ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

Implementación y análisis de un nodo bridge en la red TOR

(Implementation and analysis of a bridge relay in the TOR network)

Para acceder al Título de

Graduado en

Ingeniería de Tecnologías de Telecomunicación

Autor: Luis Fernando Garrido Saiz

Mayo - 2019

E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Luis Fernando Garrido Saiz
Director del TFG: Roberto Sanz Gil

Título: "Implementación y análisis de un nodo bridge en la red TOR"
Title: "Implementation and analysis of a bridge relay in the TOR network"

Presentado a examen el día: 19/06/2019

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): Alicia Casanueva López
Secretario (Apellidos, Nombre): Alberto Eloy García Gutiérrez
Vocal (Apellidos, Nombre): Roberto Sanz Gil

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº
(a asignar por Secretaría)

Agradecimientos

En primer lugar, quiero dar las gracias a Roberto, director de este TFG, por darme la idea y ayudarme en todo lo que ha podido. Muchas gracias por tu disposición.

También quiero dar las gracias a mis compañeros de clase por estar tan unidos y ayudarnos mutuamente estos años. Gracias Raúl, Alberto, Martin y Jorge.

Para acabar, muchísimas gracias a mis padres y el resto de mi familia, que sin ellos no habría llegado hasta aquí.

Luis Fernando Garrido Saiz, mayo 2019

Resumen

El proyecto trata de analizar la Tor Network. Empezamos por una visión general de la red, centrándonos en qué es Tor, un poco de su historia, los diferentes tipos de usuarios que lo utilizan y su funcionamiento, para después centrarnos en los bridges relays, uno de los diferentes tipos de relays que forman esta red, creados para luchar contra la censura de esta red.

Explicamos que son, cómo funcionan y los protocolos que utilizan, llamados transportes conectables y utilizados para camuflar el tráfico y evitar que se identifique como tráfico Tor. Después implementamos uno de estos relays en un equipo propio y a continuación, analizaremos el tráfico que pasa por él con el software Wireshark.

Para finalizar probamos las diferentes opciones de acceso a la red Tor a través de nuestro Bridge relay, tanto desde diferentes navegadores, como desde un Smartphone.

Abstract

The project tries to analyze the Tor Network. We start with an overview of the network, focusing on what Tor is, a bit of its history, the different types of users that use it and its operation, and then focus on bridges relays, one of the different types of relays that form this network, created to fight against the censorship of this network.

We explain what they are, how they work and the protocols they use, called pluggable transport and used to camouflage traffic and prevent it from being identified as Tor traffic. Then we implement one of these relays in our own computer and then we will analyze the traffic that passes through it with the Wireshark software.

To finish we tested the different access options to the Tor network through our Bridge Relay, both from different browsers, as well as from a Smartphone.

Índice general

Índice de figuras

Índice de tablas

Acrónimos

AES	-	Advanced Encryption Standard
CPU	-	Central Processing Unit
DARPA	-	Defense Advanced Research Projects Agency
DH	-	Diffie Hellman
DHCP	-	Dynamic Host Configuration Protocol
DNS	-	Domain Name System
DPI	-	Deep packet inspection
FTE	-	Format-Transforming Encryption
HTTP	-	HyperText Transfer Protocol
HTTPS	-	Hypertext Transfer Protocol Secure
IP	-	Internet Protocol
IPv4	-	Internet Protocol version 4
IPv6	-	Internet Protocol version 6
ISP	-	Internet service provider
LAN	-	Local Area Network
MAC	-	Media Access Control
NAT	-	Network Address Translation
OP	-	Onion Proxy
OR	-	Onion Router
RAM	-	Random-access memory
RSA	-	Rivest–Shamir–Adleman
SATA	-	Serial Advanced Technology Attachment
SHA	-	Secure Hash Algorithms
SSL	-	Secure Service Layer
TCP	-	Transmission Control Protocol
TLS	-	Transport Layer Security
TOR	-	The Onion Router
URL	-	Uniform Resource Locator
USB	-	Universal Serial Bus
VGA	-	Video Graphics Array
WiFi	-	Wireless Fidelity
WWW	-	World Wide Web

Palabras clave

-Red Tor

-Internet

-Deep Web

-Bridge Relay

-Anonimato

-Censura

-Tráfico Tor

-Transportes conectables

-Tor Browser

-Tor Proxy

1 Introducción

En la actualidad, la tecnología está adentrada en todos los ámbitos de la vida de una persona de nuestra sociedad.

La sociedad avanza y toda persona vive desde sus inicios rodeada de una gran cantidad de dispositivos conectados a la red que nos permiten realizar tareas de una forma más ágil, cómoda y eficaz.

La nuestra es una sociedad red, es decir, una sociedad construida en torno a redes personales y corporativas operadas por redes digitales que se comunican a través de Internet. Y como las redes son globales y no conocen límites, la sociedad red es una sociedad de redes globales. Esta estructura social propia de este momento histórico es el resultado de la interacción entre el paradigma tecnológico emergente basado en la revolución digital y determinados cambios socioculturales. **[1]**

La última colección de informes 2019 de We Are Social y Hootsuite en 2019 revela que los usuarios de Internet están creciendo en un promedio de más de un millón de usuarios nuevos cada día desde enero de 2018, hasta asegurar que el mundo cuenta con 4.388 millones de usuarios de Internet, un 57% de la población mundial. **[2]**

Figura 1.1 Estadísticas de Internet

En la figura 1.1. Estadísticas de Internet vemos que cada vez aumentan los usuarios de Internet. Podemos encontrar actividades muy variadas, las cuales precisan de una conexión con internet para poder ser realizadas. Hoy en día ya no existe un sector que no trabaje apoyándose en la red.

A la vez que los usuarios crecen también crece su inseguridad. Cada vez se cuestionan más la seguridad de su información, les preocupa más su privacidad y proteger su identidad.

A partir de estas inseguridades nacen herramientas para dar privacidad y anonimato en la red como puede ser la red Tor, sobre la que profundizaremos en este proyecto.

1.1 Motivación

La motivación principal para el desarrollo de este proyecto es dar a conocer un poco a la red Tor, centrándonos un poco más los Bridge Relays, uno de los diferentes tipos de relays que tiene la red Tor y su principal herramienta contra la censura, explicar cómo poder conseguir uno, como funciona y como utilizarlo. Quizás en un futuro nos encontremos con una censura de Internet y de acceso a la red Tor y no podamos conectarnos a los relays de entrada convencionales de esta red. En esta situación un bridge relay nos pueda servir de gran ayuda.

Otra de las motivaciones es analizar un poco los protocolos que utilizan estos bridge relays para camuflar su tráfico, un tema del que hay muy poco escrito, quizás porque cuanto menos se publique acerca de estos protocolos más difícil lo tienen los censores de la red Tor para descubrir el tráfico Tor.

1.2 Objetivos

Como primer objetivo tenemos el aprendizaje teórico sobre la red Tor, centrándonos en qué es, su historia, el tipo de usuarios que suele utilizar esta red y con qué fines, los componentes que tiene y el funcionamiento de la red.

Una vez tengamos claros los primeros conceptos sobre la red Tor, pasamos al principal objetivo, los bridge relays, con el objetivo de saber qué son, aprender cómo funcionan y qué protocolos utilizan.

En la parte más práctica, el objetivo será la implementación, configuración y mantenimiento de un bridge relay sobre el sistema operativo Debian, en un equipo propio y analizar el tráfico que pasa por él.

Otro objetivo práctico será aprender a acceder a la red Tor a través de un Bridge relay, desde diferentes herramientas como puede ser el navegador propio de Tor, otros navegadores más comunes, como Google Chrome o Firefox e incluso desde un Smartphone.

2 Estado del arte

Con la aparición de las nuevas tecnologías y servicios se ha experimentado un gran cambio en la sociedad haciendo que los usuarios estén constantemente conectados a Internet. Esto nos ha aportado grandes ventajas en el día a día, pero también nos encontramos con algunos problemas importantes.

La información que consultamos en Internet está toda indexada en buscadores, por lo tanto, se está produciendo un seguimiento y análisis de datos por estos buscadores. Para evitar esto se buscan alternativas que nos aporten ese anonimato y privacidad que no se tiene en la red. Una de estas alternativas puede ser la Deep Web o Web profunda.

La Deep Web, o la internet oculta, es el contenido web que no está en la superficie de Internet. El contenido de la superficie corresponde a las páginas que pueden ser indexadas por los motores de búsqueda que utilizamos normalmente y contienen enlaces a otros sitios. En cambio, en la Deep Web se encuentran páginas que no contienen enlaces a otros sitios ni tienen enlaces provenientes de otros y no pueden ser indexados por los motores de búsqueda. La única manera de acceder a la Deep Web es saber la URL del sitio que se desea visitar. **[3]**

Se calcula que la Deep Web es actualmente 400 a 500 veces más grande que la superficie de Internet, contiene 7500 terabytes de información y 550 billones de documentos comparado con 19 terabytes y un billón de documentos que son propios de la superficie. **[4]**

Para representar metafóricamente Internet podemos utilizar la imagen de un Iceberg.

Figura 2.2 Representación metafórica de Internet

La primera parte es accesible para todos los usuarios corrientes de la red y se correspondería con el internet “visible” englobando únicamente un 4% del volumen total de la red, lo que conocemos como “Surface Web”.

La segunda en cambio es la parte profunda que engloba un porcentaje del 96% sobre el tamaño total de internet y es la zona que necesita de unas condiciones especiales para su acceso.

El acceso a esta parte de la red no se consigue directamente desde la Web Superficial, sino que es necesario un software específico, como es por ejemplo Tor, que nos da acceso a esta parte de la red de forma anónima.

Tor va a aportar al usuario un anonimato y privacidad en su navegación mediante un enrutamiento especial llamado “enrutamiento cebolla” estructurado en capas de cifrado. Tanto los usuarios como los servidores de la red van a conseguir anonimidad.

Por lo tanto, Tor va a ser una herramienta utilizada por los usuarios que quieran navegar por la Deep Web ya que necesitaran ocultar su identidad y también utilizada por los usuarios que quieran navegar por la parte visible, pero ocultando su identidad.

2.1 ¿Qué es Tor?

Tor es un proyecto cuyo objetivo principal es el desarrollo de una red de comunicaciones virtual distribuida de baja latencia y superpuesta sobre internet, que permite la comunicación anónima en Internet, es decir, oculta la dirección IP de sus usuarios (anonimato a nivel de red) y que, además mantiene la integridad y confidencialidad de los mensajes que circulan a través de la red ya que posee un mecanismo de encriptación muy sofisticado, impidiendo que un mensaje interceptado en el camino pueda ser entendido. **[5] [6]**

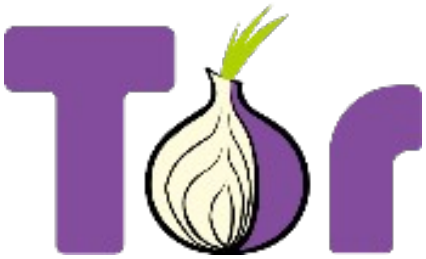


Figura 2.3 Logotipo “The Onion Router”

En la red Tor, un mensaje es encriptado múltiples veces antes de ser enviado a través de la red, por capas como las de una cebolla, y hace que los mensajes viajen desde el origen a su destino a través de una serie de nodos o relays que se encargan de decodificar parte del mismo, revelando el siguiente relay al que hay que mandar la información encriptada, hasta que llega al último relay, que se encarga de decodificar por completo el resto del mensaje y así poder enviar la información original al destinatario, sin revelar, o incluso conocer, la IP de origen.

El nombre de TOR viene de las siglas de The Onion Router (enrutador de cebolla en español) ya que Tor propone el uso de un encaminamiento de forma que los mensajes viajen desde el origen al destino a través de una serie de nodos o relays especiales llamados 'routers de cebolla'.

Para la consecución de estos objetivos se ha desarrollado un software libre específico. El sistema está diseñado con la flexibilidad necesaria para que pueda implementar mejoras, se despliegue en el mundo real y pueda resistir diferentes tipos de ataque. Sin embargo, tiene puntos débiles y no puede considerarse un sistema infalible.

También debemos dejar claro que Tor no es una red entre iguales (peer-to-peer) ya que por un lado están los usuarios de la red y por otro lado los relays encaminadores del tráfico y algunos de los cuales hacen una función de servicio de directorio.

La red funciona a partir de un conjunto de organizaciones e individuos que donan su ancho de banda y poder de procesamiento, convirtiéndose en relays o routers cebolla de esta red. Actualmente hay alrededor de 6000 ordenadores de voluntarios vinculados en la red Tor, que hacen las funciones de routers y servicios de directorio. **[6]**

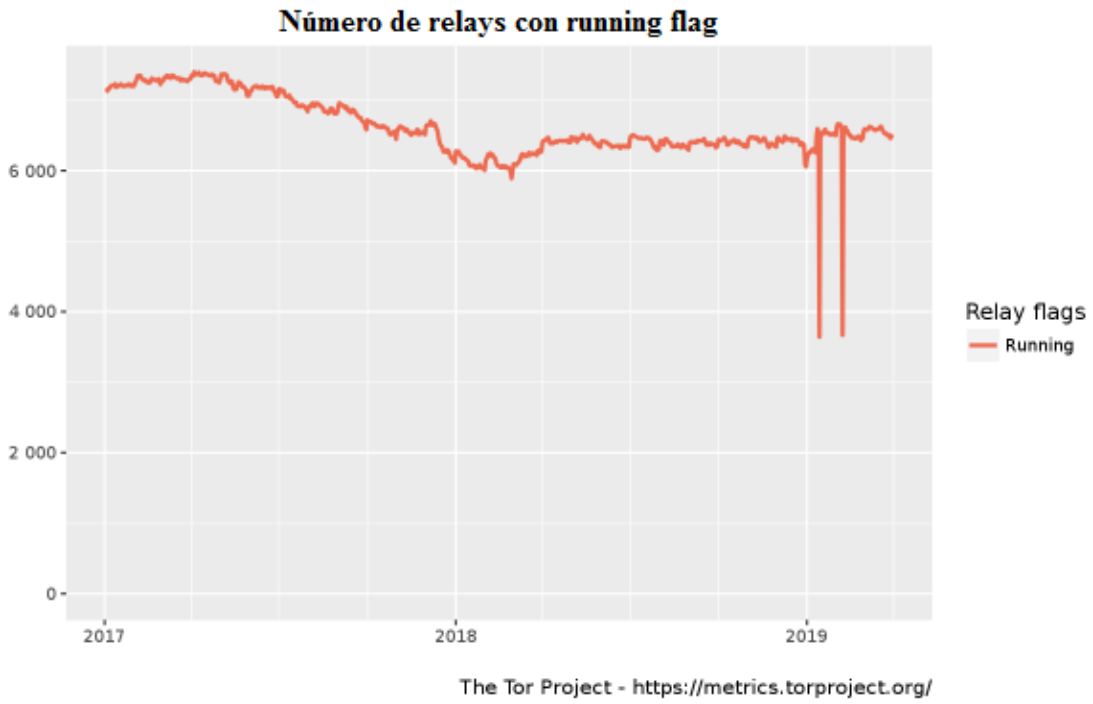


Figura 2.4 Número de relays activos de Tor

2.2 Historia de Tor

Los orígenes de Tor se remontan a 1995 cuando se inician los trabajos de investigación sobre enrutamiento por capas (Onion Routing) en la Oficina de Investigación Naval de la Marina de los Estados Unidos. La motivación principal fue la de proteger las comunicaciones del gobierno. A esta versión se le conoce como generación cero de Onion Router y fue desarrollada por Michael Reed, Paul Syverson y David Goldschlag.

En 1997, el proyecto consigue financiamiento de la Agencia de Investigación de Proyectos Avanzados de Defensa (DARPA) bajo el programa de Redes de Alta Confidencialidad. En el Simposio de Seguridad y Privacidad de la IEEE se publica el diseño de la primera generación de Onion Routing, que incluye mejoras como el número de rutas variable, la separación del proxy de navegación del router, introducción de políticas de seguridad para los relays de salida, módulo de criptografía separado de la aplicación, permitiendo que el mismo se ejecute en hardware especializado, entre otras mejoras.

En 1999 el desarrollo de Onion Routing es suspendido debido a la falta de fondos y a que los desarrolladores abandonaron el Laboratorio de Investigación.

En 2001 se reactiva el desarrollo gracias a los fondos de DARPA, esta vez bajo el programa de Redes Tolerantes a Fallas, con el objetivo de completar el código de la primera generación lo suficiente como para levantar una red de prueba y además agregar tolerancia a fallos y manejo de recursos.

En 2002 se abandona el código de la primera generación por ser antiguo. Se inicia el desarrollo de la segunda generación bajo la dirección de Roger Dingledine, Nick Mathewson y Paul Syverson.

En 2003, se aportan fondos desde la Oficina de Investigación Naval para el desarrollo y la implementación de la segunda generación. En octubre, la red Tor es desplegada y el código fuente se libera bajo licencia MIT. **[5]**

A finales del año 2004 pasó a ser patrocinado por la Electronic Frontier Foundation, la organización de defensa de las libertades en el mundo digital.

Desde noviembre de 2005 el proyecto Tor está en manos de The Tor Project una organización sin ánimo de lucro orientada a la investigación y la educación, dirigido por Roger Dingledine, radicada en Massachusetts y que ha sido financiada por distintas organizaciones. **[4]**

La lista actual de los patrocinadores activos en 2019 es la siguiente: **[7]**

- Decenas de miles de donaciones personales de personas como usted (2006-presente)
- Fundación Rose para las Comunidades y el Medio Ambiente (2017-2019)
- Mozilla (2016-2019)
- Fondo de tecnología abierta (2012-2019)
- Sida - Agencia Sueca de Cooperación para el Desarrollo Internacional (2018-2019)
- Media Democracy Fund (2016-2019)
- Handshake - La fundación del apretón de manos (2006-2019)
- National Science Foundation a través de la Universidad de Minnesota (2013-2019)
- National Science Foundation junto con Georgetown (2015-2019)
- National Science Foundation junto con Rochester Institute of Technology (2016-2019)
- Fastly (2016-2019)
- Departamento de Estado de los Estados Unidos, Oficina de Democracia, Derechos Humanos y Trabajo (2013-2019)
- DARPA a través de la Universidad de Pennsylvania (2018-2019)
- Instituto de servicios de museos y bibliotecas a través de la Universidad de Nueva York (2018-2019)
- Equipo Cymru (2009-2019)

Patrocinadores anteriores

- Torfox (2009)
- SRI Internacional (2011-2017)
- Tecnología Shinjiru (2009-2011)
- Reddit (2015)
- Omidyar Network Enzyme Grant (2006)
- Fundación Nacional de Ciencias a través de la Universidad Rice (2006-2007)
- Fundación Nacional de Ciencias junto con la Universidad de Princeton (2012-2018)
- Fundación Nacional de Ciencias junto con la Universidad de Illinois en Chicago (2016-2018)
- Fundación Nacional de Ciencia a través de la Universidad de Drexel (2009-2011)
- Laboratorio de Investigación Naval (2006-2010)
- Fundación NLnet (2008-2009)
- Fundación Nacional Cristiana (2010-2012), (2014)
- Internews Europe (2006-2008)
- Human Rights Watch (2007)
- Ministerio Federal de Relaciones Exteriores de Alemania (2015)
- Google (2008-2009)
- Google Summer of Code (2007-2014) y (2016-2017)
- La libertad de prensa de la Fundación (2014)
- La Fundación Ford (2013-2014)
- Hivos / The Digital Defenders Partnership (2014-2015)
- Fundación Electronic Frontier (2004- 2005)
- Desconectar (2014)
- Proyecto Cyber-TA (2006-2008)
- Bell Security Solutions Inc (2006)
- Junta de Gobernadores de Radiodifusión (2006-2013)
- Accede Ahora (2012)
- Internews (2008-2013)
- DARPA y ONR a través del Laboratorio de Investigación Naval (2001-2006)

En marzo de 2011, Tor recibió de la Free Software Foundation el premio para proyectos de beneficio social correspondiente a 2010 por «haber permitido que, aproximadamente, 36 millones de personas de todo el mundo, usando software libre, hayan experimentado libertad de acceso y de expresión en Internet manteniendo su privacidad y anonimato. Su red ha resultado crucial en los movimientos disidentes de Irán y Egipto». **[8]**

2.3 Usuarios de la red Tor

Tor fue originalmente diseñado, implementado y desplegado como un proyecto de enrutamiento de cebolla del Laboratorio de Investigación Naval. Fue desarrollado originalmente con la Marina de los EE. UU, con el propósito principal de proteger las comunicaciones gubernamentales. Hoy en día, se usa todos los días para una amplia variedad de propósitos por militares, periodistas, agentes del orden público, activistas y muchos otros.

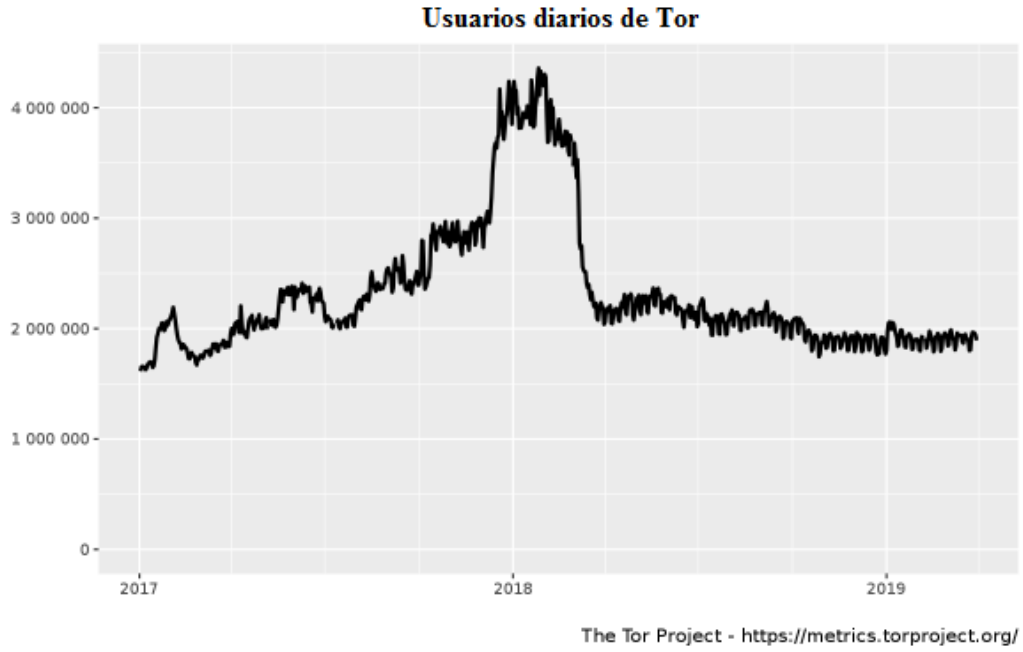


Figura 2.5 Número de usuarios al día de la Tor Network, excluyendo usuarios de bridge relay

Como podemos ver en la figura 2.4 el número de usuarios de Tor ha aumentado en los últimos años. Actualmente estamos sobre los 2.000.000 pero a finales de 2017 y principios de 2018 hemos llegado a superar los 4.000.000 de usuarios al día.

País	Promedio de usuarios diarios
Estados Unidos	371886 (17.64%)
Emiratos Árabes Unidos	289188 (13.72%)
Rusia	243432 (11.55%)
Alemania	155529 (7,38%)
Francia	113745 (5,40%)
Indonesia	82315 (3.90%)
Ucrania	80776 (3.83%)
Reino Unido	61812 (2.93%)
India	45784 (2,17%)
Países Bajos	41546 (1,97%)

Tabla 2.1 Top-10 países por usuario de Tor

La tabla 2.1 muestra los 10 principales países según la cantidad estimada del número de clientes que se conectan a la Tor Network. Estos números se derivan de las solicitudes de directorio contadas en las autoridades de directorio y las réplicas. **[9]**

Estos son algunos de los usos específicos de los usuarios que utilizan Tor:

La gente normal usa Tor

Protegen su privacidad de mercadólogos y ladrones de identidad. Los proveedores de servicios de Internet venden sus registros de navegación de Internet a los vendedores o cualquier otra persona que esté dispuesta a pagar por ellos. Un registro completo de cada sitio que visita, el texto de cada búsqueda que realiza, y la información de usuario e incluso la contraseña pueden ser parte de estos datos. Además de su ISP, los sitios web (y los motores de búsqueda) que visita tienen sus propios registros, que contienen la misma o más información.

Protegen sus comunicaciones de corporaciones irresponsables. A través de Internet, Tor está siendo recomendado a personas recientemente preocupadas por su privacidad frente a las crecientes infracciones y traiciones de datos privados. Desde las cintas de copia de seguridad perdidas, hasta la entrega de los datos a los investigadores, sus datos a menudo no están bien protegidos por aquellos en los que se supone que deben confiar para mantenerlos seguros.

Proteger información de identificación personal en línea y su ubicación simplemente ocultando su dirección IP. Cada vez más, las direcciones IP se pueden asignar a una ciudad o incluso a una calle, y pueden revelar otra información sobre cómo se está conectado a Internet.

Eludir la censura. Investigan temas delicados o simplemente acceder a Facebook o YouTube. Hay una gran cantidad de información disponible en línea. Pero tal vez en su país, el acceso a esta información está detrás de un cortafuegos.

Los periodistas y su audiencia usan Tor

Multitud de reporteros sin fronteras han sido encarcelados por todo el mundo. Aconsejan a los periodistas, las fuentes, los blogueros y los disidentes que usen Tor para garantizar su privacidad y seguridad ya que su trabajo les dirige en muchas ocasiones a temas controvertidos sobre democracia, economía y religión.

Tor es parte de SecureDrop, un sistema de envío de documentos de forma segura destinado a comunicarse con fuentes anónimas. Muchas organizaciones de noticias usan SecureDrop.

Los ciudadanos y periodistas usan Tor para investigar propaganda estatal y puntos de vista opuestos, para presentar historias con medios no controlados por el Estado y para evitar arriesgarse a las consecuencias personales de la curiosidad intelectual.

Los agentes del orden usan Tor

Vigilancia en línea: Tor permite a los funcionarios navegar por sitios web y servicios cuestionables sin dejar pistas reveladoras. Si el administrador del sistema de un sitio de apuestas ilegales, por ejemplo, viera múltiples conexiones de direcciones IP gubernamentales o policiales en los registros de uso, las investigaciones podrían verse obstaculizadas.

Operaciones de Sting: Del mismo modo, el anonimato permite a los oficiales de la ley participar en operaciones "encubiertas" en línea. Independientemente de cuán buena sea la "credibilidad callejera" de un oficial encubierto, si las comunicaciones incluyen rangos de IP de direcciones de la policía, la tapadera es volada.

Líneas de información verdaderamente anónimas: si bien las líneas de información anónima en línea son populares, sin software de anonimato, son mucho menos útiles. Las fuentes sofisticadas comprenden que, aunque un nombre o dirección de correo electrónico no está adjunta a la información, los registros del servidor pueden identificarlos muy rápidamente. Como resultado, los sitios web de línea de punta que no fomentan el anonimato están limitando las fuentes de sus consejos.

Activistas y denunciantes usan Tor

Los activistas de derechos humanos usan Tor para reportar anónimamente los abusos de las zonas de peligro. Internacionalmente, los trabajadores de los derechos laborales usan Tor y otras formas de anonimato en línea y fuera de línea para organizar a los trabajadores de acuerdo con la Declaración Universal de los Derechos Humanos. Aunque están dentro de la ley, no significa que estén a salvo. Tor proporciona la capacidad de evitar la persecución mientras sigue elevando una voz.

En el este de Asia, algunos organizadores laborales usan el anonimato para revelar información sobre talleres clandestinos que producen bienes para los países occidentales y para organizar la mano de obra local.

Tor puede ayudar a los activistas a evitar la censura gubernamental o corporativa que obstaculiza la organización. En uno de esos casos, un ISP canadiense bloqueó el acceso a un sitio web sindical utilizado por sus propios empleados para ayudar a organizar una huelga.

Las personas de perfil alto y bajo usan Tor

Personas con un puesto de trabajo de alto nivel, normalmente pueden estar expuestas a otras personas, las cuales pueden juzgar sus creencias políticas o religiosas. Sin embargo, estas personas puede que no quieran permanecer en silencio sobre temas que les importan, pero pueden afectarle sobre su carrera profesional. Tor ayuda a sentirse seguro de que pueden expresar su opinión sin consecuencias para su función.

Las personas que viven en la pobreza o una clase baja a menudo no participan plenamente en la sociedad civil por miedo a la opinión de sus superiores, perder su trabajo… Tor da voz a muchas de estas personas en esta situación.

Los ejecutivos de negocios usan Tor

Se emplea Tor para el intercambio de información sobre violaciones de seguridad entre empresas. Así la empresa que informa a centros de información a los que pertenece, mantiene segura su dirección IP y da anonimato por si un atacante está escuchando esta información.

Ver su competencia: si trata de verificar los precios de un competidor, es posible que no encuentre información o información engañosa en su sitio web. Esto se debe a que su servidor web puede estar codificado para detectar conexiones de la competencia y bloquear o difundir

desinformación a su personal. Tor le permite a una empresa ver su sector tal como lo vería el público en general.

Mantener la confidencialidad de las estrategias: un banco de inversión, por ejemplo, podría no querer que los fisgones de la industria puedan rastrear qué sitios web están viendo sus analistas. La importancia estratégica de los patrones de tráfico, y la vulnerabilidad de la vigilancia de dichos datos, están comenzando a ser más ampliamente reconocidos en varias áreas del mundo de los negocios.

Rendición de cuentas: en una época en que la actividad corporativa irresponsable y no denunciada ha socavado negocios de miles de millones de dólares, un ejecutivo que ejerza una verdadera administración quiere que todo el personal se sienta libre de divulgar la malversación interna. Tor facilita la responsabilidad interna antes de que se convierta en una denuncia o problema.

Los militares usan Tor

Agentes de campo: no es difícil para los insurgentes monitorear el tráfico de Internet y descubrir todos los hoteles y otras ubicaciones desde las cuales las personas se conectan a servidores militares conocidos. Los agentes de campo militares desplegados fuera de casa usan Tor para enmascarar los sitios que visitan, protegiendo los intereses y operaciones militares, así como también protegiéndose de daños físicos.

Servicios de cebolla: cuando Internet fue diseñado por DARPA, su objetivo principal era poder facilitar comunicaciones distribuidas y sólidas en caso de huelgas locales. Sin embargo, algunas funciones deben estar centralizadas, como los sitios de comando y control. La naturaleza de los protocolos de Internet es revelar la ubicación geográfica de cualquier servidor accesible en línea. La capacidad de los servicios de cebolla de Tor permite que el comando y el control militares estén físicamente seguros contra el descubrimiento y el derribo.

Reunión de inteligencia: el personal militar necesita usar recursos electrónicos ejecutados y monitoreados por los insurgentes. No quieren que los registros del servidor web en un sitio web insurgente graben una dirección militar, lo que revela la vigilancia.

Los profesionales de TI usan Tor

Para verificar las reglas de firewall basadas en IP: Un firewall puede tener algunas políticas que solo permiten ciertas direcciones IP o rangos. Tor se puede utilizar para verificar esas configuraciones mediante el uso de un número de IP fuera del bloque de IP asignado de la empresa.

Para eludir sus propios sistemas de seguridad para actividades profesionales delicadas: por ejemplo, una empresa puede tener una política estricta con respecto al material que los empleados pueden ver en Internet. Una revisión de registro revela una posible violación. Tor se puede usar para verificar la información sin poner una excepción en los sistemas de seguridad corporativos.

Para volver a conectarse a los servicios implementados: un ingeniero de red puede usar Tor para volver a conectarse de forma remota a los servicios, sin la necesidad de una máquina externa y una cuenta de usuario, como parte de las pruebas operacionales.

Para evitar las interrupciones de la red del ISP: a veces, cuando un ISP tiene problemas de enrutamiento o DNS, Tor puede poner a disposición recursos de Internet cuando el ISP real no funciona correctamente. Esto puede ser invaluable en situaciones de crisis.

Hackers

El principal objetivo de un hacker en la red Tor es evolucionar recopilando información útil que se encuentra esperando en las profundidades donde no quieren dejar rastro de haber estado.

Además, los hackers éticos pueden emplear el anonimato de la red Tor para poner al descubierto a usuarios que realizan actividades ilegales o éticamente dudosas.

Ciberdelincuentes

Los delincuentes emplean la red Tor por la protección y seguridad que el ser anónimo en la red les brinda para realizar sus actividades ilegales. En sus negocios dentro de la red tienen la tranquilidad de no necesitar un contacto cara a cara con sus clientes para realizar la transacción lo que conlleva un mayor porcentaje de éxito a la vez que se reduce la posibilidad de ser capturado.

2.4 Funcionamiento de Tor

Tor es una red que implementa una técnica llamada Onion Routing (enrutado cebolla en castellano), diseñada con vistas a proteger las comunicaciones en la Marina de los Estados Unidos. La idea es cambiar el modo de enrutado tradicional de Internet para garantizar el anonimato y privacidad de los datos.

El enrutado tradicional que usamos para conectarnos a servidores en Internet es directo. Por ejemplo, si quieres leer el sitio web “El País”: tu ordenador se conecta de forma directa a los servidores de “El País”. La ruta es, a grandes rasgos, sencilla: de tu ordenador a tu router, de ahí a los enrutadores de tu ISP (proveedor de Internet) y después directos a los servidores de “El País”.

Fácil y sencillo, salvo por el hecho de que si alguien intercepta los paquetes de datos en un punto intermedio sabrá perfectamente de dónde vienen y a dónde van. Incluso aunque se cifren los datos de cada paquete (por ejemplo, visitando una página HTTPS) las cabeceras de este no se cifran, y los campos del remitente y destinatario (entre otros) siguen siendo visibles.

Ahí es donde entra el Onion Routing o enrutamiento de Cebolla. Consiste en enviar el paquete por un camino no directo, a través de varios nodos o relays, pero en realidad es algo más complejo que eso. **[5]**

Para entender bien el funcionamiento del enrutamiento de Cebolla lo podemos dividir en tres partes:

-La primera parte consiste en el establecimiento de la ruta a seguir para poder obtener o enviar un contenido.

-La segunda etapa será el proceso de cifrado de la totalidad de la información que queremos enviar u obtener.

-La tercera y última etapa consistirá en ir descifrando de forma progresiva la totalidad de información que queremos enviar u obtener.

2.4.1 Establecimiento e inicialización de la ruta

El primer paso que hace el software Tor en el proceso de establecimiento de la ruta es conectarse a internet para recoger información acerca de los relays disponibles de los servicios de directorio de Tor.

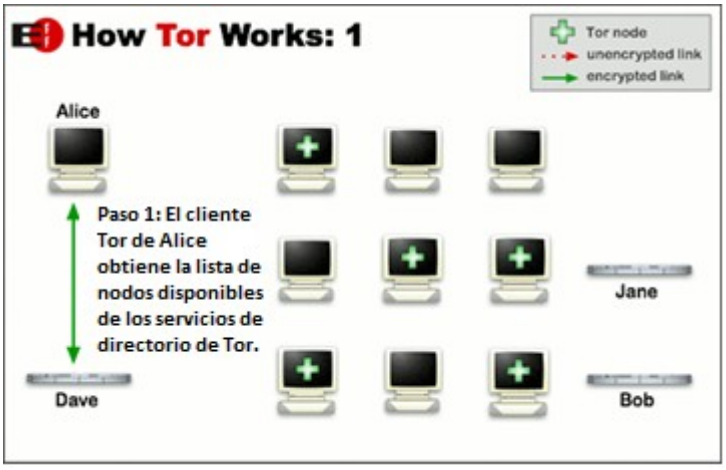


Figura 2.6 Cómo funciona Tor 1

A partir de la información obtenida y de su configuración el software Tor decide un circuito aleatorio por el que van a circular los paquetes. Por defecto el circuito tiene 3 relays, el entry guard que será el primer relay, el middle relay que será el segundo y el exit relay que será el último, antes de que el mensaje llegue a su destino. **[4]**

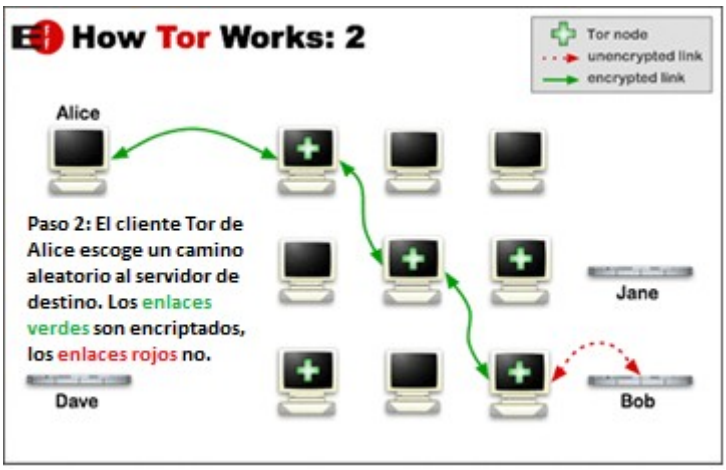


Figura 2.7 Cómo funciona Tor 2

El siguiente paso es establecer una conexión TLS con todos los relays del circuito:

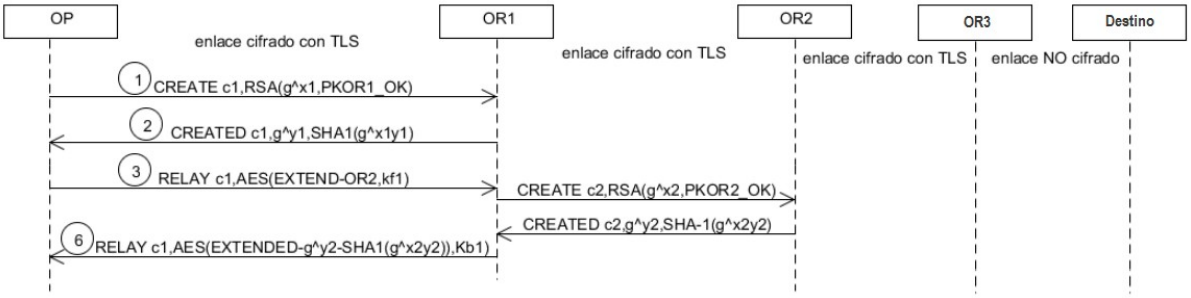


Figura 2.8 Intercambio de mensajes para establecimiento de circuito

- Conexión con el entry guard o relay de entrada (OR1):

A continuación, el software TOR (OP) se conectará de forma segura con el entry guard usando el protocolo TLS.

El software Tor inicia un intercambio de clave Diffie-Hellman con el OR1 enviando una celda de tipo CREATE.

El entry guard le devuelve una célula de tipo CREATED y de esta forma se obtiene una clave compartida Diffie-Hellman (g^x1y1). De esta clave se derivan dos claves simétricas, una para cifrar tráfico en cada sentido:

kf1 (forward key). - Se usará en la comunicación OP->OR1.

kb1 (backward key). - Se usará en la comunicación OR1->OP.

- Conexión con el middle relay o relay intermedio (OR2):

A continuación, el OP envía una petición a el OR1 para extender el circuito mediante una célula de tipo RELAY_EXTEND. Esta célula le indica al OR1 cuál va a ser el nuevo relay OR2, su puerto y también contiene estructuras de datos que le tendrá que reenviar el OR1 al OR2 y que permitirán establecer un protocolo D-H de intercambio de claves entre el OP y OR2 a través de OR1 sin que el OR1 se entere de nada.

El OR1 realiza el procedimiento Diffie-Hellman con OR2. Observar que ahora las células CREATE/CREATED se intercambian entre el OR1 y el OR2.

El OR1 manda al OP una célula de tipo RELAY_EXTENDED para mandarle lo que ha respondido el OR2 y así el OP tiene conocimiento de la clave D-H. La información está cifrada de tal forma que el relay OR1 no puede acceder la información que se están intercambiando indirectamente el OP y OR2. De esta forma se establecen la clave compartida Diffie-Hellman (g^x2y2) de la que derivan las nuevas claves simétricas: kf2 y kb2.

Kf2 (forward key). - Se usará en la comunicación OP->OR2.

Kb2 (backward key). - Se usará en la comunicación OR2->OP.

- Conexión con el exit relay o relay de salida:

A continuación, el OP envía una petición a el OR2 para extender el circuito mediante una célula de tipo RELAY_EXTEND cifrado con kf1 y kf2 de tal forma que el relay OR1 no puede acceder la información que se están intercambiando indirectamente el OP y OR2.

Esta célula le indica al OR2 cuál va a ser el nuevo relay OR3, su puerto y también contiene estructuras de datos que le tendrá que reenviar el OR2 al OR3 y que permitirán establecer un protocolo D-H de intercambio de claves entre el OP y OR3 a través de OR1 y OR2 sin se enteren de nada.

El OR2 realiza el procedimiento Diffie-Hellman con OR3. Observar que ahora las células CREATE/CREATED se intercambian entre el OR2 y el OR3.

El OR2 manda al OP una célula RELAY_EXTENDED para mandarle lo que ha respondido el OR3 y así el OP tiene conocimiento de la clave D-H. La información está cifrada de tal forma que el relay OR2 y OR1 no pueden acceder la información que se están intercambiando indirectamente el OP y OR3. De esta forma se establecen la clave compartida Diffie-Hellman (g^x3y3) de la que derivan las nuevas claves simétricas: kf3 y kb3.

Kf3 (forward key). - Se usará en la comunicación OP->OR3.

Kb3 (backward key). - Se usará en la comunicación OR3->OP.

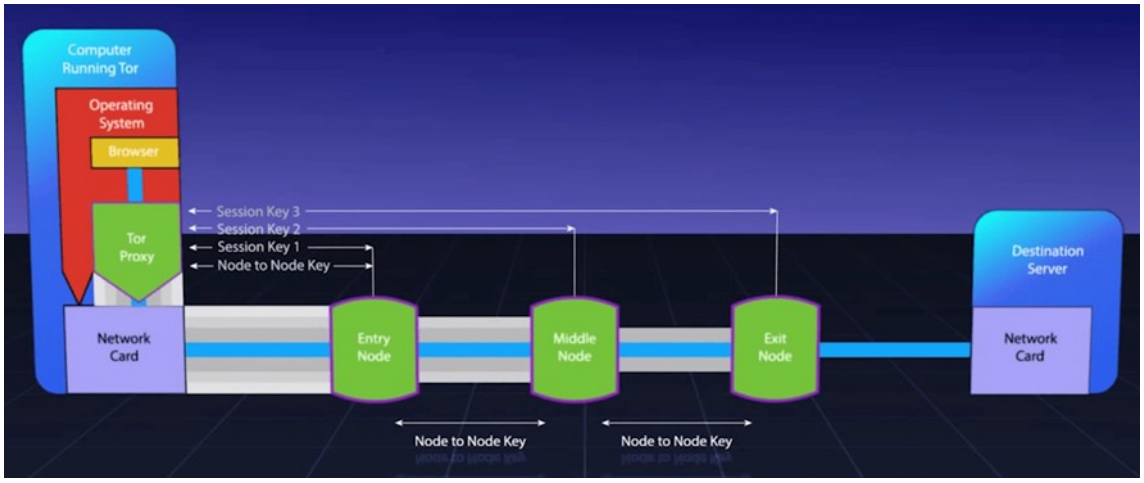


Figura 2.9 Representación gráfica del establecimiento de la ruta

Como medida de seguridad cada 10 minutos se establecerá una nueva ruta de conexión de forma automática. En los archivos de configuración de TOR podremos modificar este tiempo o si lo precisamos también podemos forzar que se establezca una ruta de forma manual.

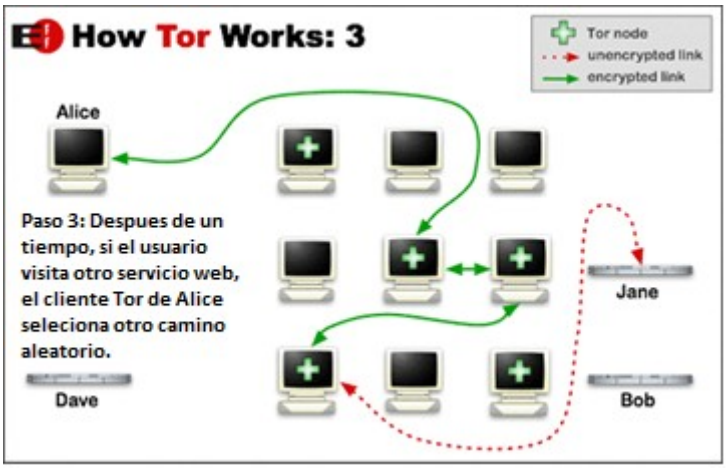


Figura 2.10 Cómo funciona Tor 3

Una vez definida la ruta ya podemos pasar a ver el procedimiento de cifrado usado por TOR.

2.4.1.1 Células utilizadas en el establecimiento de circuito

Célula CREATE

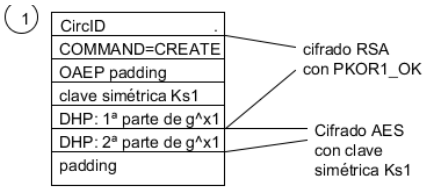


Figura 2.11 Establecimiento del circuito: Célula CREATE

Sobre el formato de la célula CREATE cabe señalar que el cifrado de g^x se hace de forma híbrida (con RSA se cifra una clave de sesión y parte de g^x, con la clave de sesión se cifra el resto de g^x) para permitir que en una sola célula podamos hacer el intercambio del g^x completo.

Célula CREATED

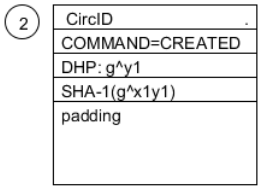


Figura 2.12 Establecimiento del circuito: Célula CREATED

En la célula CREATED se envía g^y y el valor resumen SHA-1 de la clave establecida g^xy para poder hacer una verificación de que ambos extremos de la comunicación comparten la misma clave.

Célula RELAY_EXTEND

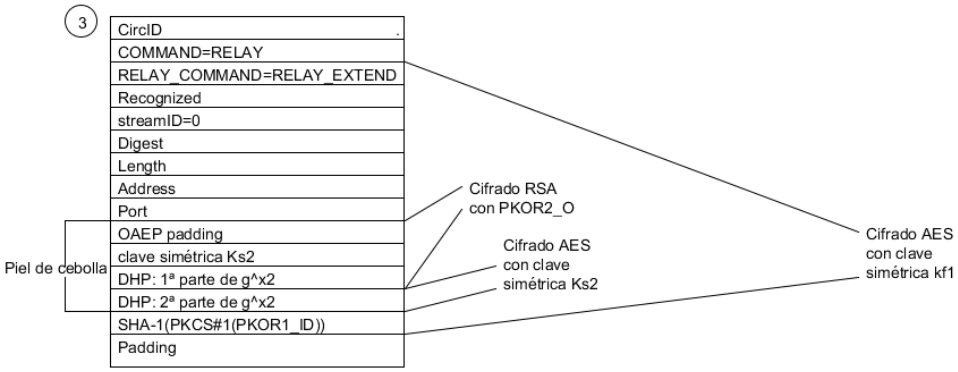


Figura 2.13 Establecimiento del circuito: Célula RELAY_EXTEND

La célula va cifrada por las claves simétricas kf que descifrada cada OR. Cuando el OR descifra con su clave AES obtiene distintos campos y entre ellos obtiene la siguiente piel de cebolla que puede utilizar directamente en la célula CREATE que tiene que construir para enviársela al siguiente OR.

En la célula se envía g^x, se hace de forma híbrida (con RSA se cifra una clave de sesión y parte de g^x, con la clave de sesión se cifra el resto de g^x) para permitir que en una sola célula podamos hacer el intercambio del g^x completo.

El campo SHA-1(PCKS#1(PKOR1_ID)) es el hash SHA-1 del PKCS#1 de la clave de identidad (PKOR_ID) del próximo OR. Esto permite prevenir cierto tipo de ataques man-in-the-middle.

Observar que cuando el OR1 descifra con su clave AES obtiene distintos campos y entre ellos obtiene la siguiente piel de cebolla que puede utilizar directamente en la célula CREATE que tiene que construir para enviársela al OR2.

Célula RELAY_EXTENDED

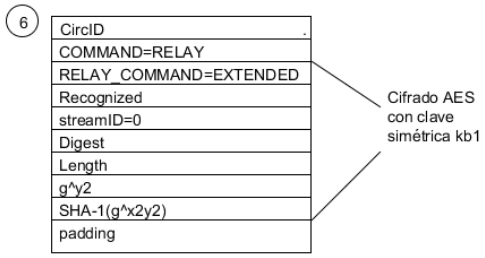


Figura 2.14 Establecimiento del circuito: Célula RELAY_EXTENDED

La célula RELAY_EXTENDED consiste en la comunicación de la respuesta al protocolo Diffie-Hellman que hace el OR al OP pasando por los OR intermedios sin que estos puedan inferir información alguna.

2.4.2 Proceso de cifrado

Una vez establecida la ruta y obtenidas las claves simétricas de cifrado, cifraremos la totalidad de contenido de nuestra petición.

La totalidad del contenido de nuestra petición estará cifrado mediante criptografía asimétrica. Su funcionamiento es el siguiente:

Antes de entrar en el relay de entrada, la totalidad de nuestra información se cifra por capas mediante el protocolo de cifrado AES-128 del siguiente modo:

1. Se usa la clave pública del exit relay (Kf3) para cifrar la totalidad de contenido de nuestra petición. De esta forma añadimos una capa de cifrado y aseguramos que únicamente el último relay podrá descifrar nuestro mensaje.
2. Al mismo tiempo se usa la clave pública del middle relay (kf2) y de este modo se aplica una segunda capa de cifrado a nuestra petición.
3. Finalmente se usa la clave pública del entry guard (kf1) para añadir una tercera capa de cifrado a nuestra petición.

Una vez finalizada la etapa de cifrado nuestra petición entra en el entry guard y empezará la transmisión y descifrado de nuestra petición.

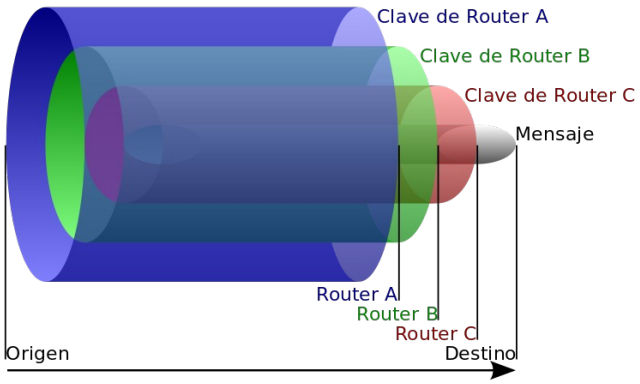


Figura 2.15 Capas de cifrado mensaje Tor

2.4.3 Proceso de descifrado

Una vez seleccionada la ruta y cifrada la totalidad de nuestra información mediante el protocolo AES, el cliente Tor envía el mensaje cifrado a través de los tres relays de la ruta de Tor y cada relay ira descifrando capa a capa el mensaje mediante el protocolo AES:

1. La totalidad de información de nuestra petición y/o mensaje entra dentro del entry guard. Usando la clave privada del entry guard quitaremos la primera de las 3 capas de

cifrado. Como el mensaje aún tiene dos capas de cifrado será completamente imposible que el relay inicial pueda ver el contenido.

2. Seguidamente nuestra petición y/o mensaje se dirigirá al middle relay. Allí usaremos la clave privada del middle relay para quitar la segunda de las 3 capas de cifrado.

3. Para finalizar, el exit relay usará su clave privada para quitar la última de las capas de cifrado de nuestra petición. Una vez nuestro mensaje está sin cifrar se procederá a la entrega de nuestra petición al servidor, y el servidor nos dará respuesta repitiendo de nuevo el procedimiento que acabamos de describir.

Figura 2.16 Representación descifrado TOR

Como conclusión, podemos decir que el Onion Routing nos proporciona más privacidad que el enrutado normal y corriente. Exceptuando el primer y último relay, nadie sabe de dónde viene o a dónde va la información que enviamos. El mensaje va encapsulado en muchas capas encriptadas, lo cual nos proporciona un extra en la privacidad buscada.

2.4.4 Seguridad de Tor

A continuación, veremos los principales protocolos que hacen a Tor una red tan segura como es.

TLS

Tor transmite sus datos a través de una serie de relays que se comunican mediante el protocolo TLS sobre TCP/IP manteniendo así secreta e íntegra, sin modificaciones externas, la información desde un relay a otro.

TLS es un protocolo mediante el cual se establece una conexión segura por medio de un canal cifrado entre el cliente y servidor. Así el intercambio de información se realiza en un entorno seguro y libre de ataques. **[10]**

El protocolo TLS se basa en tres fases básicas:

Negociación: Los dos extremos de la comunicación negocian que algoritmos criptográficos utilizarán para autenticarse y cifrar la información.

Autenticación y Claves: Los extremos se autentican mediante certificados digitales e intercambian las claves para el cifrado, según la negociación.

Transmisión Segura: los extremos pueden iniciar el tráfico de información cifrada y autentica.

Tor, para establecer las conexiones TLS, usa TLS/SSLv3. Todos los OR y OP tienen que soportar SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA y deberían tener disponible TLS_DHE_RSA_WITH_AES_128_CBC_SHA.

Los OP para comunicarse con los OR pueden usar:

TLS_DHE_RSA_WITH_AES_256_CBC_SHA

TLS_DHE_RSA_WITH_AES_128_CBC_SHA,

SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA

SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA

Diffie- Hellman

Tor usa el protocolo Diffie-Hellman para establecimiento de claves simétricas para el posterior cifrado de los datos.

Diffie-Hellman es una forma de establecer un secreto compartido entre dos partes en un canal inseguro.

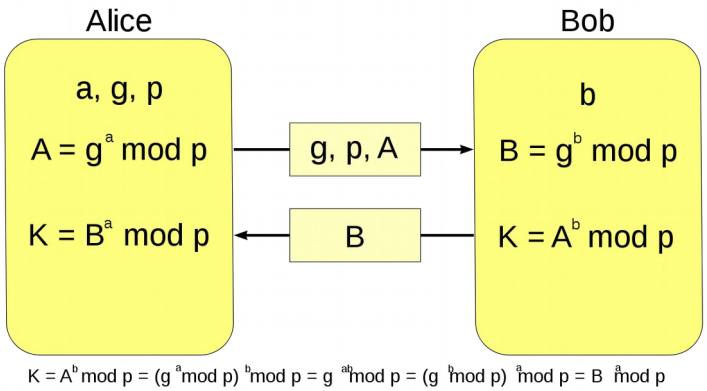


Figura 2.17 Funcionamiento Diffie-Hellman

Para comenzar, las dos partes deciden públicamente dos números primos, g y p.

Una de las partes decide secretamente otro número, a. y la otra decide en secreto sobre un número, b. Ninguna de las dos partes envía estos números, se guardan para sí mismos. La primera parte realiza un cálculo, g ^ a mod p, lo llamaremos A. La segunda parte entonces realiza g ^ b mod p que llamaremos B.

Las dos partes comparten entre ella A y B. Un observador ahora podría tener 4 números, A, B, g y p, pero no a o b. Por último, la primera parte toma la B y realiza B ^ a mod p. De manera similar, la segunda parte toma la A y realiza la modulación A ^ b mod p. Esto da como resultado el mismo número, es decir, B ^ a mod n = A ^ b mod p. Ahora tienen un número compartido K del que se derivan las claves kf y kb y el observador no lo puede averiguar. **[11]**

Tor usa Diffie-Hellman con g=2 y para p usamos el primo seguro de 1024 bits obtenido con valor hexadecimal:

FFFFFFFFFFFFFFFFC90FDAA22168C234C4C6628B80DC1CD129024E088A67CC74020BBEA63B139B22514A0879
8E3404DDEF9519B3CD3A431B302B0A6DF25F14374FE1356D6D51C245E485B576625E7EC6F44C42E9A637ED6B
0BFF5CB6F406B7EDEE386BFB5A899FA5AE9F24117C4B1FE649286651ECE65381FFFFFFFFFFFFFFFF

AES

El algoritmo principal empleado a la hora de encriptar los datos en Tor es AES (Advanced Encryption Standard) con claves de 128 bits, el algoritmo simétrico más utilizado hoy en día.

AES es un cifrador en bloque de criptografía simétrica, es decir, trabaja cifrando y descifrando bloque a bloque, utilizando la misma clave privada para ambos procesos. **[12]**

Según el estándar, divide los datos de entrada en bloques de 4 palabras de 32 bits, es decir, 4 × 32 = 128 bits.

En cuanto a la longitud de clave el algoritmo trabaja con longitudes de 128 (4 × 32), 192 (6 × 32) o 256 (8 × 32) bits, pero en nuestro caso Tor utiliza 128.

Si observamos la figura 2.17, veremos que, tanto el texto claro como el texto cifrado se dividen en bloques de 128 bits.

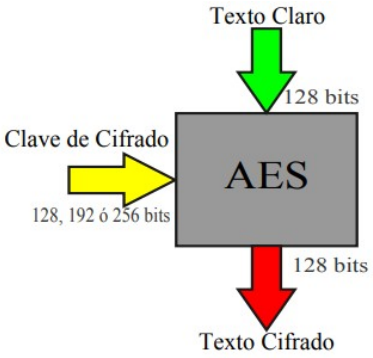


Figura 2.18 Cifrador AES

Básicamente, el cifrador aplica al Estado cuatro operaciones durante un número determinado de rondas.

Dicho número de rondas (Nr) viene definido por la longitud de clave utilizada, siendo Nr = 10 para una longitud de clave de 128 bits, Nr = 12 para 192 bits y Nr = 14 para 256 bits.

Las cuatro operaciones realizadas en el cifrado son denominadas:

SubBytes.

ShiftRows.

MixColumns.

AddRoundKey.

En la figura 2.18 se explica cómo se distribuyen las operaciones realizadas en el cifrado a lo largo de las 10 rondas necesarias para una clave de 128 bits.

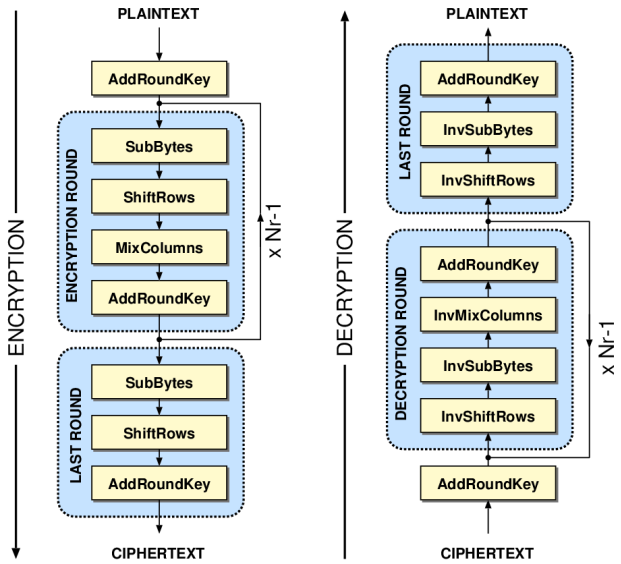


Figura 2.19 Proceso de cifrado AES

2.5 Tipos de Relays

Debido a la forma en que funciona la red Tor, no todos los relays que la conforman son iguales. Dependiendo de las características y configuración, un relay puede cumplir algunas funciones particulares.

Tenemos cinco tipos de relays Tor: Entry Guard, Middle Relays, Exit Relays, Bridge Relays y Directory Authorities.

Para una mayor seguridad, todo tráfico Tor pasa por al menos tres relays antes de alcanzar su destino, un Entry Guard o Bridge, un Middle Relay y un Exit Relay.

Entry Guard

Un entry guard es el primer relay utilizado en los circuitos de Tor. Es una posición privilegiada porque verá la IP real de los usuarios de Tor. Para reducir el potencial de ataques, no es fácil convertirse en un entry guard.

Para conseguir un entry guard deberemos correr un middle relay y las autoridades del directorio de Tor son las encargadas de dar a un middle relay la bandera de Guard Node si cumple tres características: "ancho de banda" (necesitan tener un peso de consenso lo suficientemente grande), "tiempo de actividad fraccional ponderado" (necesitan estar trabajando la mayor parte del tiempo) y "tiempo conocido" (para hacer los ataques más caros, no queremos dar la bandera de Entry Guard a relays que han existido por poco tiempo). **[13]**

Middle relay

El middle relay ocupa la segunda posición en este camino. Los dos primeros relays, los middle y los entry relays, tienen la función de recibir tráfico y pasarlo a otro relay, se suman a la velocidad y robustez de la red Tor sin hacer que el propietario del relay parezca el origen del tráfico. Los middle relays anuncian su presencia al resto de la red Tor, para que cualquier usuario Tor pueda conectarse a ellos. Incluso si un usuario malintencionado emplea la red Tor para hacer algo ilegal, la dirección IP de un middle relay no se mostrará como la fuente del tráfico. Eso significa que un middle relay generalmente es seguro para ejecutar en su hogar, junto con otros servicios, o en una computadora con sus archivos personales. **[16]**

Exit relay

Un exit relay es el relay final por el que pasa el tráfico Tor antes de que llegue a su destino. Los exit relay también anuncian su presencia a toda la red Tor, por lo que pueden ser utilizados por cualquier usuario de Tor. Debido a que el tráfico Tor sale a través de estos relays, la dirección IP del exit relay se interpreta como la fuente del tráfico. Si un usuario malintencionado emplea la red Tor para hacer algo que pueda ser objetable o ilegal, el exit relay puede ser el culpable. Las personas que administran los exit relays deben estar preparadas para tratar las quejas, los avisos de eliminación de derechos de autor y la posibilidad de que sus servidores puedan atraer la atención de las agencias encargadas de hacer cumplir la ley. Si no está preparado para lidiar con posibles problemas como este, es posible que no desee ejecutar un exit relay. Se recomienda que un exit relay se opere en una máquina dedicada. **[14]**

Si queremos un exit relay debemos configurar un middle relay y cambiarle las políticas de salida.

Bridge relay

Los bridge relays, también llamados puentes de Tor, son puntos alternativos de entrada a la red de Tor por lo que harían la función de un entry guard en un circuito Tor.

A diferencia del resto de relays Tor, los bridges no están listados en los servicios de directorio de Tor. **[15]**

Directory Authorities

Los Directory Authorities o servicios de directorio son relays que publican una base de datos que asocia a cada relay una serie de información. Esta información es accesible a todos los relays y a todos los usuarios finales y la usan para tener un conocimiento de la red. Si se tienen pocos servidores de directorio se corre el riesgo tener un punto cuyo fallo puede ocasionar el fallo del sistema completo. Por motivos de backup y de latencia, los relays que dan el servicio de directorio mantienen duplicada la información pasándosela de unos a otros. Hay una serie de relays principales (autoridades de directorio) y luego hay otros secundarios que hacen de caches y backup (directory caches). Los servidores de directorio son en realidad un grupo establecido de relays confiables. Para dar fiabilidad a la información que da el servicio de directorio, las entradas son protegidas criptográficamente con firmas y sólo la información que proviene de relays aprobados será publicada en la base de datos. Por tanto, todo relay nuevo tiene que ser previamente aprobado y de esta forma se evitan ataques en los que alguien añade muchos relays no confiables.

Cuando un relay se arranca, recolecta un conjunto de datos que lo describen a él, a su modo de funcionamiento y capacidades. Ejemplos de este tipo de atributos son la dirección IP, nombre amigable para el usuario, versión del software TOR, sistema operativo, clave pública, exit policies…Toda esta información se publica a través del servicio de directorio. **[17]**

La siguiente figura 2.19 muestra la cantidad de relays en ejecución que han tenido ciertas banderas asignadas por las autoridades del directorio Tor. Podemos ver que, actualmente, el número de entry guards está cerca de los 3.000, en cambio los exit relays, que son los relays que parecen el origen del tráfico, están un poco por debajo de los 1.000. El resto de relays son middle relays y servicios de directorio. Los bridge relays no están incluidos en esta figura ya que no están listados en los servicios de directorio de donde se obtienen estos datos. **[18]**

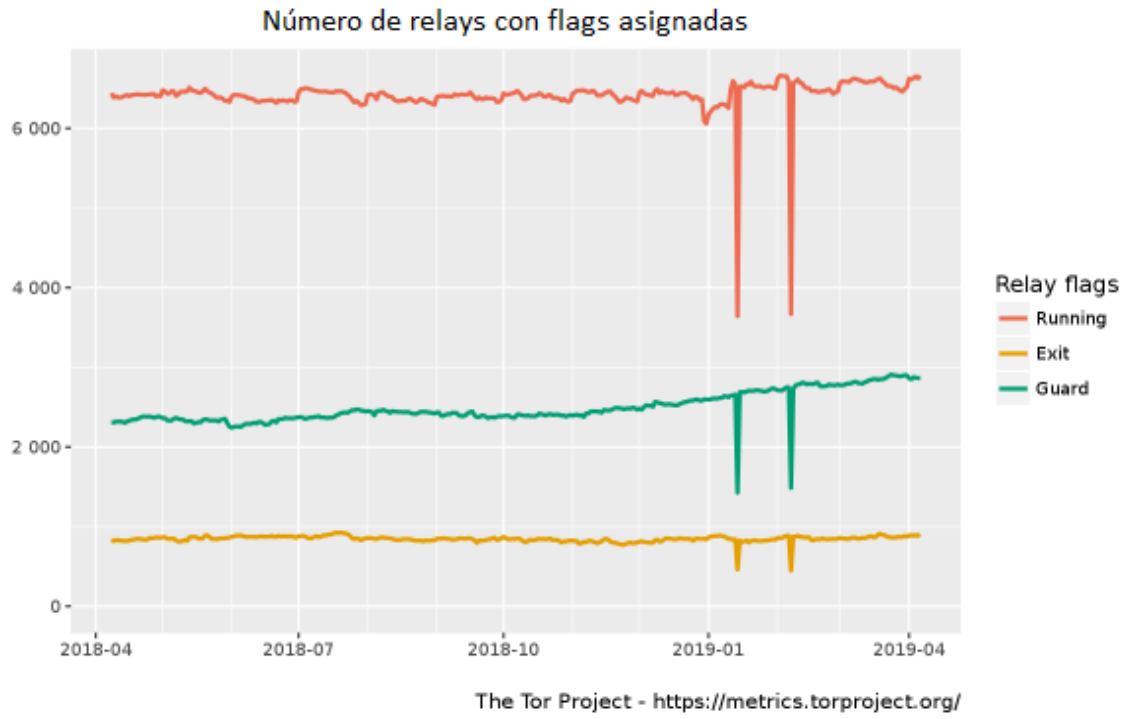


Figura 2.20 Número de relays con flags asignadas

2.6 Censura de Tor

Censura de Internet

El bloqueo y la censura en Internet es algo cada vez más predominante por desgracia en la red. Mientras que gobiernos de países como China bloquean ciertas páginas web a nivel nacional, como es el caso de Facebook o Twitter, en países donde prima más la libertad de expresión como los europeos la responsabilidad del bloqueo acaba recayendo en los operadores.

Son diversos motivos los que llevan a que se censure un determinado contenido en un país. Principalmente suelen ser razones políticas en países con gobiernos opresores, como puede ser el caso de Arabia Saudí, que utiliza la religión como herramienta de opresión para censurar también aspectos sociales, como el contenido para adultos o la libertad de ideas. También encontramos motivos de seguridad nacional en países como Estados Unidos.

En Europa el principal motivo por el que se bloquea una página web es porque ésta pueda ofrecer contenido que ponga en peligro a la población o que fomente la realización de actividades ilegales, o que en esa página se distribuya contenido que esté bajo la protección de derechos de autor, es decir, páginas de torrent o descarga directa.

El bloqueo que realizan los operadores suele ser de tres tipos. Entre las opciones disponibles, se encuentra el bloqueo por DNS (la más débil), bloqueo de la dirección IP (o redirección de IP), o bloqueo de URL.

Tor contra la censura

Tor ciertamente puede ayudar a que la gente acceda a un sitio web desde lugares donde está bloqueado. La mayoría de las veces se logrará el acceso simplemente descargando el navegador Tor y luego usándolo para navegar hacia el sitio bloqueado.

Como ya sabemos, Tor, para conectarse a una página Web, sigue una ruta de tres relays, entry guard, middle relay y exit relay. Este exit relay será el que se conecte en realidad al sitio Web.

Por lo tanto, para tener acceso a una web a través de Tor, desde un lugar donde está bloqueado dicho sitio, necesitamos que se cumplan dos requisitos:

- Es necesario que nuestro ISP nos permita el acceso a la dirección IP del entry guard de la ruta Tor que se nos ha asignado.
- También es necesario que el exit relay, que se nos ha asignado, tenga acceso al sitio Web al que nos queremos conectar.

Si la ruta asignada no cumple esos dos requisitos, Tor tiene la opción de “Utilizar otro circuito para este sitio” donde se nos asignara otra ruta diferente.

Tor nos permite conectarnos a un sitio Web bloqueado porque transparentemente solo estamos manteniendo una conexión con una dirección IP que en principio será accesible desde nuestra red y no será visible que a través de ese relay estamos alcanzando un sitio Web bloqueado.

Censura de Tor

Mientras que Tor provee muchas propiedades de seguridad y privacidad, no todas las personas alrededor del mundo tienen el lujo de poder conectarse a esta red para utilizarla. Actualmente los gobiernos y autoridades no solo bloquean sitios web si no que intentan bloquear la propia red Tor.

Por defecto, Tor no oculta el hecho de que estás conectado a la red Tor, una red abierta en la que todos pueden tener acceso a la lista de relays. Esta transparencia de la red tiene muchos beneficios, pero también cosas negativas: Muchos gobiernos represivos y autoridades gubernamentales, simplemente, toman la lista de relays y bloquean sus direcciones IP.

La solución de Tor a este bloqueo es el uso de bridges, en lugar de Entry Guards, para conectarse a la red Tor. Los bridges son relays de Tor alternativos que no están publicados en las listas de Tor, por lo que para los censores es más difícil bloquearlos.

Lamentablemente algunos gobiernos y proveedores de servicio no solo bloquean las direcciones IP asociadas a la red Tor, sino que toman medidas un poco más avanzadas para monitorear la red y detectar tráfico de internet que corresponda a la red Tor.

Así, aunque no conozcan previamente las direcciones IP de los bridge relays de Tor, sí pueden bloquear estas conexiones, impidiendo finalmente que los usuarios puedan conectarse a la red.

Pero Tor también tiene solución para evadir la inspección de paquetes, los llamados pluggable transports o transportes conectables. Estos transportes conectables son distintos protocolos cuya función es disfrazar y ocultar la conexión Tor entre un cliente Tor y un bridge relay y hacerla parecer otro tipo de tráfico que probablemente e idealmente no genere alertas y no sea bloqueado por el proveedor del servicio de internet.

Son pluggable transports en plural porque existen varios métodos que se han generado y revisado para hacer pasar desapercibido el tráfico de Tor y, además, en caso de que alguno de ellos sea finalmente identificable por la inspección de paquetes, pueda ser reemplazado por otro método que aún sea efectivo contra el bloqueo.

La siguiente tabla 2.2 muestra los 10 primeros países por posibles eventos de censura en el año 2018, obtenidos a partir de un sistema de detección de censura basado en anomalías. **[19]**

Country	Downturns	Upturns
Kyrgyzstan	55	72
Mongolia	47	59
Armenia	38	51
Egypt	30	55
Albania	30	35
Guyana	27	34
East Timor	27	34
Georgia	25	20
Venezuela	24	44
China	24	33

Tabla 2.2 Top-10 países con posibles eventos de censura 2018

En conclusión, podemos ver que el proyecto Tor es un proyecto que se encuentra bastante actualizado y está en una constante “guerra” con los censores para conseguir nuevos métodos para luchar contra el bloqueo de la red.

En el siguiente apartado nos centraremos en los Tor Bridge o puentes de Tor, el elemento más importante contra la censura de la red Tor.

3 Bridge relay

3.1 ¿Qué es un bridge relay?

Los bridge relays, también llamados puentes de Tor, son puntos alternativos de entrada a la red Tor, sustituyen y hacen la función de un entry guard en un circuito de Tor.

Son los relays que más se diferencian del resto ya que son relays privados y no están anunciados públicamente a la red en los servicios de directorio de Tor (Directory Authorities). Al no haber una lista pública de todos los bridges, aunque el ISP del cliente filtre conexiones a los relays de la red Tor, este no podrá bloquear todos los bridges en línea. Por este motivo los bridges son herramientas esenciales contra censura y elusión en países que bloquean regularmente las direcciones IP de todos los relays de Tor.

Otra característica muy importante de los bridge relays es que permiten la opción de utilizar unos protocolos que camuflan el tráfico Tor entre el software Tor del cliente y el bridge relay, llamados transportes conectables. Aun así, debemos admitir que usar un bridge hace más difícil, pero no imposible, que tu proveedor de internet sepa que estás usando Tor.

Generalmente, un bridge relay es seguro para ejecutarlo en su hogar, en conjunto con otros servicios o en una computadora con sus archivos personales ya que no suele ser un relay muy amenazado contra ataques hacia la red Tor. **[20]**

La figura 3.1 muestra la cantidad de relays y bridges en ejecución en la red.

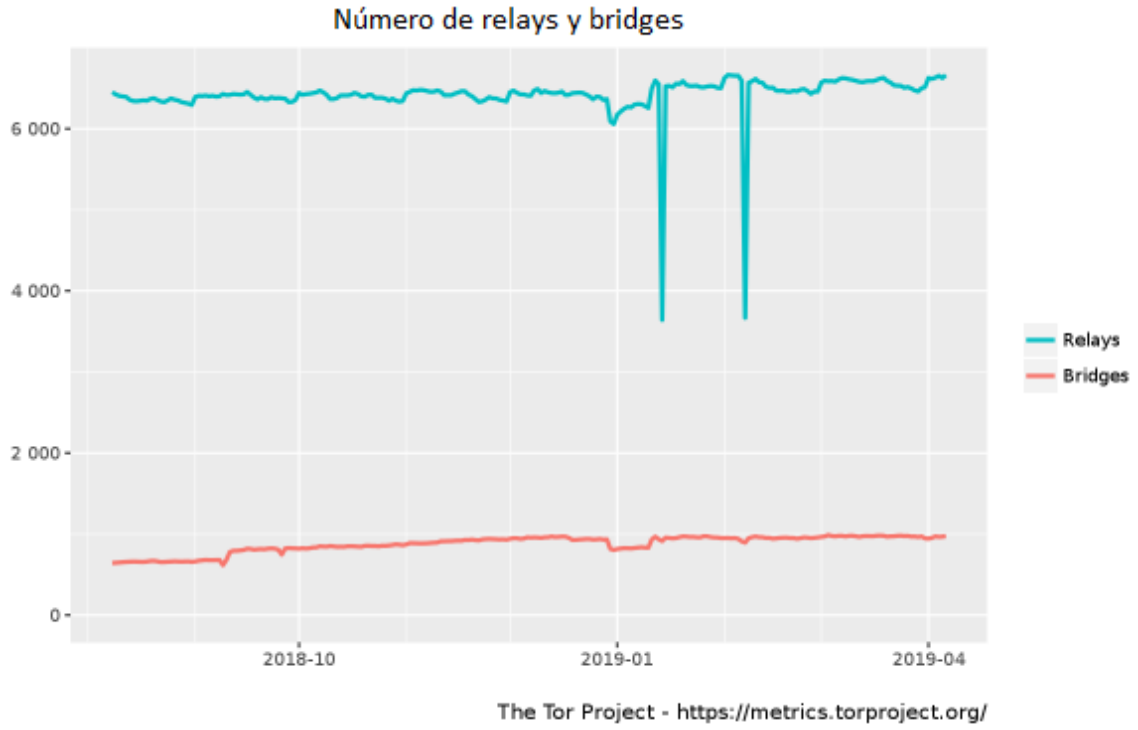


Figura 3.21 Número de relays y bridges

En la figura 3.1 vemos que existen sobre 6500 relays de Tor, los cuales son, entry guard, middle relays y exit relays, y existen sobre 1000 bridge relays, lo que es un número bastante significante. Más o menos el 15% de los relays de la red son Bridge Relays.

Ahora vamos a centrarnos en el número de usuarios de bridge relays. La figura 3.2 muestra los usuarios diarios de bridge relays.

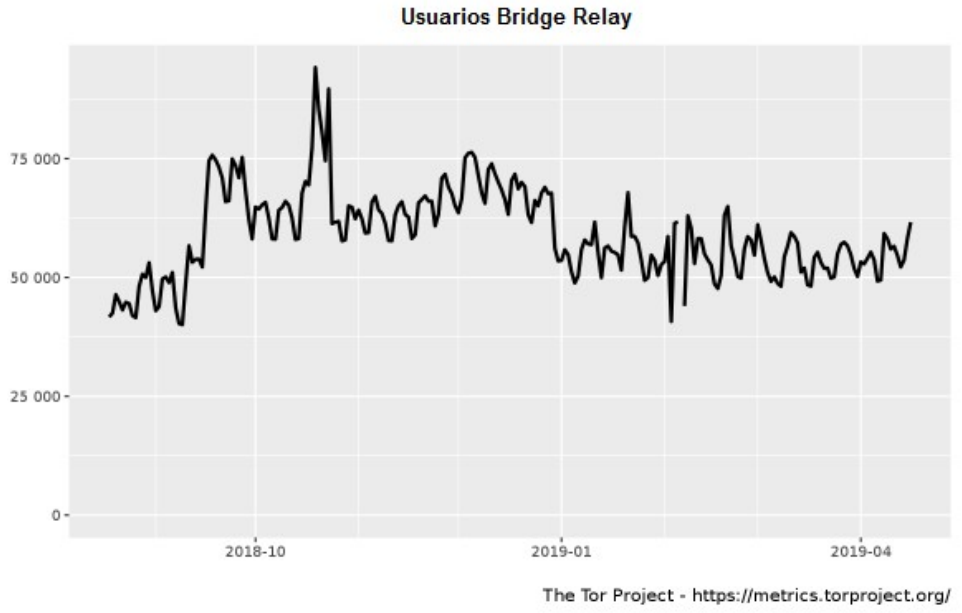


Figura 3.22 Usuarios bridge relay al día

Actualmente sobre 60.000 usuarios utilizan bridges relays diariamente frente a los cerca de los 2.000.000 de usuarios diarios que tiene la red Tor. Más o menos el 3% de los usuarios de Tor utilizan bridge relays.

En la tabla 3.1 se muestran los 10 primeros países por el número estimado medio de clientes que se conectan a través de bridges al día.

Country	Mean daily users
Russia	10566 (17.01 %)
United States	5122 (8.25 %)
Iran	4861 (7.83 %)
Turkey	3591 (5.78 %)
India	2797 (4.50 %)
Indonesia	2623 (4.22 %)
Ukraine	1995 (3.21 %)
Brazil	1963 (3.16 %)
Vietnam	1937 (3.12 %)
Egypt	1930 (3.11 %)

Tabla 3.3 Los 10 principales países por usuarios de bridges

Como hemos comentado los bridge relays son relays que luchan contra la censura de Tor por lo que estas cifras nos sirven para hacernos una idea de la cantidad de usuarios que intentan acceder a la red a través de estos relays y que países son donde más se utiliza esta herramienta contra la censura.

Comparándolo con la tabla 3.2, que muestra los usuarios medios diarios sin utilizar bridge relays como relay de entrada, vemos que pueden parecer cifras pequeñas, pero en Rusia, Estados Unidos, Irán, Turquía, etc. hacen una gran labor para luchar contra la censura que hay en esos países.

País	Usuarios medios diarios
Estados Unidos	366976 (18.95%)
Rusia	247603 (12.79%)
Alemania	156939 (8.11%)
Indonesia	108068 (5.58%)
Emiratos Árabes Unidos	105984 (5.47%)
Francia	81715 (4.22%)
Ucrania	77605 (4.01%)
Reino Unido	62794 (3.24%)
India	48396 (2,50%)
Países Bajos	43435 (2.24%)

Tabla 3.4 Los 10 principales países por usuarios de relays

3.2 Distribución de Bridge relays

Para usar un bridge relay, los clientes Tor necesitan obtener su información, es decir, la dirección IP, el puerto donde el bridge escucha las conexiones y en ocasiones su fingerprint o identificador personal. Además, el usuario puede necesitar información adicional (por ejemplo, el secreto al usar protocolos de ofuscación).

Como hemos dicho anteriormente estos relays no están anunciados públicamente a la red como el resto de relays, que podemos encontrar su información en los servicios de directorio de

Tor. No debe ser posible para un adversario encontrar toda la información de un bridge relay, incluyendo su dirección IP. Su información debe distribuirse cuidadosamente a los clientes.

Existen dos clases de bridges: públicos y privados.

Los bridges públicos pueden ser utilizados por cualquier cliente de Tor. Suben su información a los Bridge Authority (o autoridad de directorio de bridge), que mantiene una lista de bridges públicos disponibles en la red Tor. La información de bridges públicos se distribuye a los usuarios mediante el servicio BridgeDB, que periódicamente la recibe de los Bridge Authority.

BridgeDB soporta dos canales de distribución diferentes. Los usuarios pueden visitar su sitio web https://bridges.torproject.org/ o solicitarlos por correo electrónico a la dirección bridges@torproject.org. En ambos casos, los usuarios pueden especificar el tipo de transporte conectable que desean y si necesitan un bridge que admita IPv6.

El algoritmo de distribución adoptado por BridgeDB tiene como objetivo evitar el listado de una fracción significativa de bridges públicos: solo distribuye algunos bridges a cada dirección IP o cuenta de correo electrónico solicitantes, restringe la distribución a un subconjunto del grupo de bridges que cambia y limita las solicitudes de correo electrónico a direcciones de proveedores de correo específicos (Gmail, Yahoo, RiseUp). **[21]**

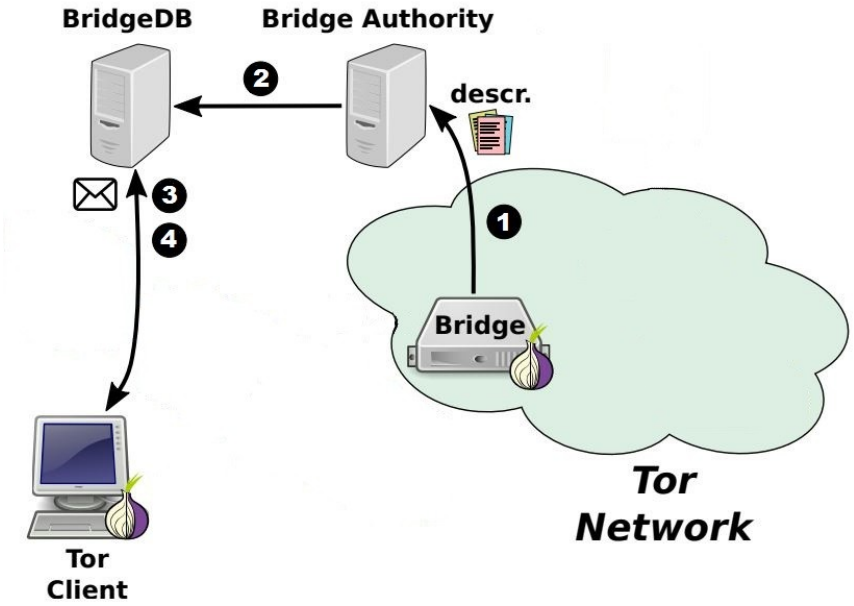


Figura 3.23 Distribución de bridges Relays

❶ Tras la instalación, el bridge envía su información a los Bridge Authority

❷ Los Bridge Authority envían la información de los bridges públicos a los BridgeDB

❸ El usuario solicita bridges a un BridgeDB a través de uno de los canales

❹ BridgeDB envía la información del bridge al cliente de Tor

Para facilitar el uso de bridges sin tener que pasar por los canales de distribución de BridgeDB, el software Tor se envía con una lista de bridges predeterminados para diferentes transportes conectables. Las direcciones IP de estos bridges son triviales de obtener, ya que están codificadas en los archivos de configuración del navegador Tor. Por lo tanto, estos bridges pueden ser fácilmente bloqueados por los adversarios. Cuando el Proyecto Tor detecta el bloqueo en un bridge predeterminado, el bridge se reemplaza por un nuevo valor.

En cambio, el otro tipo de bridges, los bridges privados, no comparten su información con los Bridge Authority y, por lo tanto, son opacos para los mantenedores del Proyecto Tor. Dado que no cargan su información en los Bridge Authority, no se distribuyen y solo sus propietarios tienen su información para poder utilizarlos.

3.3 Transportes conectables

Los transportes conectables o pluggable transport, también llamados protocolos de ofuscación, son una herramienta fundamental para eludir la censura de internet con Tor. Son un sistema que permite a un cliente Tor comunicarse con un bridge relay a través de protocolos que son mucho más difíciles de identificar y, por tanto, de censurar. De esta forma, los censores que controlan el tráfico entre el cliente y el bridge verán tráfico transformado de aspecto inocente en lugar del tráfico real de Tor. Esto puede ser útil en situaciones donde un Proveedor de Servicios de Internet (ISP), u otra autoridad está bloqueando activamente conexiones a la red Tor. **[22]**

Los transportes conectables son una interfaz separada en la cual se crea un túnel por el que va el tráfico Tor ofuscado. Tanto el cliente Tor como el bridge relay de Tor deben soportar la implementación del transporte conectable para que funcione.

La figura 3.4 representa el funcionamiento de los transportes conectables, que están destinados a la ofuscación en el primer enlace de una conexión Tor. El modelo asume que el bridge relay está fuera del alcance del agente de censura de Internet y esta estacionado en una red sin censura. **[23]**

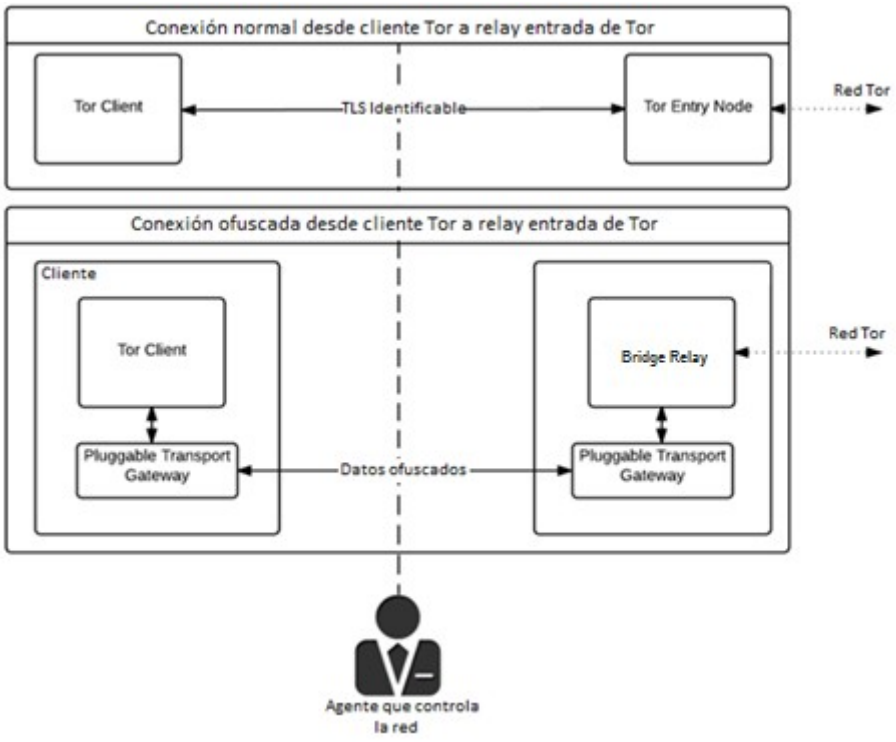


Figura 3.24 Representación transportes conectables

En la primera situación (conexión normal desde el cliente Tor a un relay de entrada) un observador que controle la conexión podrá ver que el tráfico tiene los patrones de conexión de Tor, en cambio en la segunda situación (datos ofuscados) tendrá más difícil reconocer el tráfico Tor.

Los transportes conectables más importantes en la actualidad son: Obfs3, Obfs4, ScrambleSuit, FTE, y Meek.

En la figura 3.5 se muestra la cantidad estimada de clientes que se conectan a través de bridges a la Tor Network utilizando los principales protocolos de ofuscación o utilizando el protocolo por defecto de Tor sin ninguna ofuscación.

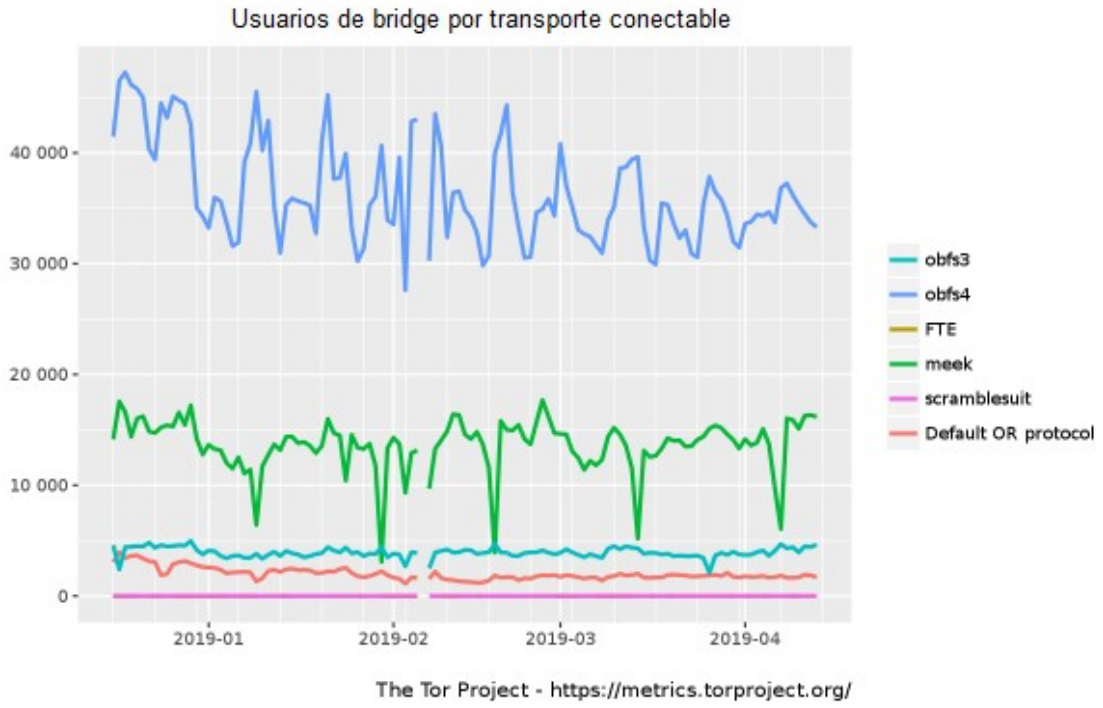


Figura 3.25 Usuarios de bridge por transporte conectable

En la actualidad, obfs4 es el transporte conectable más utilizado. **[24]**

3.3.1 Obfs3

El Proyecto Tor desarrolló e implementó el protocolo obfs3 a principios de 2013 en respuesta a la vulnerabilidad del protocolo obfs2 (su antecesor, ya en desuso) a la detección a través de los mecanismos de DPI o inspección de paquetes y la posterior censura del protocolo. En la actualidad, ha sido superado por el protocolo obfs4, pero todavía es utilizado.

El desarrollo del protocolo está inactivo. Su versión actual es obfs4. **[25]**

Obfs3 es una capa de ofuscación de protocolo para protocolos TCP. Su propósito es evitar que un tercero le diga qué protocolo se está utilizando según el contenido del mensaje. Disfraza el tráfico para que parezca información aleatoria, mientras que TOR tiene una estructura diferente.

El protocolo no proporciona autenticación o integridad de datos y tampoco oculta las longitudes de los datos. Es más adecuado para proporcionar una capa de ofuscación para un protocolo autenticado existente, como TLS.

El protocolo tiene dos fases: en la primera fase, las partes establecen claves. En el segundo, las partes intercambian tráfico supercifrado.

El anterior protocolo, Obfs2, encriptaba el tráfico mediante una clave negociada durante el protocolo, pero no usaba un intercambio de claves robusto. La clave podría ser recuperada por cualquier adversario pasivo que monitoreara el handshake inicial de obfs2.

Para defenderse de este ataque, Obfs3 negocia las claves mediante un intercambio anónimo de claves Diffie Hellman para que un adversario pasivo no pueda recuperar la clave de sesión de Obfs3.

Desafortunadamente, el DH tradicional no se ajusta a nuestro modelo de amenaza ya que sus claves públicas son distinguibles de cadenas aleatorias del mismo tamaño. Por esta razón, se propuso un protocolo DH personalizado que ofrece claves públicas que parecen cadenas aleatorias. **[26]**

El esquema UniformDH fue propuesto por Ian Goldberg en:

https://lists.torproject.org/pipermail/tor-dev/2012-December/004245.html

3.3.2 Scramblesuit

El protocolo ScrambleSuit fue desarrollado por Philipp Winter, Tobias Pulls, and Jergen Fuss como respuesta a la censura basada en DPI (inspección de paquetes) con el objetivo de contrarrestar los ataques cada vez más sofisticados contra Tor y el protocolo Obfs3. Estuvo disponible a partir de febrero de 2014.

Mientras que los paquetes de software que proporcionan la funcionalidad de ScrambleSuit se mantienen y desarrollan de forma activa, el protocolo de ScrambleSuit está superado casi por completo por el protocolo obfs4. Toda su funcionalidad está presente en el protocolo obfs4, por lo tanto, se considera obsoleta. **[27]**

ScrambleSuit es un protocolo de transporte conectable para el marco de ofuscación obfsproxy. Toda su carga útil es computacionalmente indistinguible de la aleatoriedad, modifica su firma de flujo y emplea cifrado autenticado para disfrazar el protocolo transportado.

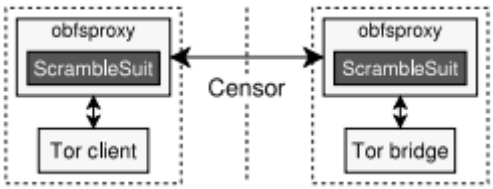


Figura 3.26 ScrambleSuit

El protocolo proporciona dos características principales:

ScrambleSuit protege contra ataques de sondeo activo mediante una forma efectiva de evitar que las cajas de DPI identifiquen protocolos encriptados.

El protocolo implementa además técnicas de transformación que hacen posible que cada servidor ScrambleSuit genere una firma de flujo única. En particular, modifica los tiempos de llegada de paquetes y la distribución de longitud de paquetes del protocolo transportado.

Al igual que Obfs3, para defenderse del ataque de inspección de paquetes para recuperar la clave de sesión, ScrambleSuit utiliza el protocolo UniformDH para negociar las claves.

Hasta ahora, hemos visto las defensas contra los censores con el objetivo de analizar la carga útil del paquete o realizar ataques activos para revelar la presencia de tráfico Tor. Sin embargo, un censor podría hacer uso del análisis de tráfico, es decir, analizar aspectos de comunicación

distintos a la carga útil. Este protocolo propone contramedidas ligeras para disminuir, pero no para derrotar, tales ataques. En particular, cada servidor ScrambleSuit genera su propia y exclusiva "forma de protocolo".

ScrambleSuit considera la longitud de los paquetes y los tiempos entre llegadas. Si bien el cifrado de los mensajes de protocolo hace inútil el análisis de la carga útil, estas dos métricas de flujo aún filtran información sobre la aplicación transportada. Como resultado, ScrambleSuit busca disfrazar estas características para disminuir la precisión de los clasificadores de protocolo para identificar el protocolo. El enfoque a este problema es el polimorfismo de protocolo. Logra el polimorfismo creando una forma de protocolo para cada servidor. Cuando un servidor ScrambleSuit arranca por primera vez, genera aleatoriamente una semilla de 256 bits. Esta semilla se usa para obtener dos distribuciones de probabilidad discretas. Estas dos distribuciones dictan la forma deseada de la longitud de los paquetes y los tiempos entre llegadas. Además, un servidor comunica su semilla única a los clientes después de la autenticación exitosa. Como la semilla es compartida por ambas partes, pueden generar distribuciones de probabilidad idénticas y, por lo tanto, configurar su tráfico de la misma manera. Un censor que supervise dos servidores distintos de ScrambleSuit observará diferentes distribuciones para la longitud de los paquetes y los tiempos entre llegadas. **[28]**

3.3.3 Obfs4

El protocolo obfs4 se desarrolló a lo largo de 2014 y está implementado por el Proyecto Tor en respuesta a la vulnerabilidad del protocolo obfs3 a la detección mediante ataques de sondeo activo. Es el transporte conectable de primera línea utilizado por la mayoría de los usuarios de Tor.

Desde el punto de vista del diseño, el protocolo obfs4 está significativamente más cerca del protocolo ScrambleSuit, y puede describirse como un descendiente directo con mejoras criptográficas incrementales.

El proyecto ha tenido una cantidad moderada de revisiones por parte de terceros y está en desarrollo activo. Es poco probable que el protocolo obfs4 en sí cambie significativamente. En el futuro quizás se realicen nuevas mejoras en forma de nuevos protocolos. **[29]**

A diferencia de obfs3, obfs4 intenta proporcionar autenticación e integridad de datos, aunque todavía está diseñado principalmente para proporcionar una capa de ofuscación para un protocolo autenticado existente como TLS.

Al igual que obfs3 y ScrambleSuit, el protocolo tiene 2 fases: en la primera fase ambas partes establecen claves. En el segundo, las partes intercambian tráfico supercifrado.

ScrambleSuit y obfs3 utilizan UniformDH para el protocolo de enlace criptográfico, pero tiene graves implicaciones en el rendimiento debido a que el exponenciación modular es una operación costosa. Además, el intercambio de claves no se autentica, por lo que es posible que un atacante activo central pueda conocer el secreto compartido entre cliente y bridge.

Obfs4 intenta solucionar estos inconvenientes mediante el uso de un mecanismo de intercambio de claves autenticado basado en el protocolo de enlace de Ntor del Proyecto Tor.

El protocolo Ntor se desarrolló para sustituir el protocolo de enlace anterior. Utiliza curvas elípticas Diffie-Hellman y una función hash para realizar un intercambio de claves autenticado en un solo sentido. **[30]**

Obfs4, al igual que ScrambleSuit, implementa el polimorfismo de protocolo para ofuscar la firma de flujo. Obfs4 considera la longitud de los paquetes y los tiempos entre llegadas ya que estas dos métricas de flujo aún filtran información sobre la aplicación transportada. Como resultado, Obfs4 busca disfrazar estas características. Logra el polimorfismo creando una forma de protocolo para cada servidor. Cuando un servidor Obfs4 arranca por primera vez, genera

aleatoriamente una semilla de 256 bits que dicta la forma deseada de la longitud de los paquetes y los tiempos entre llegadas. La implementación sigue la misma especificación de ScrambleSuit. **[31]**

3.3.4 FTE

FTE (Format Transforming Encryption) fue desarrollado por Kevin P. Dyer, presentado en noviembre del año 2013 y actualmente está en uso. Utiliza una familia criptográfica primitiva que transforma texto plano en texto cifrado que se ajusta a un formato predefinido, generalmente el de otro protocolo. **[32]**

Se ha realizado una cantidad moderada de revisiones de este protocolo, ya que el desarrollador principal se comprometió con el Proyecto Tor en el desarrollo del transporte. El proyecto parece mantenerse, pero no se han agregado nuevas funciones importantes recientemente.

Es el único transporte conectable actual basado en "mimetismo", todos los demás transportes transforman el tráfico para que parezca un ruido aleatorio (la familia Obfs), o incluyen implementaciones completas de otros protocolos (Meek).

Las primitivas criptográficas tradicionales toman una clave y un mensaje como entrada y generan un texto cifrado sin formato.

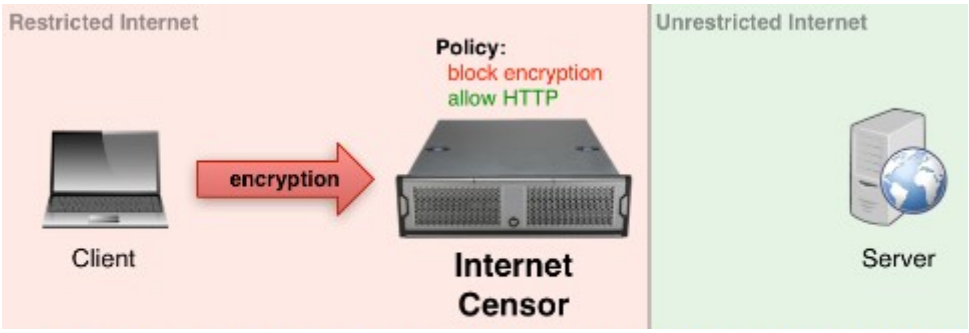


Figura 3.27 FTE 1

Un gobierno puede identificar fácilmente que un cliente y un servidor están utilizando un protocolo de cifrado y rechazar la conexión

Fteproxy toma una clave, un mensaje y un formato como entrada y genera un texto cifrado en otros formatos. Puede estar transmitiendo un protocolo censurado (por ejemplo, Tor, TLS SSH, etc.) pero en realidad simular que envía un protocolo sin censura como puede ser por ejemplo HTTP.

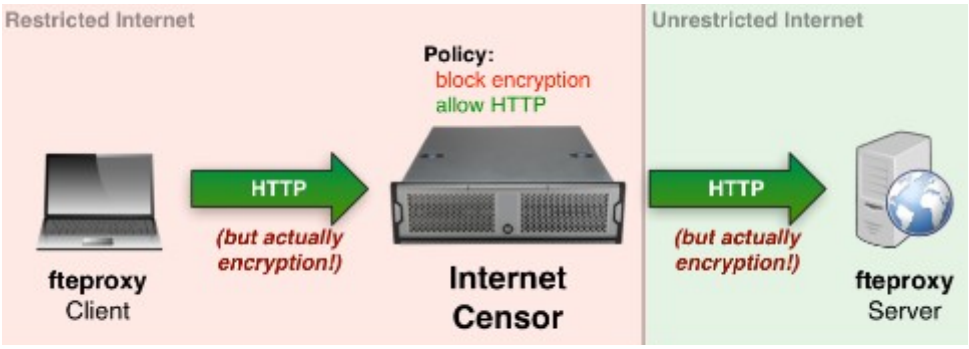


Figura 3.28 FTE 2

Fteproxy cifra de forma transparente las comunicaciones, por ejemplo, un protocolo sin censura parece un protocolo censurado.

3.3.5 Meek

El protocolo Meek fue desarrollado por David Fitfield y actualmente está en uso. Es una de las mejores opciones para los usuarios en ciertos entornos. El protocolo y el software se anunciaron en enero de 2014. **[33]**

Como proyecto que ha ganado un considerable interés externo, ha tenido importantes cantidades de revisiones externas y está en continuo desarrollo por parte del autor original.

Meek es un transporte conectable, una capa de ofuscación para Tor, diseñada para evadir la censura de Internet. El tráfico se transmite a través de un servidor de terceros que es difícil de bloquear. Se utiliza un truco llamado frente de dominio, una técnica de elusión de la censura versátil que oculta el extremo remoto de una comunicación. El frente de dominio funciona en la capa de la aplicación, usando HTTPS, para comunicarse con un host prohibido mientras parece comunicarse con algún otro host, permitido por el censor.

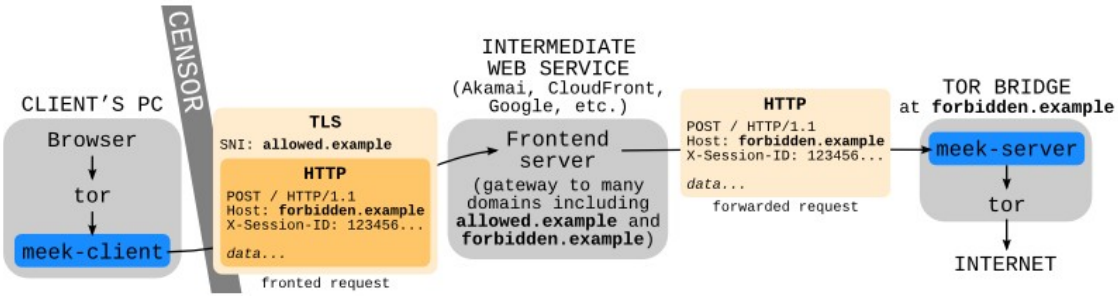


Figura 3.29 Arquitectura de meek

El cliente envía una solicitud HTTP al bridge Tor a través de un servicio web intermedio. El cliente protege el nombre de dominio prohibido del bridge del censor al enfrentarlo con otro nombre, aquí allowed.example. El servidor web intermedio descifra la capa TLS y reenvía la solicitud al bridge de acuerdo con el encabezado del host. Meek-client y meek-server son la interfaz entre Tor y el transporte conectable. Desde el punto de vista de Tor, todo lo que existe entre meek-client y meek-server es un transporte de datos opaco. El host de allowed.example no participa en la comunicación.

La secuencia de bridge a cliente se devuelve en los cuerpos de las respuestas HTTP. Después de recibir una solicitud del cliente, meek-server verifica los datos pendientes que el bridge debe enviar al cliente y los envía en la respuesta HTTP. Cuando el cliente meek recibe la respuesta, se la transmite al cliente Tor. **[34]**

Meek hace que parezca que estás navegando por uno de los principales sitios web en lugar de estar usando Tor. Los servidores intermedios más importantes son: meek-amazon que hace que parezca que estás usando Amazon Web Services, meek-azure que hace que parezca que estás usando un sitio web de Microsoft, y meek-google que hace que parezca que estás haciendo una búsqueda de Google.

3.4 Ventajas de utilizar un Bridge Relay

Utilizar un Bridge Relay como primer relay de un circuito Tor tiene algunas ventajas respecto a utilizar un Entry Guard.

Una de las principales ventajas de los bridge relays es que nos pueden ayudar a eludir la censura si las IP conocidas de Tor están bloqueadas.

Muchos gobiernos y autoridades toman la lista de relays publicadas en los Directory Autorithies y bloquean sus direcciones IP, por lo que nos sería imposible establecer una conexión con Entry Guard y acceder a Tor. La ventaja de los bridge relays es que no hay una lista publicada de estos relays, por lo que no es tan sencillo bloquearlos.

Los bridges son un paso adelante en la carrera de resistencia de bloqueo de Tor.

Otra de las principales ventajas de los bridges es que tienen la posibilidad de utilizar unos protocolos, llamados transportes conectables, para camuflar Tor.

Algunos gobiernos y proveedores de servicio toman medidas un poco más avanzadas para bloquear Tor, monitorean la red para intentar detectar tráfico que corresponda a la red Tor y bloquearlo.

Los transportes conectables son protocolos que camuflan el tráfico Tor para que si alguien monitoriza nuestro tráfico tenga mucho más difícil saber que es tráfico Tor.

Podemos decir que una ventaja de los bridge relays junto a los transportes conectables puede ser la menor detectabilidad de su conexión / sesión.

3.5 Establecimiento de un circuito a través de un Bridge Relay

El establecimiento de un circuito a través de un Bridge Relay no funciona de manera tradicional como cuando nos conectamos a Tor a través de un circuito convencional como vimos en el funcionamiento de Tor, en el apartado 2.4.1.

Una de las principales diferencias es que, de la forma tradicional, el software Tor elige de forma aleatoria todos los relays del circuito Tor después de recoger la información acerca de los relays disponibles de los Directory Authorities y cuando queremos conectarnos a través de un bridge relay, este relay es pre asignado en la configuración inicial del software de Tor, manualmente por el cliente o seleccionando la opción de que el software escoja un bridge de una lista de bridges que el software descarga y actualiza automáticamente. En posteriores apartados de este proyecto veremos cómo utilizar y configurar el software Tor para el uso de Bridge Relays.

Otra diferencia es que si configuramos un bridge relay con un protocolo de ofuscación o transporte conectable el protocolo TLS tradicional ira transportando sobre otro protocolo entre el software del cliente Tor y el bridge relay.

Establecimiento de la ruta

El primer paso que hace el software Tor en el proceso de establecimiento de la ruta a través de un bridge es la comprobación, en la configuración del software, de que está configurado para conectarse a Tor a través de un bridge relay (“UseBridges 1” en el archivo de configuración torrc). En la configuración del software también está la lista de los bridge relays disponibles y su información para la conexión, de la que escogerá el primer relay del circuito.

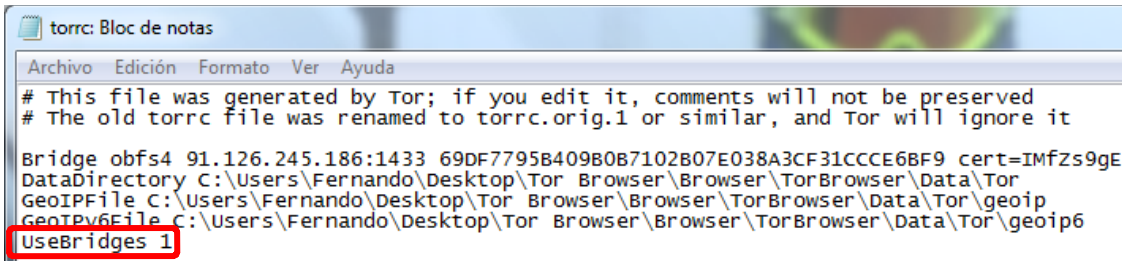


Figura 3.30 Archivo de configuración torrc cliente bridge

A continuación, el software Tor recoge información de los Directory Authorities acerca de los relays disponibles. A partir de la información obtenida el software Tor decide el segundo y tercer relay del circuito de Tor. Por defecto el circuito tiene 3 relays, el bridge relay que será el primero, el middle relay que será el segundo y el exit relay que será el último, antes de que el mensaje llegue a su destino.

Conexión con el Bridge Relay:

El software Tor se conectará de forma segura con el primer bridge disponible de la lista de bridges en la configuración, utilizando una conexión TLS normal o camuflándola con el protocolo de ofuscación o transporte conectable (PT) correspondiente.

Una vez establecida la conexión segura se creará una clave de sesión 1 entre el software TOR del equipo del cliente y el bridge relay.

Conexión con el Middle Relay:

Para extender la ruta, el software TOR de nuestro ordenador usará la clave de sesión 1 para cifrar un mensaje que enviará al Bridge Relay. Cuando el Bridge Relay reciba el mensaje lo descifrará y de esta forma descubrirá el Middle Relay al que se tiene que contactar.

A continuación, el bridge establece una conexión segura con el middle relay mediante el protocolo TLS. Una vez establecida la conexión el bridge relay cifra un mensaje con la clave de sesión 1 en el que informa al software TOR que se ha establecido la conexión entre el bridge relay y el middle relay.

Al llegar el mensaje del bridge relay al Software Tor se descifra y al confirmarse la conexión entre relays se establece una clave de sesión 2 entre el software Tor y el Middle Relay.

Conexión con el Exit Relay:

Finalmente, el mensaje que contiene el relay de salida se envía desde el Software Tor al Bridge Relay cifrado con la clave de sesión 1 y con la clave de sesión 2. En el Bridge Relay se descifra parte del mensaje con la clave de sesión 1 y a posteriori se envía al Middle Relay.

Cuando el mensaje llega al Middle Relay se usa la clave de sesión 2 para descifrar totalmente el mensaje. Acto seguido el Middle Relay establecerá una conexión segura con el Exit Relay mediante el protocolo TLS.

Al establecerse la conexión se informará de forma segura al Software Tor del cliente que la conexión entre el Middle Relay y el Exit Relay se ha establecido. Acto seguido se creará una clave de sesión 3 entre el software Tor y el Exit Relay.

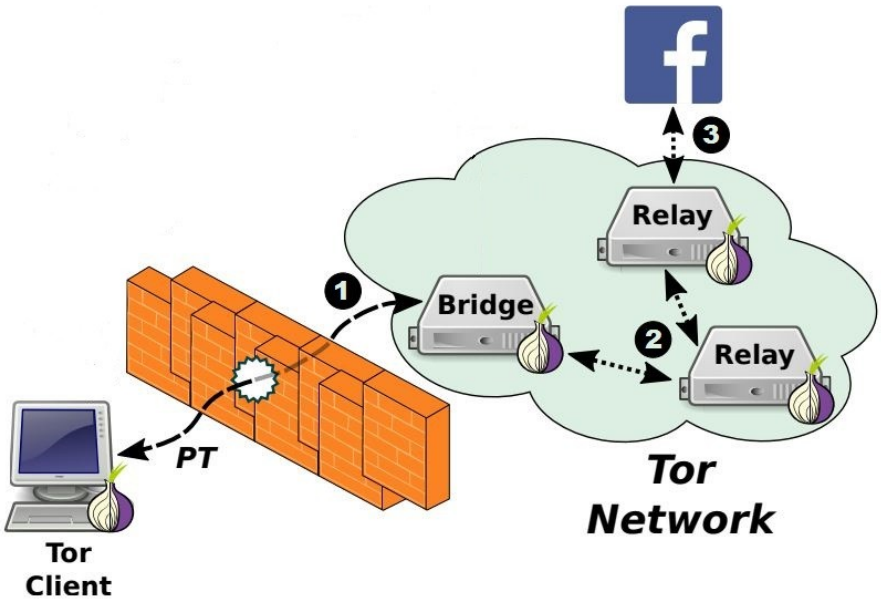


Figura 3.31 Establecimiento circuito bridge

❶ El cliente se conecta al bridge usando TLS identificable o un transporte conectable.

❷ Se construye un circuito de la red Tor a través del bridge relay.

❸ El cliente puede comunicarse con el servicio a través de Tor.

4 Implementación de un Bridge Relay

La mayor parte de la sociedad piensa que se necesita material muy costoso y unos conocimientos expertos para trabajar con herramientas como podría ser un relay, y nada más lejos de la realidad ya que se precisa de muy pocos prerrequisitos para lograrlo.

Cualquier futuro poseedor de un relay de Tor debe cumplir con los siguientes requisitos:

- La conexión necesita un ancho de banda de por lo menos 20 kilobytes/segundo en ambas direcciones.
- Necesitará una conexión con una dirección IP que sea enrutable públicamente.
- Si el dispositivo se encuentra tras un cortafuego NAT (Network Address Translation) y no tiene acceso público o dirección IP pública, necesitaremos configurar una regla de reenvío de puertos en nuestro router. Esto se puede realizar mediante “Tor Universal Plug & Play” o manualmente en el panel de control de nuestro router.

Teniendo en cuenta lo anterior, el futuro poseedor de un relay Tor tiene la ventaja de acceder a un enorme abanico de posibilidades de configuración e instalación.

A continuación, vamos a ver la parte más práctica de este proyecto, la instalación y configuración de un bridge relay.

Cuando tengamos nuestro relay operativo analizaremos el tráfico de Tor sin utilizar un transporte conectable y utilizando él más actual, obfs4.

4.1 Instalación de un Bridge Relay

El dispositivo donde vamos a correr nuestro Bridge Tor es un antiguo Netbook Acer que tenía en desuso. No es un equipo que ofrezca un gran rendimiento, pero nos será más que suficiente, ya que para correr un relay de Tor no necesitamos una gran capacidad de procesamiento. Este Netbook nos da la ventaja de poseer teclado y pantalla integrados por lo que nos facilita mucho la interacción con nuestro relay, a diferencia de si utilizaríamos un dispositivo como por ejemplo una Raspberry.

Figura 4.32: Netbook Acer Aspire One

Las características de hardware de este Netbook son:

- **Modelo**: Netbook Acer Aspire One.
- **Procesador**: Intel Atom N270 1.6GHz.
- **Memoria RAM**: DDR2 de 1GB.
- **Almacenamiento**: SATA 160 GB.
- **Conectividad de red**: Red inalámbrica Acer InviLink 802.11B/G y Red Fast Ethernet 10/100 Mbps por cable RJ-45.
- **Salida de video**: pantalla de 10.1“ y salida de video VGA.
- **Interfaces de entrada**: teclado y ratón integrados.
- **Puertos USB**: 3 USB 2.0.

La elección del sistema operativo donde correrá nuestro relay es Debian 9.4 <<Strech>> de 32 bits, una de las últimas versiones de este sistema operativo libre, publicada el 10 de marzo de 2018. **[35]**

Figura 4.33 Debian

Las características principales para su elección son:

Versión actual.

Instalación sencilla.

Actualizaciones fáciles.

Soporta un impresionante número de arquitecturas CPU y es difícil que encuentre un dispositivo que no pueda correr Debian.

Rápido y ligero en memoria, ideal para nuestro Netbook. **[36]**

En el anexo de este documento podemos ver el proceso de instalación de Debian.

Una vez ya tenemos Debian 9 instalado y podemos empezar con la configuración de nuestro Bridge Relay.

4.1.1 Configuración inicial y actualizaciones

Lo primero que tenemos que tener en cuenta es que Debian 9 no tiene instalado por defecto el comando sudo que nos permitirá instalar aplicaciones desde el usuario común que hemos creado en la instalación de Debian, sin tener que acceder con la cuenta de usuario root. Por lo que vamos a instalarlo: **[39]**

Abrimos una ventana del terminal donde introducimos el comando ($su) e introducimos la contraseña de root que nos pedirá.

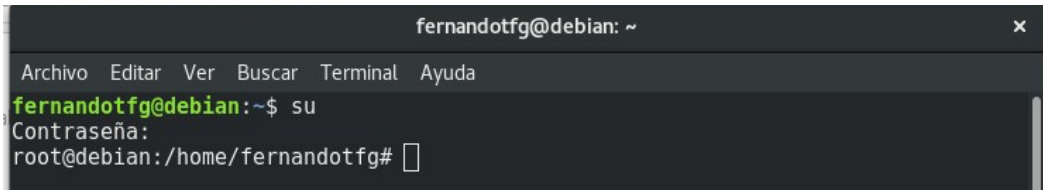


Figura 4.34 Comando $su

Instalamos el paquete sudo, agregamos nuestro usuario (en mi caso fernandotfg) al grupo sudo con los siguientes comandos y reiniciamos para que se apliquen los cambios:

```
#apt install –y sudo

#gpasswd –a fernandotfg sudo

#reboot
```

Abrimos de nuevo una ventana del terminal e introducimos el siguiente comando que nos abrirá un archivo de configuración que debemos modificar para que cada vez que ejecutemos un

comando con sudo no nos pida la contraseña del usuario:

#sudo visudo

Nos pide la contraseña de nuestro usuario, la introducimos y se nos abre un archivo de configuración. Donde cambiamos la siguiente línea:

%sudo ALL=(ALL:ALL) ALL	por	**%sudo ALL=(ALL:ALL) NOPASSWD:ALL**

A continuación, vamos a actualizar nuestro sistema con los siguientes comandos:

#sudo apt-get update **#sudo apt-get dist-upgrade**

4.1.2 Abrir puertos en el router

Para el correcto funcionamiento de nuestro Bridge Relay necesitamos que dos puertos sean accesibles desde el exterior de nuestra red. Vamos a utilizar el puerto 9001 con el protocolo TCP que hará la función de puerto ORPort y el 1433 con el protocolo TCP que lo utilizaremos para el transporte conectable obfs4. En el siguiente apartado, en la configuración de Tor, explicaremos un poco más para que se utilizan estos puertos.

El proceso de abrir los puertos dependerá del router que tengamos, en mi caso mi router es un Huawei hg532s con un firmware de la compañía Orange. Desde el navegador de cualquier equipo que esté conectado a la red local podremos entrar en la configuración accediendo a la dirección 192.168.1.1 e introduciendo el usuario y la contraseña.

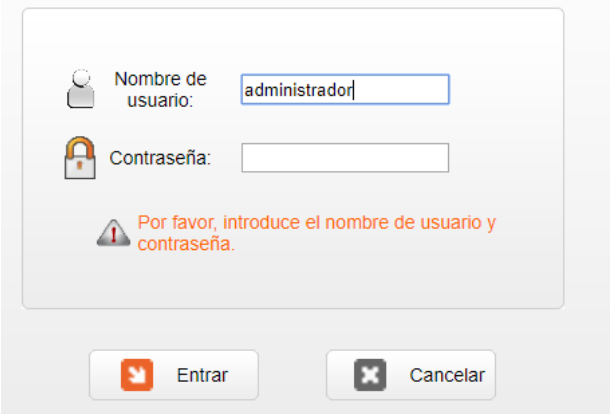


Figura 4.35 Acceso router

Lo primero que tenemos que hacer es asignar una dirección IP fija al equipo. El equipo irá conectado a la red por Wifi por lo que podemos hacerlo de dos formas, mediante comandos desde el terminal del propio equipo o por reserva de MAC desde la configuración de nuestro router.

Hemos escogido la segunda opción. Por lo que en primer lugar iremos al equipo donde vamos a instalar nuestro TOR bridge y con el comando (**# ip address**) obtendremos la dirección MAC de su tarjeta de red Wifi.

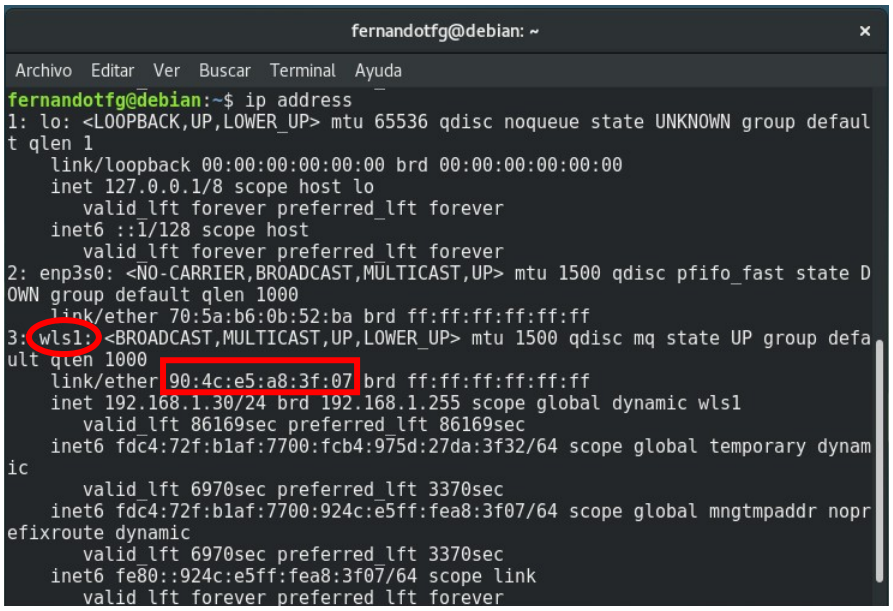


Figura 4.36 MAC Debian

Ahora iremos a la configuración de nuestro router y en el apartado de la configuración “Básica” está la configuración de “DHCP” donde tenemos la opción de “Reserva de dirección IP”. Allí añadimos una línea nueva con la MAC del equipo, en este caso 90:4c:e5:a8:3f:07, y la dirección IP que le queramos dar, en este caso le damos la 192.168.1.30.

Figura 4.37 IP fija

Una vez tengamos una dirección IP local fija en el equipo iremos a la configuración del router para abrir los puertos que necesitamos que sean accesibles desde el exterior.

En la configuración avanzada del router vamos a la pestaña NAT y asignación de puertos. Añadiremos dos líneas, una para el ORPort que utilizaremos el puerto 9001 con el protocolo TCP y otra para obfs4 en el puerto 1433 con el protocolo TCP.

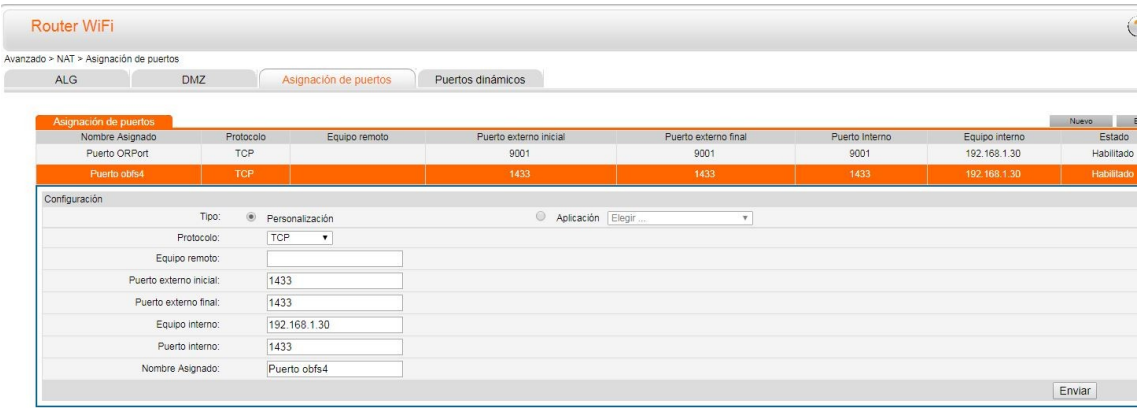


Figura 4.38 Asignación de puertos

4.1.3 Configuración firewall Debian

Para hacer más seguro nuestro Bridge Relay vamos a instalar y configurar un firewall que tan solo permita la entrada de conexiones a través de los puertos específicos Tor.

Hemos el firewall UFW que tiene una interfaz orientada a simplificar el proceso de configuración de un firewall. **[40]**

Debian no tiene instalado UFW por defecto, por lo que vamos a instalarlo con el siguiente comando:

```
#sudo apt install ufw
```

De forma predeterminada, UFW está configurado para denegar todas las conexiones entrantes y permitir todas las conexiones salientes. Esto significa que cualquier persona que intente acceder a su equipo no podrá conectarse, mientras que cualquier aplicación dentro del equipo podrá acceder al mundo exterior.

Para asegurarnos de que estas reglas predeterminadas están configuradas utilizamos los siguientes comandos:

```
#sudo ufw default deny incoming

#sudo ufw default allow outgoing
```

Por defecto, tras la instalación, el firewall UFW está desactivado, con el siguiente comando lo activaremos:

```
#sudo ufw enable
```

Mediante varias herramientas Web, como https://www.testdevelocidad.es/test-de-puertos/, podemos comprobar el estado de los puertos.

Como vemos, en la figura 4.8, ahora mismo no están permitidas las conexiones a los puertos 9001 y 1433.

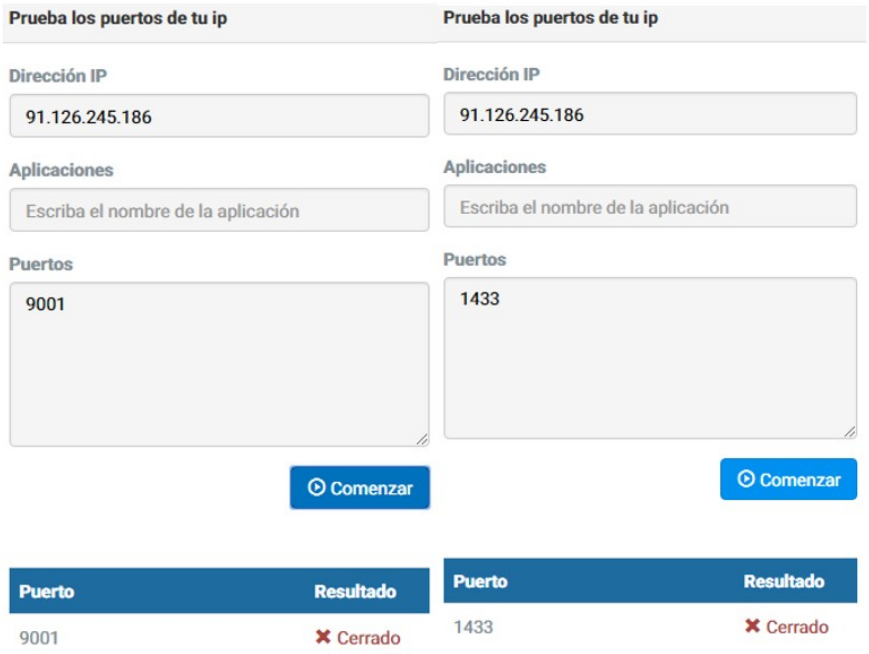


Figura 4.39 Comprobación de puertos

Para el funcionamiento del bridge Relay, nuestro equipo debe permitir las conexiones a los puertos 9001 y 1433.

Con los siguientes comandos permitiremos estas conexiones:

```
#sudo ufw allow 9001

#sudo ufw allow 1433
```

A continuación, en la figura 4.9, podemos ver que ahora si están permitidas las conexiones a los puertos 9001 y 1433:

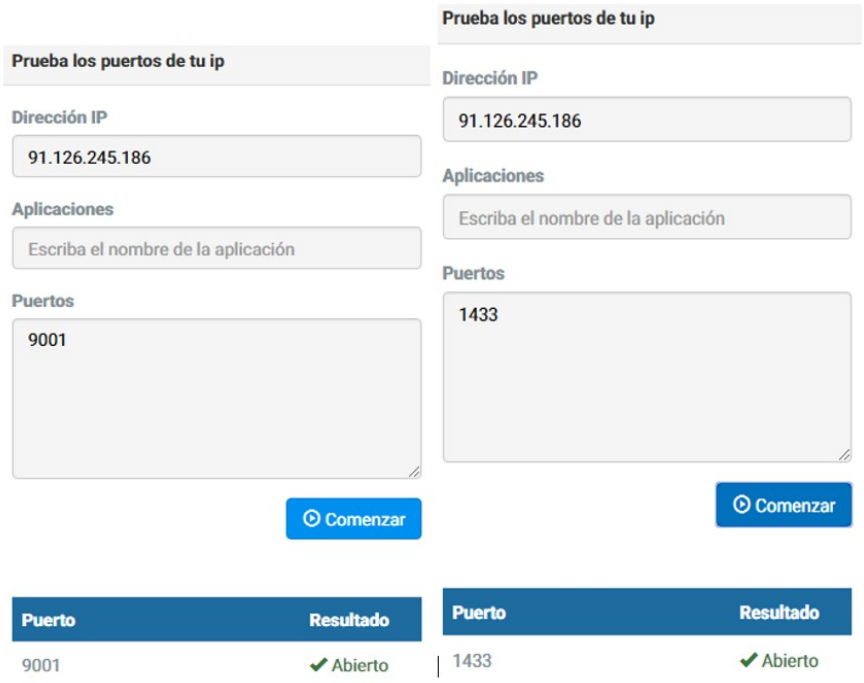


Figura 4.40 Comprobación de puertos 2

Para ver las reglas que tenemos creadas podemos utilizar el siguiente comando:

#sudo ufw status numbered

```
fernandotfg@debian:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 1433                       ALLOW IN    Anywhere
[ 2] 9001                       ALLOW IN    Anywhere
[ 3] 1433 (v6)                  ALLOW IN    Anywhere (v6)
[ 4] 9001 (v6)                  ALLOW IN    Anywhere (v6)
```

Figura 4.41 Reglas firewall UFW

Si deseamos eliminar alguna de las reglas podemos hacerlo con el siguiente comando, sustituyendo el último número por el número de regla que queremos eliminar:

#sudo ufw delete <número de regla>

Si creemos que el firewall nos está dando algún problema en las conexiones, podemos deshabilitarlo con el siguiente comando:

#sudo ufw disable

4.1.4 Instalación Tor

Lo primero que tenemos que hacer es editar nuestra lista de fuentes de Debian con el comando:

```
#sudo nano /etc/apt/sources.list
```

Se nos abrirá un documento, y al final añadimos las siguientes dos líneas para los repositorios del proyecto TOR:

```
deb http://deb.torproject.org/torproject.org stretch main

deb-src http://deb.torproject.org/torproject.org stretch main
```

Presionamos las teclas Ctrl+X para salir y confirmamos el guardado.

A continuación, tenemos que agregar la clave gpg utilizada para asegurar que el paquete de Tor que se va a descargar fue creado por los desarrolladores de TOR.

Ejecutando lo siguientes comandos: **[37]**

```
#sudo apt-get install curl

# curl
https://deb.torproject.org/torproject.org/A3C4F0F979CAA22CDBA8F512EE8CBC9E886
DDD89.asc | gpg --import

# gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-key add -
```

Lo siguiente será instalar Tor y obfs4 con los siguientes comandos:

```
#sudo apt-get install tor
```

Aceptaremos los paquetes opcionales que nos ofrece instalar.

```
#sudo apt-get install obfs4proxy
```

Ahora vamos a configurar Tor para que funcione como un bridge relay editando el archivo de configuración Tor:

```
#sudo nano /etc/tor/torrc
```

Al final del archivo añadiremos: **[38]**

```
ORPort 9001

ExtORPort auto

ExitPolicy reject *:*

BridgeRelay 1

PublishServerDescriptor 0

ServerTransportPlugin obfs4 exec /usr/bin/obfs4proxy

ServerTransportListenAddr obfs4 0.0.0.0:1433

RunAsDaemon 1

Nickname FernandoTFG
```

ORPort: Indica el puerto que escucha conexiones para clientes y servidores Tor. Esta opción es requerida si queremos correr nuestro propio Tor relay. Si está configurado en auto, Tor automáticamente elegirá un puerto por nosotros. Pero nosotros elegimos el puerto 9001 porque necesitamos un puerto fijo y debemos hacer que sea accesible desde el exterior de nuestra red.

ExtorPort: Abra este puerto para escuchar las conexiones en ORPort desde los transportes conectables.

ExitPolicy: Para conseguir un bridge relay configuraremos las políticas de salida como reject. Si estamos corriendo un exit relay, entonces esta política la configuraríamos como acept. Los valores más básicos de configuración son aceptar o rechazar (acept/reject) paquetes basados en su número de puerto destino.

BridgeRelay: Configura el relay para que actúe como un "bridge" con respecto a la retransmisión de las conexiones de los usuarios del bridge a la red Tor. Principalmente, hace que Tor publique el relay en los Bridge Authority, en lugar de en las Directory Authorities.

PublishServerDescriptor: Esta opción especifica si Tor publicará el bridge en los Bridge Authority. Si esta opción se establece en 0, Tor no publicará su bridge en ningún directorio de Tor.

ServerTransportPlugin: El relay Tor lanza el proxy de transporte conectable desde la ruta y espera recibir de él el tráfico del cliente proxy.

ServerTransportListenAddr: Cuando se establezca esta opción, Tor sugerirá IP:PORT como la dirección de escucha de cualquier proxy de transporte conectable que intente iniciar el transporte. En nuestro caso con la IP 0.0.0.0 indicaremos que escuchamos en todas las direcciones IP y en el puerto 1433 que debemos asegurar que es accesible desde el exterior.

RunAsDaemon: Si es 1, Tor funciona como un programa que se ejecuta como un proceso en segundo plano, en lugar de estar bajo el control directo de un usuario.

Nickname: Define el alias de nuestro relay.

Reiniciamos Tor para aplicar la configuración:

```
#sudo service tor restart
```

Si abrimos el registro:

```
#sudo cat /var/log/tor/log
```

En unos minutos deberíamos ver algo como esto:

```
Mar 12 14:19:23.000 [notice] Your Tor server's identity key fingerprint is 'FernandoTFG
3DBFD173EE603BCBB36DE89729CCF5556C3BD29F'
Mar 12 14:19:23.000 [notice] Your Tor bridge's hashed identity key fingerprint is 'FernandoTFG
1FDE5ED6E6D227E5D612AEFC8A6056845ECF27B9'
Mar 12 14:19:23.000 [notice] Bootstrapped 0%: Starting
Mar 12 14:19:24.000 [notice] Signaled readiness to systemd
Mar 12 14:19:24.000 [notice] Opening Socks listener on /var/run/tor/socks
Mar 12 14:19:24.000 [notice] Opening Control listener on /var/run/tor/control
Mar 12 14:19:24.000 [notice] Bootstrapped 5%: Connecting to directory server
Mar 12 14:19:24.000 [notice] Bootstrapped 10%: Finishing handshake with directory server
Mar 12 14:19:25.000 [notice] Bootstrapped 15%: Establishing an encrypted directory connection
Mar 12 14:19:25.000 [notice] Bootstrapped 20%: Asking for networkstatus consensus
Mar 12 14:19:25.000 [notice] Bootstrapped 50%: Loading relay descriptors
Mar 12 14:19:25.000 [notice] Guessed our IP address as 91.126.245.186 (source: 163.172.149.155).
Mar 12 14:19:26.000 [notice] Registered server transport 'obfs4' at '[::]:1433'
Mar 12 14:19:55.000 [notice] Bootstrapped 56%: Loading relay descriptors
Mar 12 14:20:03.000 [notice] Bootstrapped 64%: Loading relay descriptors
Mar 12 14:20:07.000 [notice] Bootstrapped 70%: Loading relay descriptors
Mar 12 14:20:12.000 [notice] Bootstrapped 76%: Loading relay descriptors
Mar 12 14:20:16.000 [notice] Bootstrapped 80%: Connecting to the Tor network
Mar 12 14:20:19.000 [notice] Bootstrapped 90%: Establishing a Tor circuit
Mar 12 14:20:21.000 [notice] Tor has successfully opened a circuit. Looks like client functionality is working.
Mar 12 14:20:21.000 [notice] Bootstrapped 100%: Done
Mar 12 14:20:21.000 [notice] Now checking whether ORPort 91.126.245.186:9001 is reachable... (this may take up to 20
minutes -- look for log messages indicating success)
Mar 12 14:20:21.000 [notice] Self-testing indicates your ORPort is reachable from the outside. Excellent.
Mar 12 14:20:23.000 [notice] Performing bandwidth self-test...done.
```

Ya tendríamos nuestro bridge relay en funcionamiento. Solo nos falta obtener la línea de nuestro bridge que necesitaremos saber para poder conectarnos a Tor a través de él.

Para obtener la línea de nuestro bridge introducimos el siguiente comando:

#sudo cat /var/lib/tor/pt_state/obfs4_bridgeline.txt

El comando nos muestra la siguiente plantilla:

```
fernando@debian:~$ sudo cat /var/lib/tor/pt_state/obfs4_bridgeline.txt
# obfs4 torrc client bridge line
#
# This file is an automatically generated bridge line based on
# the current obfs4proxy configuration.  EDITING IT WILL HAVE
# NO EFFECT.
#
# Before distributing this Bridge, edit the placeholder fields
# to contain the actual values:
#  <IP ADDRESS>  - The public IP address of your obfs4 bridge.
#  <PORT>        - The TCP/IP port of your obfs4 bridge.
#  <FINGERPRINT> - The bridge's fingerprint.

Bridge obfs4 <IP ADDRESS>:<PORT> <FINGERPRINT> cert=RAkY8RfGo8hU7WXXg0A6fPp1R6nVrcua8D
uQ1SCPmpjIKmLMhSe9LgjaRVp3hRBD1X6UKw iat-mode=0
```

Figura 4.42 Bridge line

En esta plantilla deberemos sustituir <IP ADDRESS> por nuestra IP pública, <PORT> por el puerto que obfs4 está escuchando, en este caso 1433 y el <FINGERPRINT>, un identificador único de nuestro servidor que lo obtendremos con el siguiente comando:

#sudo cat /var/lib/tor/fingerprint

Figura 4.43 Fingerprint

Por lo que la línea de nuestro bridge utilizando el transporte conectable obfs4 es:

obfs4 <IP ADDRESS>:1433 3DBFD173EE603BCBB36DE89729CCF5556C3BD29F cert=RAkY8RfGo8hU7WXXg0A6fPp1R6nVrcua8DuQ1SCPmpjlKmLMhSe9LqjaRVp3h RBD1X6UKw iat-mode=0

Pero tendremos una segunda opción que es utilizar nuestro bridge sin utilizar un transporte conectable con la línea:

<IP ADDRESS>:9001 3DBFD173EE603BCBB36DE89729CCF5556C3BD29F

4.2 Análisis con Nyx

Nyx es un monitor de línea de comandos para Tor. Una aplicación anteriormente llamada anonymizing relay monitor (arm) pero que en su última versión cambio su nombre por Nyx. **[41]**

Con esta aplicación podemos obtener información detallada en tiempo real sobre nuestro relé, como:

- Uso de recursos (ancho de banda de subida y bajada, CPU, uso de memoria…)
- Información general (nickname, fingerprint, IP, puertos, flags…)
- Registro de eventos
- Conexiones (IP, tipos de conexión, detalles de retransmisión, etc.)
- Configuración Tor (archivo torrc)

A continuación, vamos a instalar Nyx en nuestro relé desde el Terminal.

La instalación normal sería con el comando:

#sudo apt-get install

Pero Nyx aún no está disponible en los repositorios de distribución de Debian, por lo que nos dan la opcion de instalarlo con pip. Pip es un sistema de administración de paquetes que se puede usar para instalar y administrar paquetes escritos en Python.

#sudo apt-get install python-pip

#sudo pip install nyx

Para ejecutarlo utilizamos el comando " nyx " en la línea de comandos como usuario root.

#nyx

La aplicación Nyx está dividida en 5 pantallas por las que podemos navegar utilizando las flechas de dirección de izquierda y derecha.

La parte superior de la pantalla es común para las 5 pantallas. Donde vemos información como la versión de sistema operativo en el que está instalado, la versión de Tor instalada (si está

actualizada nos lo indicara con el mensaje “recommended”), Nickname de nuestro relay, nuestra IP publica, el puerto que hemos configurado previamente como ORPort, el uso de CPU y memoria, el fingerprint de nuestro relay y las banderas (flags) que tiene asignado nuestro relay.

Figura 4.44 Nyx parte superior

Es interesante destacar que nuestro relay no posee ninguna flag asignada, dado que al ser un bridge relay y no estar publicado en los directorios de Tor, ningún directorio de Tor le asigna ninguna flag.

Primera pantalla

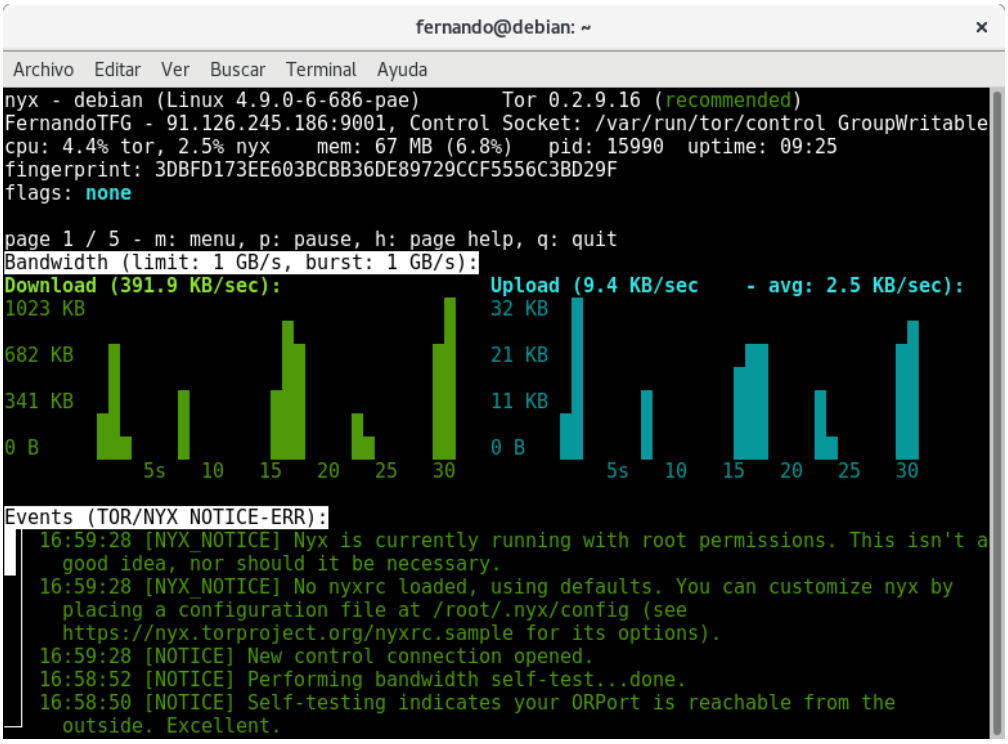


Figura 4.45 Nyx pantalla 1

En primera pantalla de Nyx tenemos una gráfica en tiempo real del ancho de banda de Tor, dividido en tráfico de bajada y tráfico de subida y abajo tenemos los eventos donde por ejemplo nos puede indicar si no se puede conectar, si tenemos una versión desactualizada, información acerca de nuestro fichero torrc…en nuestro caso nos muestra mensajes donde podemos ver que nuestro relay está funcionando correctamente.

Segunda pantalla

En la segunda página de arm, podemos ver las conexiones y los circuitos establecidos en la red Tor.

Las conexiones de entrada son conexiones de equipos de mi propia red a nuestro relay y las conexiones salientes son los siguientes saltos de la red por los que nos estamos conectando.

El caso de los circuitos, al iniciar nuestro relay, Tor nos incluye siempre circuitos como método de seguridad, que son de confirmación para la propia red y que se utilizarían en el caso de que actuáramos como clientes de Tor, pero como no es el caso al poco tiempo desaparecen.

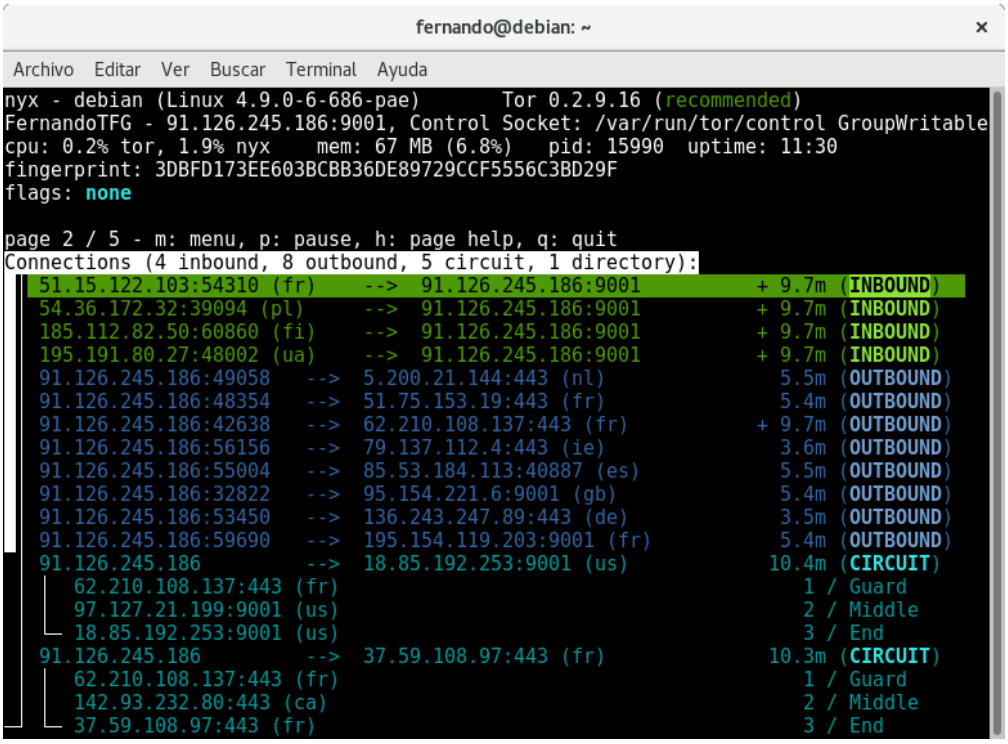


Figura 4.46 Nyx pantalla 2

Tercera pantalla

Desde esta pantalla podemos configurar algunos parámetros del fichero de configuración torrc sobre la marcha. Presionamos 'enter' para cambiar la configuración de Tor y 'w' para escribir sus cambios en el disco.

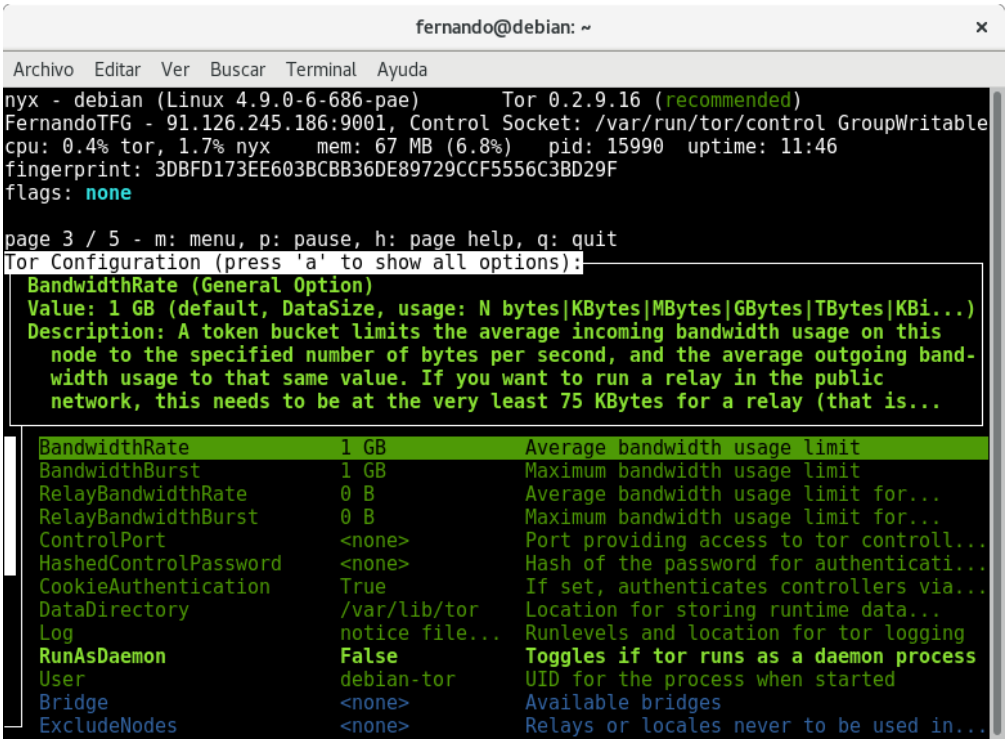


Figura 4.47 Nyx pantalla 3

Cuarta pantalla

En la cuarta pantalla nos permitirá modificar el fichero /etc/tor/torrc a mano como si lo abriríamos con un editor de texto. La principal ventaja es que puedes reiniciar el servicio rápidamente pulsando una tecla.

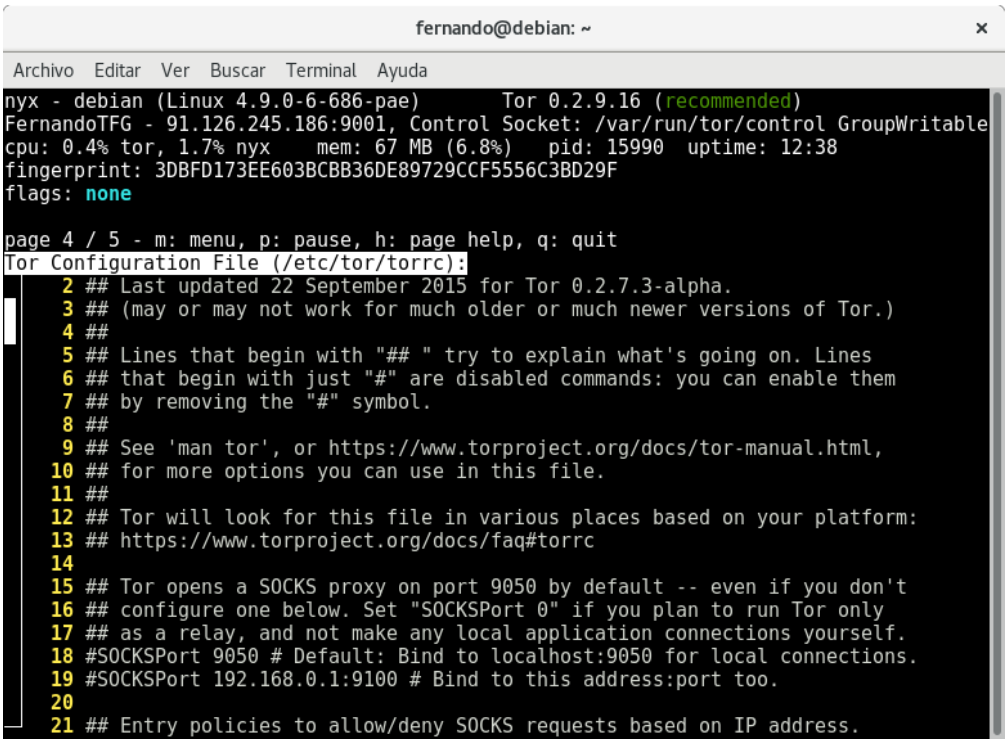


Figura 4.48 Nyx pantalla 4

Quinta pantalla

El Control Interpretor es una consola de comandos interactiva con la cuál podremos administrar nuestro servicio.

Mediante el comando /help podemos ver una lista con los comandos disponibles y una breve descripción de los mismos.

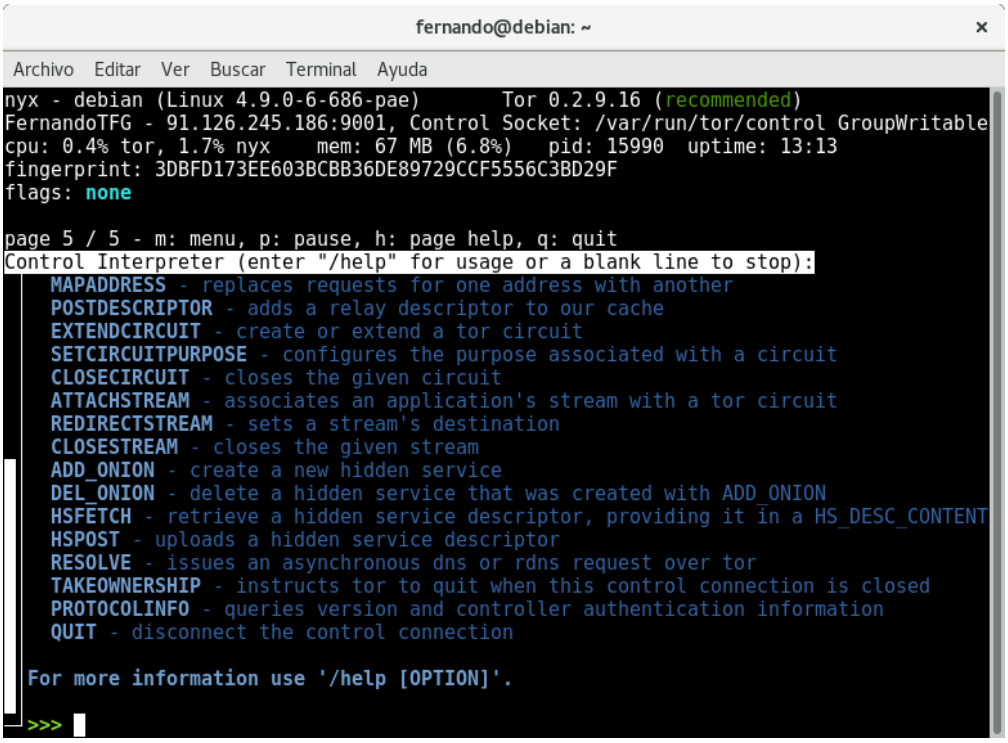


Figura 4.49 Nyx pantalla 5

4.3 Análisis de tráfico con Wireshark

Wireshark es un analizador de paquetes gratuito y de código abierto. Se utiliza para la solución de problemas de red, análisis, desarrollo de software y protocolo de comunicaciones y educación. **[42]**

Instalación Wireshark

Lo primero que vamos a hacer es proceder a instalar la aplicación Wireshark en el equipo que tenemos instalado nuestro bridge relay.

Para instalar la versión más reciente y estable de Wireshark en Debian 9, usamos los siguientes comandos:

```
#sudo apt-get update
#sudo apt-get install wireshark -y
```

Durante la instalación, nos preguntará si deseamos crear un grupo <wireshark> en el que los usuarios que sean miembros de este grupo tendrán los permisos necesarios para poder capturar paquetes.

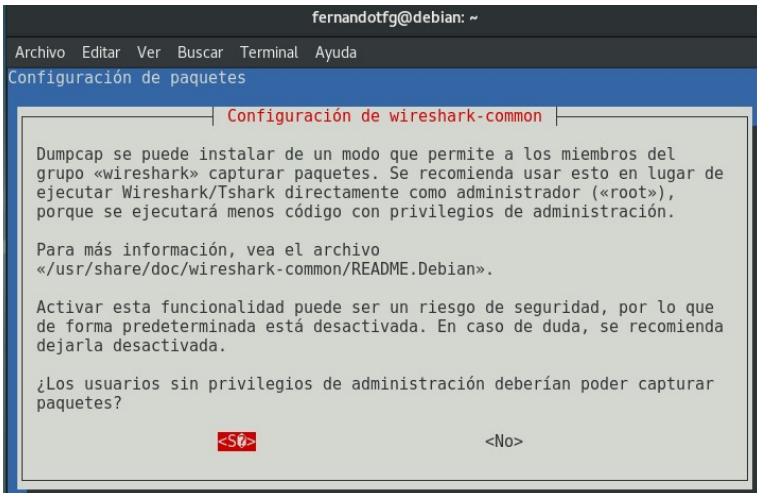


Figura 4.50 Configuración Wireshark

Seleccionamos “Sí” y luego presionamos “Enter”.

Una vez completada la instalación, ejecutamos el siguiente comando para añadir a nuestro usuario al grupo <wireshark>.

```
#sudo adduser <Usuario> wireshark
```

Reiniciamos o cerramos sesión para que se apliquen los cambios y ya estaremos listos para capturar tráfico.

4.3.1 Sin ofuscación

En este apartado vamos a realizar una captura de la conexión entre el software Tor y nuestro bridge relay sin utilizar ningún protocolo de ofuscación para a continuación analizar el tráfico capturado.

Iniciamos Wireshark con el siguiente comando:

```
#wireshark
```

Se abre la siguiente ventana donde elegimos la interfaz wifi wls1, que es la interfaz que estamos utilizando para conectarnos a la red, y la ponemos a capturar tráfico pulsando en el icono .

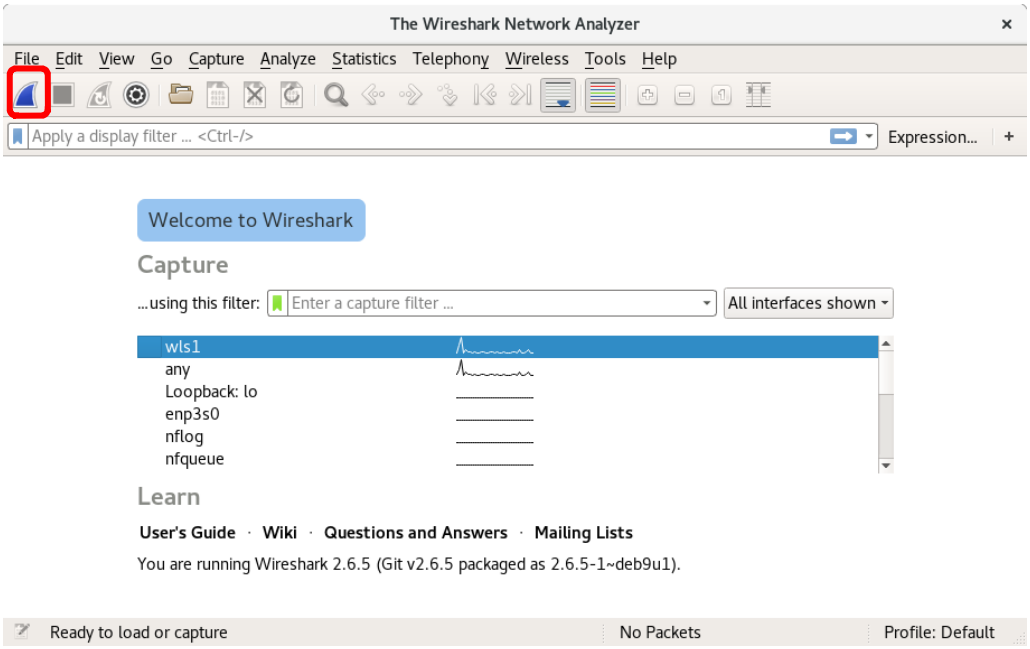


Figura 4.51 Welcome to Wireshark

A continuacion,desde otro equipo iniciamos una conexión desde el navegador Tor utilizando nuestro brige relay pero sin utilizar ningun transporte conectable como explicamos en el apartado 5.1.

Ya podemos parar la captura desde el boton de stop de la aplicación wireshark.

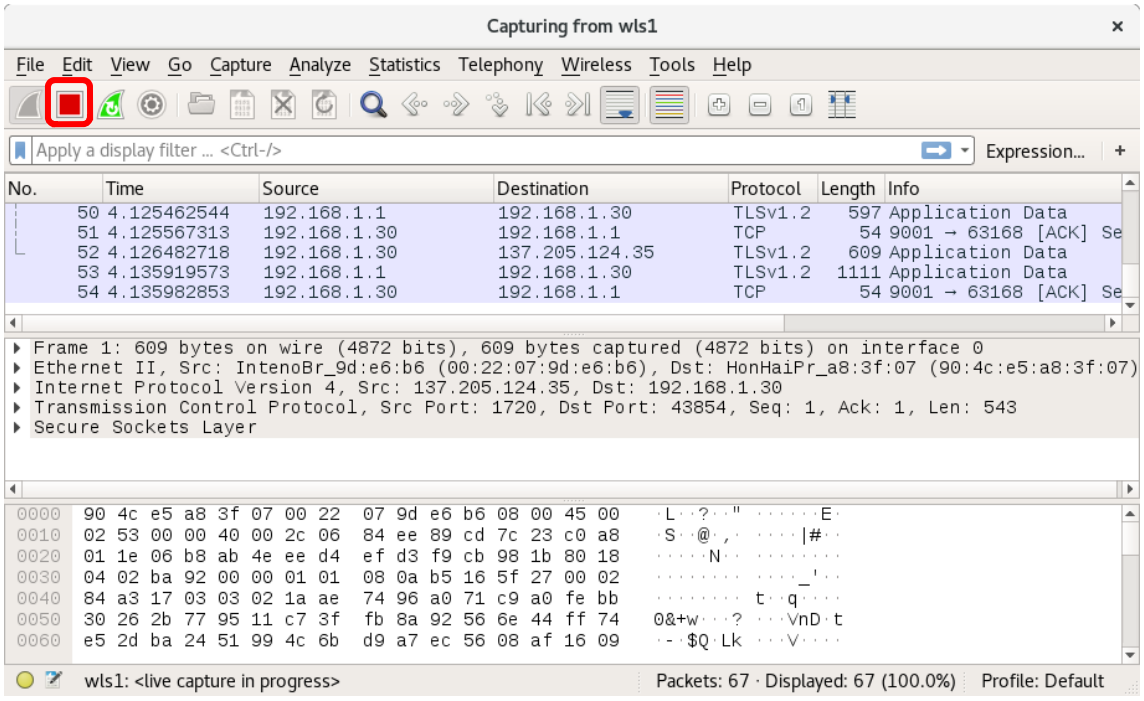


Figura 4.52 Pantalla Wireshark

Guardamos la captura y ya la podemos analizar.

Para filtrar la captura y quitarnos el tráfico de tipo broadcast podemos utilizar el filtro “ip.addr == 192.168.1.30” en wireshark, para así ver solo el tráfico que tiene como origen o destino la dirección IP de nuestro bridge relay. Obtenemos lo siguiente:

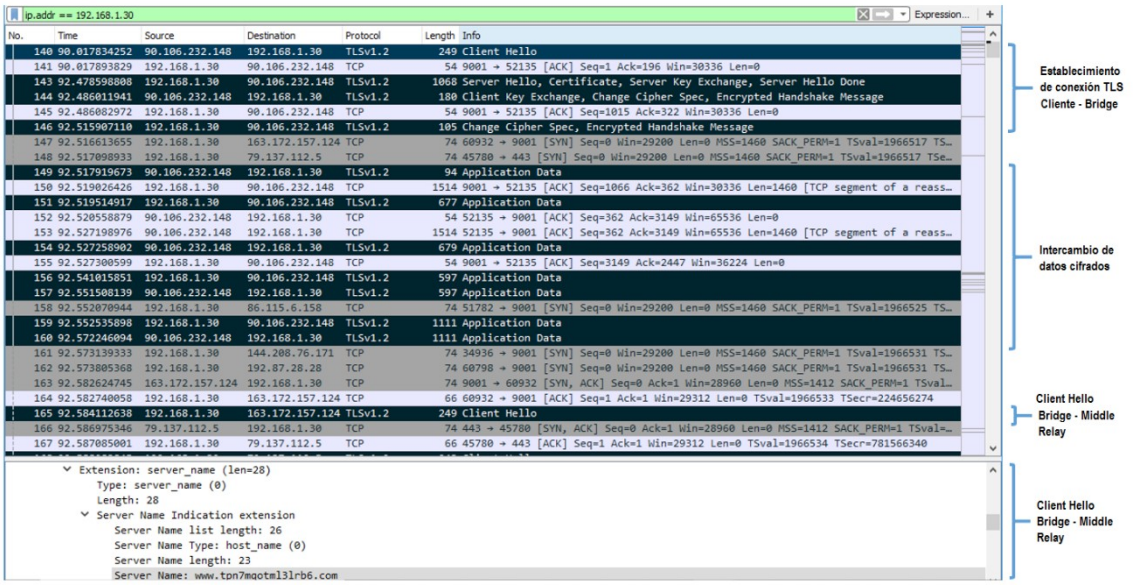


Figura 4.53 Wireshark: tráfico sin ofuscación (cliente-bridge)

En la figura 4.22 podemos identificar fácilmente el establecimiento de la conexión TLS de Tor. Vemos como la dirección IP “90.106.232.148”, que es la de nuestro cliente Tor, inicia la conexión con un “Client Hello” hacia la direccion IP “192.168.1.30” que es la dirección IP local de nuestro bridge relay.

En la parte inferior de la figura 4.22 vemos el desglose de la trama “Client Hello”, donde podemos ver que el nombre del servidor de destino es falso, algo carecterístico de Tor.

A continuación, el cliente y el bridge intercambian datos cifrados en los que entre ellos el cliente de Tor le indica al bridge relay cual sera el siguiente relay del camino, el middle relay.

Una vez intercambiados estos datos, el bridge relay inicia una conexión TLS con el middle relay, con un “Client Hello” hacia la dirección IP “163.172.157.124”, que será la de middle relay.

Para filtrar la captura y ver el tráfico entre el bridge y el middle relay podemos utilizar el filtro “ip.addr == 163.172.157.124” en wireshark. Obtenemos lo siguiente:

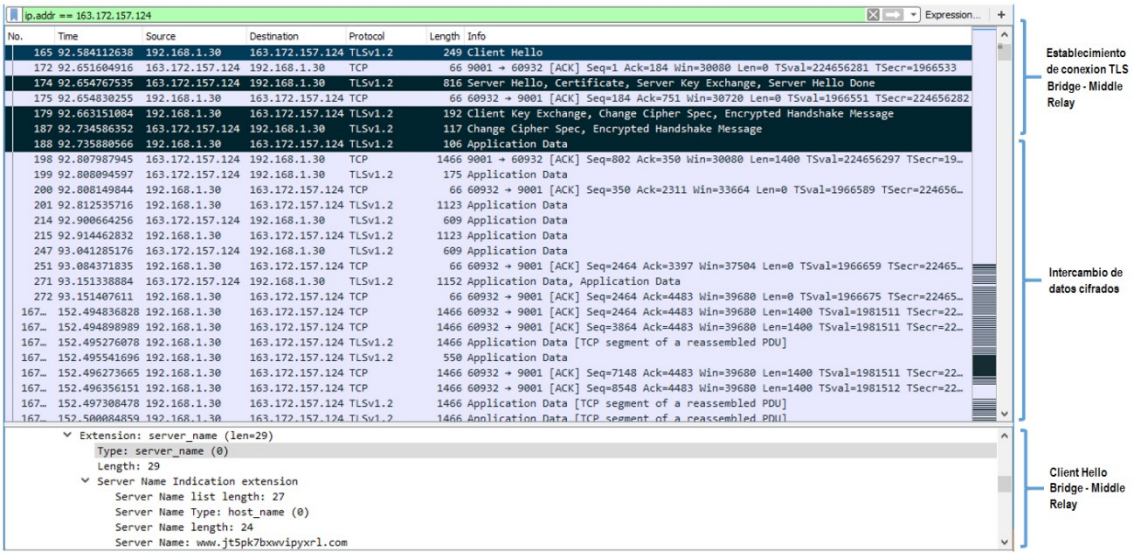


Figura 4.54 Wireshark: tráfico sin ofuscación (bridge-middle relay)

En la figura 4.23, tenemos el intercambio de tramas para el establecimiento de la conexión TLS entre el bridge relay y el middle relay, donde el bridge inicia la conexión TLS con una trama

“Client Hello”.

Como vemos no es muy difícil identificar la estructura del tráfico de Tor observando una captura del tráfico.

4.3.2 Detección y bloqueo de tráfico Tor por los censores

En un principio, la red Tor se creó como una forma de acceder a Internet de forma anónima, no estaba diseñado para eludir la censura de Internet. Esto queda claro si nos fijamos en la infraestructura básica de Tor. Un cliente Tor descarga una lista disponible públicamente con relays de entrada de Tor para conectarse. Bloquear Tor sería tan simple como automatizar la descarga de estas listas y agregar estos relays a una lista negra.

Cuando el proyecto Tor se dio cuenta de esto, crearon Tor Bridges, un proyecto que permite la existencia de relays Tor que no están en la lista pública. Estas listas se comparten de forma segura. Ya no se puede identificar los relays Tor a través de una lista pública. Pero cada vez se adaptan técnicas de filtrado más avanzadas.

Una de las técnicas de filtrado más recientes incluye el sondeo activo de los relays de Tor. Este método, diseñado para filtrar los bridges Tor, utiliza sistemas informáticos activos para verificar si la configuración de una conexión TLS es un relay Tor. Tor usa TLS, pero el protocolo de enlace puede identificarse como Tor debido a una estructura y conjunto de parámetros específicos.

Como hemos visto en el apartado anterior, la identificación del apretón de manos TLS de Tor se puede hacer fácilmente. Sí, abrimos Wireshark e iniciamos una conexión Tor, podemos ver un saludo del cliente con una estructura identificable. Además, el protocolo de enlace también incluye un nombre de dominio falso (como se ve en la figura 4.24). Tor intenta imitar el tráfico TLS del sitio web (actuando como una extensión que visita www.5rglbnpa3tck.com), pero esto no impide que se identifique.

Figura 4.55 Server name

Después de una posible identificación de una conexión TLS de Tor, podemos probar activamente el relay de entrada Tor para verificar que el servidor sea realmente un relay Tor. Esta prueba activa se realiza conectándonos al mismo host y puerto al que se conectó el ciudadano. Después de esto, el sistema simula un saludo de Tor y, si esto tiene éxito, se desconectará del ciudadano y la IP del servidor se agregará a una lista negra.

Otra técnica más reciente identificación se basa en considerar la longitud de los paquetes y los tiempos entre llegadas. Una forma curiosa de saber quién ha enviado y recibido mensajes es analizar los tiempos. Si el ordenador A ha enviado un paquete a las 18:19:01 y 3 milisegundos, y 300 milisegundos más tarde el ordenador B ha recibido otro paquete, y se repite el patrón de latencia varias veces, es muy probable que A y B estén conectados entre sí. **[43]**

Como puede concluir, la red Tor ya ha recorrido un largo camino para mejorar la elusión de la censura en Internet. Cosas como el proyecto Tor Bridges ya han mejorado la infraestructura central de Tor para ser más resistente contra el bloqueo básico de Tor.

Sin embargo, el "vector de ataque" para los países a los que les gusta prohibir Tor es todavía bastante grande. Los principales problemas incluyen la conexión TLS identificable de Tor, que permite distinguir el tráfico de Tor debido a las técnicas distintivas de inspección profunda de paquetes.

Estos problemas de seguridad los abordan los transportes conectables, ya que intentan camuflar el tráfico Tor y tienen como objetivo resolver estos problemas de identificación de Tor.

En el siguiente apartado, veremos una captura del protocolo obfs4 donde vemos que es mucho más difícil identificar tráfico Tor. **[23]**

4.3.3 Con ofuscación

En este apartado vamos a realizar una captura de la conexión entre el software Tor y nuestro bridge relay utilizando el transporte conectable Obfs4.

Iniciamos wireshark y capturamos tráfico de Tor utilizando el protocolo obfs4, para ello iniciamos desde un navegador Tor Browser una conexión Tor utilizando el protocolo obfs4 como hicimos en el apartado anterior 4.3.1.

Captura de tráfico Tor obfs4:

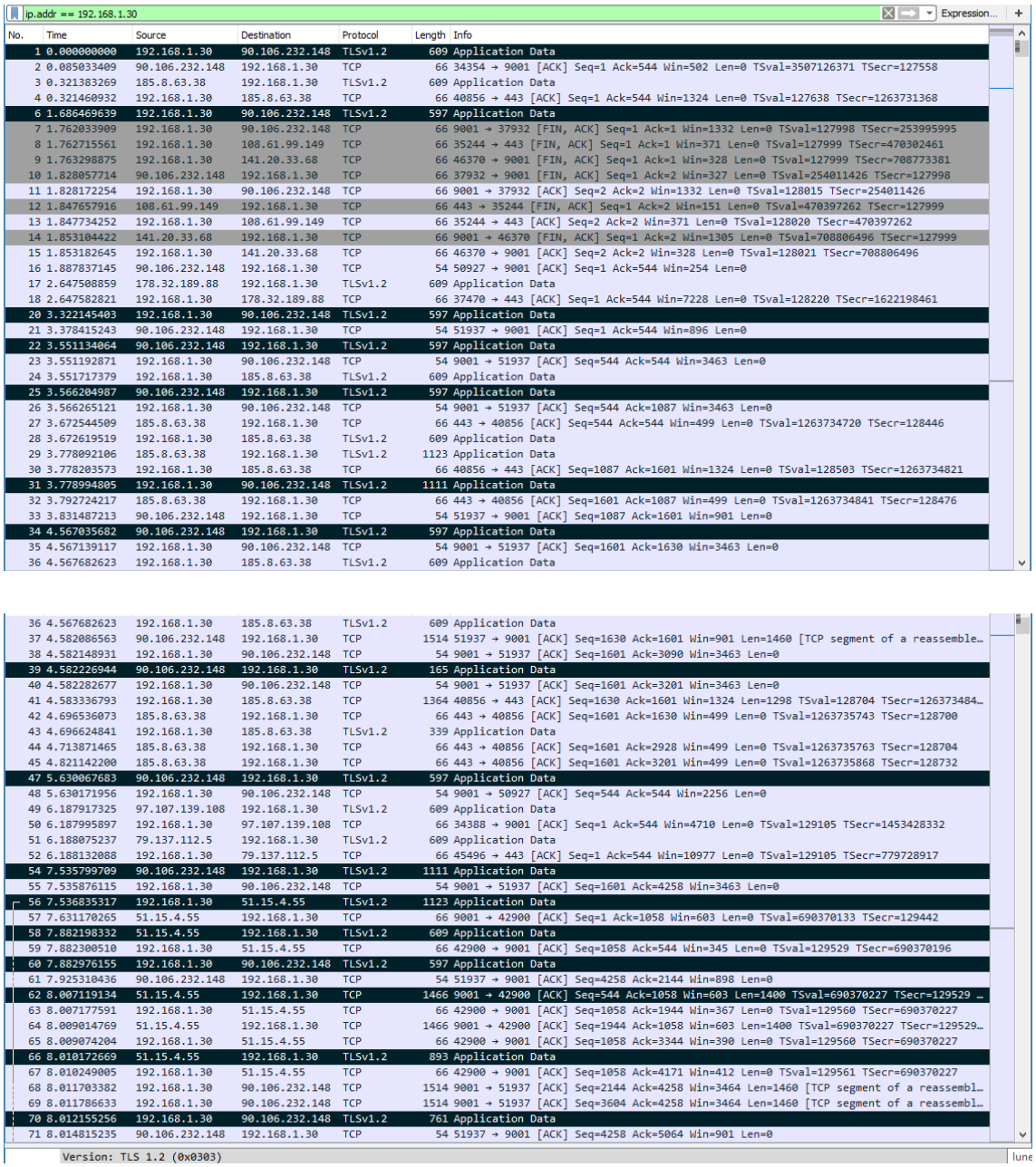


Figura 4.56 Wireshark: Tráfico Tor obfs4

Al contrario que sin utilizar un transporte conectable, en esta captura de tráfico, utilizando obfs4, no podemos identificar tráfico de Tor a simplevista.

Para filtrar la captura y ver el tráfico entre software Tor y el bridge relay podemos utilizar el filtro “ip.addr == 90.106.232.148 && ip-addr == 192.168.1.30” en wireshark. Obtenemos lo siguiente:

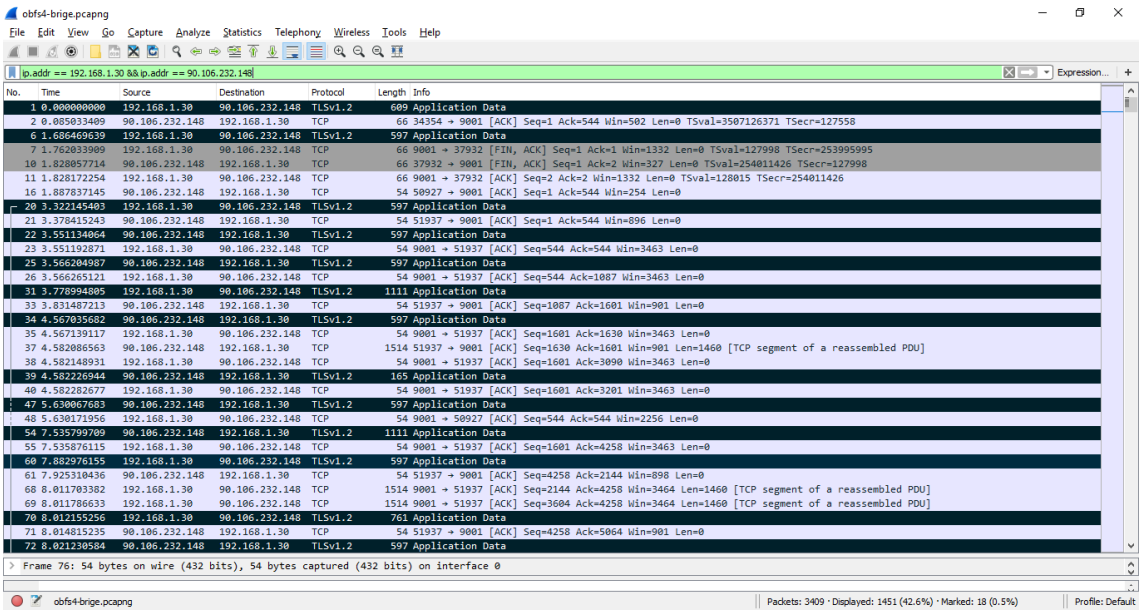


Figura 4.57 Wireshark: Tráfico obfs4 (cliente - bridge)

En el tráfico obfs4 entre el cliente y el bridge relay, ya no podemos ver claramente la estructura de establecimiento de una conexión TLS como si veíamos sin utilizar ningún transporte conectable. Ahora vemos tráfico de aspecto inocente entre los dos, pero no tenemos ninguna estructura ni parámetros específicos para relacionarlo con tráfico real de Tor.

4.4 Problemas técnicos en el desarrollo

En la ejecución de este proyecto nos ha surgido algún que otro problema que explicamos a continuación:

IP dinámica

Nuestro proveedor de internet nos proporciona una IP dinámica que cambia aproximadamente una vez a la semana por lo que hemos podido experimentar.

Esto nos afecta a la hora de conectarnos a la red Tor a través de nuestro bridge porque no tendrá la misma IP siempre y deberemos cambiar la IP de la línea del bridge de la configuración de Tor desde el navegador que nos estemos conectando.

El problema fue que de repente de un día para otro intentábamos conectarnos a través de nuestro bridge y no funcionaba. Pensando que el problema venia del bridge, probamos reiniciando los servicios, revisando la configuración y los log, los puertos abiertos en el router… y seguía sin funcionar hasta que nos dimos cuenta que el problema era que la IP pública de nuestra red había cambiado y había que modificar la IP pública en la línea del bridge en la configuración del Tor Browser porque estábamos indicando que nuestro bridge estaba en una IP que ya no tenía.

La solución para el desarrollo del proyecto fue comprobar la IP cada vez, antes de intentar conectarnos a nuestro relay, pero hemos visto que por una cuota mensual nuestro ISP nos ofrece una IP fija.

Orange:

Servicios adicionales de Internet	concepto	precio sin impuestos	Península y Baleares IVA 21%	Canarias (IGIC) 7%	Ceuta IPSI 10%	Melilla IPSI 4%
Traquilidad Orange para tu ADSL (precio especial lanzamiento)	Cuota mensual	2,89 €	3,50 €	3,10 €	3,18 €	3,01 €
Traquilidad Orange para tu ADSL	Cuota mensual	4,00 €	4,84 €	4,28 €	4,40 €	4,16 €
Kit Internet Everywhere prepago (Canales no presenciales)	Precio Final	29,00 €	35,09 €	31,03 €	31,90 €	30,16 €
Kit Internet Everywhere prepago (Puntos de venta)	Precio Final	9,00 €	10,89 €	9,63 €	9,90 €	9,36 €
Tarifa diaria por defecto Internet Everywhere	Por día	2,97 €	3,59 €	3,18 €	3,27 €	3,09 €
Bono mensual Internet Everywhere	Cuota mensual	29,67 €	35,90 €	31,75 €	32,64 €	30,86 €
IP fija	Cuota mensual	12,00 €	14,52 €	12,84 €	13,20 €	12,48 €

Tabla 4.5 Cuotas servicios adicionales

Cambio de ISP en el transcurso del proyecto

En el transcurso del proyecto, en mi casa, que ha sido el lugar donde hemos realizado la parte práctica del proyecto, hicimos un cambio de proveedor de servicios de Internet.

El quehacer que nos provocó fue conectar a la nueva red el equipo donde instalamos nuestro bridge relay, darle de nuevo una dirección IP fija y abrir nuevamente los puertos para el funcionamiento de Tor.

Para dar una dirección IP fija accedemos a la configuración de LAN de nuestro nuevo router y asignamos una IP estática:

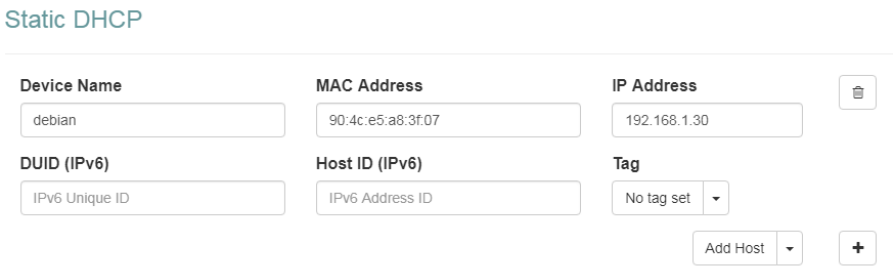


Figura 4.58 DHCP Estático

Para abrir los puertos accedemos a la configuración de red y abrimos los puertos necesarios:

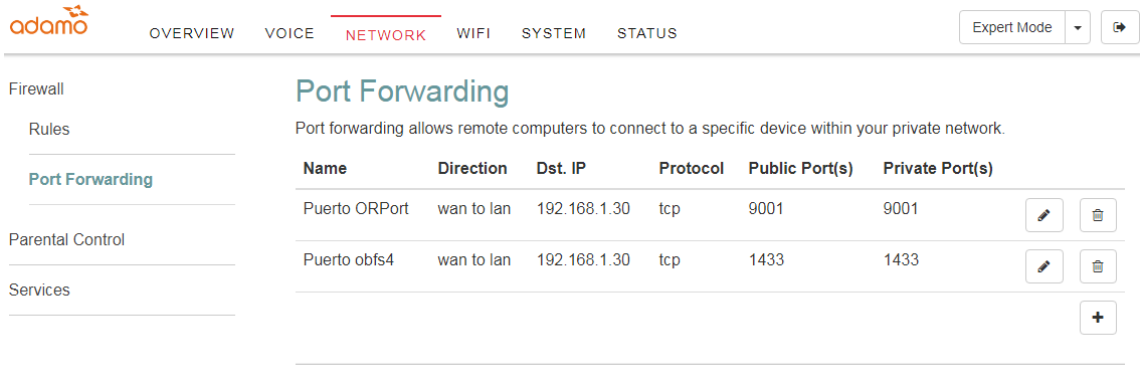


Figura 4.59 Asignación de puertos

Pero el cambio de ISP también nos dio una ventaja, el problema de la IP dinámica se solucionó. Aunque nuestro nuevo proveedor de Internet ofrece también una cuota para contratar una IP fija, hemos comprobado que desde hace unos meses que nos proporcionan los servicios de Internet nuestra dirección IP no ha variado ninguna vez.

Adamo:

Figura 4.60 IP fija

5 Acceso a Tor a través de un Bridge Relay

En este apartado vamos a ver como conectarnos a la red Tor a través de nuestro bridge.

Para acceder a Tor a través de nuestro bridge deberemos conseguir nuestra línea de bridge como hicimos en el apartado 4.1.4. Instalación Tor.

```
Bridge obfs4 <IP ADDRESS>:<PORT> <FINGERPRINT> cert=RAkY8RfGo8hU7WXXg0A6fPp1R6nVrcua8D
uQ1SCPmpjIKmLMhSe9LqjaRVp3hRBD1X6UKw iat-mode=0
```

Figura 5.61 Bridge line obfs4

Tendremos dos líneas de bridge diferentes dependiendo si queremos camuflar nuestro tráfico Tor utilizando el transporte conectable obfs4 o no:

-Con obfs4, utilizando la línea completa:

obfs4 <*IP del bridge*>:1433 3DBFD173EE603BCBB36DE89729CCF5556C3BD29F cert=RAkY8RfGo8hU7WXXg0A6fPp1R6nVrcua8DuQ1SCPmpjIKmLMhSe9LqjaRVp3hRBD1X6UKw iat-mode=0

-Sin transporte conectable, bastará con una parte de la línea y cambiando el puerto obfs4 por el ORPort:

<*IP del bridge*>:9001 3DBFD173EE603BCBB36DE89729CCF5556C3BD29F

5.1 Tor Browser

Tor Project ofrece un navegador que integra el software necesario para conectarse a la red Tor, llamado Tor Browser.

El Tor Browser es un navegador libre y de código abierto que permite el anonimato en línea y evasión de censura. A diferencia de otros navegadores, el Navegador Tor: **[44]**

- Ofrece anonimato en línea ocultado las direcciones IP de los usuarios
- Evade la censura en línea permitiendo a los usuarios accedan a sitios web o páginas web bloqueados
- No incluye funciones de rastreo en línea por defecto
- No gana dinero con la información de los usuarios
- Funciona y está recomendado por algunos de los expertos en seguridad más renombrados del mundo

El Navegador Tor inicia automáticamente los procesos en segundo plano de Tor y enruta el tráfico a través de la red Tor. Al finalizar una sesión, el navegador borra los datos confidenciales de la privacidad, como las cookies HTTP y el historial de navegación. **[45]**

El Navegador Tor está disponible para usar Tor en Microsoft Windows, Apple MacOS o GNU / Linux. Es un software portable, que no hace falta instalar en el equipo, directamente lo podemos ejecutar. **[46]**

Tor Browser

En primer lugar, descargamos desde la página de Tor Project (https://www.torproject.org) el instalador del navegador Tor en el idioma y para el sistema operativo correspondiente.

Una vez descargado, lo ejecutamos.

En la primera ventana que nos encontramos, tenemos la opción “Conectar” y la opción “Configurar”.

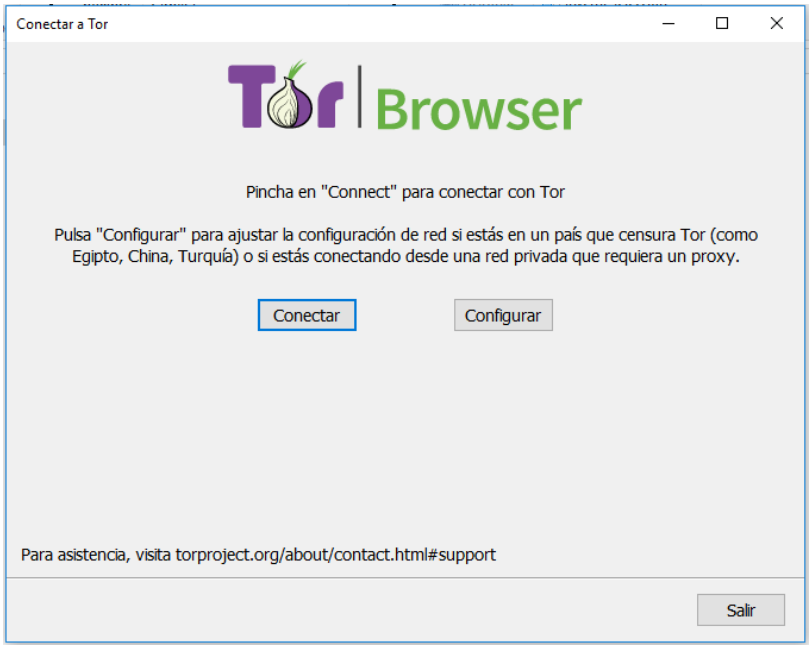


Figura 5.62 Tor Browser 1

En la opción “Conectar” nos conectaremos a Tor de manera convencional, estableciendo un circuito a través de un entry guard listado en los directorios de Tor.

En la otra opción, “Configurar”, es donde podemos configurar la opción de establecer un circuito Tor a través de un bridge relay. Por lo que pulsamos en “Configurar”.

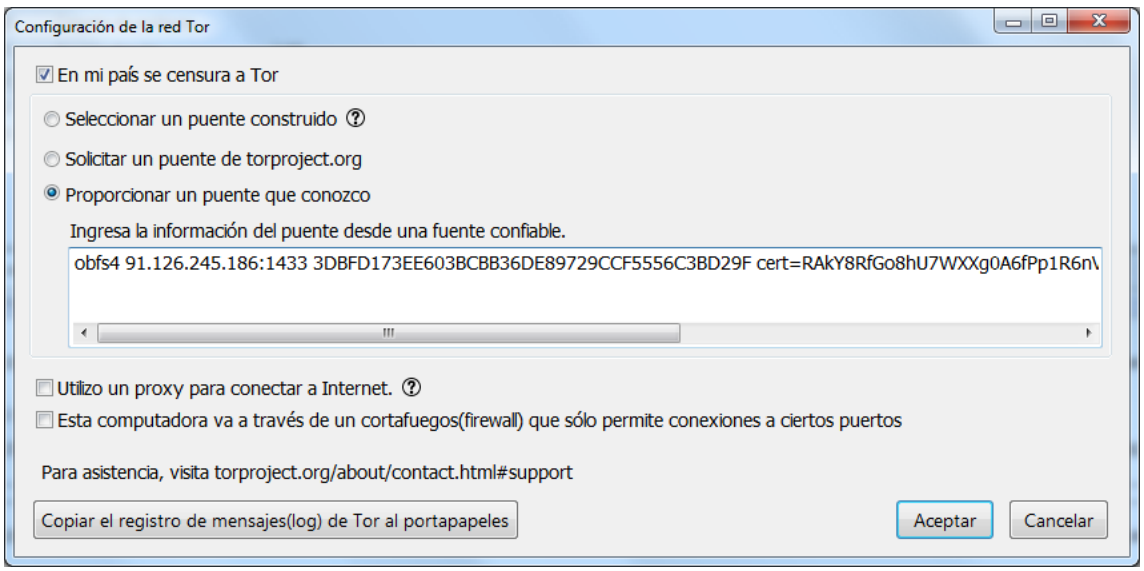


Figura 5.63 Tor Browser: Proporcionar un puente que conozco obfs4

En la ventana de configuración marcamos la opción “En mi país se censura a Tor” donde tenemos dos opciones:

- “Seleccionar un puente construido”: donde Tor nos asignará un bridge ya construido.

- “Proporcionar un puente que conozco”, la opción que a nosotros nos interesa, donde podemos introducir la línea de un bridge que conozcamos, en nuestro caso, introducimos la de nuestro propio bridge, en este caso utilizando obfs4, y hacemos clic en conectar y en unos minutos se establecerá un circuito a través de Tor en el que el primer relay del camino será nuestro bridge.

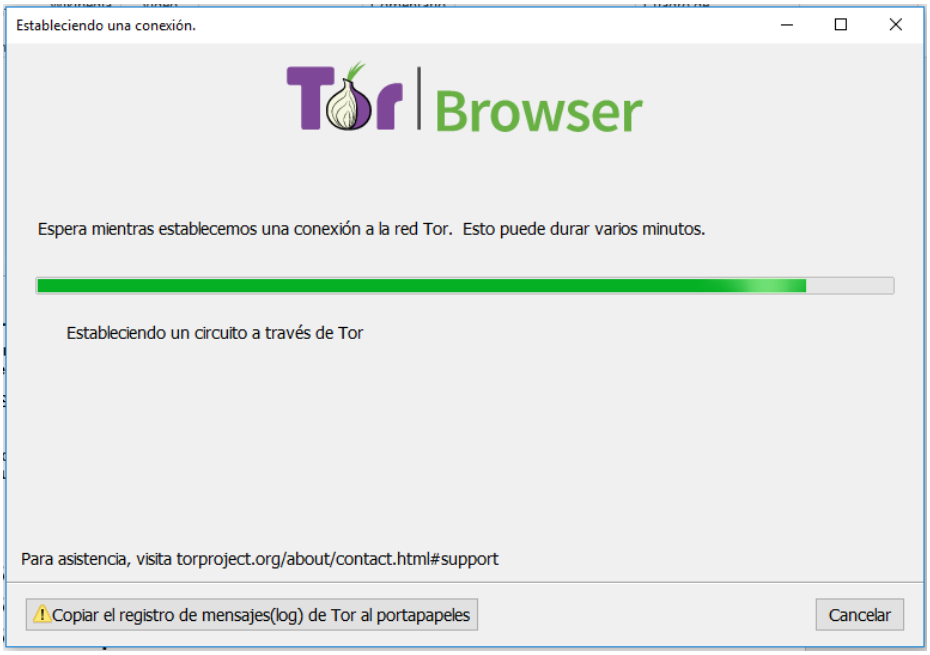


Figura 5.64 Tor Browser: Estableciendo un circuito a través de Tor

Una vez termine el proceso de establecimiento del circuito, ya tendremos listo el navegador Tor para navegar por la red Tor.

Para ver más detalles de la conexión podemos ver el Log de Tor yendo a la pestaña “Configuración de la red TOR” y pulsando en “Copiar el registro de mensajes(log) de Tor al portapapeles”.

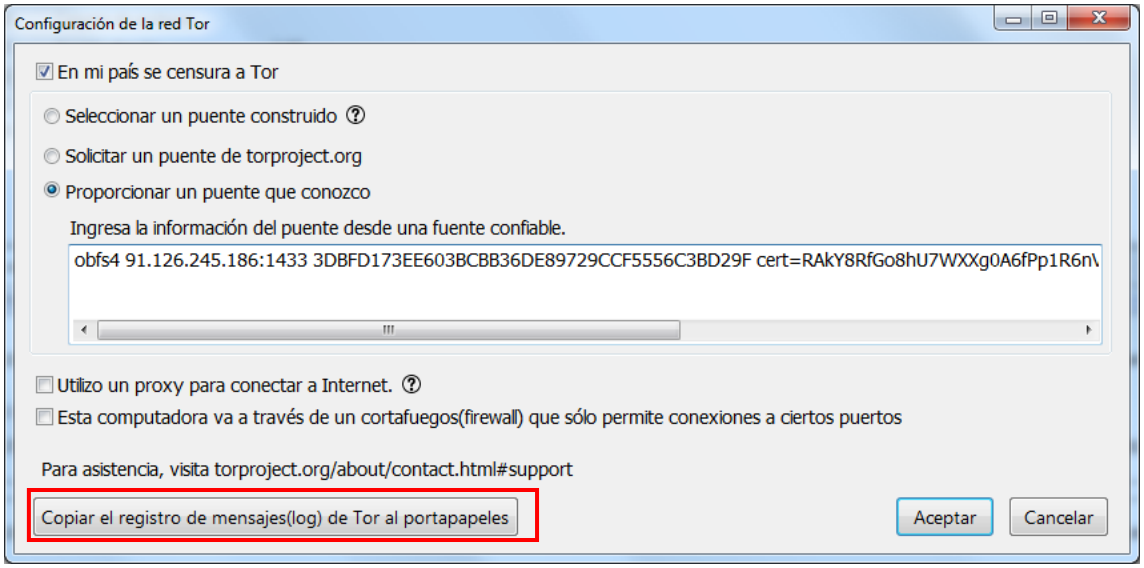


Figura 5.65 Tor Browser: Registro de mensajes

Donde vemos algo parecido a lo siguiente:

```
7/6/2018 10:14:36 AM.600 [NOTICE] Switching to guard context "bridges" (was using "default")
7/6/2018 10:14:36 AM.600 [NOTICE] Opening Socks listener on 127.0.0.1:9150
7/6/2018 10:14:36 AM.600 [NOTICE] Renaming old configuration file to "C:\Users\lfgs9\Desktop\Tor Browser\Browser\TorBrowser\Data\Tor\torrc.orig.1"
7/6/2018 10:14:43 AM.500 [NOTICE] Bootstrapped 5%: Connecting to directory server
7/6/2018 10:14:43 AM.500 [NOTICE] Bootstrapped 10%: Finishing handshake with directory server
7/6/2018 10:14:43 AM.600 [NOTICE] Bootstrapped 15%: Establishing an encrypted directory connection
7/6/2018 10:14:43 AM.600 [NOTICE] Bootstrapped 20%: Asking for networkstatus consensus
7/6/2018 10:14:43 AM.600 [NOTICE] new bridge descriptor 'FernandoTFG' (fresh): $69DF7795B409B0B7102B07E038A3CF31CCCE6BF9~FernandoTFG at 91.126.245.186
7/6/2018 10:14:44 AM.100 [NOTICE] Bootstrapped 25%: Loading networkstatus consensus
7/6/2018 10:14:45 AM.800 [NOTICE] Bootstrapped 40%: Loading authority key certs
7/6/2018 10:14:47 AM.400 [NOTICE] Bootstrapped 45%: Asking for relay descriptors
7/6/2018 10:14:47 AM.600 [NOTICE] Bootstrapped 50%: Loading relay descriptors
7/6/2018 10:14:49 AM.200 [NOTICE] Bootstrapped 55%: Loading relay descriptors
7/6/2018 10:14:49 AM.800 [NOTICE] Bootstrapped 63%: Loading relay descriptors
7/6/2018 10:14:49 AM.800 [NOTICE] Bootstrapped 68%: Loading relay descriptors
7/6/2018 10:14:49 AM.800 [NOTICE] Bootstrapped 73%: Loading relay descriptors
7/6/2018 10:14:49 AM.900 [NOTICE] Bootstrapped 80%: Connecting to the Tor network
7/6/2018 10:14:51 AM.400 [NOTICE] Bootstrapped 90%: Establishing a Tor circuit
7/6/2018 10:14:52 AM.200 [NOTICE] Tor has successfully opened a circuit. Looks like client functionality is working.
7/6/2018 10:14:52 AM.200 [NOTICE] Bootstrapped 100%: Done
7/6/2018 10:14:54 AM.500 [NOTICE] New control connection opened from 127.0.0.1.
7/6/2018 10:14:54 AM.600 [NOTICE] New control connection opened from 127.0.0.1.
```

Podemos ver que nos estamos conectando a nuestro bridge “FernandoTFG”.

Desde el navegador TOR, si realizamos una busca en la web podemos ver información del circuito a través del cual nos estamos conectando a un sitio web.

En la figura 5.6 podemos ver que, al acceder a ese sitio web, el primer relay de nuestro camino, “Repetidor puente(bridge): Obfs4 (91.126.245.186)”, es nuestro bridge, el segundo relay es un middle relay situado en Alemania y el exit relay está situado en Estados Unidos.

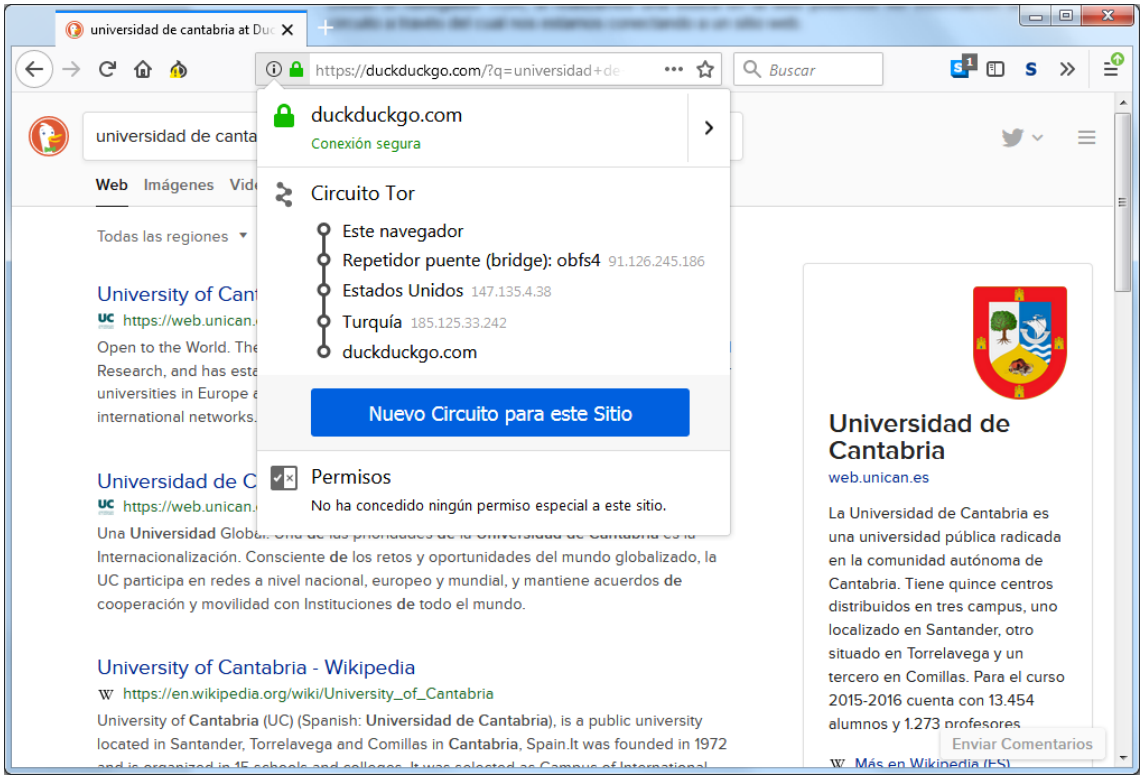


Figura 5.66 Tor Browser: Circuito Tor obfs4

Hemos visto como conectarnos a la red Tor a través de nuestro bridge relay utilizando el transporte conectable obfs4, ahora vamos a ver como conectarnos sin utilizar ningún transporte conectable, con una conexión TLS Tor identificable por un observador.

En la ventana de configuración introducimos la línea de nuestro propio bridge, en este caso sin utilizar obfs4, y hacemos clic en conectar y en unos minutos se establecerá el circuito Tor.

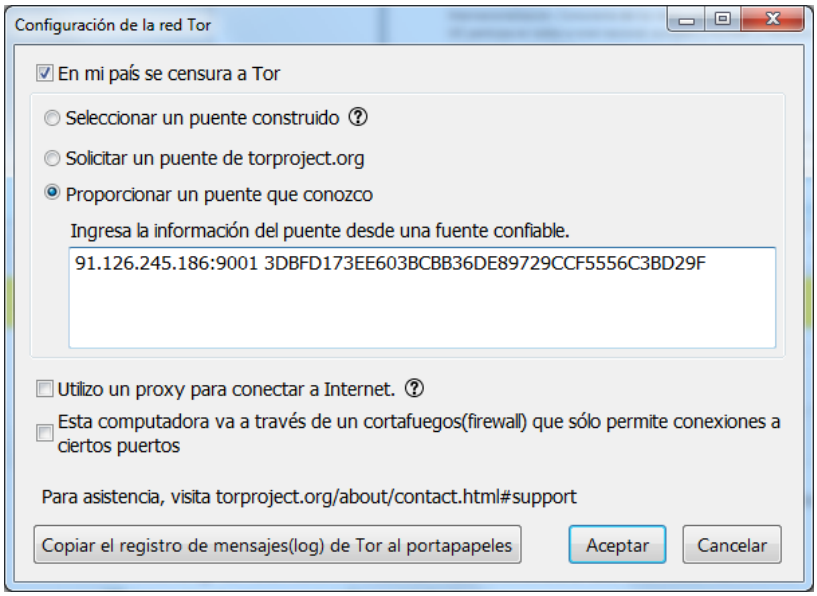


Figura 5.67 Tor Browser: Proporcionar un puente que conozco

Desde el navegador TOR podemos ver información del circuito a través del cual nos estamos conectando a un sitio web.

En la figura 5.8 podemos ver que el primer relay de nuestro camino, “Repetidor puente(bridge): 91.126.245.186”, es nuestro bridge, el segundo relay es un middle relay situado en Alemania y el exit relay está situado en Suecia.

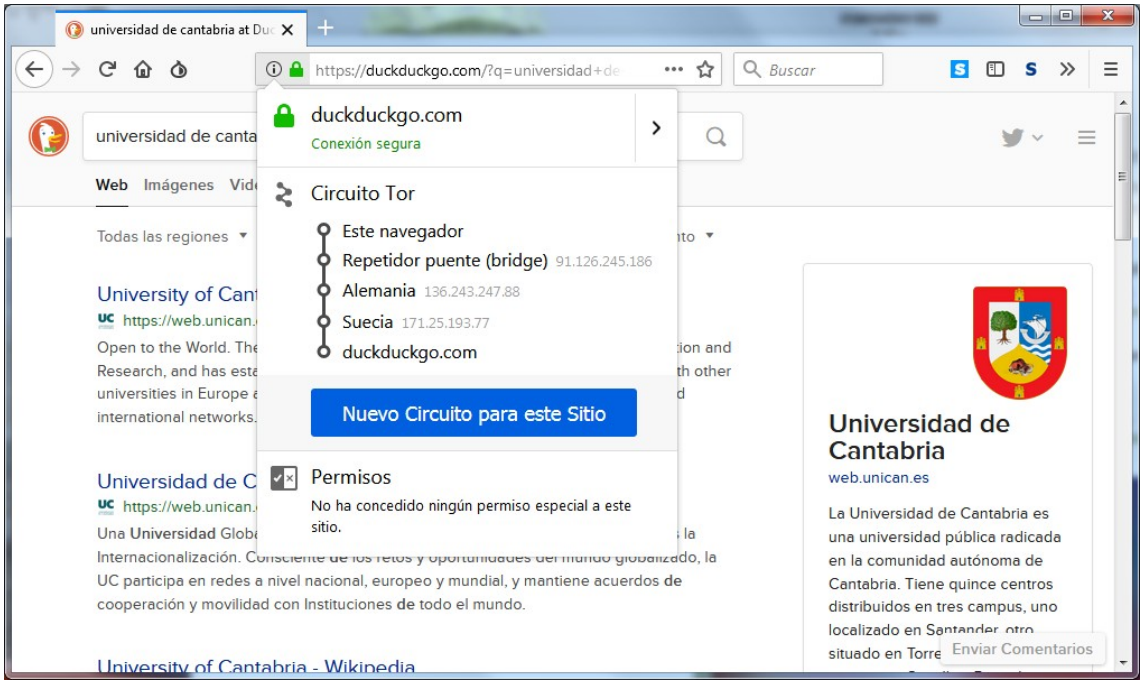


Figura 5.68 Tor Browser: Circuito Tor

5.2 Navegador común

Este apartado está orientado a personas que no están a gusto utilizando el Tor browser y prefieren su navegador convencional, pero con las ventajas de Tor.

Utilizaremos la aplicación Tor instalada en el mismo ordenador como un servidor proxy que hará de intermediario entre nuestro ordenador y los servidores externos a los que enviamos solicitudes y peticiones, por lo que todo nuestro tráfico irá por la red Tor.

Lo primero es instalar TOR y configurar nuestro bridge, igual que en el apartado anterior. Deberemos dejar activo Tor durante todo el proceso.

Lo siguiente será ir a Panel de control – Opciones de Internet – Conexiones – Configuración de LAN, donde marcaremos la opción: “Usar un servidor proxy para la LAN”

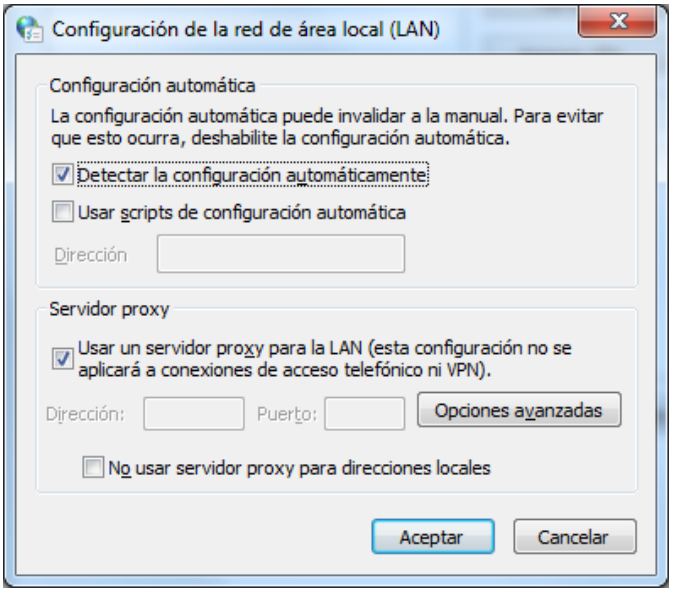


Figura 5.69 Configuración LAN Windows

A continuación, vamos a Opciones avanzadas y rellenamos el campo "Socks" con los siguientes valores:

Dirección: 127.0.0.1

Puerto: 9150

Tor utiliza por defecto la dirección 127.0.0.1 para esperar conexiones locales de aplicaciones. Podemos modificar esta dirección en el archivo de configuración torrc mediante la variable SocksListenAddress (ej. SocksListenAddress 192.168.0.1). Nosotros no hemos configurado esa variable en el archivo torrc por eso utilizamos la dirección por defecto.

Tampoco hemos configurado en el archivo torrc el puerto para la escucha de conexiones locales de aplicaciones, así que utilizamos el que Tor utiliza por defecto el 9150. Si quisiéramos utilizar otro utilizaríamos la variable **SocksPort** (ej. SocksPort 9005). **[47]**

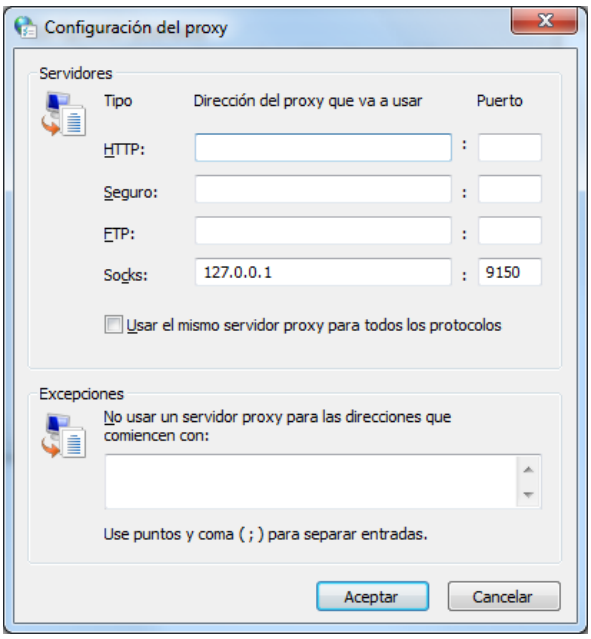


Figura 5.70 Configuración de proxy Windows

Ahora podemos usar nuestro navegador habitual (Google Chrome, Firefox, Opera, etc) y nos estaremos conectando a la red Tor a través de nuestro bridge.

Podemos comprobar por ejemplo desde “el navegador Opera” como cambia nuestra dirección IP:

Con la configuración de proxy por defecto:

Figura 5.71 Navegador Opera: IP real

Utilizando Tor:

Figura 5.72 Navegador Opera IP Tor

5.2.1 Chrome

Desde Google Chrome tenemos otras opciones:

Extensión de Chrome Proxy SwitchyOmega

Proxy SwitchyOmega es una extension para Google Chrome que nos permite cambiar la “configuracion de proxy “ unicamente para Chrome, sin afectar al resto de navegadores

instalados en nuestro ordenador como pasaria si configurariamos el proxy desde el panel de control del sistema operativo. **[48]**

Utilizaremos la aplicación Tor instalada en el mismo ordenador como un servidor proxy que hará de intermediario entre nuestro navegador Google Chrome y los servidores externos a los que enviamos solicitudes para navegar por las páginas web.

Lo primero que haremos es instalar TOR y configurar nuestro bridge igual que en el apartado anterior. Deberemos dejar activo Tor durante todo el proceso.

A continuación, desde Google Chrome instalamos la extensión de Chrome Proxy SwitchyOmega que lo encontramos gratuitamente en la tienda virtual de Chrome.

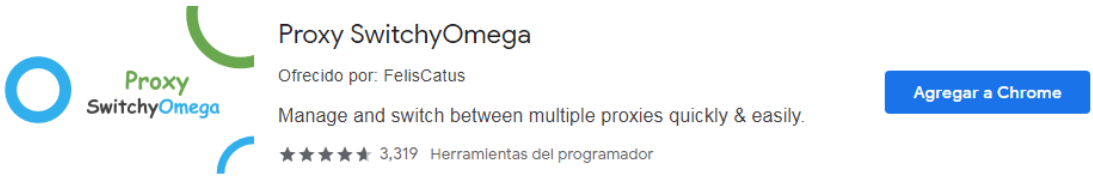


Figura 5.73 Proxy SwitchyOmega

Pulsamos en Agregar a Chrome y nos aparecerá en la parte superior derecha el botón de la extensión de Proxy.

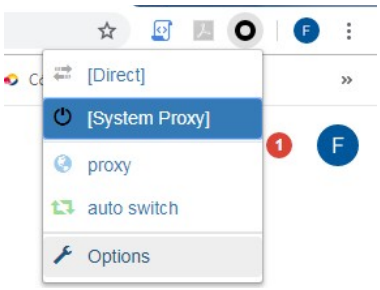


Figura 5.74 Botón SwitchyOmega

A continuación, vamos a opciones donde creamos un nuevo perfil llamado Tor con la siguiente configuración y aplicamos los cambios:

Protocolo: SOCKS5

Servidor: 127.0.0.1

Puerto: 9150

Figura 5.75 Configuración Proxy SwitchyOmega

Para activar y desactivar el proxy Tor lo haremos desde el siguiente menú:

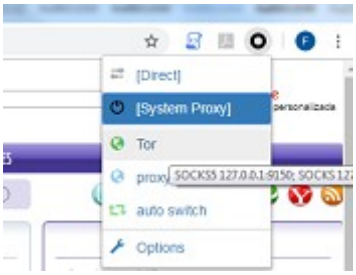


Figura 5.76 ON/OFF SwitchyOmega

Podemos comprobar cómo cambia nuestra dirección IP:

Sin Proxy Tor:

Figura 5.77 SwitchyOmega IP real

Con Proxy Tor:

Figura 5.78 SwitchyOmega IP Tor

5.2.2 Firefox

A diferencia de Google Chrome, Firefox nos permite cambiar su "configuración de proxy" desde su propia configuración, sin instalar ninguna extensión.

Como en el apartado anterior necesitamos tener instalado Tor en el mismo ordenador y configurar nuestro bridge relay igual que en los apartados anteriores. Deberemos dejar activo Tor durante todo el proceso.

A continuación, abrimos el navegador Firefox y vamos al menú de configuración pulsando en "Opciones".

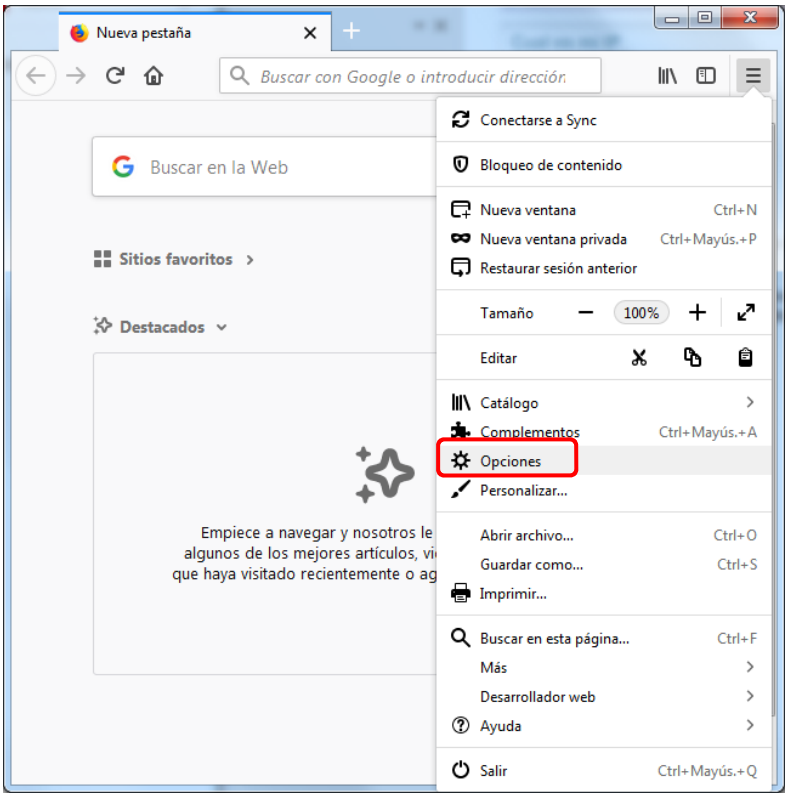


Figura 5.79 Firefox: Opciones

En la opción "Configuración de Red" pulsamos en "configuración"

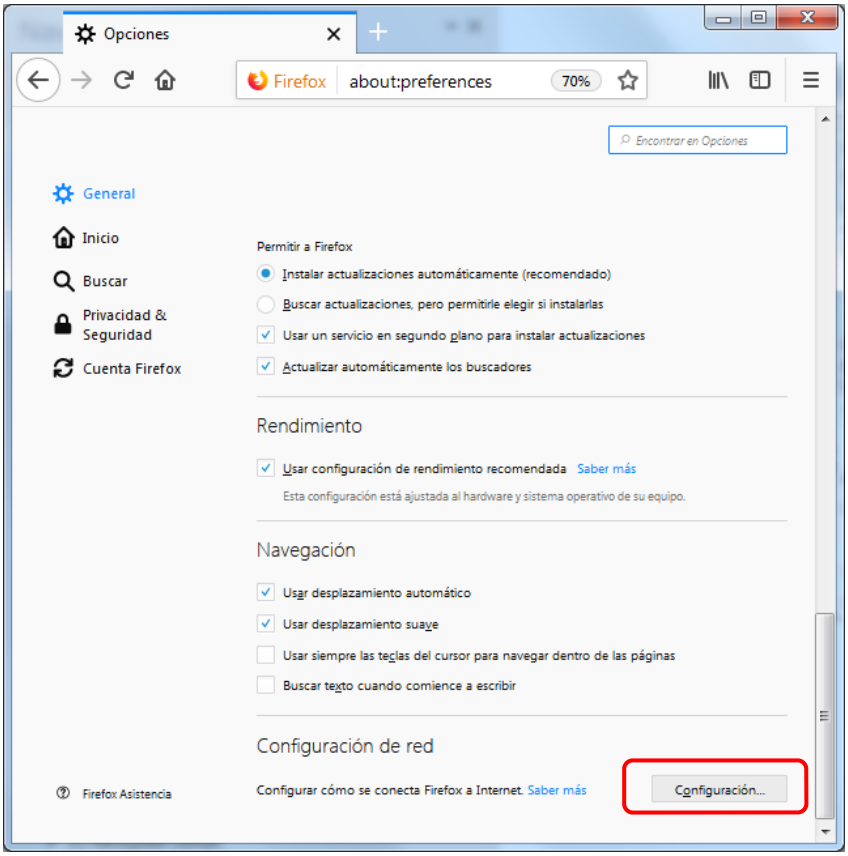


Figura 5.80 Firefox: Configuración

Configuramos la “Configuración manual del proxy”. Con los valores:

Host SOCKS: 127.0.0.1

Puerto: 9150

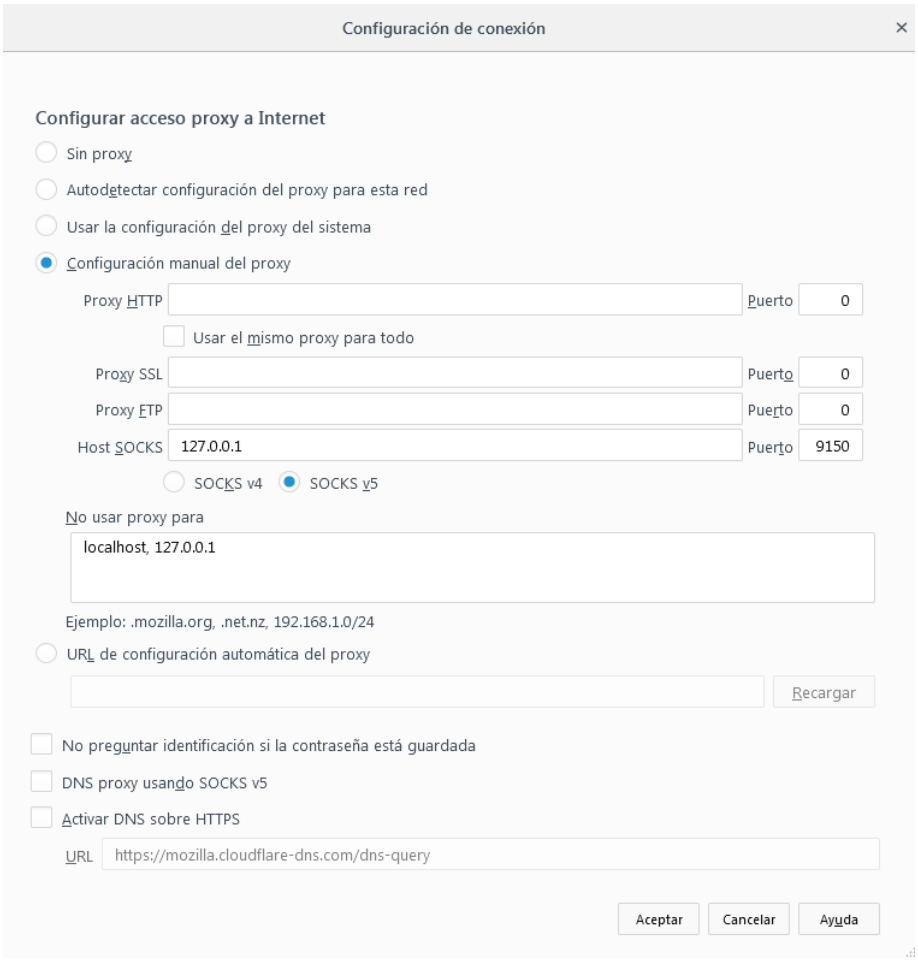


Figura 5.81 Firefox: Configuración proxy

Pulsamos en “Aceptar” y ya estaremos navegando utilizando Tor como proxy.

Podemos comprobar cómo cambia nuestra dirección IP:

Sin Tor:

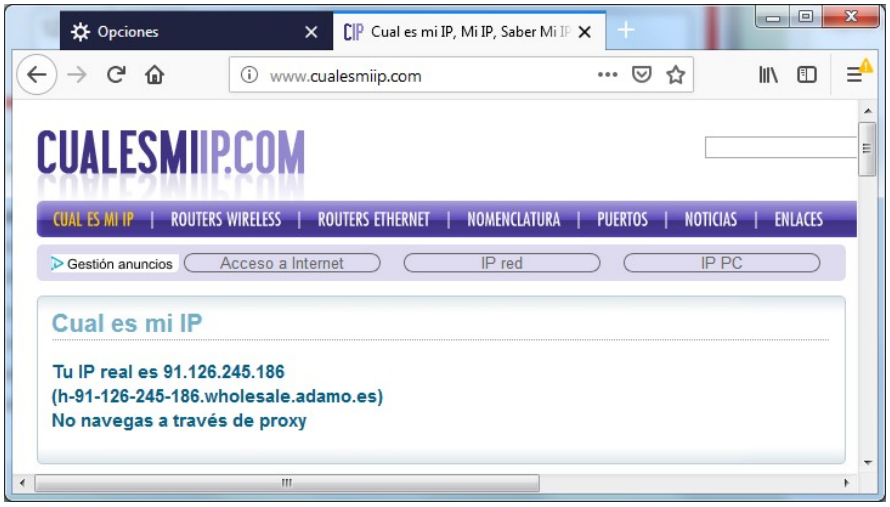


Figura 5.82 Firefox proxy IP real

Con Tor:

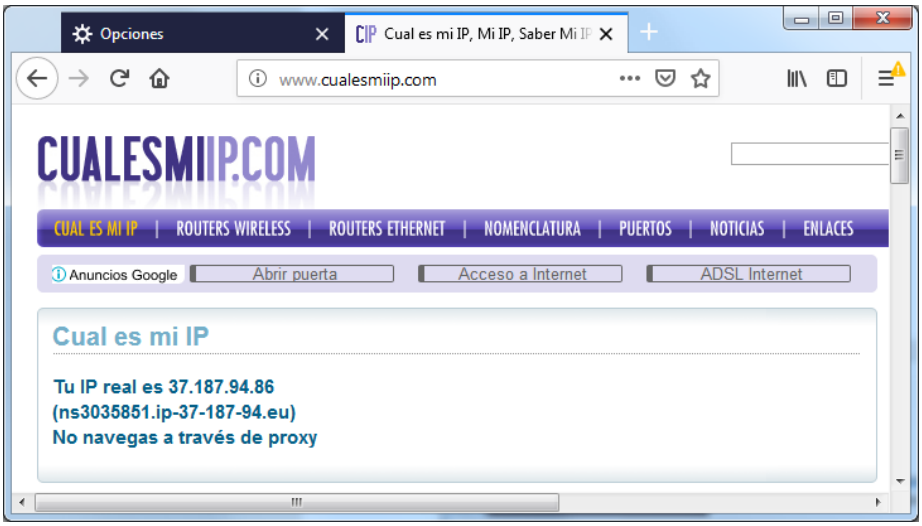


Figura 5.83 Firefox proxy IP Tor

5.3 Smartphone

En este apartado vamos a explicar cómo usar Tor en Android y conectarnos a través de nuestro bridge a la red Tor. Lo vamos a hacer mediante la aplicación Tor Browser para Android que acaba de ser lanzada en fase alpha, una versión de prueba para ponerla a punto de cara a su lanzamiento definitivo en 2019.

Lo primero que vamos a hacer es buscar la aplicación Tor Browser for Android en Google Play y pulsamos en instalar.

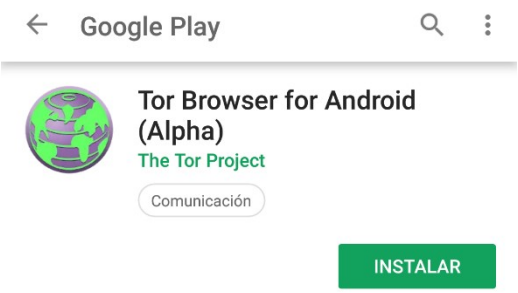


Figura 5.84 Tor Browser for Android

Una vez termine la instalación, abrimos la aplicación. Lo primero que nos dice al abrirla por primera vez es que necesitamos instalar también la aplicación Orbot Proxy con Tor, que está también en Google Play. La buscamos e instalamos.

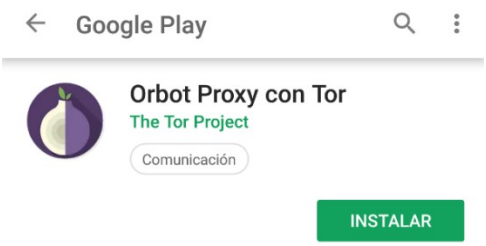


Figura 5.85 Orbot Proxy con Tor

Una vez instaladas las dos aplicaciones, abrimos Orbot Proxy.

En la aplicación en la parte superior derecha, tenemos el menú de configuración donde podemos hacer las configuraciones para acceder a Tor a traves de nuestro bridge, por lo que entramos en este menú.

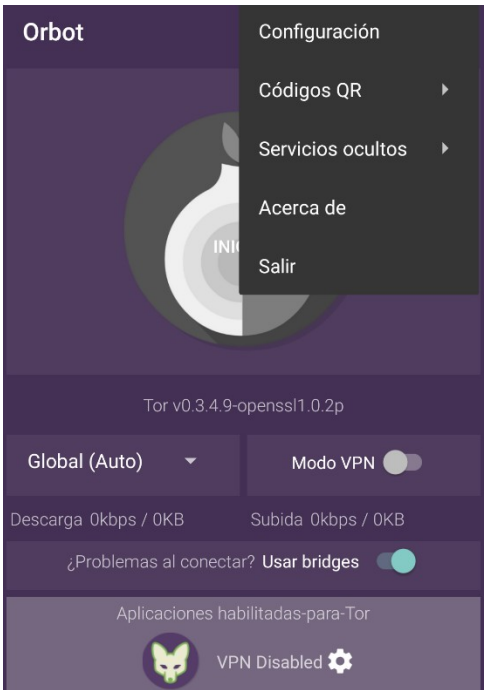


Figura 5.86 Orbot Proxy: Menú

En el menú de configuración debemos activar la ópcion “Use Bridges” y en “Bridges” introducimos la línea de nuestro bridge:

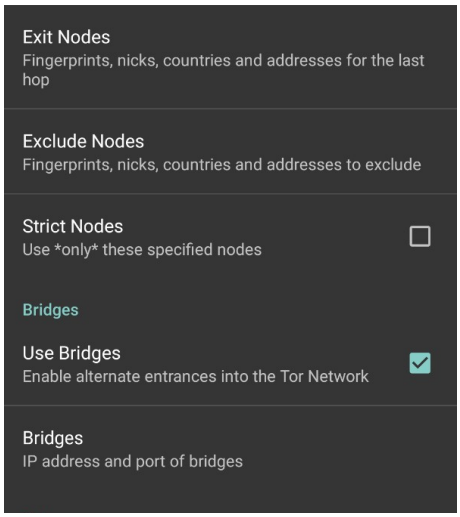


Figura 5.87 Orbot Proxy: Bridges

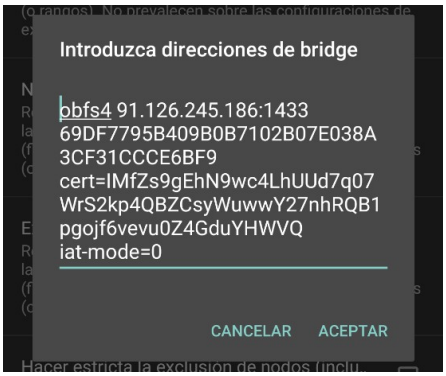


Figura 5.88 Orbot Proxy: bridge line

Por último, en la pantalla principal de la aplicación, pulsamos en el icono de la cebolla para iniciar la conexión.

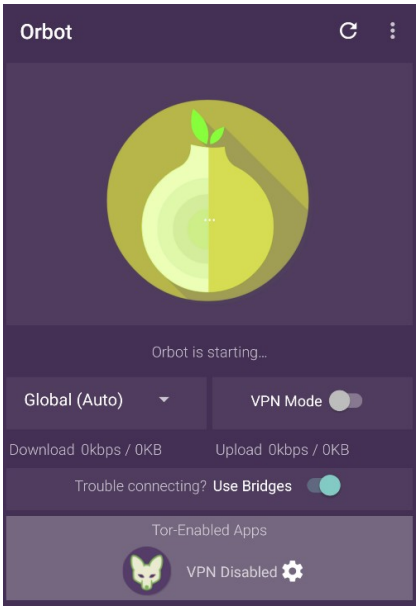


Figura 5.89 Orbot Proxy: Start

En los log de la aplicación podemos ver como el primer relay de todos los circuitos Tor es nuestro bridge “FernandoTFG”.

Figura 5.90 Orbot Proxy: Log

Una vez configurado el “Orbot Proxy” y dejarlo en funcionamiento ya podemos utilizar el navegador “Tor Browser for Android” y navegar por la red ocultando nuestra identidad. **[49]**

6 Conclusiones y líneas futuras

6.1 Conclusiones

Durante el transcurso de este proyecto hemos podido conocer un poco más la red Tor. Por suerte hemos podido investigar sus entresijos y con ello somos capaces de afirmar que es una invención muy útil para toda la sociedad. Hemos visto la gran variedad de usuarios que se benefician de la anonimidad y seguridad que da la red Tor y la diversidad de usos que tiene.

Centrándonos más en los bridge relays, tema principal de nuestro proyecto, hemos visto que son un elemento esencial contra la censura de la red Tor que hay por parte de muchos gobiernos. Continuamente se siguen desarrollando mejoras para los bridge relays, en forma de protocolos conectables, muy necesarios, dado que a medida que los censores avanzan en sus métodos de filtrado de tráfico Tor, es necesario mejoras para evitar la censura.

Por suerte, en España no hay bloqueo de conexiones directas de sus clientes con esta red, pero quien sabe… quizás en un futuro estemos en la misma situación que existe en muchos países del mundo actualmente.

Como hemos visto, la implementación de un bridge relay es sencilla y precisa de unos requerimientos mínimos bastante bajos que cualquiera podemos tener. Por lo que, para cualquier persona, sin demasiados conocimientos informáticos, podría implementar un bridge relay.

Otro punto importante de la red Tor es la variedad de acceso que tenemos. Las opciones van desde acceder desde su propio navegador, Tor Browser, o por medio de otros navegadores como pueden ser Google Chrome, Firefox, Opera… utilizando Tor como un proxy para conectarse a la red.

También Tor se adentra en el mundo de los dispositivos móviles y nos da una opción muy útil actualmente para conectarnos a Tor desde un Smartphone, descargando una aplicación que hará de proxy y el navegador Tor Browser.

Para finalizar, debemos decir que esperamos que la red Tor siga evolucionando y ayudando diariamente a muchas personas que viven detrás de una censura de Internet y sin esta red no podrían acceder a Internet.

6.2 Líneas futuras

Las posibles continuaciones de este proyecto podrían ir en dos sentidos:

La primera puede ser la implementación de un Exit relay y analizar el tráfico que pasa por él. Si capturamos en un exit relay podemos ver la comunicación entre el middle relay y el exit relay que será un enlace cifrado con TLS, y la conexión entre en exit relay y el servicio web, que será un enlace no cifrado. Podría ser interesante analizar esta transformación de tráfico de Tor.

El segundo camino por el que seguir en las investigaciones futuras pasaría por el estudio más profundo de los diferentes transportes conectables, comparando Obfs3, Obfs4, ScrambleSuit, FTE y Meek, por ejemplo. Existe la posibilidad de implementar un bridge relay en el que instalemos cada uno de los transportes conectables que queremos capturar o simplemente utilizar bridges que nos proporcione la red Tor y capturar el tráfico desde el cliente para a continuación analizar y comparar los protocolos.

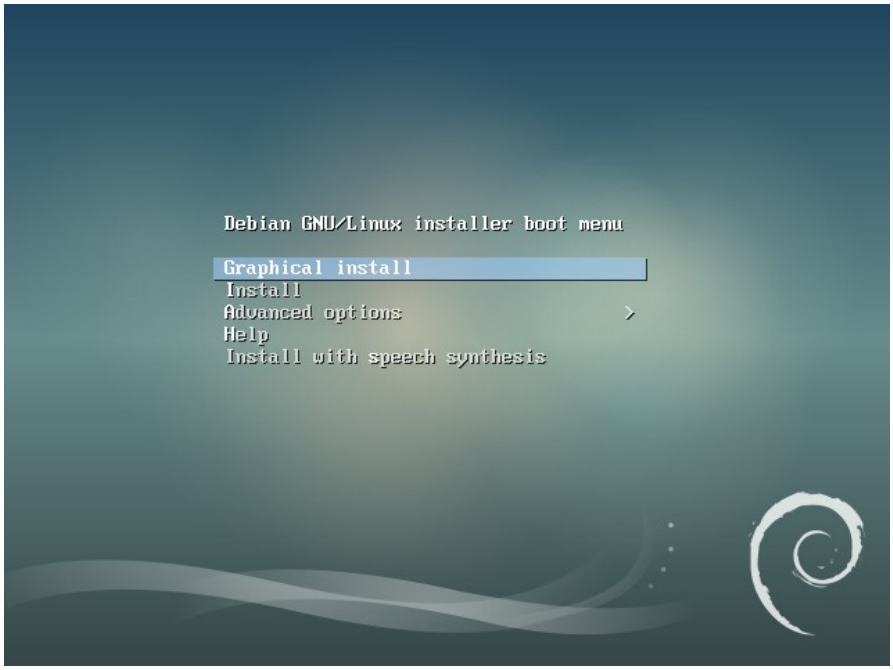
Anexo

Instalación de Debian

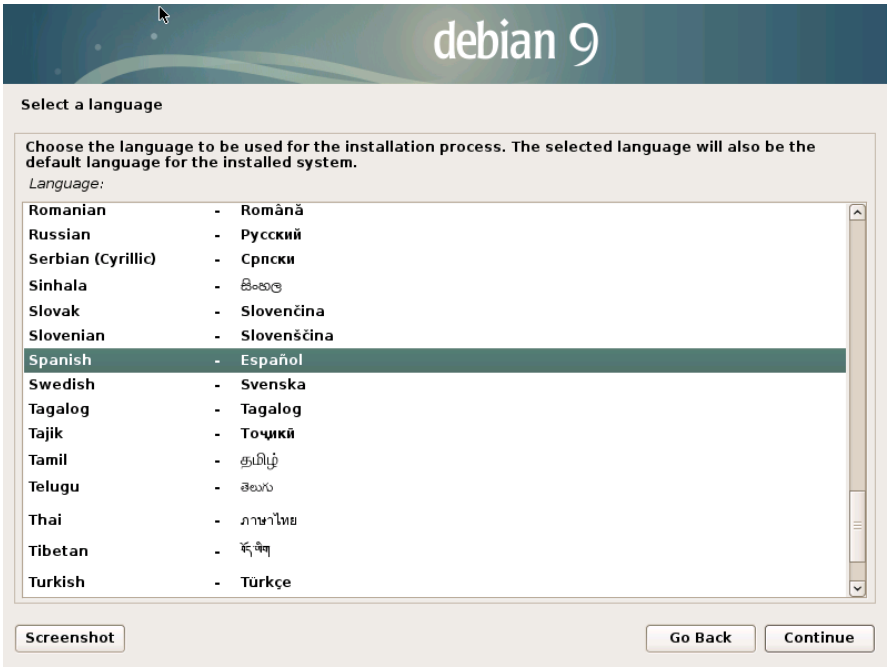
Lo primero, descargamos una imagen pequeña de Debian 32 bits desde la página Web oficial **[3]** y desde un ordenador Windows con la ayuda de un software booteador creamos un medio de instalación de Debian en un USB, en este caso hemos utilizado UNetbootin.

Una vez esté listo el USB lo pincharemos en nuestro Netbook y arrancaremos el equipo desde el USB.

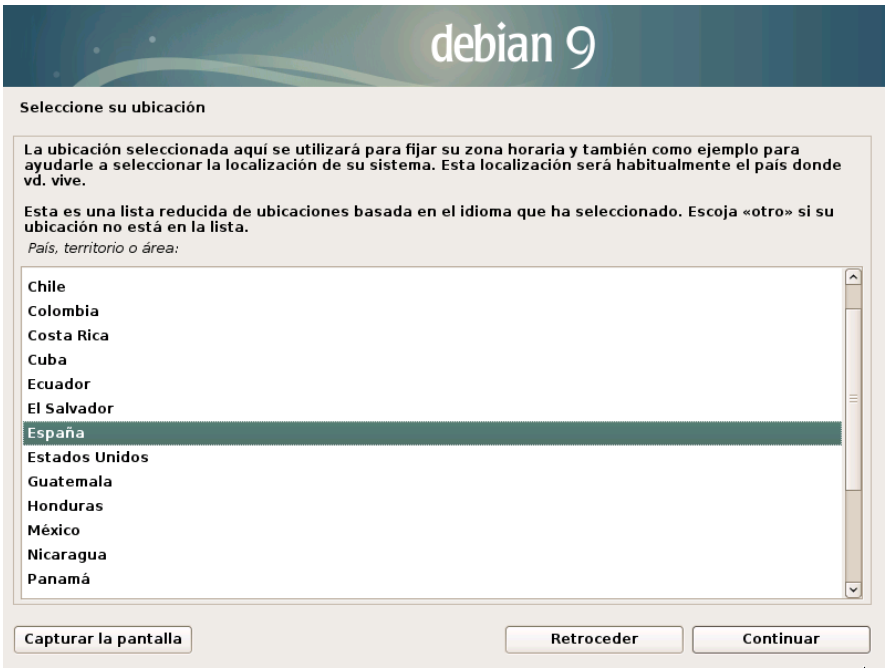
En la primera pantalla se puede elegir el método de instalación y elegimos la instalación gráfica.



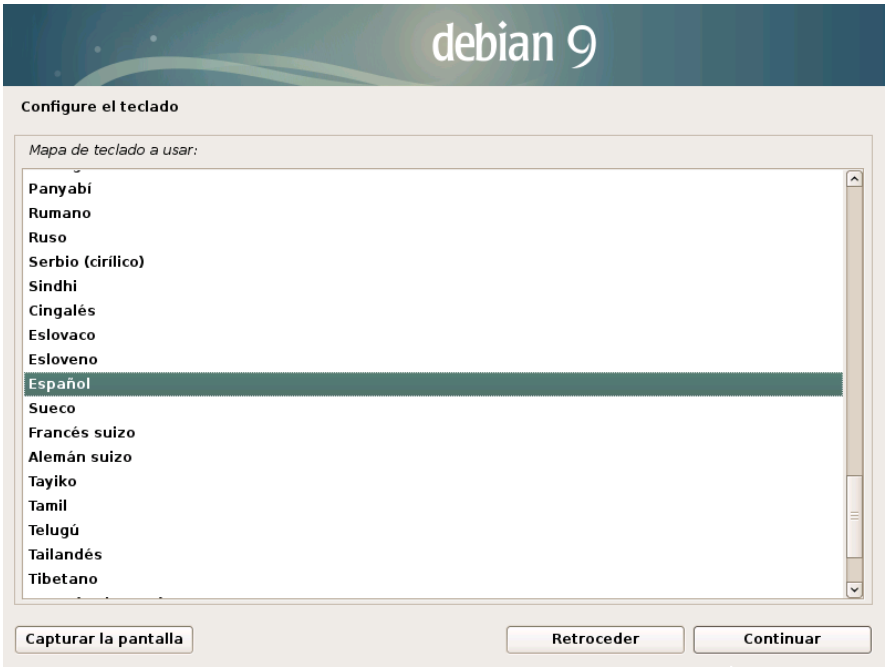
Seleccionamos el idioma del proceso de instalación y del sistema operativo.



Seleccionamos nuestra ubicación.



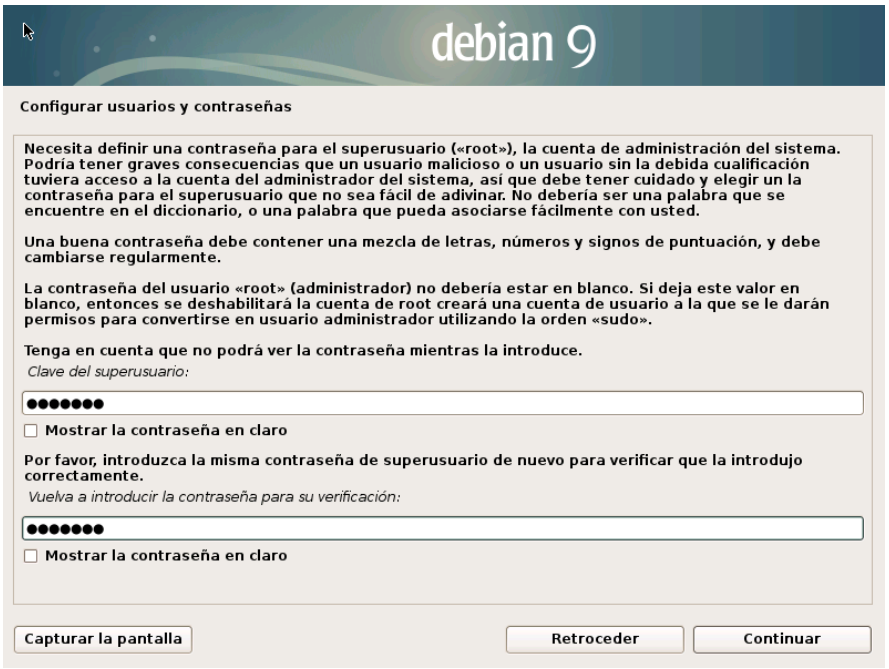
Seleccionamos la distribución del teclado.



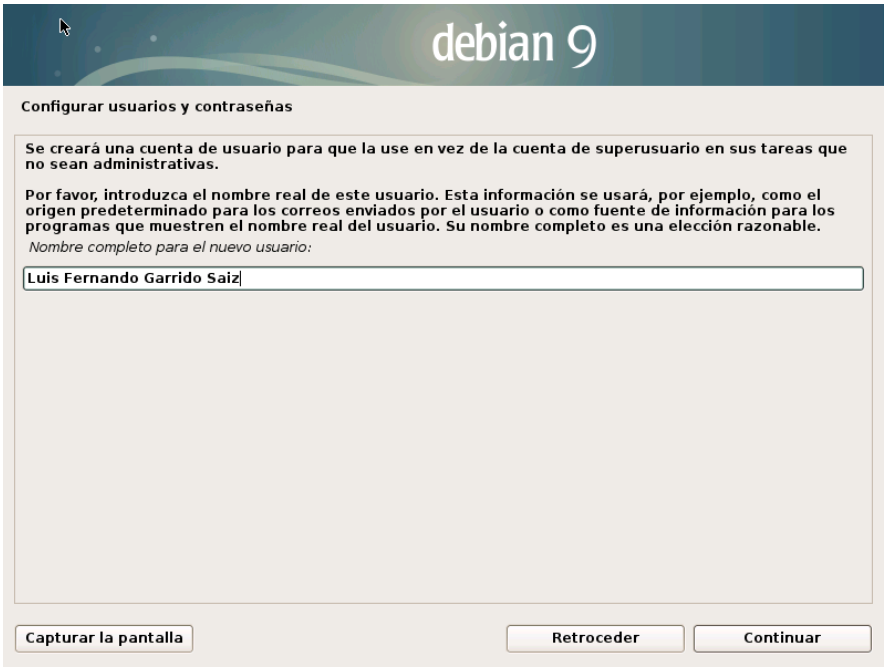
Asignamos un nombre a la máquina.

Añadimos el nombre de dominio.

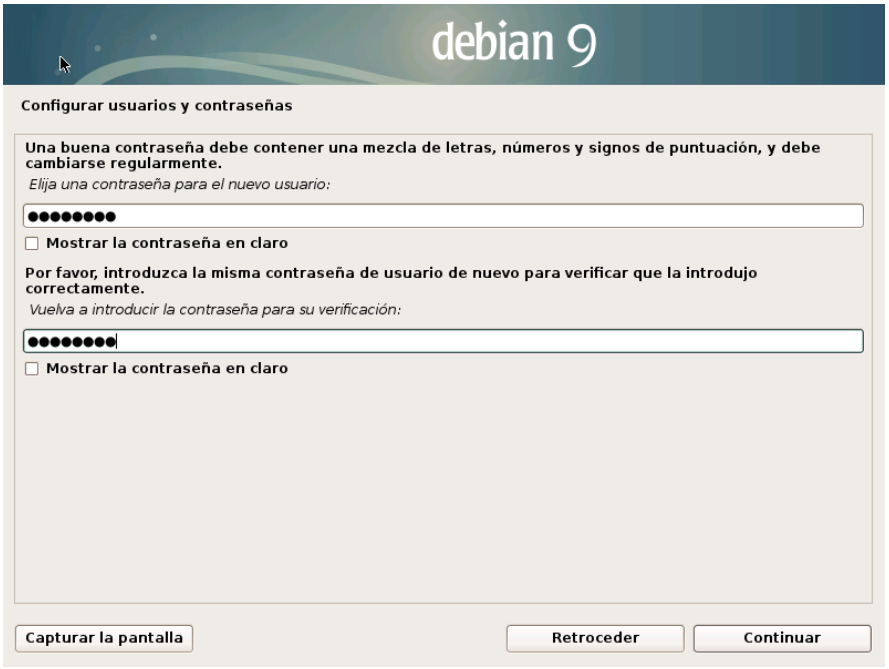
Insertamos la contraseña del superusuario root.



Creamos un usuario.

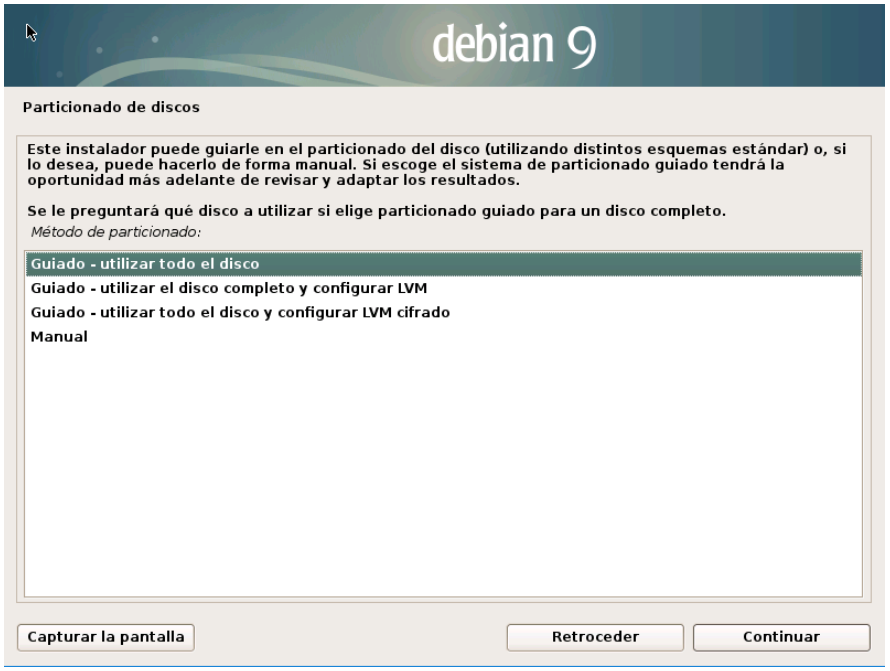


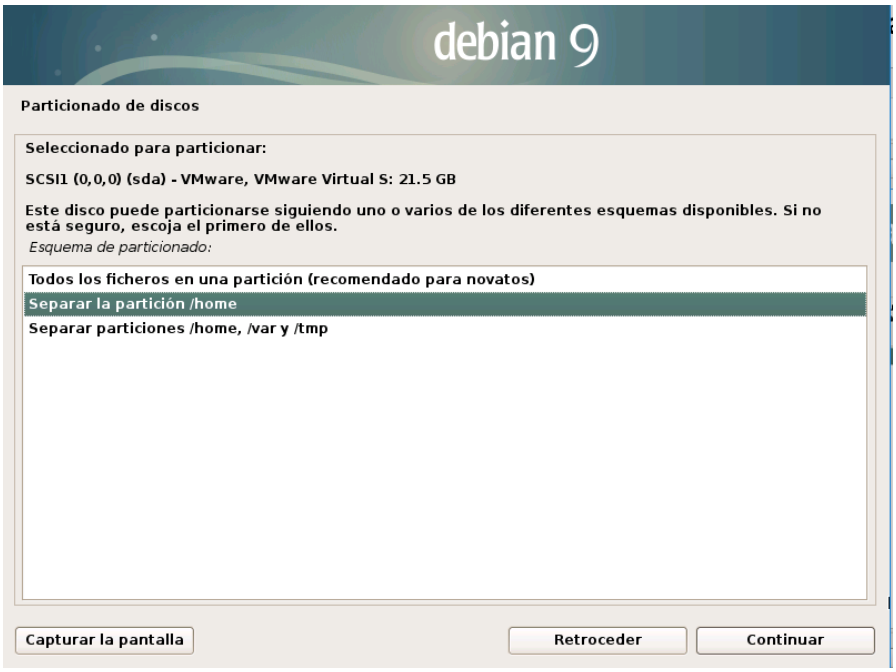
Le asignamos una contraseña diferente a la de root.



Seleccionamos la zona horaria correspondiente.

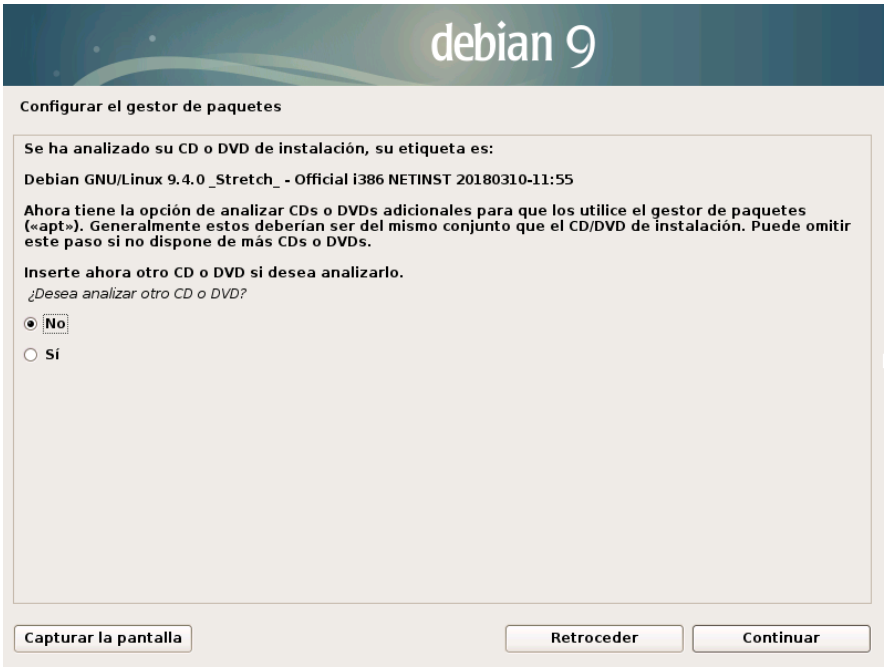
Ahora toca particionar el disco, en este caso elegimos el particionado guiado utilizando todo el disco con la partición /home separada.


debian 9
Particionado de discos
Este instalador puede guiarle en el particionado del disco (utilizando distintos esquemas estándar) o, si lo desea, puede hacerlo de forma manual. Si escoge el sistema de particionado guiado tendrá la oportunidad más adelante de revisar y adaptar los resultados.
Se le preguntará qué disco a utilizar si elige particionado guiado para un disco completo.
Método de particionado:
Guiado - utilizar todo el disco
Guiado - utilizar el disco completo y configurar LVM
Guiado - utilizar todo el disco y configurar LVM cifrado
Manual
Capturar la pantalla
Retroceder
Continuar


debian 9
Particionado de discos
Seleccionado para particionar:
SCSI1 (0,0,0) (sda) - VMware, VMware Virtual S: 21.5 GB
Este disco puede particionarse siguiendo uno o varios de los diferentes esquemas disponibles. Si no está seguro, escoja el primero de ellos.
Esquema de particionado:
Todos los ficheros en una partición (recomendado para novatos)
Separar la partición /home
Separar particiones /home, /var y /tmp
Capturar la pantalla
Retroceder
Continuar

Una vez terminado el particionado confirmamos los cambios.

Si se tienen paquetes para añadir al sistema en un medio extraíble de manera adicional, se pueden agregar ahora. Nosotros seleccionamos No y continuamos.

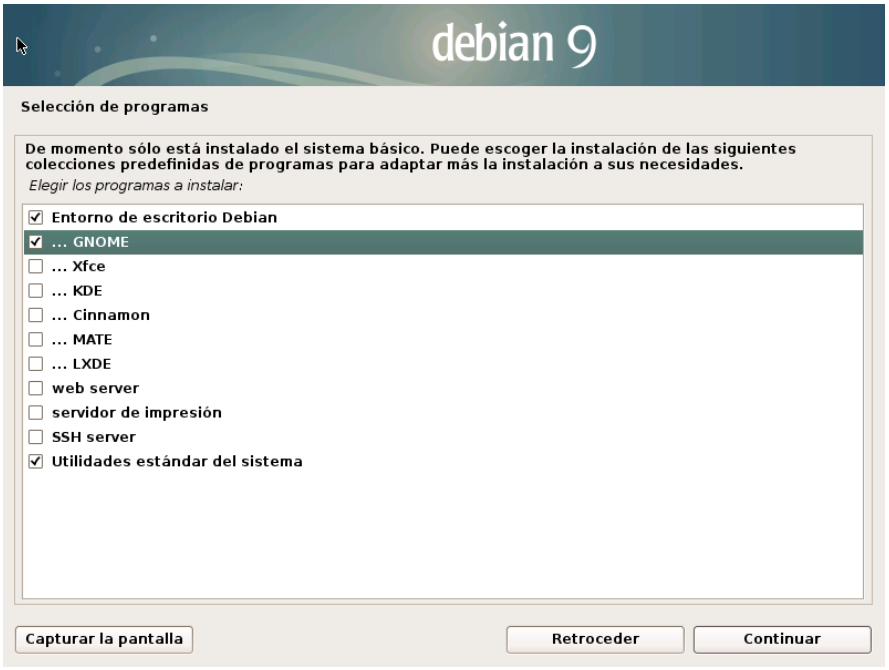


Elegimos la réplica correspondiente a nuestra ubicación y el repositorio.

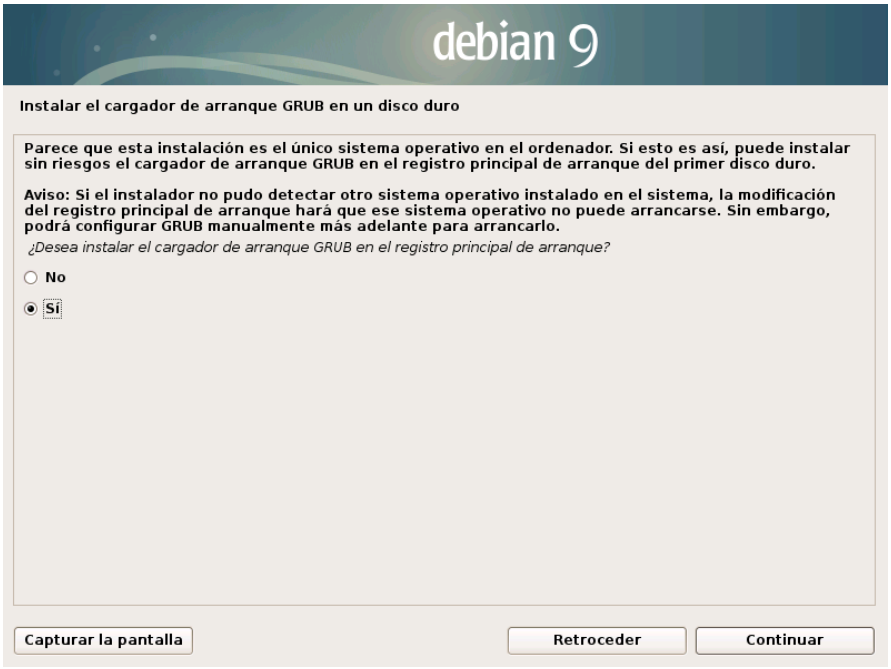
Como no utilizamos Proxy dejamos el siguiente campo en blanco.

Se puede participar o no en la encuesta sobre el uso de paquetes. En este caso no lo haremos.

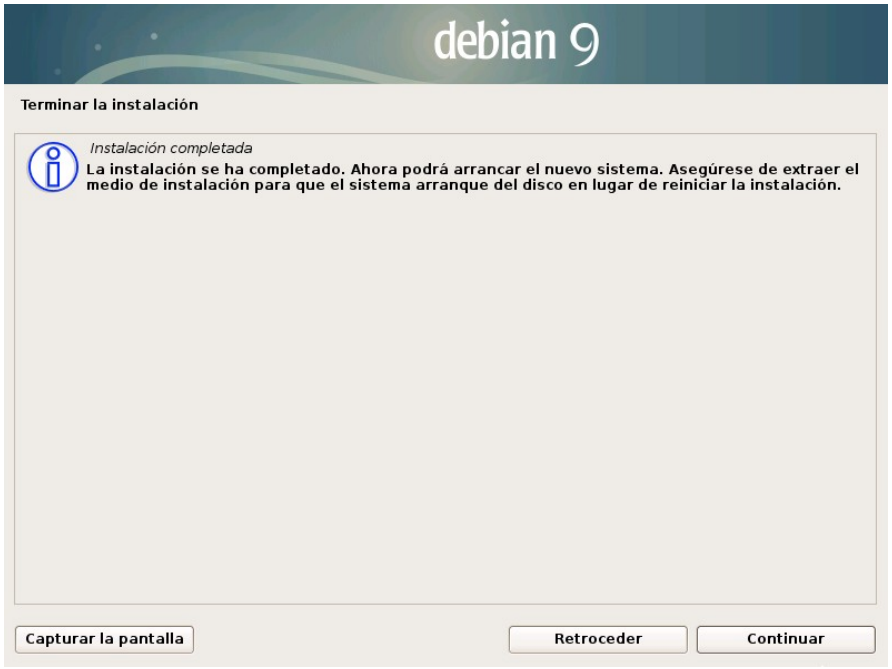
Ahora escogemos la instalación de programas para adaptar más la instalación a nuestras necesidades. Marcamos las opciones: *Entorno de escritorio Debian*, *GNOME* y *Utilidades estándar del sistema*.



Confirmamos la instalación del cargador de arranque GRUB.



Una vez finalizado el proceso, reiniciamos.



Ya tenemos Debian 9 instalado y podemos empezar con la configuración de nuestro Bridge Relay.

Bibliografía

[1] *El impacto de internet en la sociedad*
https://www.bbvaopenmind.com/articulos/el-impacto-de-internet-en-la-sociedad-una-perspectiva-global/
20 de marzo de 2019

[2] *Digital 2019: Global Internet Use Accelerates - We Are Social*
https://wearesocial.com/blog/2019/01/digital-2019-global-internet-use-accelerates
20 de marzo de 2019

[3] *La red Tor: Un análisis desde el punto de vista técnico*
http://repositori.uji.es/xmlui/bitstream/handle/10234/168731/TFG_2017_EstellerVidalClara.pdf?sequence=1
26 de diciembre 2018

[4] *TOR The Onion Router*
http://jeuazarru.com/wp-content/uploads/2014/10/tor.pdf
26 de diciembre de 2018

[5] *Tor (red de anonimato)*
https://es.wikipedia.org/wiki/Tor_(red_de_anonimato)
28 de diciembre de 2018

[6] *Tor*
https://wikis.fdi.ucm.es/ELP/Tor
28 de diciembre de 2018

[7] *Tor Project | Sponsors*
https://www.torproject.org/about/sponsors.html.en
28 de diciembre de 2018

[8] *El Proyecto Tor, galardonado por su papel en las revoluciones de Oriente Medio*
https://hipertextual.com/archivo/2011/04/el-proyecto-tor-galardonado-por-su-papel-en-las-revoluciones-de-oriente-medio/
29 de diciembre de 2018

[9] *Welcome to Tor Metrics*
https://metrics.torproject.org/
30 de diciembre de 2018

[10] *Protocolo TLS (Transport Layer Security)*
https://www.monografias.com/trabajos74/protocolo-tls-transport-layer-security/protocolo-tls-transport-layer-security2.shtml
23 de abril de 2019

[11] *Algorithms Explained: Diffie-Hellman – Hacker Noon*
https://hackernoon.com/algorithms-explained-diffie-hellman-1034210d5100
16 de abril de 2019

[12] *tor/aes.c GitHub*
https://github.com/arlolra/tor/blob/master/src/common/aes.c
18 de abril de 2019

[13] *The lifecycle of a new relay | Tor Blog*
https://blog.torproject.org/lifecycle-new-relay
17 de diciembre de 2018

[14] *What is a Tor Relay? | Tor Challenge*
https://www.eff.org/torchallenge/what-is-tor.html
17 de diciembre de 2018

[15] *Tails - Modo Puente (bridge) de Tor*
https://tails.boum.org/doc/first_steps/startup_options/bridge_mode/index.es.html
18 de diciembre de 2018

[16] *TorRelayGuide – Tor Bug Tracker & Wiki*
https://trac.torproject.org/projects/tor/wiki/TorRelayGuide
17 de diciembre de 2018

[17] *Tor (red de anonimato) #Componentes*
https://es.wikipedia.org/wiki/Tor_(red_de_anonimato)#Componentes
18 de diciembre de 2018

[18] *Servers – Tor Metrics*
https://metrics.torproject.org/relayflags.html
19 de diciembre de 2018

[19] *Users – Tor Metrics*
https://metrics.torproject.org/userstats-censorship-events.html
12 de febrero de 2019

[20] *Tor Project: Bridges*
https://www.torproject.org/docs/bridges
13 de febrero de 2019

[21] *BridgeDB*
https://bridges.torproject.org/
13 de febrero de 2019

[22] *Tor Project: Pluggable Transports*
https://2019.www.torproject.org/docs/pluggable-transports.html.en#user
18 de febrero de 2019

[23] *Progress in censorship circumvention: overview of Tor and Pluggable transports*
https://maikel.pro/published/progress-tor-pluggable-transports.pdf
7 de mayo de 2019

[24] *En mi región Tor está bloqueado · Torifícate*
https://tor.derechosdigitales.org/torificate//p2.1/
18 de febrero 2019

[25] *doc / PluggableTransports / Obfs3Evaluation - Tor Bug Tracker & Wiki*
https://trac.torproject.org/projects/tor/wiki/doc/PluggableTransports/Obfs3Evaluation
21 de febrero 2019

[26] *obfs3-protocol-spec.txt \ obfs3 \ doc - pluggable-transports / obfsproxy*
https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/obfs3/obfs3-protocol-spec.txt
21 de febrero de 2019

[27] *doc/PluggableTransports/ScrambleSuitEvaluation – Tor Bug Tracker & Wiki*
https://trac.torproject.org/projects/tor/wiki/doc/PluggableTransports/ScrambleSuitEvaluation
25 de febrero de 2019

[28] *scramblesuit-spec.txt\scramblesuit\doc - pluggable-transports/obfsproxy*
https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/scramblesuit/scramblesuit-spec.txt
25 de febrero de 2019

[29] *doc/PluggableTransports/Obfs4Evaluation – Tor Bug Tracker & Wiki*
https://trac.torproject.org/projects/tor/wiki/doc/PluggableTransports/Obfs4Evaluation
4 de marzo de 2019

[30] *torspec/216-ntor-handshake.txt at master · torproject/torspec · GitHub*
https://github.com/torproject/torspec/blob/master/proposals/216-ntor-handshake.txt
4 de marzo de 2019

[31] *obfs4-spec.txt \ doc - pluggable-transports / obfs4 - El obfourscator*
https://gitweb.torproject.org/pluggable-transports/obfs4.git/tree/doc/obfs4-spec.txt
4 de marzo de 2019

[32] *doc / PluggableTransports / FteEvaluation - Tor Bug Tracker & Wiki*
https://trac.torproject.org/projects/tor/wiki/doc/PluggableTransports/FteEvaluation
28 de enero de 2019

[33] *doc / PluggableTransports / MeekEvaluation - Tor Bug Tracker & Wiki*
https://trac.torproject.org/projects/tor/wiki/doc/PluggableTransports/MeekEvaluation
28 de enero de 2019

[34] Blocking-resistant communication through domain fronting
https://www.bamsoftware.com/papers/fronting/#sec:deploy-tor
28 de enero de 2019

[35] *Información sobre la versión de Debian "stretch"*
https://www.debian.org/releases/stable/
15 de octubre de 2018

[36] *Razones para escoger Debian*
https://www.debian.org/intro/why_debian.es.html
15 de octubre de 2018

[37] *Tor Project: Debian/Ubuntu Instructions*
https://www.torproject.org/docs/debian.html.en
18 de octubre de 2018

[38] *Tor Project: manual*
https://www.torproject.org/docs/tor-manual.html.en
18 de octubre de 2018

[39]*Debian 9 sudo sin password*
https://www.hiroom2.com/2017/06/19/debian-9-sudo-without-password/
16 de octubre de 2018

[40] *UFW - Community Help Wiki*
https://help.ubuntu.com/community/UFW
17 de octubre de 2018

[41] *Nyx*
https://nyx.torproject.org/#home
22 de octubre de 2018

[42] *Wireshark · Go Deep.*
https://www.wireshark.org/
3 de marzo de 2019

[43] *¿Cómo funciona la red Tor?*
https://www.genbeta.com/seguridad/como-funciona-la-red-tor
14 de marzo de 2019

[44] *Navegador Tor para Windows - Anonimato en línea y evasión de censura*
https://securityinabox.org/es/guide/torbrowser/windows/
29 de octubre de 2018

[45] *Tor Browser Bundle Ubuntu PPA ~ Web Upd8: Ubuntu / Linux blog*
http://www.webupd8.org/2013/12/tor-browser-bundle-ubuntu-ppa.html
29 de octubre de 2018

[46] *Tor Browser*
https://2019.www.torproject.org/projects/torbrowser.html.en
29 de octubre de 2018

[47] *Tor Project: manual*
https://www.torproject.org/docs/tor-manual.html.en
31 de octubre de 2018

[48] *Proxy SwitchyOmega - Chrome Web Store*
https://chrome.google.com/webstore/detail/proxy-switchyomega/padekgcemlokbadohgkifijomclgjgif?hl=en
31 de octubre de 2018

[49] *Cómo usar TOR en Android para entrar en la Dark Web y Deep Web*
https://www.xataka.com/basics/como-usar-tor-android-para-entrar-dark-web-deep-web
5 de noviembre de 2018