374. El circuíto por el que se mueve es el número 3524028674 (ID 46), formado por los nodos relay1, exit1 y exit2. Esto se comprueba por el valor de sus fingerprints.

$3FB0BD7827C760FE7F9DD810FCB10322D63AB4CF	-	relay1
$0A9B1B207FD13A6F117F95CAFA358EEE2234F19A	-	exit1
$4EBB385C80A2CA5D671E16F1C722FBFB5F176891	-	exit2

```
Jan 01 00:21:04.000 [info] connection_ap_handshake_send_begin(): Sending relay cell 0 on circ 3524028674 to begin stream 45374.
Jan 01 00:21:04.000 [info] connection_ap_handshake_send_begin(): Address/port sent, ap socket 1313, n_circ_id 3524028674
Jan 01 00:21:05.000 [info] connection_edge_process_relay_cell_not_open(): 'connected' received for circid 3524028674 streamid 45374 after 1 seconds.
Jan 01 00:21:05.000 [info] internal (high-uptime) circ (length 3): $3FB0BD7827C760FE7F9DD810FCB10322D63AB4CF(open) $0A9B1B207FD13A6F117F95CAFA358EEE2234F19A(open)
$4EBB385C80A2CA5D671E16F1C722FBFB5F176891(open)

Jan 01 00:21:09.000 [info] connection_edge_process_relay_cell(): 1313: end cell (closed normally) for stream 45374. Removing stream.
```

Figura 100: Ejemplo de celda BEGIN

10. Conclusiones

Aunque la idea generalizada sobre la red Tor en la actualidad es que proporciona un anonimato absoluto, no se debe de tomar como algo infalible, puesto que se ha comprobado a lo largo de los años que ha sido vulnerada en diversas ocasiones demostrando que no es tan segura como se anunciaba inicialmente. Consecuentemente, se han de tomar una serie de medidas de seguridad operacionales (OPSEC) adicionales con el fin de proteger lo máximo posible la privacidad y el anonimato del usuario.

Puesto que muchos de los motivos por los que se usa Tor son con fines criminales, se podría esperar que la mayoría de los sitios alojados en dicha red puedan contener algún tipo de código dañino en sus páginas, que podría implicar la instalación de *malware* en el host del usuario, el robo de información confidencial o el minado de criptomoneda, entre otros.

Con respecto a las comunicaciones, si bien es cierto que éstas van cifradas entre los nodos que componen el circuito establecido entre un cliente y el servidor web dentro de la red Tor, esto no ocurre entre el nodo de salida y el destino final, de manera que, de no estar usando cifrado adicional para proteger las comunicaciones, los datos se transmitirían en claro en el último tramo, siendo potencialmente visibles por el propietario del nodo de salida, el ISP que da servicio en dicho tramo y el sitio web destino. Para mejorar la seguridad en este punto, se pueden seguir las siguientes recomendaciones[1]:

- Acceder a webs que posean un certificado SSL y realizar conexiones https
- Emplear el navegador Tor Browser para el acceso a la red Tor
- No habilitar o instalar plugins del navegador
- No descargar Torrents con Tor
- No abrir documentos descargados a través de Tor mientras se esté online
- Usar nodos puentes Tor

Por otro lado, es necesario destacar las repercusiones que se pueden producir tras una navegación imprudente en esta red, ya que el simple hecho de acceder a determinados sitios puede ser considerado un delito que puede conllevar penas de prisión.

Finalmente, con respecto al análisis de los servicios que ofrece Tor, es desalentador observar cómo a pesar de que las autoridades consiguen clausurar los mercados negros, vuelven a surgir de nuevo estos negocios realizando las mismas transacciones ilegales, ya que mientras siga existiendo demanda, aparecerán individuos dispuestos a suministrar el producto asumiendo los riesgos de su ilegalidad. Es por esto, que sopesando las ventajas y desventajas, la duda que se plantea es si a nivel global, la balanza realmente se inclina del lado de los beneficios que brinda esta red y si su existencia está perturbando el equilibrio de las sociedades que buscan la democracia y la paz en sus habitantes.

Lo que sí es destacable como punto de reflexión, es el concepto de libertad virtual y el derecho de privacidad y libertad de expresión en Internet, por lo que inicialmente se creó esta red. En un mundo totalmente globalizado en el que las nuevas tecnologías han irrumpido en todos los ámbitos posibles, se hace imprescindible concienciar a la población sobre estos derechos.

1 https://www.torproject.org/download/download.html.en#warning

11. Bibliografía

Zhen Ling, Junzhou Luo, Kui Wu and Xinwen Fu. Protocol-level Hidden Server Discovery. (s.f.) http://www.cs.uml.edu/~xinwenfu/paper/HiddenServer.pdf

The Tor Project. Tor's Fall Harvest: the Next Generation of Onion Services. (2017, 2 de noviembre) https://blog.torproject.org/tors-fall-harvest-next-generation-onion-services

Jansen, Rob and Hopper, Nick. Shadow Design Screencast. (2012, 2 de febrero) http://youtu.be/Tb7m8OdpD8A

Jansen, Rob. Github. Mission: Run Tor in a Box. (2017, 7 septiembre) https://github.com/shadow/shadow/wiki/0-Design-Overview

The Tor Project. Welcome to Tor Metrics! (s.f.) https://metrics.torproject.org

Agudo, Sergio. Así es el mapa del uso de Tor en cada país del mundo. (2017, 20 de febrero) https://www.genbeta.com/seguridad/asi-es-el-mapa-del-uso-de-tor-en-cada-pais-del-mundo

Guerra, Manuel y Rodríguez, Fco J. Mitos, debilidades y delitos imperfectos en Tor. (2018, 4 de enero) https://www.youtube.com/watch?v=sd4sKwm1mLU&feature=youtu.be

Richard. Further Tor Vulnerabilities Discovered: Public IP Address of Tor Hidden Sites Identified via SSL Certificates. (2018, 7 de septiembre) https://darkwebnews.com/anonymity-tools/tor/tor-hidden-sites-ip-identified-via-ssl-certificates/

Abrams, Lawrence. Public IP Addresses of Tor Sites Exposed via SSL Certificates. (2018, 4 de septiembre) https://www.bleepingcomputer.com/news/security/public-ip-addresses-of-tor-sites-exposed-via-ssl-certificates/

Barroyeta, Rosselyn. IP’s públicas de Tor son expuestas mediante certificados SSL. (2018, 4 de septiembre) https://www.tekcrispy.com/2018/09/04/ip-publica-tor-expuesta/

Amir, Waqas. Misconfigured Tor sites using SSL certificates exposing public IP addresses. (2018, 5 septiembre) https://www.hackread.com/misconfigured-tor-sites-using-ssl-certificates-exposing-public-ip-addresses/

Husam Al Jawaheri, Mashael Al Sabah, Yazan Boshmaf, Aiman Erbad. When A Small Leak Sinks A Great Ship: Deanonymizing Tor Hidden Service Users Through Bitcoin Transactions Analysis. (2018, 11 abril) https://arxiv.org/abs/1801.07501

Sameeh, Tamer. An Overview of Modern Tor Deanonymization Attacks. (2017, 12 de septiembre) ttps://www.deepdotweb.com/2017/09/12/overview-modern-tor-deanonymization-attacks/

Sameeh, Tamer. Targeting Adversaries & Deanonymization Attacks Against Tor Users. (2017, 21 de agosto). https://www.deepdotweb.com/2017/08/21/targeting-adversaries-deanonymization-attacks-tor-users/

INCIBE-CERT. Vulnerabilidad en TOR capaz de revelar la IP de los usuarios. (2017, 3 de noviembre). https://www.incibe-cert.es/alerta-temprana/bitacora-ciberseguridad/vulnerabilidad-tor-capaz-revelar-ip-los-usuarios

Amir, Waqas. Flaw in Tor Browser Leads to Leaking of Your Real IP Address. (2017, 4 de noviembre) https://www.hackread.com/flaw-in-tor-browser-leads-to-leaking-of-your-real-ip-address/

Velasco, Rubén. TorMoil, una grave vulnerabilidad que expone la IP de los usuarios de Tor. (2017, 6 de noviembre) https://www.redeszone.net/2017/11/06/tormoil-vulnerabilidad-tor/

López, Francisco. Vulnerabiidad en Tor Browser permite obtener las IPs reales de los usuarios. (2017, 6 de noviembre) https://unaaldia.hispasec.com/2017/11/vulnerabiidad-en-tor-browser-permite-obtener-las-ips-reales-de-los-usuarios.html

Alarcón Sánchez, Ramón. Las principales vulnerabilidades de Tor. (2017, 17 de agosto) https://periodismoactual.com/las-principales-vulnerabilidades-de-tor

Takuya Fukui. De-anonymizing Tor traffic with website fingerprinting. (2017, 22 de abril) https://witestlab.poly.edu/blog/de-anonymizing-tor-traffic-with-website-fingerprinting/

Goodin, Dan. Firefox 0-day in the wild is being used to attack Tor users. (2016, 30 de noviembre) https://arstechnica.com/information-technology/2016/11/firefox-0day-used-against-tor-users-almost-identical-to-one-fbi-used-in-2013/

Harbison, Cammy. Deanonymizing Tor Hidden Service Traffic Through HSDir Is A Cake Walk, Say Researchers: HITB Presenters Showcase New Threats. (2016, 29 de junio) https://www.player.one/deanonymizing-tor-hidden-service-traffic-through-hsdir-cake-walk-say-researchers-hitb-445328

Yago, Jesús. Desanonimizando nodos Tor gracias a SSH. (2015, 23 de enero) http://www.securitybydefault.com/2015/01/desanonimizando-nodos-tor-gracias-ssh.html

White, Nic. Undercover Australian police shared child porn with paedophiles for a YEAR after taking over the world's biggest abuse website to catch offenders. (2017, 8 de octubre) https://www.dailymail.co.uk/news/article-4959726/Police-shared-child-porn-paedophiles-year.html

Cox, Joseph. An Admin's Foolish Errors Helped the FBI Unmask Child Porn Site 'Playpen'. (2016, 16 de mayo) https://motherboard.vice.com/en_us/article/nz7e8x/an-admins-foolish-errors-helped-the-fbi-unmask-child-porn-site-playpen

Hoffman, Chris. How to Create a Hidden Service Tor Site to Set Up an Anonymous Website or Server. (2012, 2 de abril) https://www.makeuseof.com/tag/create-hidden-service-tor-site-set-anonymous-website-server/

Pastor, Javier. Así cayeron AlphaBay y Hansa, las sucesoras de Silk Road que dominaban la venta de productos ilegales en la Dark Web. (2017, 23 de julio) https://www.xataka.com/legislacion-y-derechos/asi-cayeron-alphabay-y-hansa-las-sucesoras-de-silk-road-que-dominaban-la-venta-de-productos-ilegales-en-la-dark-web

Brown, Aaron. VLC WARNING - Hackers can take control of YOUR computer using THIS feature. (2017, 29 de mayo) https://www.express.co.uk/life-style/science-technology/810324/VLC-Player-Download-Hackers-Subtitles

Khandelwal, Swati. Zerodium Offers $1 Million for Tor Browser 0-Days That It will Resell to Governments. (2017, 13 de septiembre) https://thehackernews.com/2017/09/tor-zero-day-exploits.html

Garrido Courel, Maite. El cierre de Freedom Hosting y por qué Tor sigue siendo segura. (2013, 9 de agosto) https://www.eldiario.es/turing/red-Tor_0_162384283.html

Kumar, Mohit. Firefox Zero-Day Exploit used by FBI to shutdown Child porn on Tor Network hosting; Tor Mail Compromised. (2013, 4 de agosto) https://thehackernews.com/2013/08/Firefox-Exploit-Tor-Network-child-pornography-Freedom-Hosting.html

Kan, Michael. Hacker takes out dark web hosting service using well-known exploit. (2017, 6 de febrero) https://www.csoonline.com/article/3166194/security/hacker-takes-out-dark-web-hosting-service-using-well-known-exploit.html

Cid, Daniel. From a Site Compromise to Full Root Access – Symlinks to Root – Part I (2013, 23 de mayo) https://blog.sucuri.net/2013/05/from-a-site-compromise-to-full-root-access-symlinks-to-root-part-i.html

Cox, Joseph y Franceschi-Bicchierai, Lorenzo. Newly Uncovered Tor Browser Exploit Targeted Dark Web Child Porn Site. (2016, 30 de noviembre) https://motherboard.vice.com/en_us/article/9a3mq7/tor-browser-zero-day-exploit-targeted-dark-web-child-porn-site-giftbox

Fobian, Andreas y Bender, Carl-Benedikt. Firefox 0-Day targeting Tor-Users (2016, 30 de noviembre) https://www.gdatasoftware.com/blog/2016/11/29346-firefox-0-day-targeting-tor-users

Nurmi, Juha. Tor de-anonymization techniques. (2017, 5 de agosto) https://media.ccc.de/v/SHA2017-102-tor_de-anonymization_techniques

Onieva, David. La configuración de los servidores web Apache podría revelar detalles del tráfico en Tor. (2016, 31 de enero) https://www.adslzone.net/2016/01/31/la-configuracion-predeterminada-de-los-servidores-web-de-apache-podria-revelar-detalles-del-trafico-en-tor/

Seoane Pedreira, Alejandro. BITCOINS PARA BLANQUEAR DINERO. (s.f.) http://seoanepedreira.es/bitcoins-para-blanquear-dinero/

Olvera Rodríguez, Patricia. Web profunda, Darknet, Tor. (2017, 10 de abril) http://crimina.es/crimipedia/topics/web-profunda-darnet-tor/

Pagnotta, Sabrina. Navegación anónima en Tor: ¿herramienta para cuidadosos o para cibercriminales? (2014, 2 de julio) https://www.welivesecurity.com/la-es/2014/07/02/navegacion-anonima-tor-herramienta-cuidadosos-o-cibercriminales/

Paganini, Pierluigi. Malware in Dark Web. (s.f.) https://resources.infosecinstitute.com/malware-dark-web/#gref

Comunidad Tecnobineros. Clases de Carding. (s.f.) http://tecnobineros.blogspot.com/p/gggg.html

Krebs, Brian. Carding Sites Turn to the 'Dark Cloud'. (2016, 12 de mayo) https://krebsonsecurity.com/2016/05/carding-sites-turn-to-the-dark-cloud/

Yúbal FM. Así de fácil me ha resultado entrar en la Darknet y encontrar cuentas robadas de casi todo. (2016, 7 de noviembre) https://www.genbeta.com/a-fondo/asi-de-facil-o-dificil-me-ha-resultado-entrar-en-la-darknet-y-encontrar-cuentas-robadas-de-casi-todo

S. Zavia, Matías. Una semana en la deep web. Esto es lo que me he encontrado. (2018, 19 de diciembre) https://www.xataka.com/analisis/una-semana-en-la-deep-web-esto-es-lo-que-me-he-encontrado

A Gomez, Boris. Privacidad y Seguridad en Internet: La Web Oscura (DeepWeb). (2017, 2 de febrero) http://hackeruna.com/2017/02/02/privacidad-y-seguridad-en-internet-la-web-oscura-deepweb/

García, Alberto. Así se envían las armas ilegales de la Dark Web desde EE.UU. a Europa. (2017, 20 de julio) https://www.adslzone.net/2017/07/20/asi-se-envian-las-armas-ilegales-de-la-dark-web-desde-ee-uu-europa/

E. Hall, Gabriel. CryptoWall is a ransomware family that encrypts important files on the affected computers. (2018, 1 de diciembre) https://www.2-spyware.com/remove-cryptowall-virus.html

Luft, Benedikt. El lucrativo alcance de los ataques 'ransomware' en Bitcoin. (2018, 7 de mayo) https://www.technologyreview.es/s/10173/el-lucrativo-alcance-de-los-ataques-ransomware-en-bitcoin

Mirada Profesional. ¿Qué es la Deep Web y cuál es su relación con la venta ilegal de medicamentos? (2015, 15 de junio) https://miradaprofesional.com/ampliarpagina.php?id=47661

Europa Press. Cae la mayor red de venta ilegal online de medicamentos de disfunción eréctil y tratamiento adelgazante. (2018, 15 de febrero) https://www.europapress.es/nacional/noticia-cae-mayor-red-venta-ilegal-online-medicamentos-disfuncion-erectil-tratamiento-adelgazante-20180215114728.html

Yúbal FM. Qué es la Dark Web, en qué se diferencia de la Deep Web y cómo puedes navegar por ella. (2018, 17 de abril) https://www.xataka.com/basics/que-es-la-dark-web-en-que-se-diferencia-de-la-deep-web-y-como-puedes-navegar-por-ella

Bit2Me. ¿Qué es la Cadena de Bloques (Blockchain)? (s.f.) https://academy.bit2me.com/que-es-cadena-de-bloques-blockchain/

Elaine. ¿Qué es doxing? (2017, 18 de julio) https://onretrieval.com/que-es-doxing/

Fukui, Takuya. De-anonymizing Tor traffic with website fingerprinting (2017, 22 de abril) https://witestlab.poly.edu/blog/de-anonymizing-tor-traffic-with-website-fingerprinting/

Alonso, Chema. RAPTOR - El anonimato en la red TOR (Deep Web) comprometido desde los Sistemas Autónomos (2015, 17 de marzo) http://www.elladodelmal.com/2015/03/raptor-el-anonimato-en-la-red-tor-deep.html

Castillo, Pedro. Vulnerabilidades en Tor ponen en peligro el anonimato en los servicios ocultos (2015, 27 de julio) https://securityinside.info/vulnerabilidades-en-tor-anonimato-servicios-ocultos/

ANEXO A. Glosario

Blockchain: cadena de bloques de Bitcoin que contiene un registro certero y verificable de todas las transacciones que se han hecho en su historia.

Bridge o puente: es un nodo intermedio Tor que no está listado en el directorio público de Tor y así es útil en países donde los nodos públicos están bloqueados. A diferencia del caso de los nodos salida, las direcciones IP de los nodos puentes nunca aparecen en los ficheros de log de los servidores y nunca pasan a través de los nodos de monitoreo de forma que puedan ser conectados con la evasión de censura.

Buffer overflows: un desbordamiento de búfer. Es un error de software que se produce cuando un programa no controla adecuadamente la cantidad de datos que se copian sobre un área de memoria reservada a tal efecto (buffer): Si dicha cantidad es superior a la capacidad preasignada, los bytes sobrantes se almacenan en zonas de memoria adyacentes, sobrescribiendo su contenido original, que probablemente pertenecían a datos o código almacenados en memoria. Esto constituye un fallo de programación, vulnerabilidad que puede ser aprovechada por un usuario malintencionado para influir en el funcionamiento del sistema.

Bug: error que se produce en un programa informático.

Carding: robo de datos de tarjetas de crédito para luego venderlos.

Clearnet o surface web: para referirse a la red de Internet normal, los servicios públicos de Internet.

Cookie: Una cookie es una cadena de texto que envía un servidor Web al navegador del usuario para almacenarla en el ordenador del usuario, y que contiene la información necesaria para mantener la continuidad de la sesiones a través de múltiples páginas web, o a través de sesiones múltiples. Algunos sitios Web no pueden usarse sin aceptar y almacenar una cookie. Algunas personas consideran esto una invasión de la privacidad o un riesgo de seguridad.

Cross site scripting (XSS): tipo de vulnerabilidad que puede permitir a una tercera persona inyectar en páginas web visitadas por el usuario código JavaScript o en otro lenguaje similar. Su finalidad es robar información delicada, secuestrar sesiones de usuario, y comprometer el navegador, subyugando la integridad del sistema.

Darknet: mientras que la Dark Web es todo ese contenido deliberadamente oculto en Internet, las darknets son esas redes específicas como Tor, Freenet, Zeronet o I2P que alojan esas páginas. En ocasiones se usa este término para referirse también a Dark Web o específicamente a la red Tor.

Dark Web: colección de redes y tecnologías usadas para compartir información y contenidos digitales (por ejemplo, textos, software, canciones, imágenes, películas) que está "distribuida" entre los distintos nodos y que trata de preservar el anonimato de las identidades de quienes intercambian dicha información, es decir, persiguen el anonimato del origen y del destino cuando se produce la transferencia de información. La Dark Web representa una pequeña parte de la Deep Web: ha sido ocultada intencionadamente, usa direcciones IP enmascaradas y es

inaccesible a través de los navegadores web estándar. Necesita un navegador web especial para acceder a ella.

Deep Web: también conocida como Internet profunda, Internet invisible o Internet oculta. Es el contenido de Internet que no está indexado por los motores de búsqueda convencionales, debido a diversos factores. Por una parte, pueden tratarse de páginas convencionales que han sido protegidas por un paywall (término incluido en este glosario), pero también archivos guardados en Dropbox o correos electrónicos guardados en los servidores del proveedor de correo. La Deep Web incluye la Dark Web.

Dirección MAC: identificador de 48 bits (representados por dígitos hexadecimales) que corresponde de forma única a una tarjeta o dispositivo de red. La dirección está formada por 12 dígitos agrupados en seis parejas separadas generalmente por dos puntos, aunque también puede haber un guión o nada en absoluto. Un ejemplo de dirección MAC podría ser 00:1e:c2:9e:28:6b.

Doxing: conseguir información pública y privada acerca de una persona, empresa, etc., a través de Internet, con intenciones maliciosas.

Escrow: un contrato de depósito en garantía en el que el dinero queda en reserva a través de un tercero. Este se encargará de liberar el pago cuando se haya recibido el servicio o el producto comprado.

Exploits 0-Day: es un ataque contra una aplicación o sistema que tiene como objetivo la ejecución de código malicioso gracias al conocimiento de vulnerabilidades que, por lo general, son desconocidas para la gente y el fabricante del producto. Esto supone que aún no hayan sido arregladas.

Fingerprint: huella digital. Se refiere a cualquier rastro de información dejada por alguien. A menudo, si alguien ha obtenido acceso no autorizado a un ordenador o red, un administrador o agente de seguridad puede buscar cualquier "huella digital" dejada por el atacante. Esta evidencia identificativa puede incluir direcciones IP, nombres de host, etc.

Forum: En un sitio web, un fórum es un lugar de discusión, donde los usuarios pueden escribir mensajes y comentar mensajes enviados anteriormente. Se distingue por la persistencia de las páginas que contienen los encabezados. Los archivos de grupos de noticias y listas de correo, a diferencia, típicamente muestran los mensajes uno por página, con páginas de navegación que listan solo los encabezados de los mensajes en un tema.

Hacktivismo: utilización no-violenta de herramientas digitales ilegales o legalmente ambiguas persiguiendo fines políticos. Estas herramientas incluyen desfiguraciones de webs, redirecciones, ataques de denegación de servicio, robo de información, parodias de sitios web, sustituciones virtuales, sabotajes virtuales y desarrollo de software.

Hidden Service: servicios ocultos en Tor o servicios onion. Son lo equivalente a las páginas webs en Tor. Los servicios que se ofrecen en Tor comprenden desde empresas que venden drogas, armas, pornografía y otros delitos, hasta chats, foros, blogs y servicios de correo electrónico.

Hosting: alojamiento de páginas web o ficheros.

HSDir (Hidden Service Directories) o Directorios de servicios onion: son nodos Tor que almacenan los descriptores o extractos firmados por los servidores de servicios onions para que los usuarios puedan contactar con ellos.

Jitter: es la variabilidad del tiempo de ejecución de los paquetes de datos que se envían en las comunicaciones en Internet. Este efecto es especialmente molesto en aplicaciones multimedia en Internet como radio por Internet o telefonía IP, ya que provoca que algunos paquetes lleguen demasiado pronto o tarde para poder entregarlos a tiempo.

Keylogger: tipo de *software* o un dispositivo *hardware* específico que se encarga de registrar las pulsaciones que se realizan en el teclado, para posteriormente memorizarlas en un fichero o enviarlas a través de Internet. Suele usarse como *malware* del tipo daemon, permitiendo que otros usuarios tengan acceso a contraseñas importantes, como los números de una tarjeta de crédito, u otro tipo de información privada que se quiera obtener.

Latencia: La latencia es una medida de la demora de tiempo experimentada en un sistema, en una red de ordenadores. Está medida por el tiempo entre el inicio de la transmisión del paquete al inicio de la recepción del paquete, entre un extremo de la conexión y el otro.

Malware: término general para programas maliciosos, incluyendo los virus, que pueden ser instalados o ejecutados sin el conocimiento del usuario. El malware puede tomar el control del host entre otras cosas para enviar spam.

Paywall: un sistema que restringe el acceso a sitios web a usuarios que no cuentan con una subscripción pagada.

Phising: técnica de ingeniería social utilizada por los delincuentes para obtener información confidencial como nombres de usuario, contraseñas y detalles de tarjetas de crédito haciéndose pasar por una comunicación confiable y legítima.

SSL: SSL (o Secure Sockects Layer), es uno de varios estandares de la criptografía usado para hacer seguras las transacciones de Internet. Se usó como base para la creación de Transport Layer Security (TLS). Se puede ver fácilmente si se está usando SSL/TSL buscando en la URL del navegador: Si comienza con https en lugar de http, la conexión está cifrada.

Wallet de Bitcoin: un wallet o billetero es el lugar donde se almacenan los bitcoins. Es el equivalente a la cuenta bancaria, pues consiste en un número desde donde se pueden realizar transferencias o recibirlas.

Warez: software ilegal copiado.

Wireshark es una herramienta multiplataforma utilizada para realizar análisis sobre paquetes de red, de modo que brinda información en cuanto a las conexiones que se llevan a cabo en dicha red.

ANEXO B. Índice de figuras