Trabajo Fin de Grado en Ingeniería Informática

Modelización del protocolo Tor y extracción de características de servicios ocultos

Jorge García de Quirós Giménez

Director: Ricardo J. Rodríguez Fernández

Ponente: José Merseguer Hernáiz

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Junio de 2018
Curso 2017/2018

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Jorge García de Quirós Giménez,

con nº de DNI 25209644B en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado, (Título del Trabajo) Modelización del protocolo Tor y extracción de características de servicios ocultos,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, a 28 de Junio de 2018

Fdo: Jorge García de Quirós Giménez

Agradecimientos

A José por guiarme.
A Francisco Monserrat, de Rediris, por apoyar este proyecto.
Por último a Ricardo, por brindarme esta oportunidad.

RESUMEN

Tor (The Onion Router) garantiza el anonimato y la privacidad de sus usuarios durante la navegación por Internet, mediante el establecimiento de circuitos virtuales entre los diferentes nodos integrantes de la red Tor. Además, Tor permite que los usuarios publiquen páginas web que son únicamente alcanzables a través de la propia red Tor, quedando entonces no accesibles desde la red de Internet convencional. Estos sitios se conocen como *servicios ocultos* de la red Tor, o sitios de la *dark web*. Su peculiaridad radica en que no funcionan a través de un sistema de resolución de nombres clásico como los servicios de Internet (a través de DNS), por lo que la dirección IP del servidor que los aloja es desconocida.

El trabajo realizado en este proyecto se dividen en dos partes. En primer lugar, se ha realizado la modelización y formalización del funcionamiento de la red Tor, así como los protocolos que utiliza (*o protocolo Tor*, por simplificar). Mediante diagramas de secuencia y actividad UML, se describen qué participantes existen en la red Tor, cómo es la comunicación y cómo es el comportamiento de cada uno de los participantes. Estos modelos servirán para entender desde un punto de vista más abstracto el funcionamiento del protocolo y además ayudarán a una futura verificación formal del protocolo de la red Tor. La segunda parte trata sobre la extracción de características de los servicios ocultos. Para ello se ha desarrollado una herramienta para la obtención de direcciones de servicios ocultos y la extracción de información de los mismos a través de estas direcciones. Esta herramienta pretende ser de utilidad para las Fuerzas y Cuerpos de Seguridad del Estado.

La información recopilada de estos servicios ocultos se ha estudiado con el objetivo de encontrar características que lleven a la desanonimización del servicio oculto (es decir, que ayuden a localizar la dirección IP). Para ello, se han utilizado concretamente los metadatos suministrados por las cabeceras HTTP y los certificados digitales obtenidos a través del establecimiento de una conexión HTTPS. Estos metadatos se han contrastado con un motor de búsqueda de servicios en Internet, que contiene información de cualquier dirección IPv4 alcanzable en Internet.

Los resultados obtenidos muestran que en un número importante de casos se han acotado los dispositivos que pueden estar relacionados con cada servicio oculto, llegando incluso a su desanonimización directa. Sobre la categorización de servicios ocultos, se han encontrado algunos relacionados con actividades ilegítimas.

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

La red Tor [10] (*The Onion Router*) es una red formada por nodos que voluntariamente se ponen a disposición de los usuarios con el objetivo de mejorar su anonimato y la privacidad durante la navegación por Internet. De hecho, el proyecto Tor se mantiene por una organización sin ánimo de lucro que se encarga de gestionar el desarrollo de la aplicación y de los protocolos de la red Tor (o *protocolo Tor*, por simplificar), así como de controlar el estado de la red.

Originalmente, la red Tor fue diseñada por el U.S. Naval Research Laboratory, del U.S. Navy, con la idea de proteger las comunicaciones gubernamentales. Hoy en día, se usa por una amplia comunidad de individuos y con diferentes propósitos, como por ejemplo periodistas para contactar con sus fuentes de manera anónima, activistas para denunciar abusos en zonas de conflicto, miembros y fuerzas de los cuerpos de seguridad de los estados para la llevar a cabo operaciones encubiertas o de vigilancia, e incluso propósitos militares (por ejemplo, proteger a militares o activos desplegados en zonas donde insurgentes pueden monitorizar el tráfico de Internet y descubrir localizaciones u otras operaciones de interés que supongan un riesgo tanto a la propia operativa como a los activos en la zona).

La red Tor proporciona un servicio de comunicación anónima de baja latencia basado en circuitos. En concreto, se forma un circuito virtual entre un nodo origen y un nodo destino garantizando así que las organizaciones y usuarios puedan compartir información sin comprometer su privacidad. Este circuito virtual se compone de numerosos nodos dentro de la red Tor que retransmiten el tráfico entre ellos, de modo que finalmente se consiguen comunicar el nodo origen y el nodo destino, pero a través de múltiples "saltos" intermedios (cada uno de estos saltos es un nodo de la red). Estos circuitos virtuales garantizan que no hay una conexión directa entre cliente y servidor, sino a través de los diferentes nodos de la red donde cada uno conoce únicamente a su anterior y siguiente nodo en la cadena.

El objetivo de estos circuitos virtuales es ocultar mediante las retransmisiones intermedias la dirección IP de los partícipes en la comunicación. Recuérdese que una dirección IP está compuesta por cuatro octetos y se le asigna a cada dispositivo cuando se conecta a Internet, permitiendo así identificarlo de manera inequívoca. Para mantener la privacidad

en la navegación, todo el tráfico que discurre por un circuito Tor se encuentra cifrado. Además, cada par de nodos que se comunican dentro del circuito se mandan también entre ellos el tráfico cifrado punto a punto. Es decir, el tráfico de red se va cifrando por capas – es por ello que el tipo de encaminamiento que realiza la red Tor recibe el nombre de *encaminamiento de cebolla*.

Dentro de la red Tor existen *servicios ocultos*, que son páginas web únicamente accesibles a través de la red Tor, y no accesibles por los buscadores de la red de Internet convencional. Estos sitios se conocen comúnmente como sitios de la *dark web* [2, 6] y no utilizan un sistema basado en nombres de dominio convencional, por lo que no existe una dirección IP asociada a la dirección de acceso. En concreto tienen asociada una dirección con terminación de dominio `.onion`, que permite conectarse al servicio gracias un sistema de tablas distribuidas (DHT) utilizado por Tor. De esta manera, junto al protocolo de conexión a los servicios ocultos, se garantiza que la IP del servidor permanece oculta al usuario que se conecta.

Cabe destacar también que la motivación principal del uso de la red Tor radica en aras de la defensa de los derechos y libertades del individuo, como por el ejemplo el acceso a servicios de contenido prohibidos por los países de origen del usuario (por ejemplo, imagínese un usuario intentando acceder desde China a información sobre la masacre ocurrida en la Plaza de Tiananmen en 1989), acceso a foros o salas de chat para víctimas de abusos y/o violaciones, o incluso la denuncia y el filtrado anónimo de información sensible de empresas o estados.

Sin embargo, aunque el uso de la red Tor y los servicios ocultos tienen un objetivo y usos legítimos, desgraciadamente la red Tor se ha explotado con objetivos ilegítimos. El caso más conocido es el famoso servicio oculto SilkRoad, un mercado negro online que operaba en la red Tor y que fue mayormente conocido por la venta de drogas ilegales a través de Internet [3,8]. SilkRoad fue creado en 2011 y clausurado dos años más tarde por el FBI, siendo arrestada una persona bajo la acusación de ser su fundador – esta persona actualmente cumple condena de cadena perpetua sin posibilidad de libertad condicional en Estados Unidos.

Respecto a estadísticas de uso, según [18] en este último mes se han contabilizado unos 8,000 ordenadores formando parte de la red Tor, siendo el número de usuarios estimado cercano a dos millones y medio.

1.1. Objetivo

El objetivo de este trabajo es analizar el funcionamiento de la red Tor para comprender su funcionamiento y encontrar, si existe, alguna deficiencia en el diseño del protocolo. Estas deficiencias podrían ser aprovechadas para limitar las capacidades de privacidad y anonimato que proporciona Tor. Para ello, en este trabajo se hará uso de un modelado mediante técnicas UML que posteriormente pueden usarse para la verificación del modelo. Este tipo de aproximación se ha usado con éxito en otros campos, como en [4], donde se modelizó formalmente el protocolo usado por las tarjetas de crédito encontrando diversas vulnerabilidades, o en [11], donde se modelizó el protocolo industrial MODBUS y se

verificó formalmente la existencia de ataques *man-in-the-middle*.

De manera adicional, también es un objetivo de este trabajo, el desarrollo de una herramienta que recopile direcciones `.onion` y que extraiga información relevante sobre ellos. Posteriormente, esta información se empleará para intentar desanonimizar y categorizar los servicios ocultos. La aproximación llevada a cabo, para la desanonimización, utilizará los metadatos obtenidos en cabeceras HTTP y de certificados digitales, consultándolos en un servicio que puedan llevar a una identificación de la dirección IP tras el servicio oculto. Para la categorización se utilizarán diccionarios de palabras relacionadas con actividades ilegítimas.

Implicaciones éticas

Se ha accedido a los servicios ocultos de manera automática y sólo a su página principal, descartando cualquier contenido gráfico y procesando sólo texto. Se ha realizado de esta manera dado que no se conoce el contenido al que se accede en la *dark web* y así se evita la descarga de contenido ilegal y no deseado. Se espera que este modelo de herramienta pueda ayudar a las Fuerzas y Cuerpos de Seguridad del Estado a encontrar servicios ocultos con fines ilegítimos, evitando que actúen bajo el amparo del anonimato proporcionado por la red Tor.

1.2. Motivación

Este trabajo nace ante la necesidad de conocer en profundidad el funcionamiento de Tor. En la literatura, la red Tor se estudia siempre analizando y midiendo las fortalezas inherentes de la red Tor (privacidad y anonimato). Un buen número de artículos relacionados con estos extremos se encuentran accesibles desde la propia página web del proyecto Tor [7, 16]. Sin embargo, ningún trabajo se centra en la modelización y formalización de la red Tor, como se hace en primer lugar en este trabajo.

Siguiendo la motivación de Tor, se pretende ayudar a los usuarios que actúan de manera legítima a seguir bajo el amparo de Tor. Del mismo modo, se busca detectar a aquellos usuarios que realizan un uso ilegítimo, para que puedan ser detectados y expulsados de la red, e incluso perseguidos por sus actividades.

1.3. Organización

Este documento se encuentra dividido en siete capítulos y un anexo. El Capítulo 2 introduce algunos conceptos útiles para facilitar la comprensión del resto del documento: UML y metadatos en servidores web. En el Capítulo 3 se describen los diferentes participantes de la red Tor, los mensajes y el cifrado que existen en la red Tor, cómo sucede el intercambio de mensajes en diferentes escenarios y muestra diagramas de actividad UML para describir el comportamiento de los diferentes participantes. A continuación, en el Capítulo 4 se describe el sistema implementado para la extracción de características

de los servicios ocultos. El Capítulo 5 analiza la información obtenida por el sistema. El Capítulo 6 recoge una serie de aproximaciones similares a la realizada en este trabajo y herramientas útiles en este ámbito. El Capítulo 7 concluye el trabajo y presenta el trabajo futuro.

Al final del documento se encuentra el Anexo A, que muestra detalladamente el tiempo invertido en este trabajo.

Capítulo 2

Conocimientos previos

A continuación se explicará UML, así como los diferentes tipos de diagramas UML utlizados en este trabajo. Además, se introducirán los metadatos que se han obtenido de los servicios ocultos.

2.1. UML

El lenguaje unificado de modelado (UML, en inglés *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos, etc. Es sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos.

UML usa elementos y los asocia de diferentes formas para realizar diagramas que representan aspectos estáticos o estructurales de un sistema, diagramas de comportamiento, que captan los aspectos dinámicos de un sistema, y diagramas de interacción, que muestran como colaboran los distintos objetos del sistema de manera específica. En este trabajo se han usado:

- **Diagrama de clases**, que es un tipo de diagrama de estructura estática que describe las clases del sistema;
- **Diagrama de actividades**, que son un diagrama de comportamiento que representan los flujos de trabajo de forma gráfica; y
- **Diagrama de secuencia**, que en UML muestran cómo los objetos interactúan entre sí y el orden en que se producen estas interacciones para un escenario en concreto.

2.2. Metadatos en servidores web

Los metadatos son datos que describen otros datos. Por ejemplo un fichero de texto puede contener información sobre su autor, fecha de creación, etc. Estos metadatos pueden ser un riesgo potencial para el usuario si no son gestionados adecuadamente, como ha sucedido con los querellados por la Fiscalía del Estado por la Declaración Unilateral de Independencia en Cataluña [5].

Como ejemplo de metadatos, los servidores que utilizan el protocolo de páginas web HTTP o HTTPS (HTTP en versión "segura"), envían unas cabeceras durante la comunicación para proporcionar información adicional sobre la transacción en curso. Estas cabeceras utilizan la sintaxis textual `Cabecera: Valor`. Por ejemplo, la cabecera HTTP que se obtiene al conectar a la página web de `http://www.unizar.es` es:

```
HTTP/1.1 200 OK
Date: Mon, 25 Jun 2018 09:23:04 GMT
Server: Apache
X-Powered-By: PHP/5.4.16
X-Drupal-Cache: HIT
Etag: "1529916407-0"
Content-Language: es
X-Frame-Options: SAMEORIGIN
X-Generator: Drupal 7 (http://drupal.org)
Cache-Control: public, max-age=600
Last-Modified: Mon, 25 Jun 2018 08:46:47 GMT
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Vary: Cookie,Accept-Encoding
Content-Type: text/html; charset=utf-8
```

Como se puede observar, se obtiene información del servidor web que está ejecutando (Server: Apache), del lenguaje del contenido (Content-Language: es) y del tipo de codificación usado (charset=utf-8), por mostrar algunos ejemplos.

Si el servidor utiliza el protocolo HTTPS, esté dispone de un certificado digital. Un certificado digital es un documento que garantiza que la conexión establecida es segura, y que la comunicación va a ser cifrada punto a punto. El certificado digital contiene información sobre la clave, la identidad del propietario de la clave y la firma de la entidad que expide del certificado. Por ejemplo parte de la información del certificado del servidor de la página web de `http://www.unizar.es` es:

```
Server certificate:
subject: C=ES; ST=Zaragoza; L=Zaragoza; O=Universidad de Zaragoza;
OU=SICUZ; CN=*.unizar.es
start date: Jan 19 00:00:00 2016 GMT
expire date: Jan 23 12:00:00 2019 GMT
subjectAltName: host "www.unizar.es" matched cert's "*.unizar.es"
issuer: C=NL; ST=Noord-Holland; L=Amsterdam; O=TERENA; CN=TERENA SSL CA 3
```

En este caso se detalla la fecha de inicio (start date: Jan 19 00:00:00 2016 GMT) y fin del certificado (expire date: Jan 23 12:00:00 2019 GMT), la entidad encargada de expedirlo (O=TERENA) y el sujeto del certificado (O=Universidad de Zaragoza), aparte de otra información que no aparece en este ejemplo, como la versión, el algoritmo de clave pública utilizado, número de serie, entre otros.

Para cotejar este tipo de metadatos existen motores de búsqueda que permite encontrar equipos conectados a Internet a través de una variedad de filtros. Estos motores recopilan los *banners* de servicios, que son metadatos que el servidor envía de vuelta al cliente (es decir, software de servidor, qué opciones admite el servicio, la localización, etc.)

Uno de estos motores de búsqueda más conocido es Shodan [1]. El motor de Shodan monitoriza y escanea Internet en su rango de direcciones IPv4, almacenando la información de los servicios accesibles. Al buscar las cabeceras HTTP, se cotejan con la base de datos de *banners* de Shodan, permitiendo filtrar en la propia búsqueda mediante filtros preestablecidos, con la notación `filtro:valor` (entre los filtros se encuentra el puerto del servicio, el software del servicio, la localización, etc.).

Capítulo 3

Modelización de Tor

En este capítulo se describe en detalle la estructura de la red Tor, definiendo los tipos de nodos que la conforman, los roles específicos de cada nodo, así como las comunicaciones que sucedan en la red Tor.

3.1. Documentación de consulta y suposiciones

La mayor parte de documentación consultada para realizar este capítulo se ha extraído de la página web del proyecto Tor [17], en donde se describe ampliamente y de manera granular el funcionamiento del protocolo.

Nótese que a pesar de que esta información se encuentra públicamente disponible, es complicada de analizar dado que se divide en un total de 20 ficheros de texto (sin formato alguno, ni imágenes gráficas) sumando unas 18000 líneas aproximadamente. Además en la misma página web conviven diferentes versiones, lo que ha permitido encontrar ambigüedades y partes poco detalladas durante su estudio. Para poder llevar a cabo su modelización se han realizado algunas suposiciones:

1. Se ha supuesto que Tor ejecuta de manera paralela las diferentes tareas que realizan los nodos, dado que es la implementación lógica que debería tener una aplicación de estas características.

2. Se ha asumido que algunos nodos realizan actividades no especificadas en la documentación, dado que son necesarias para poder llevar a cabo el resto de acciones que realizan.

3.2. Elementos de la red Tor

La red de Tor funciona mediante el establecimiento de circuitos entre los diferentes elementos (llamados *nodos*) que componen la red. En Tor, cualquier ordenador conectado a la red es un nodo, tanto los que colaboran activamente construyendo la red como los usuarios que se benefician de ella.

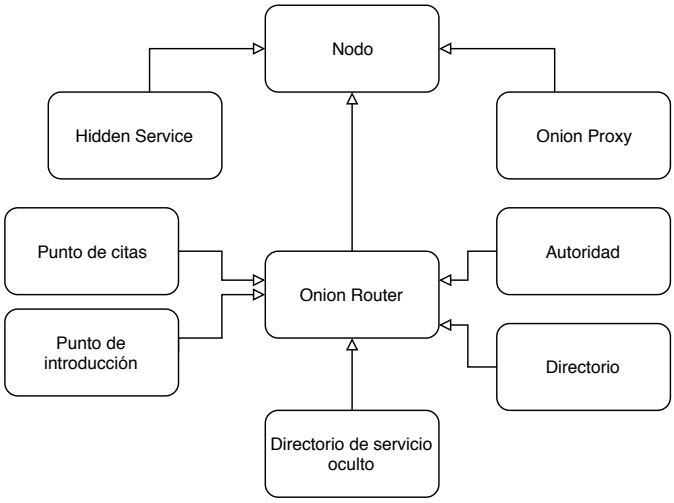

Figura 3.1: Participantes en la red Tor (tipos de nodos).

Cada uno de los nodos que se encuentran en la red Tor tienen un rol diferente, con lo que se pueden distinguir diferentes tipos de nodos. Una clasificación de los diferentes nodos Tor, usando una representación de diagrama de clases UML, se muestra en la Figura 3.1. Se distinguen principalmente tres tipos de nodos:

Onion Proxy. Los *nodos Onion Proxy* (OP) representan a un cliente que está ejecutando Tor en su máquina. Así, este cliente usará Tor para lograr establecer conexión a otra máquina a través de la red Tor, preservando así su privacidad y disfrutando de la seguridad garantizada por la propia red Tor.

Onion Router. Los *nodos Onion Router* (OR) son nodos que mantienen accesible un puerto determinado, el 9050 por defecto, y representan la pieza básica del funcionamiento de la red Tor. Estos nodos son mantenidos por voluntarios y son los encargados de, mediante conexiones entre ellos, establecer los circuitos por los que posteriormente fluirá la información de datos.

Hidden Service. Los *nodos Hidden Service* (HS) son nodos de la red Tor que proveen algún tipo de servicio oculto. Estos servicios usan siempre la dirección de dominio `.onion`, aprobada oficialmente por la IETF/IANA (agencia encargada de gestionar los nombres de dominio en Internet) en 2015.

Los elementos clave de la red Tor, los nodos OR, pueden tener a su vez otros roles diferenciados dentro de la red:

Directorio. Los *nodos directorio* (Dir) son nodos OR con un puerto concreto accesible que se encargan de obtener la información del estado actual de la red. Esta información la consiguen a través de las autoridades o a través de otros directorios

que ya hayan obtenido esta información para suministrarla a otros nodos que la soliciten. Algunos actúan como directorio de respaldo (*fallback directory mirror*), estos están distribuidos en el propio software de Tor, para disponer de un directorio al que realizar la primera conexión sin tener información previa de la red.

Directorio autoridad. Los *nodos autoridad* (Auth) son nodos directorio con permiso de autoridad. Este rol es minoritario, y forman parte de este grupo de nodos aquellos miembros de la red Tor que llevan un largo período de tiempo ininterrumpido de conexión en la red y que además gozan de buenas características (es decir, ancho de banda, largo tiempo de actividad ininterrumpida, etc.). El número de nodos autoridad es limitado (únicamente 10) y son elegidos por la organización Tor. Estos nodos se encargan de elaborar el estado actual de la red a través de la información que recopila de la misma.

Punto de introducción. Los *nodos punto de introducción* (IP) son nodos OR con una tarea específica durante la creación de un circuito para la conexión a un HS. Se explican en más detalle en la Sección 3.4.2.

Punto de citas. Los *nodos punto de citas* (RP) son, como los nodos IP, unos nodos OR que tienen asignada una tarea específica durante la creación de un circuito para la conexión a un HS. Se explican también en más detalle en la Sección 3.4.2.

Bridge. Un *nodo bridge* es un nodo OR que no está listado en el estado de red. Estos nodos no se encuentran en los listados para garantizar que no son censurados por parte de entidades o agentes estatales que tienen intereses en bloquear el acceso a la red Tor. En países donde existe esta censura, los nodos bridge son con los que se establece la primera conexión para entrar a la red Tor.

Bridge autoridad. Un *nodo bridge autoridad* es similar a un nodo Auth, pero sólo aplicado en nodos Bridge.

Directorio de servicio oculto. Los *nodos directorio de servicio oculto* (HsDir) son nodos OR con un funcionamiento específico. En concreto, se encargan de suministrar la información a los clientes que quieran establecer una conexión con un HS.

Atendiendo a la posición de un OR cuando se establece un circuito de comunicación, se pueden distinguir también tres tipos de nodos: *guard node*, que es el primer nodo de entrada de un OP (es decir, es el primer nodo dentro del circuito virtual); *middle node*, que es un nodo OR cualquiera dentro del circuito que no es ni el primero ni el último; y *exit node*, que es el último nodo del circuito donde la información deja la red Tor.

3.3. Mensajes y cifrado

Como se ha descrito anteriormente, toda la comunicación que sucede entre los diferentes nodos dentro de la red Tor se realiza de forma cifrada a través de una conexión basada

en el establecimiento de circuitos. Respecto a los tipos de mensajes que se intercambian entre los nodos, existen dos tipos:

Control Cell. El mensaje *control cell* es un mensaje formado por una cabecera y una carga útil (*payload*). La cabecera es una estructura que contiene información sobre el identificador del circuito (considerándose como circuito la conexión establecida uno a uno; no confundirlo con el circuito Tor) y un comando que indica qué se ha de hacer con la carga útil. Estos mensajes son interpretados siempre directamente por el receptor de los mismos. Pertenecen a este grupo de mensajes los mensajes `CREATE`, `CREATED`, `DESTROY` y `PADDING`.

Relay Cell. El mensaje *relay cell* es un mensaje formado por una cabecera, una carga útil y una cabecera adicional (antes de la carga útil) que se usa para identificar a qué flujo de datos pertenece. Nótese que el flujo de datos viajará a través de un circuito Tor, pero estos conceptos no son equivalentes dado que a través de un circuito pueden viajar varios flujos de datos diferentes. Para cada uno de los mensajes de este grupo se realiza una acción específica cuando se recibe. Algunos ejemplos de los mensajes de este grupo son `DATA`, `BEGIN`, `END`, `CONNECTED`, `EXTEND`, o `EXTENDED`, entre otros.

En el resto de este documento, se considera que el intercambio de mensajes mostrado en los diagramas se realiza correctamente y que los mensajes son válidos, es decir, que los mensajes están bien formados y están firmados por la clave correcta.

Cifrado

Como se ha comentado anteriormente, todas las conexiones de la red Tor van cifradas. La conexión entre dos nodos utiliza TLS/SSLv3. Hay tres formas de realizar *handshakes* TLS con un servidor Tor:

1. ***Certificates-up-front***. El emisor y el receptor mandan una cadena de dos certificados como parte de su *handshake* inicial.

2. ***Renegotiaton***. El receptor proporciona un certificado individual y el emisor de inmediato realiza una renegociación TLS.

3. ***In-protocol***. Una vez se completa el *handshake* inicial, se comienza una autenticación mediante el protocolo Tor, sin utilizar más *handskahes* TLS.

Para el intercambio de claves se utiliza Diffie-Hellman con un generador (g) igual a dos, como modulo (p) un número primo, públicamente conocido de 1024-bit y claves privadas (x,y) de 320 bits. En el diagrama de secuencia de la Figura 3.2 definiendo los mensajes cifrados mediante la notación Enc($clave, textoplano$) y asumiendo que g^e implica $g^e \mod p$, se aprecia como establecer una clave simétrica g^{xy} mediante el intercambio de claves Diffie-Hellman. Se utiliza SHA-1, SHA-256 o SHA3-256 como función *hash*, dependiendo del protocolo de intercambio de claves utilizado (que puede

ser protocolo TAP o ntor).

Los nodos OR hacen uso de 7 claves diferentes. En concreto, esas claves son:

1. **Cifrado RSA de 1024-bits**:

   a) Una `Clave identidad` a largo plazo y sólo para firmado de documentos y certificados. Se usa para establecer la identidad del nodo.

   b) Una `Clave cebolla` para el protocolo TAP a medio plazo, se utiliza en el desencriptado de la capas de cuando se aceptan intentos de extensión del circuito.

   c) Una `Clave de conexión` a corto plazo para negociar las conexiones TLS.

2. **Cifrado de clave elíptica Curve25519**:

   a) Una `Clave cebolla` para el protocolo ntor a medio plazo. Misma finalidad que la clave generada con algoritmo RSA previamente explicada.

3. **Cifrado de clave pública Ed25519**:

   a) Una `Identidad maestra` a largo plazo, nunca cambia, solo se utiliza para firmar el resto de claves de este grupo. Debe guardarse *offline*.

   b) Una `Clave de firmado` a medio plazo, firmada por la *identidad maestra*, debe guardarse *offline* y es la que firma prácticamente todo.

   c) Una `Autenticación de enlace`, para autenticar el establecimiento de conexiones entre nodos, está firmada por la *Clave de firmado*.

La Clave de identidad y la Identidad maestra, juntas, identifican inequívocamente un nodo OR.

3.4. Conexiones

Como se ha comentado anteriormente, la conexión que se establece entre los nodos es una conexión basada en circuitos. Por tanto, la primera acción que se realiza cuando un cliente se conecta a la red Tor es la creación y extensión de un circuito Tor. La secuencia de acciones y mensajes que suceden entre el cliente OP y el número de nodos $\mathrm{OR}_i, i \geq 3$, que conforman el circuito se muestra en el diagrama de secuencia de UML de la Figura 3.2.

Como se observa en la Figura 3.2, el nodo OP manda un primer mensaje `CREATE` al nodo OR_1 para crear el circuito punto a punto identificado por la célula de control C_1. El mensaje que le manda el nodo OP está cifrado con la clave pública del nodo OR_1, que se expresa mediante la notación $\mathrm{Enc}(K_{OR_1}, g_1^x)$. La clave K_{OR_1} es la clave pública del nodo OR_1, que es un dato del nodo OR_1 conocido en el estado consensuado de la red. El

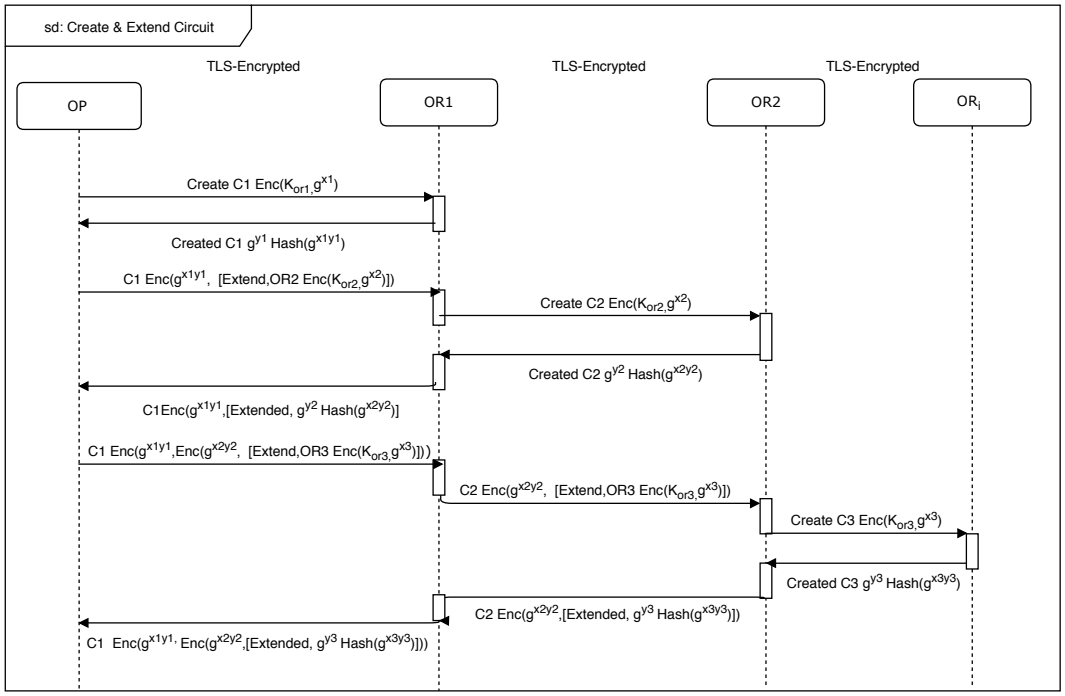

Figura 3.2: Diagrama de secuencia UML *Create & Extend Circuit*.

nodo OR_1 responde con un mensaje `CREATED` indicando que el circuito se ha establecido correctamente.

Después, el nodo OP envía un mensaje `EXTEND` a través de ese circuito, indicándole al nodo OR_1 que tiene que se quiere extender el circuito creado hasta el nodo OR_2. Cuando el nodo OR_1 decodifica el mensaje enviado por el nodo OP y verifica que se trata de un mensaje `EXTEND`, enviará un mensaje `CREATE` al nodo OR_2 para crear un circuito C_2 entre ellos. Tras ello, el nodo OR_2 responderá con un mensaje de `CREATED` al nodo OR_1, quien responderá con un mensaje `EXTENDED` al nodo OP.

Por último, el nodo OP volverá a mandar un mensaje `EXTEND` pero en este caso dirigido al nodo OR_2, para que este nodo extienda el circuito a otro nodo OR_3. Lo más importante en este paso es destacar cómo es el mensaje que se envía a través del circuito C_1. Como se muestra en la Figura 3.2, la carga útil del mensaje que se envía al nodo OR_1 está cifrada con la clave pública del nodo OR_2, que es a su vez el mensaje que se retransmite a través del circuito C_2 para que llegue al nodo OR_2. Este nodo descifra el mensaje y extiende el circuito al nodo OR_3, devolviendo este último un mensaje de `CREATED` al nodo OR_2 tras completar la creación. De nuevo, se envía un mensaje de `EXTENDED` como respuesta a la petición de extensión de circuito a través del circuito C_2. Fíjese que este mensaje va de nuevo encapsulado y cifrado hacia atrás, hasta llegar al nodo OP.

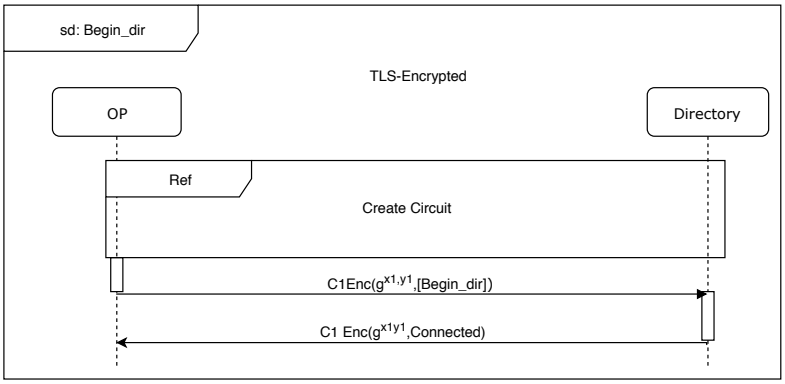

Figura 3.3: Diagrama de secuencia UML *Begin_ dir*.

3.4.1. Comunicaciones internas y externas

Respecto a las comunicaciones, se puede distinguir entre las comunicaciones internas (es decir, que ocurren dentro de la red Tor) y las comunicaciones externas.

Las comunicaciones internas son aquellas que requieren datos de un nodo interno de la red Tor, como son las conexiones a un nodo Auth o a un nodo Dir para solicitar información. Las comunicaciones para solicitar y publicar descriptores de servicios ocultos no se consideran comunicaciones internas, dado que se realizan de manera anónima y similar a las conexiones a un servidor fuera de la red (conexiones externas, explicadas más adelante).

A modo de ejemplo, se explica la comunicación que puede realizar un nodo OP con un nodo Dir. El diagrama de secuencia UML referente a esta acción se muestra en la Figura 3.3. Tras establecer el circuito, se envía un mensaje `BEGIN_DIR` al directorio para establecer una conexión con él, quien responderá con un comando `CONNECTED`. Después, se enviarían más mensajes para la obtención de información del directorio a través de comandos `DATA`.

Las comunicaciones externas son aquellas que se producen hacia fuera de la red Tor. Para conectar con un servidor fuera de Tor, el nodo salida del circuito establecido es el que se encargará de realizar la conexión al servidor que interese y devolver la respuesta, enrutándola hacia atrás hasta llegar al cliente. En la Figura 3.4 se muestra la secuencia de acciones necesaria. Primero será necesario crear un circuito Tor, de manera similar a la que se ha explicado anteriormente (véase la Figura 3.2. El circuito que se establece tiene por defecto 3 saltos. Después, el nodo OP mandará el mensaje `BEGIN` junto con la dirección IP a la que se quiere realizar la conexión. Este mensaje será procesado por el nodo de salida, que conectará con el servidor solicitado y retransmitirá de vuelta la confirmación de conexión mediante el mensaje `CONNECTED`.

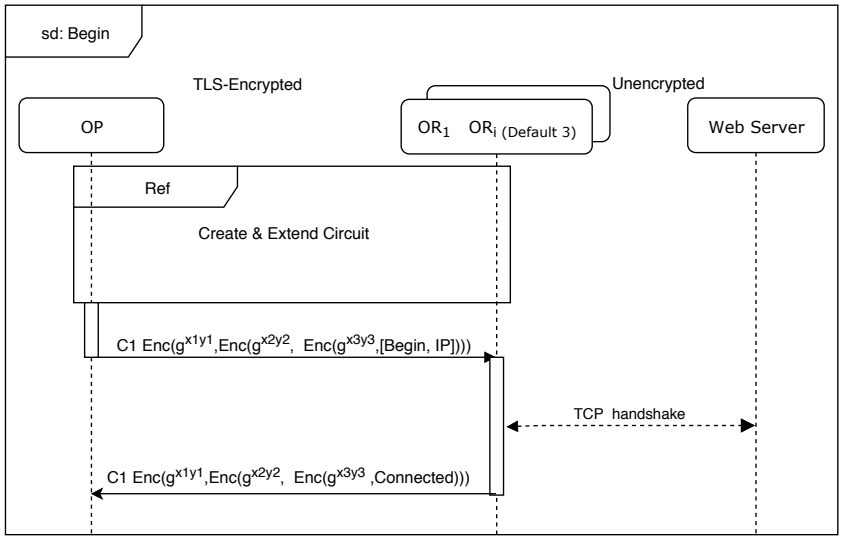

Figura 3.4: Diagrama de secuencia UML *Begin*.

3.4.2. Conexión a servicios ocultos

Respecto a la conexión a servicios ocultos, la secuencia de acciones que suceden es un poco más compleja.

En primer lugar, el servicio oculto ha de anunciar su servicio. El servicio oculto, servido por un nodo HS, crea un circuito a través de una serie de nodos OR (al menos dos) y acuerda con otros nodos OR su rol como nodos *introduction point* (IP) o nodos punto de introducción. El servicio oculto notificará a un nodo HsDir junto con su dirección `.onion` o una cadena derivada de esta, dependiendo de la versión, la identidad de estos nodos IP que servirán como nodos de entrada al propio servicio oculto. Esta secuencia de acciones se muestra en la Figura 3.5.

El diagrama de secuencia relativo a la conexión a un servicio oculto se muestra en la Figura 3.6. Para conectar a un servicio oculto, el nodo OP en primer lugar, y de la forma habitual, crea un circuito con una serie de nodos OR de la red, extendiendo también el circuito hasta un nodo HsDir. La petición de conexión a una dirección `.onion` le llegará al nodo HsDir, que responderá al nodo OP con la información relativa al nodo IP correspondiente a la dirección solicitada. Entonces, el nodo OP escogerá un nodo OR de la red que actuará como nodo *rendezvous point* (RP) o nodo de punto de citas. Entre ellos, se intercambian los mensajes `ESTABLISH RP` y `RP ESTABLISHED` para establecer esta conexión.

Una vez escogido el nodo RP, el nodo OP crea un nuevo circuito para conectar al nodo IP. Establecida la conexión, se manda un mensaje al nodo HS con el mensaje `INTRODUCE`. Este mensaje incluye información sobre el nodo escogido como nodo RP. Además, este mensaje también puede contener algún tipo de clave en el caso de que el nodo HS únicamente permita la conexión de nodos OP que conozcan la clave. Una vez

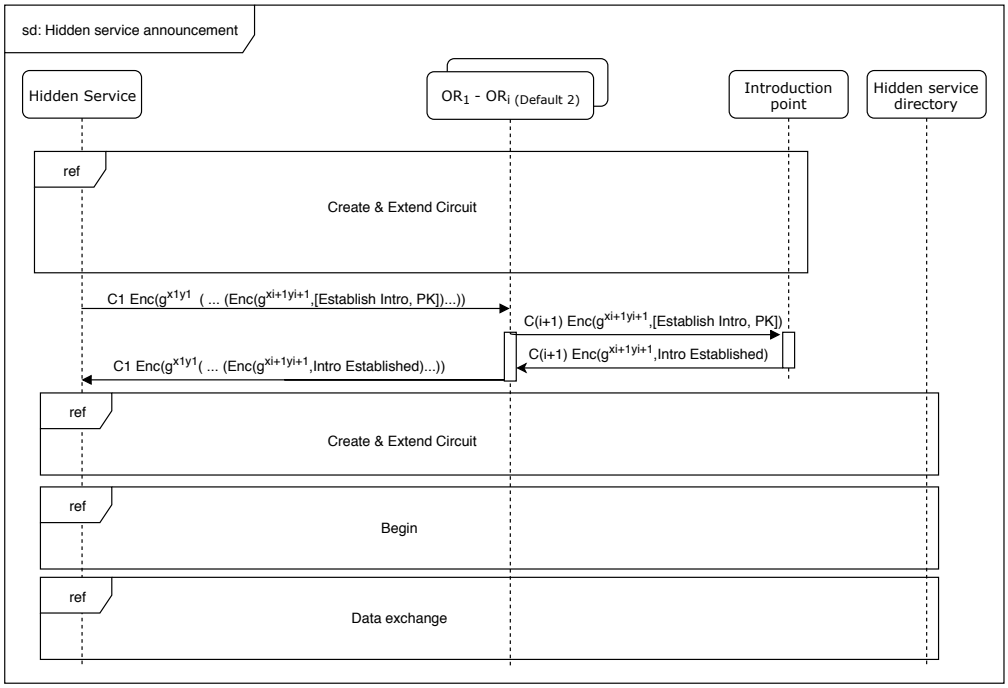

Figura 3.5: Diagrama de secuencia UML *Hidden Service announcement*.

que el nodo OP recibe el ACK del mensaje `INTRODUCE`, el nodo HS crea un circuito con el nodo RP. Tras ello, se informa al nodo OP y comienza la comunicación entre el nodo OP y el nodo HS a través del nodo RP (y los respectivos circuitos creados a cada lado del nodo RP).

3.5. Comportamientos de los nodos

3.5.1. Comportamientos por defecto

La Figura 3.7 muestra las cuatro actividades que cualquier nodo en la red Tor realiza. Estas actividades se realizan de manera concurrente durante el funcionamiento del protocolo Tor. La actividad más crítica es *actualizar el estado consensuado de red*, en la que el nodo espera hasta que le toque solicitar el documento que contiene el estado consensuado de la red. Sin un estado consensuado de red reciente (menos de 24 horas), el resto de actividades no se podrán realizar. La siguiente actividad más relevante es *actualizar descriptores*. En esta tarea, a partir del estado consensuado de red obtenido, el nodo solicita los descriptores de los nodos listados dentro de la red. Esos descriptores contienen más información acerca del nodo, su puerto accesible, las claves públicas, u otros datos acerca del estado del nodo. La actividad de *control de circuitos* abstrae la actividad de creación, eliminación y mantenimiento de los circuitos de conexión. Por último, las acciones de

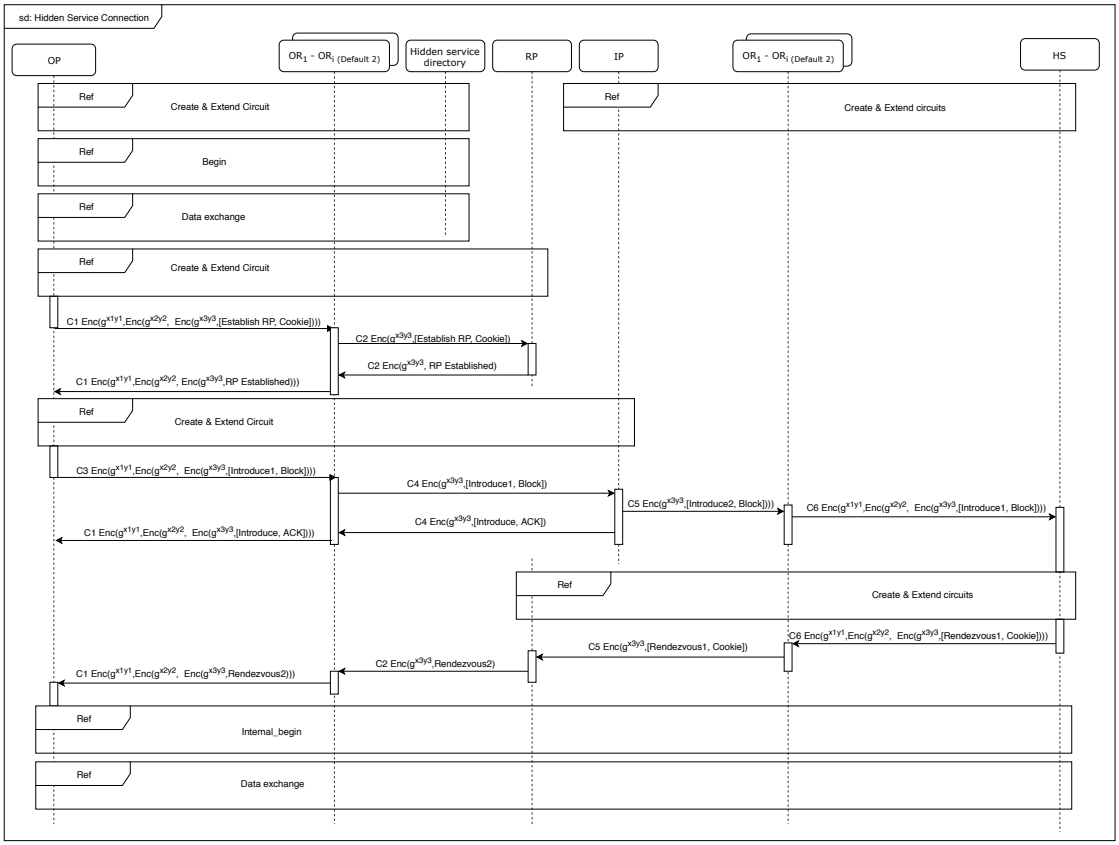

Figura 3.6: Diagrama de secuencia UML *HS connection*.

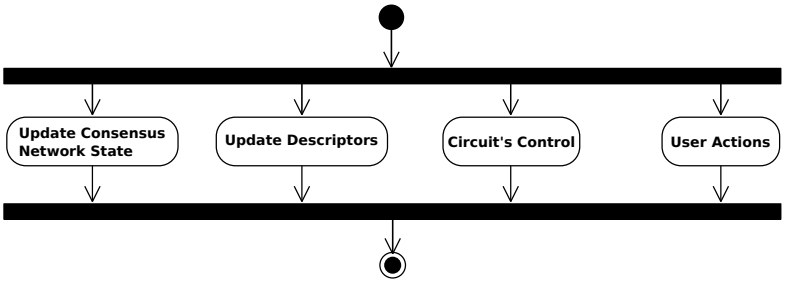

Figura 3.7: Diagrama de actividad UML *Default Behavior*.

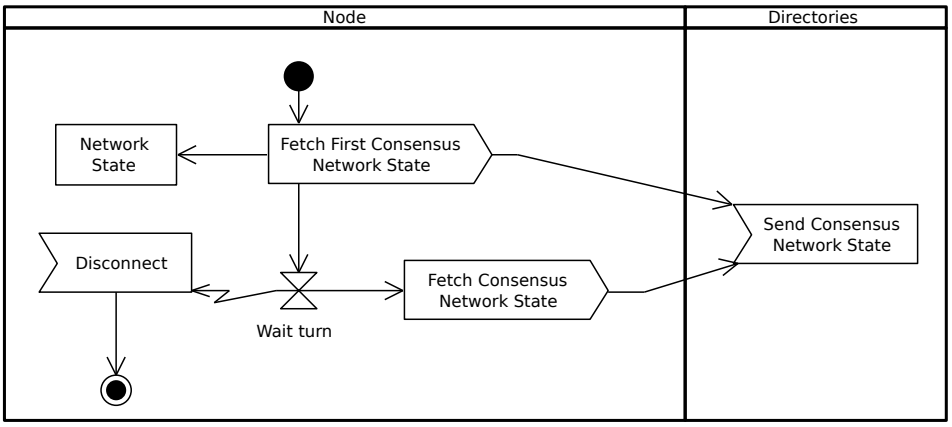

Figura 3.8: Diagrama de actividad UML *Update Consensus Network State*.

usuario son las actividades que realiza un usuario, como reiniciar el nodo, conectarse a un servidor de servicio oculto como nodo OP dentro de la red, o simplemente desconectar el nodo (esta última representa el final de ejecución del protocolo Tor).

Cada una de estas actividades pueden describirse más en detalle. En la Figura 3.8 se representa la actividad de *actualizar el estado consensuado de red*. La primera vez que se solicita el estado consensuado de red esta actividad se realiza de manera diferente, dado que no se tiene información de un estado anterior. El nodo se conecta a uno de los nodos directorios especificados en el propio código de la aplicación a través de una conexión cifrada. En caso de que haya fallo en la conexión, se solicita a otro de los nodos directorios del listado. Posteriormente, el nodo espera hasta que llegue de nuevo su turno de descargar otra vez el estado consensuado de red. Tras cada descarga, se asigna un periodo de tiempo aleatorio entre la hora prevista de publicación de un nuevo descriptor y la hora de caducidad de estado actual. La comunicación después de establecer la primera conexión será un comando `RELAY BEGIN_DIR` hacia un nodo Dir o Auth, seguido de un intercambio de datos. Este proceso se repite hasta que el usuario decida terminar la ejecución del protocolo Tor.

La Figura 3.9 muestra la actividad de *actualizar descriptores*. Como se puede observar, en primer lugar se necesita el estado consensuado de red para saber los nodos actuales de

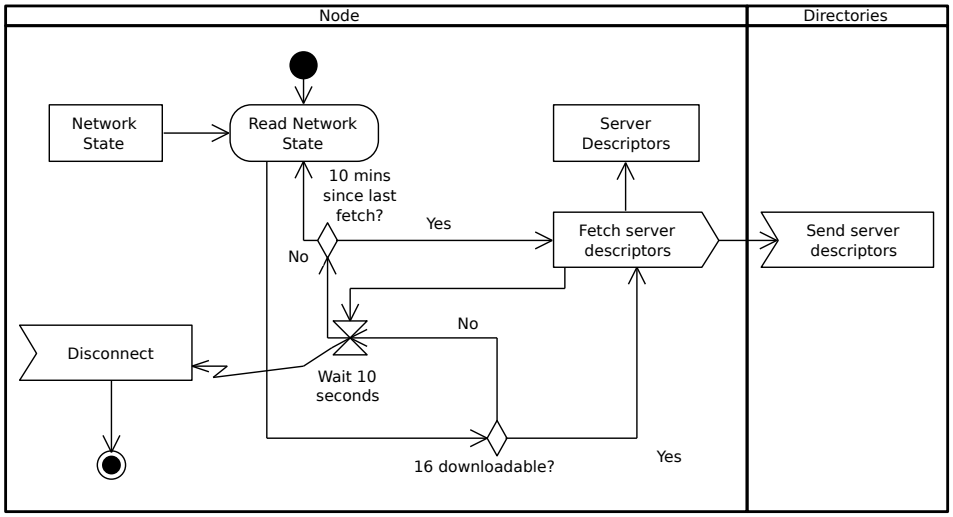

Figura 3.9: Diagrama de actividad UML *Update Descriptors*.

la red y el estado de los mismos. Después, se comprueba si hay al menos 16 descriptores nuevos descargables para su descarga. Si no los hay, se espera 10 segundos para una nueva comprobación. Si pasan 10 minutos desde la última descarga de descriptores, se vuelven a descargar todos los descriptores. La comunicación en este caso sucede de manera idéntica a la de la actividad anterior. Este proceso sucede hasta que se acaba con la ejecución del protocolo Tor.

La red Tor crea circuitos de manera preventiva y para realizar comprobaciones internas. Por ello, todos los nodos realizan la actividad de *control de circuitos* que se muestra en la Figura 3.10. De nuevo, para esta actividad será necesario tener el estado consensuado de red y un número suficiente de descriptores para empezar a construir circuitos. Por tanto, en la primera ejecución la primera decisión que se muestra será negativa, dado que no hay circuitos que requieran atención, así como la segunda decisión también será negativa, puesto que no se posee información suficiente para empezar a crearlos. El ciclo anteriormente descrito se repite hasta que haya información suficiente, momento en el que la decisión *circuits needed* es afirmativa y se crea entonces un circuito (explicado anteriormente en la Sección 3.4). Una vez existan circuitos, cada pocos segundos se realiza un control mediante el envío del comando `PADDING` para comprobar que sigue vivo. En el caso de que no se reciba el comando `PADDING` de un nodo OR o se necesite cambiar el circuito de alguna manera (ya sea mediante una destrucción del mismo o un truncamiento), se actualizarán los circuitos como corresponda. Después, se vuelve a la situación de si se necesitan más circuitos, repitiéndose así el ciclo.

3.5.2. Comportamiento de los nodos OR

Adicionalmente a las actividades comunes descritas anteriormente, un nodo OR realiza también un par más de actividades: una de ellas, denominada *publish descriptor*,

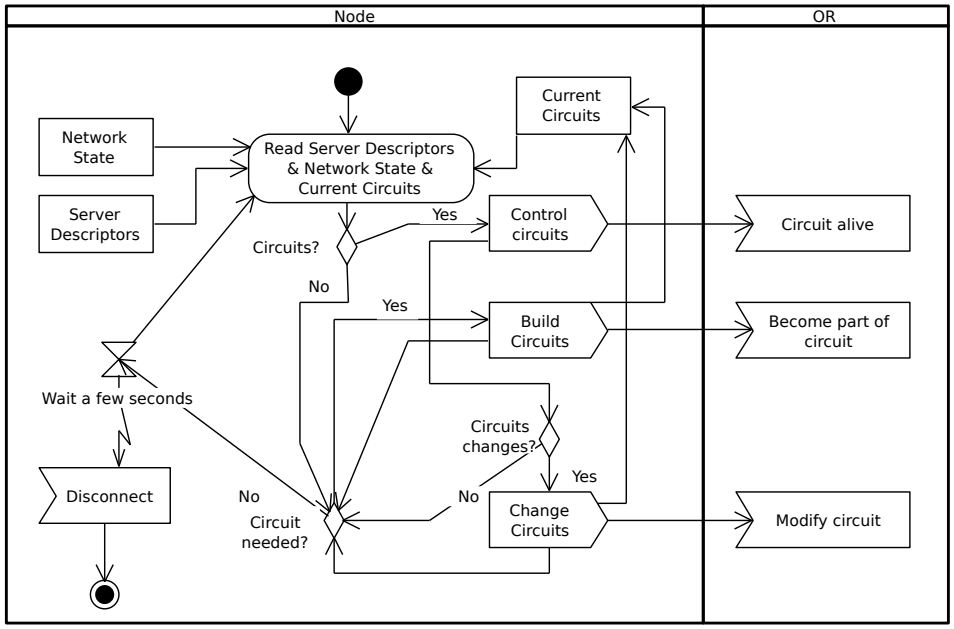

Figura 3.10: Diagrama de actividad UML *Circuits' Control*.

consiste en publicar en intervalos de al menos 12 horas su descriptor a los nodos Auth. Este descriptor sirve para elaborar el estado consensuado de red. La otra actividad, *manage relay's connection*, es el procesado de los mensajes que recibe para retransmitirlos. Recuérdese que el nodo OR es el nodo en el que se basan los circuitos Tor, con lo que un nodo OR está constantemente estableciendo conexiones a otros nodos OR y respondiendo/retransmitiendo los mensajes recibidos.

3.5.3. Comportamiento de los nodos Auth

Los nodos Auth realizan cinco actividades adicionales a las actividades de un nodo OR, como se muestra en la Figura 3.11. La actividad fundamental que realiza es elaborar el estado consensuado de red, a partir de la información recabada por las distintas autoridades. Esta actividad se repite de manera ininterrumpida. Cada vez que se genera un estado consensuado de red, este estado se almacena y se suministra a los nodos Dirs y los nodos OR cuando lo soliciten. Para no sobrecargarse con las peticiones de los nodos OR, los nodos Auth envían los descriptores recogidos a los nodos Dir que lo soliciten (es decir, los nodos Dir actúan como unos nodos secundarios de los nodos Auth). Los nodos Auth también solicitan los descriptores de otros nodos Auth que aún no tienen en su listado. Por último, los nodos Auth reciben los descriptores que publiquen los demás nodos de la red Tor.

Todas las comunicaciones que se realizan se reducen a enviar o recibir un mensaje `RELAY BEGIN_DIR` y posteriormente, enviar o recibir datos con un mensaje `RELAY DATA`.

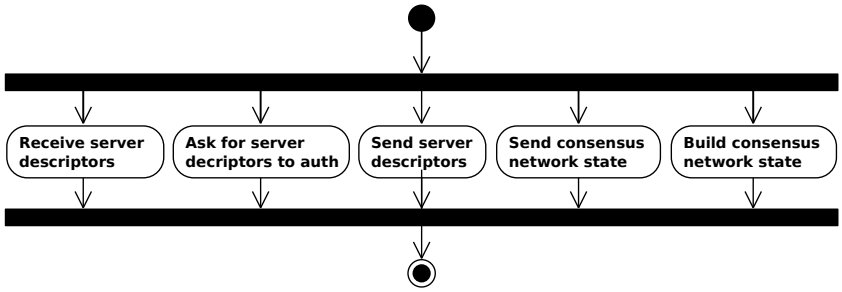

Figura 3.11: Diagrama de actividad UML del comportamiento de los nodos Auth.

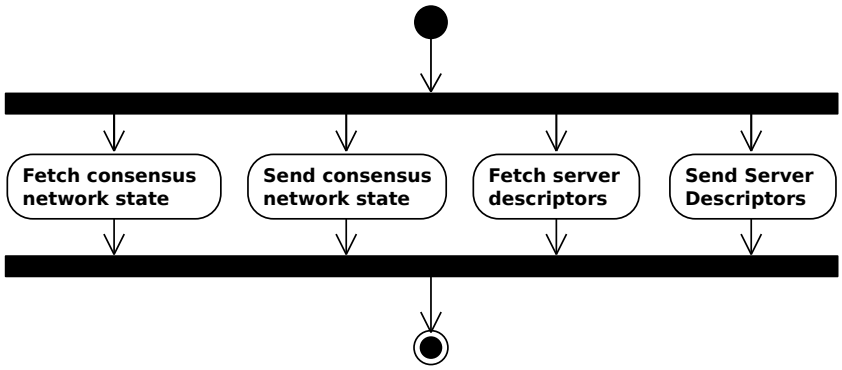

Figura 3.12: Diagrama de actividad UML del comportamiento de los nodos Dir.

3.5.4. Comportamiento de los nodos Dir

Como se ha descrito anteriormente, los nodos Dir fueron creados con la intención de aligerar el tráfico de los nodos Auth y reducir su sobrecarga de acciones. Las actividades que realiza un nodo Dir se muestran en la Figura 3.12. Como se muestra, sus actividades son únicamente obtener y suministrar el estado consensuado de la red y los descriptores de otros nodos de la red Tor, que pueden ser solicitados por otros nodos OR.

Capítulo 4

Sistema de extracción de características de servicios ocultos

En este apartado se explica en detalle cómo se ha llevado a cabo la obtención de direcciones `.onion` (recuérdese que son las direcciones para acceder a los servicios ocultos), así como la extracción de la información de los servicios ocultos encontrados. También se discuten las diferentes alternativas para lograr la desanonimización de un servicio oculto, profundizando en la opción escogida en este trabajo.

4.1. Descripción del sistema

El sistema completo se muestra en la Figura 4.1 y se puede dividir en dos partes principales:

- **La recopilación de direcciones de servicios ocultos**, que permiten el acceso a los servicios ocultos a través de Tor. Estas direcciones no están publicadas en ningún documento de la red Tor, así que hay que obtenerlas por otros medios. Esta parte del sistema, denominada *crawler*, dado un repositorio de direcciones `.onion` establece conexiones a través de un OP con los servicios ocultos asociados a esas direcciones, almacenando en la base de datos el código fuente de la página principal, las cabeceras HTTP e información adicional. Además, se analiza de manera automática la página principal del servicio oculto en busca de nuevas direcciones, que serán procesadas del mismo modo.

- **El análisis de características de los servicios ocultos**. A partir de la información obtenida, la herramienta de análisis denominada *Onion tool* extrae las características de servicio, con especial interés en aquellas que puedan llevar a la desanonimización. Para esta parte se utiliza el motor de búsqueda Shodan (véase la Sección 2.2) y también se analiza el contenido del servicio para determinar el uso de un servicio oculto.

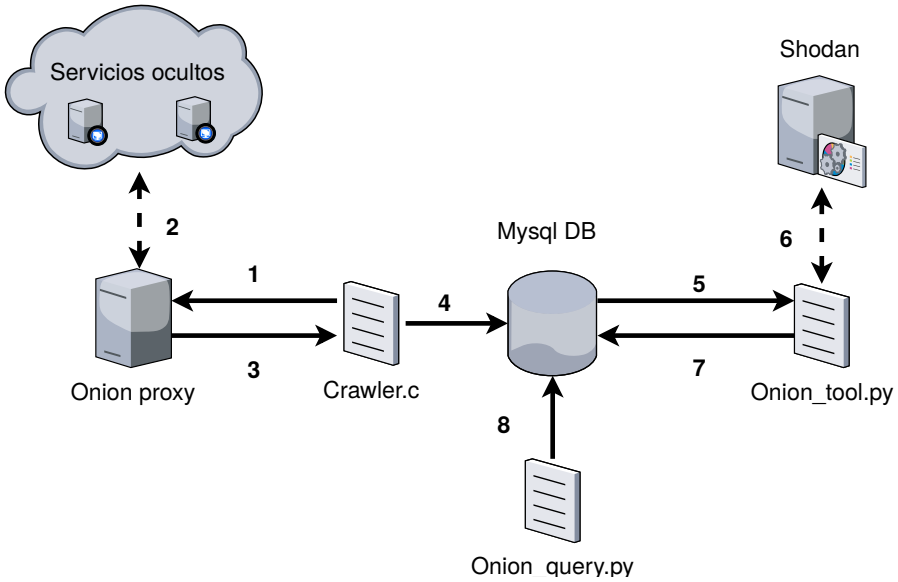

Figura 4.1: Diagramas en alto nivel del sistema.

Además, se ha desarrollado un *script* denominado *Onion query* que permite consultar la información extraída de los servicios ocultos. Este *script* suministra el número de dispositivos acotados, las direcciones IP de los dispositivos y sus cabeceras. Adicionalmente, también suministra información del idioma utilizado en el servicio oculto y el número de coincidencias con los diccionarios de palabras establecidos.

4.2. Recopilación de direcciones de servicios ocultos

El primer problema a abordar es recopilar un número considerable de direcciones de servicios ocultos. Se plantean dos soluciones a este problema:

Nodo dentro de la red Tor

Levantar un nodo OR en la red Tor, que se encargue de suministrar los descriptores de servicio oculto y conozca sus direcciones, para modificar su código y guardar estas direcciones. Recuérdese que los servicios ocultos no siguen un sistema de nombres de dominio convencional. En concreto utilizan un sistema DHT, en el que los nodos HsDir almacenan los descriptores asociados al servicio oculto correspondiente.

Las condiciones para ser un nodo HsDir son exigentes, ya que el nodo necesita estar en línea durante al menos 96 horas, ser estable (no sufrir caídas) y estar entre los 8 nodos con mejor banda ancha o disponer de al menos 100KB/s. Dadas las dificultades para conseguir una máquina en la Universidad de Zaragoza de estas características y que además no se tuviera ningún problema legal por el uso de la red Tor, se dispuso

de una máquina en Rediris, que es la red española para Interconexión de los Recursos Informáticos de las universidades y centros de investigación. En esta máquina se instaló Tor versión 3.1.8.

A pesar de que era costoso obtener el reconocimiento como HsDir, la máquina obtuvo este rol durante diferentes períodos de su actividad. Sin embargo se descartó por las novedades introducidas en la versión 3 de Tor. Una de las novedades se trata de un sistema para defenderse de este método de captura de direcciones de servicios ocultos, utilizado con éxito en versiones anteriores [14]. Ahora los nodos HsDir almacenan una cadena derivada de la dirección `.onion` real. De esta manera, los usuarios que ya conocen la dirección real pueden acceder a los descriptores almacenados en los nodos HsDir, sin embargo estos nodos no pueden acceder al servicio oculto al no conocer la dirección real.

Al considerar que trabajar con una versión 2.0 de Tor no era lo más apropiado de cara al futuro, se desestimó esta posibilidad como fuente de direcciones `.onion`.

Repositorios de direcciones .onion

La idea principal fue usar un repositorio de direcciones de servicios ocultos del que extraer datos. Sin embargo, dado que muchos de los repositorios tienen direcciones `.onion` obsoletas, se ha desarrollado una herramienta en C (para facilitar su integración con el código de Tor, en caso de que fuese necesario) capaz de conectarse a un servicio oculto y extraer todas las direcciones `.onion` que aparezcan en su página principal. Así, se amplía el número de direcciones `.onion` extraídas, aumentando también la diversidad de los datos obtenidos.

En esta herramienta, aprovechando la recopilación del código fuente de la página principal del servicio oculto, se ha integrado también la parte de extracción de datos y metadatos de los servicios. Por tanto, la herramienta recopila también la cabecera HTTP del sitio, la fecha y el tipo de protocolo utilizado HTTP/HTTPS, enlaces a otras direcciones `.onion` o a páginas web del Internet convencional.

La conexión a un servicio oculto se realiza a través de la librería CURL y una conexión proxy a través de Tor, obteniéndose la cabecera HTTP del servicio oculto así como el contenido de su página principal y los datos de su certificado en caso de que el servicio oculto disponga de él. La información se almacena en una base de datos *MySQL* para su posterior análisis. La estructura de esta base de datos tiene un diseño desacoplado para que pueda ser utilizada por diferentes aproximaciones en el estudio de la información de los servicios ocultos.

4.3. Extracción de características

A partir de la información recopilada en la sección anterior, se desarrolla una herramienta que permita identificar características que puedan llevar a la desanonimización del servicio oculto o a la detección de finalidades ilegítimas. Para la parte de desanonimización se ha recurrido a Shodan, aunque es posible el uso de otras alternativas.

Desanonimización de Servicios Ocultos

La idea de desanonimización que se desarrolla se realiza a partir de los metadatos que voluntariamente o involuntariamente se dejan accesibles en un servicio oculto. A partir de estos metadatos se intentará llevar a cabo al desanonimización con herramientas como Shodan o servicios similares como Censys. La información relevante que pueda llevar a una relación directa, se buscará a través de estas herramientas, que muestra la dirección IP de aquellos servidores con características similares a las propias del servicio oculto. En esta aproximación se ha optado por utilizar Shodan, pero la herramienta es extensible a otros motores que permitan la búsqueda de metadatos y suministren un listado de dispositivos con sus direcciones IP.

Como se ha comentado anteriormente, los metadatos considerados aquí son las cabeceras HTTP y los certificados SSL/TLS. La extracción de cabeceras HTTP se realiza como se describe anteriormente en la Sección 4.2. De los certificados se ha priorizado la obtención del número de serie y la huella digital, aunque debido a la dificultad para automatizar la extracción de esta información y que su búsqueda en Shodan no ha proporcionado buenos resultados, se han excluido de la parte de análisis realizado.

Para extraer las cabeceras más importantes, se han agrupado los diferentes tipos de nombre de cabeceras. Posteriormente, se ha realizado una búsqueda en Shodan del nombre de la cabecera sin su valor, obteniendo la cantidad de dispositivos con esa cabecera. Por último, se ha establecido una relación de orden de significancia inverso al número de dispositivos que aparecen (es decir, a menor número más significativa).

Se ha utilizado un algoritmo de búsqueda voraz priorizando la búsqueda de las cabeceras más significativas. La búsqueda comenzará con la cabecera más significativa del servicio oculto que devuelva dispositivos en Shodan. A esta cabecera se le irán añadiendo las cabeceras que hagan disminuir el número de dispositivos devueltos en Shodan. Además, se comprueban las cabeceras individualmente, para que en caso de que una cabecera menos significativa acote mejor que la suma de cabeceras más significativas, sea ésta la elegida para la progresión del algoritmo.

Para entender mejor la extracción de características, se muestra la salida del *Onion query* para el buscador `https://duckduckgo.com/`, que dispone de un servicio oculto en la dirección `3g2upl4pq6kufc4m.onion`:

```
Onion address: 3g2upl4pq6kufc4m
Query: "Expect-CT: max-age=0 && Connection: keep-alive && Content-Length:
5496"
Suspects: 2
```

Las cabeceras HTTP obtenidas del servicio oculto que mejor acotan el número de dispositivos es: `Expect-CT: max-age=0`, `Connection: keep-alive` y `Content-Length: 5496`. Al realizar la búsqueda de estos pares campo-valor de las cabeceras en Shodan obtienen dos servidores de DuckDuckGo, como se muestra en la Figura 4.2.

A pesar de que este algoritmo puede no llevar a la solución óptima, se ha escogido debido a que el número de peticiones a Shodan es limitado y además está restringido a

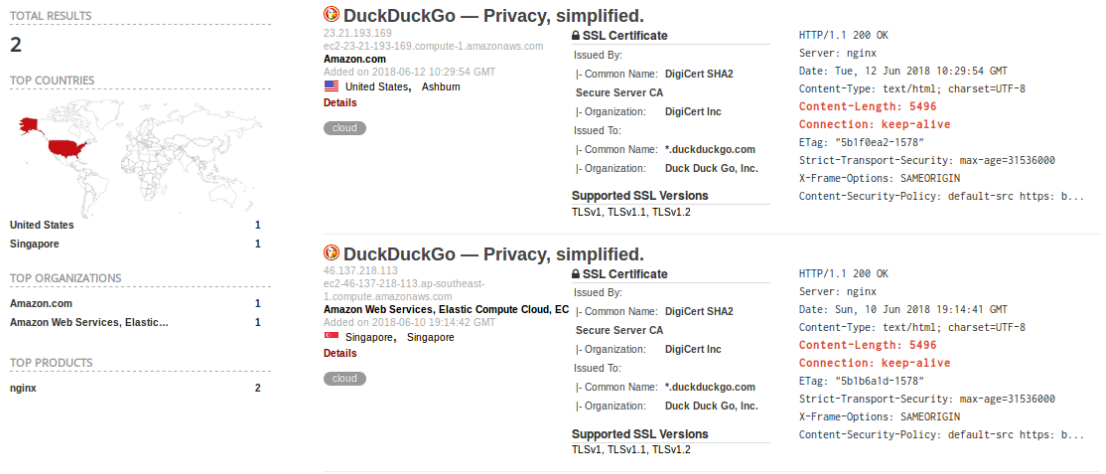

Figura 4.2: Búsqueda en Shodan de las características extraídas.

una petición por segundo. Un algoritmo que expanda todas las posibilidades puede llevar a mejores resultados.

Categorización de Servicios Ocultos

Para la categorización de los datos por razones legales, se ha decido evaluar la información de la página principal de los diferentes sitios encontrados, sin ejecutar contenido dinámico, ni descargar ningún archivo multimedia por cuestiones éticas. Se ha utilizado el lenguaje natural de estas páginas y diccionarios de palabras clave comunes en algunas actividades que pueden indicar un uso ilegítimo. El criterio para establecer que un servicio oculto pertenece a una categoría ha sido la aparición de al menos cinco palabras de los diccionarios correspondientes. Como categorías, se ha considerado *drogas*, *contenido sexual*, *criptomonedas*, *terrorismo*. Los diccionarios utilizados para cada categoría se muestran en la Tabla 4.1

Se han utilizado las librerías *BeautifulSoup* para el procesado de código HTML y CSS, y la librería NLTK para el procesado del lenguaje natural y *langdetect* para la detección de idiomas.

Drogas	*marihuana*, *cocaine*, *cocain*, *coke*, *heroine*, *heroin*, *popper*, *pills*, *lsd*, *drug*, *drugs*, *viagra*, *pills*, *amphetamines*, *amphetamine*, *hashish*, *hallucinogen*, *hallucinogens*, *opium*, *cannabis*, *peyote*, *marijuana*, *methamphetamine*, *methamphetamines*, *crack*, *crystal*, *meth*, *mdma*, *ghb*, *flunitrazepam*, *ketamine*, *salvia*, *dextromethorphan*, *dxm*, *mescaline*, *psilocybin*, *steroids*
Contenido sexual	*pornography*, *sex*, *sex-cam*, *pedophile*, *bukkake*, *gangbang*, *anal*, *blowjob*, *pov*, *creampie*, *masturbation*, *rape*, *handjob*, *cum*, *eyaculate*, *doggystyle*, *porn*, *squirting*, *cumshoot*, *orgasm*, *fuck*
Criptomonedas	*bitcoin*, *ethereum*, *ripple*, *cardano*, *stellar*, *NEO*, *litecoin*, *EOS*, *NEM*
Terrorismo	*terrorism*, *yihad*, *terrorist*, *terror*, *attack*, *target*, *enriched*, *nuclear*, *IED*, *bomb*, *islamist*, *kidnapping*

Tabla 4.1: Palabras clave de cada categoría.

Capítulo 5

Análisis de los servicios ocultos

En este capítulo se exponen los resultados obtenidos además de estadísticas de la información recopilada.

5.1. Obtención de direcciones .onion

Se han obtenido 17328 direcciones `.onion` de versión 2 y 3 (oficialmente se reconocen un total estimado de 110.000 direcciones versión 2). De estas direcciones, un 99 % son de la versión 2 y un 1 % de la versión 3. De todas ellas únicamente se ha podido establecer conexión HTTP/HTTPS al 10 %, considerando un tiempo de respuesta de alrededor de 30 segundos. Este número tan bajo de conexiones puede ser debido a que los servicios ocultos se pueden configurar para solicitar una contraseña de acceso. Además, existen otros protocolos que no se han estudiado (como conexión mediante SSH o Telnet) y por último algunas direcciones de muchos de los servicios ocultos que se estiman se encuentran caídos o eliminados. En concreto, se ha establecido la conexión HTTP con 1636 servicios y la conexión HTTPS con 160 servicios, todos ellos usando la versión 2.

5.2. Desanonimización y categorización

De las 1796 direcciones recopiladas para la desanonimización se han utilizado 1794 direcciones, dado que dos de ellas no suministaron metadatos en las cabeceras. Se ha conseguido localizar un sólo dispositivo con cabeceras idénticas o muy similares para 346 servicios ocultos, en las Figuras 5.2 y 5.3 se muestran dos ejemplos particulares de esta desanonimización. Como se aprecia la identificación del servicio oculto se ha realizado con éxito. Además, la herramienta desarrollada, consigue acotar el 30 % de los servicios ocultos analizados a menos de 10 dispositivos. El número de dispositivos acotados para los servicios ocultos se muestra en la Figura 5.1

Aunque en todos los casos no se haya encontrado una relación directa, no implica que sea un error, dado que puede que el servicio oculto sea una web diferente al servicio que se accede por el Internet convencional. Los resultados sí que implican que los metadatos

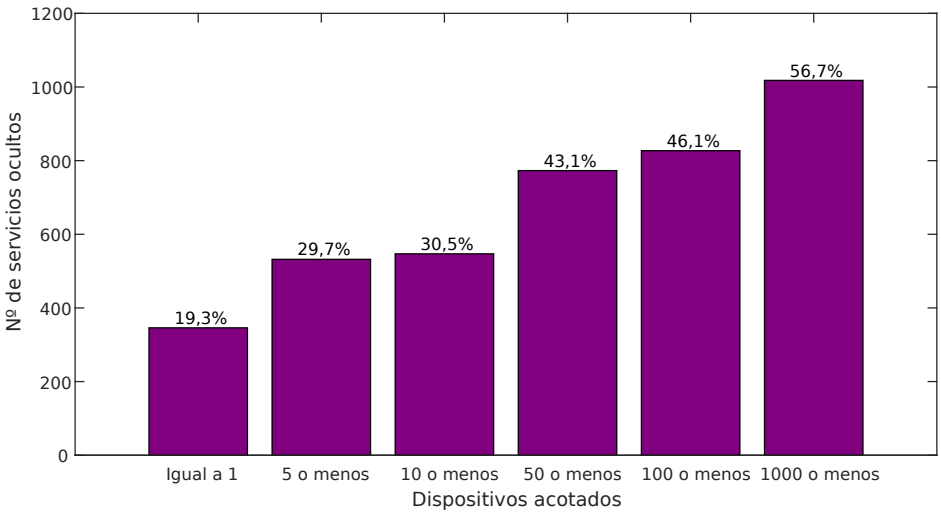

Figura 5.1: Número de dispositivos acotados para cada servicio oculto.

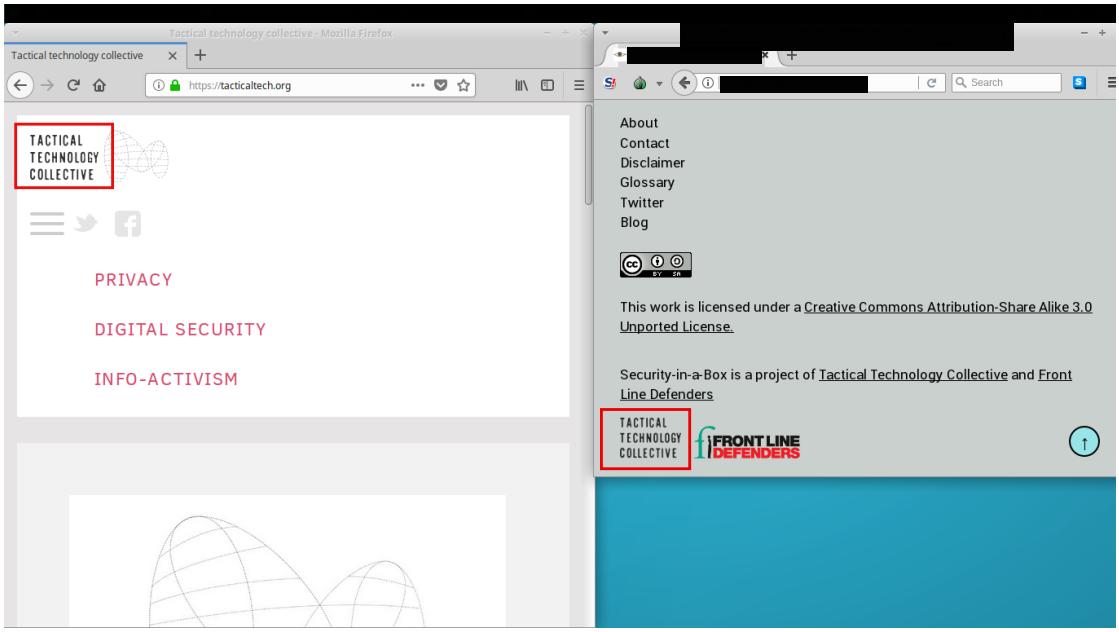

Figura 5.2: Ejemplo de desanonimización: misma organización.

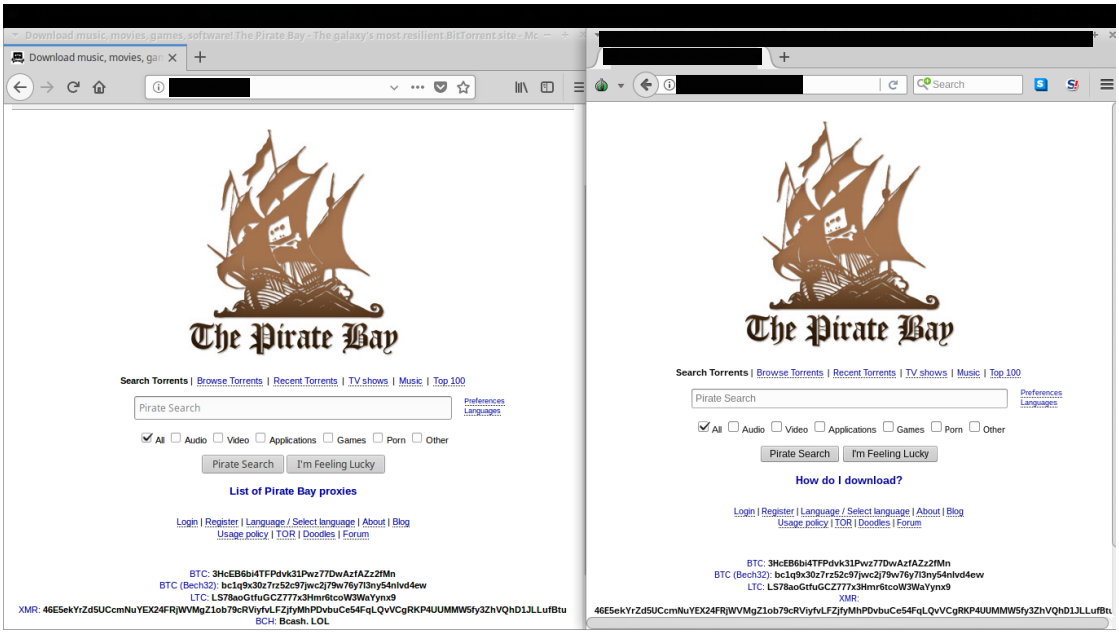

Figura 5.3: Ejemplo de desanonimización: mismo contenido.

suministrados por una cantidad considerable de los servicios ocultos accedidos, pueden llevar a una desanonimización de los mismos, en caso de que se tenga en la misma máquina otro servicio con similar configuración en otro puerto abierto a la red convencional.

En la parte de categorización, como se muestra en la Figura 5.4 se han detectado 7 posibles servicios ocultos de temática orientada al terrorismo, 13 sitios relacionados con contenido sexual, 63 asociados a temas de drogas y 124 pertenecientes al mundo de las criptomonedas. Recuérdese que esta caracterización se realiza considerando que un servicio oculto pertenece a estas categorías si tiene en su página principal al menos 5 palabras del diccionario de dicha categoría. El estudio preliminar (y fácilmente ampliable) muestra que existen en Tor servicios ocultos con finalidades cuestionables.

Respecto a los idiomas, se han detectado 28 idiomas diferentes de los 55 soportados por la librería. El más utilizado es con diferencia el inglés que se asocia 1314 sitios, apareciendo el castellano únicamente en 25. Se puede ver con más detalle los idiomas detectados en la Figura 5.1.

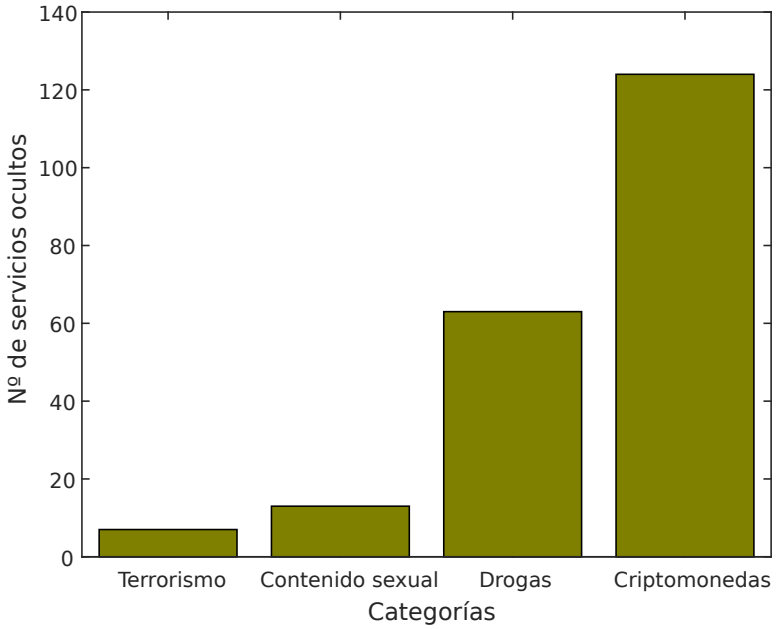

Figura 5.4: Servicios ocultos categorizados.

Idioma	**Apariciones**	**Idioma**	**Apariciones**
English	1313	Romanian	4
German	54	Turkish	4
Danish	38	Welsh	3
Portuguese	33	Slovak	3
Spanish	25	Swedish	3
French	19	Swahili	3
Italian	8	Tagalog	3
Norweigan	8	Vietnamese	3
Afrikaans	7	Indonesian	2
Dutch	7	Bulgarian	1
Somali	7	Estonian	1
Finish	6	Lithuanian	1
Polish	5	Albanian	1
Catalan	4	Unknown	229

Tabla 5.1: Idiomas de los servicios ocultos.

Capítulo 6

Trabajo relacionado

En este capítulo se introducen las publicaciones y herramientas relacionadas con el trabajo desarrollado.

Respecto a la parte de modelización del protocolo Tor, no se ha encontrado ningún trabajo similar, ya que la mayoría de los trabajos realizados están orientados a analizar las fortalezas y debilidades de privacidad y anonimato de la red Tor mediante distintas aproximaciones.

Diversos tipos de aproximaciones se han utilizado con éxito para desanonimizar un servicio oculto mediante la participación directa en el circuito de conexión a este [13]. También se ha conseguido evitar la protección de Tor identificando al servicio oculto al que se conecta un usuario mediante análisis del tráfico [12, 15].

En [14] se realiza un estudio sobre los servicios ocultos, describiendo su acceso a través de Tor y se realiza un estudio de los accesos a los servicios ocultos con una categorización de los mismos. En [9] se realiza una aproximación similar a la realizada, a partir de una herramienta que busca información suministrada por los servicios ocultos que pueda llevar a su desanonimización. Se analizan 6426 direcciones `.onion`, de las que se consigue establecer conexión con 1974 (cantidad similar a la analizada en este trabajo). El porcentaje de éxito en la aproximación es del 5 %, tras realizar un sistema propio de validación basado en la similitud de las cabeceras HTTP y el código HTML de la página.

Respecto a herramientas de análisis de direcciones `.onion` es interesante destacar la herramienta `OnionScan`. Es una herramienta desarrollada en el lenguaje Go, que analiza las debilidades de un servicio oculto en funcionamiento. La herramienta obtiene directorios accesibles, metadatos, características del servidor, o tipo de protocolos utilizados. `OnionScan` se ha utilizado con éxito en algunos servicios ocultos obteniendo directorios accesibles además de alguna dirección de correo, información que fácilmente lleva a la desanonimización de los servicios.

Capítulo 7

Conclusiones y trabajo futuro

La red Tor nació para mejorar el anonimato y la privacidad de sus usuarios durante la navegación por Internet. Basada en el establecimiento de circuitos entre los diferentes nodos de la red, y garantizando un número mínimo de nodos en cada conversación, se garantiza la no trazabilidad de las conexiones.

En este trabajo se ha estudiado el protocolo Tor, se ha definido su estructura y su comportamiento, aportando diagramas UML, a diferencia de otros estudios que se centran más en el ánalisis de las fortalezas y debilidades de Tor. De esta manera, se facilita la comprensión del funcionamiento de la red Tor. Cabe destacar que no se ha encontrado ningún fallo a nivel de diseño que permita identificar al propietario de servicio oculto ni al usuario que se conecta al mismo según la protección que garantiza Tor.

Tor permite la creación de sitios web únicamente accesibles mediante la propia red Tor y no mediante los buscadores convencionales, permaneciendo el propietario del servicio oculto bajo el anonimato que la red Tor proporciona. Este anonimato conlleva a que proliferen sitios en la red Tor de dudosa reputación.

En este trabajo además se ha analizado también la extracción de características de los servicios ocultos que puedan llevar a una desanonimización del mismo , consiguiendo acotar la búsqueda a menos de 10 dispositivos en más de un 30 % de los servicios ocultos obtenidos. Esto demuestra que, a partir de los metadatos de los protocolos web HTTP/HTTPS se puede lograr la desanonimización de un servicio oculto (en algunos casos). Respecto a la categorización, en la que no se ha podido profundizar todo lo deseado, demuestra que existe un sector de servicios ocultos potencialmente relacionados con actividades ilegítimas.

7.1. Trabajo futuro

Como trabajo futuro se plantean diferentes líneas de acción:

- ***Verificación***. A partir de la modelización realizada, se puede obtener una verificación formal del modelo para encontrar alguna deficiencia oculta y no evidente en el diseño protocolo. Otro aspecto de interés sería verificar que el modelo es fiel a la implementación del protocolo.
- ***Sistema automático de evaluación.*** Uno de los problemas encontrados es la evaluación de los resultados obtenidos de manera automática. Es necesario algún sistema basado en técnicas de aprendizaje automático que pueda determinar la relación entre los dispositivos acotados y el dispositivo del servicio oculto.
- ***Mejora de clasificación*** Dada la dimensión y la transversalidad del proyecto, no se ha llegado a profundizar en los aspectos de clasificación de servicios ocultos. De nuevo, mediante técnicas de aprendizaje automático se puede mejorar la categorización de los servicios. Además, se puede recopilar información de los servicios ocultos no considerada en este estudio, como imágenes, enlaces de la página, etc; siempre y cuando se estuviera en los límites éticos establecidos.

Bibliografía

[1] Shodan. `https://www.shodan.io/`. [Online; accedido 12-Mayo-2018].

[2] H. Chen. *Dark Web: Exploring and Data Mining the Dark Side of the Web*. Integrated Series in Information Systems. Springer New York, 2011.

[3] Nicolas Christin. Traveling the Silk Road: A Measurement Analysis of a Large Anonymous Online Marketplace. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 213–224, New York, NY, USA, 2013. ACM.

[4] Joeri de Ruiter and Erik Poll. Formal Analysis of the EMV Protocol Suite. In Sebastian Mödersheim and Catuscia Palamidessi, editors, *Theory of Security and Applications*, volume 6993 of *Lecture Notes in Computer Science*, pages 113–129. Springer Berlin Heidelberg, 2012.

[5] elPeriódico. "Más dura será la caída": así ha nombrado la Fiscalía el archivo sobre las querellas de Puigdemont. `https://www.elperiodico.com/es/politica/20171030/mas-dura-sera-caida-comunicado-fiscalia-puigdemont-6390888`. [Online; accedido 25-Junio-2018].

[6] David Goulet, Aaron Johnson, George Kadianakis, and Karsten Loesing. Hidden-service statistics reported by relays. techreport 2015-04-001, The Tor Project, April 2015.

[7] Free Haven. Papers in anonimity. `https://www.freehaven.net/anonbib/`. [Online; accedido 29-Septiembre-2017].

[8] Marie Claire Van Hout and Tim Bingham. 'Silk Road', the virtual drug marketplace: A single case study of user experiences. *International Journal of Drug Policy*, 24(5):385–391, 2013.

[9] Srdjan Matic. *Active Techniques for Revealing and Analyzing the Security of Hidden Servers*. PhD thesis, Università degli Studi di Milano, 2016.

[10] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. *Shining Light in Dark Places: Understanding the Tor Network*, pages 63–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[11] R. Nardone, R. J. Rodríguez, and S. Marrone. Formal Security Assessment of Modbus Protocol. In *Proceedings of the 11th International Conference for Internet Technology and Secured Transactions*, pages 142–147. IEEE, December 2016.

[12] Rebekah Overdorf, Mark Juarez, Gunes Acar, Rachel Greenstadt, and Claudia Diaz. How unique is your .onion?: An analysis of the fingerprintability of tor onion services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 2021–2036, New York, NY, USA, 2017. ACM.

[13] L. Overlier and P. Syverson. Locating hidden servers. In *2006 IEEE Symposium on Security and Privacy (S P'06)*, pages 15 pp.–114, May 2006.

[14] G. Owen and N. Savage. Empirical analysis of Tor Hidden Services. *IET Information Security*, 10(3):113–118, 2016.

[15] Andriy Panchenko, Asya Mitseva, Martin Henze, Fabian Lanze, Klaus Wehrle, and Thomas Engel. Analysis of Fingerprinting Techniques for Tor Hidden Services. In *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, WPES '17, pages 165–175, New York, NY, USA, 2017. ACM.

[16] Tor Project. Tor documentation. `https://www.torproject.org/docs/documentation.html.en#DesignDoc`. [Online; accedido 5-Octubre-2017].

[17] Tor Project. Tor git documentation. `https://gitweb.torproject.org/torspec.git/tree`. [Online; accedido 8-Octubre-2017].

[18] Tor Project. Tor metrics. `https://metrics.torproject.org`. [Online; accedido 21-Junio-2018].

Apéndice A

Horas de Trabajo

En la Figura A.1 se detalla el tiempo invertido en cada tarea. La Figura A.2 contiene un diagrama de Gantt ilustrando la panificación de este trabajo. Como se puede observar se comienza con una investigación y documentación de Tor. Paralelamente se levanta un nodo OR para intentar capturar las direcciones `.onion`. A continuación se comienza con la modelización de Tor y una vez completada, se inicia la investigación a cerca de la desanonimización de los servicios oculto y la documentación de Shodan. Al final se realiza el diseño e implementación del sistema y por último la elaboración de la memoria, aunque se ha utilizado trabajo redactado previamente.

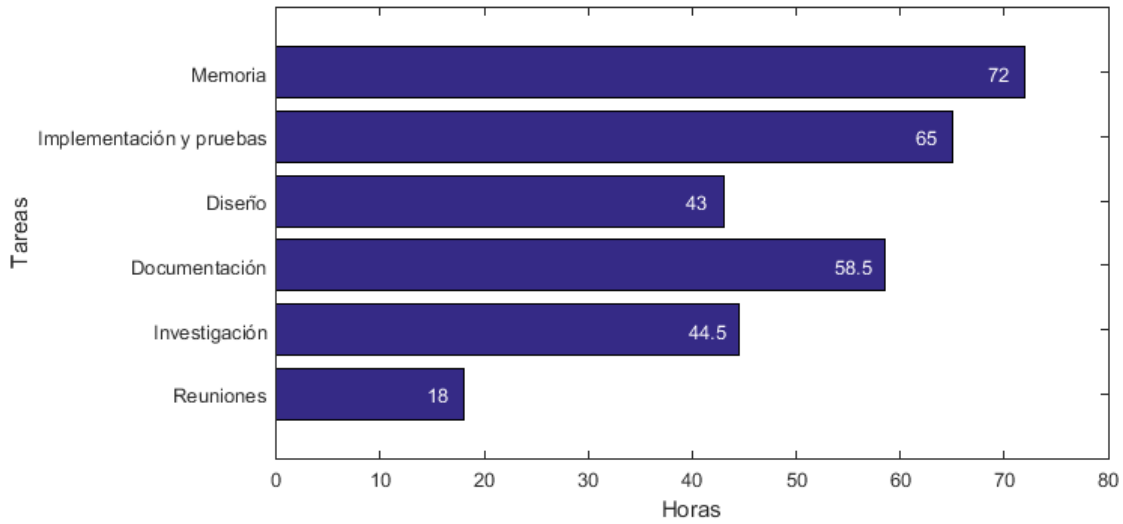

Figura A.1: Desglose de horas empleadas por tarea.

Figura A.2: Diagrama de Gantt.