Universitat Oberta de Catalunya

RED DE ANONIMIZACION TOR Y CIBERMERCADOS NEGROS

TRABAJO FINAL DE MASTER - INCIBE

Universitat Oberta de Catalunya

Universitat Autònoma de Barcelona

Universitat Rovira i Virgili

Universitat de les Illes Balears

JORGE SAMUEL HERNANDEZ CUENCA
MÁSTER INTERUNIVERSITARIO DE SEGURIDAD DE LAS TIC - MISTIC
UNIVERSIDAD OBERTA DE CATALUÑA
AREA DE TRABAJO: TFM-AD HOC INCIBE
DIRECTOR TFM: ENRIC HERNÁNDEZ JIMÉNEZ

RESUMEN

En este documento se realiza el estudio de la red de anonimización Tor y sus cibermercados negros. Inicialmente se detalla una introducción del histórico e información general de las redes de anonimización; luego del cual, se realiza un estudio técnico y explicativo del funcionamiento del protocolo Tor y del proyecto Tor en general, para dar paso sobre el uso del simulador Shadow.

Como parte del proyecto Tor y demostración práctica del funcionamiento de la red Tor, se realiza un laboratorio con varios ejemplos básicos utilizando la herramienta de simulación Shadow; así se demuestra técnicamente el enrutamiento Tor y sus componentes, como también las ventajas de uso de este simulador.

Finalmente se realiza un acceso a la red Tor mediante la herramienta Tor Browser, en el cual, se detalla consideraciones antes y durante el acceso a la red; luego del cual, de se describe detalles estadísticos de lo encontrado. Adicional y Brevemente, en este documento se realiza una introducción y funcionamiento de sobre la criptomoneda Bitcoin.

Al final de la documentación, se presentan las conclusiones y trabajos futuros en base de los conocimientos obtenidos.

Abstract

In this document, the Tor anonymization network and its black cybermarkets are studied. Initially an introduction of the historical and general information of the anonymization networks is detailed; After which, a technical and explanatory study of the operation of the Tor protocol and the Tor project in general is made, to make way for the use of the Shadow simulator.

As part of the Tor project and practical demonstration of the functioning of the Tor network, a laboratory is made with several basic examples using the Shadow simulation tool; thus, Tor routing and its components are technically demonstrated, as well as the advantages of using this simulator.

Finally, an access to the Tor network is made through the Tor Browser tool, in which considerations are detailed before and during access to the network; After which, it describes statistical details of what was found. Additionally and briefly, this document introduces and operates on Bitcoin cryptocurrency.

At the end of the documentation, the conclusions and future work are presented based on the knowledge obtained.

INDICE

Lista de Ilustraciones

1. Introducción

1.1. Contexto y justificación del Trabajo

El nombre Internet es un acrónimo formado por la palabra en Ingles de **Inter***connected* **Net***works*, el cual significa Redes Interconectadas. Así se da entender el sistema de redes de comunicación más grande que el hombre ha creado desde su aparición en 1969 como ARPAnet. El concepto fue creado con la finalidad de enlazar centros de investigación y departamentos gubernamentales con un fin de intercambiar datos científicos y militares. En la última década, el Internet ha experimentado un aumento exponencial de usuarios, el cual ha permitido el desarrollo y explotación de esta herramienta en muchos de los sectores comerciales y no comerciales. Esta explotación ha conllevado a que ciertos gobiernos restringa uso de aplicaciones en Internet geográficamente, por lo que herramientas como VPN y métodos de anonimización (TOR u otros) están siendo utilizadas para burlar estos controles. Pero no solo su uso se enfoca al desbloqueo geográfico de restricciones sino también ha creación de mercados en el cual todo está permitido: desde la venta de equipos robados, ventas de armas, hasta pornografía infantil; a este mercado de negro digital lo llamaremos “Cibermercados Negros” o “Deepweb”.

Las herramientas de anonimización permite la navegación, publicación y comercialización de servicios en internet de tal manera que no se conozca el origen de su precedencia, esto se realiza con varios propósitos como:

- Preservar la privacidad de datos personales durante la navegación de ciertos servicios web de procesamiento de datos; la mayoría de estos con fines de publicidad.
- Saltarse bloqueos geográficos; ya sea por bloqueos de servicios por gobiernos locales o servicios ofrecidos únicamente en otras zonas geográficas.
- Realizar activismo o realizar publicaciones en anonimato.
- Realizar, publicar u ofrecer servicios cuyos orígenes por naturaleza es de índole criminal como drogas, ventas de armas, etc.

En la presente TFM (Trabajo Final de Maestría) mostraremos el funcionamiento y los componentes de la red TOR (The Onion Router) de una perspectiva práctica, en el cual observaremos y estudiaremos su comportamiento.

1.2. Objetivos del Trabajo

Los objetivos del presente Trabajo Final de Maestría serán los siguientes:

1. Realizar un estudio del proyecto TOR con la finalidad de entender el funcionamiento de TOR de cada uno de sus módulos.
2. Realizar un escenario controlado de una red TOR con objetivos de investigar el funcionamiento y comportamiento de la topología The Onion Router.
3. Luego de realizar los laboratorios y entendido el funcionamiento de redes TOR mediante la metodología teórico y práctico se realizará

una investigación del estado y cual es mercado que existe en la red TOR.

1.3. Enfoque y método seguido

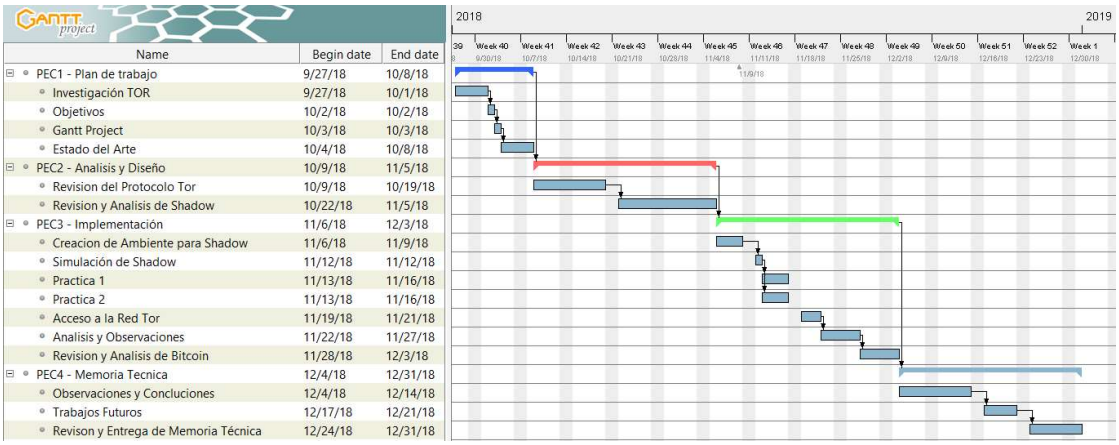
En la presente TFM se realizará un enfoque teórico y práctico al funcionamiento a la herramienta TOR mediante la investigación y recopilación de datos y finalmente realizando un laboratorio controlado del uso de una red de anonimización TOR.

Para el enfoque teórico, se realizará investigaciones en la web sobre todo referente a TOR y sus componentes. Se tomará en cuenta principalmente la página oficial del proyecto: https://www.torproject.org además de otras páginas de referencia académica como Wikipedia o Papers; adicional se investigará todo referente a los Cibermercados y la Deepweb, como son usados y cuáles son los sectores más utilizados.

En el enfoque práctico se realizará un laboratorio utilizando la herramienta Shadow del proyecto Tor. Esta herramienta se instalará y se ejecutará mediante virtualización con acceso limitado y/o restringido al internet. Esta herramienta Shadow se lo considera ideal para fines investigativos, debido a que se puede tener un escenario controlado, ya que este tipo de redes son monitoreadas con frecuencia por su nivel de criminalidad.

1.4. Planificación del Trabajo

Se especifica el siguiente diagrama de Gantt con las diferentes tareas del proyecto. Se estable 4 etapas, las cuales dan inicio el 27 de septiembre del 2018 y Entrega de la TFM el 21 de diciembre del 2018.



Name	Begin date	End date
PEC1 - Plan de trabajo	9/27/18	10/8/18
Investigación TOR	9/27/18	10/1/18
Objetivos	10/2/18	10/2/18
Gantt Project	10/3/18	10/3/18
Estado del Arte	10/4/18	10/8/18
PEC2 - Analisis y Diseño	10/9/18	11/5/18
Revision del Protocolo Tor	10/9/18	10/19/18
Revision y Analisis de Shadow	10/22/18	11/5/18
PEC3 - Implementación	11/6/18	12/3/18
Creacion de Ambiente para Shadow	11/6/18	11/9/18
Simulación de Shadow	11/12/18	11/12/18
Practica 1	11/13/18	11/16/18
Practica 2	11/13/18	11/16/18
Acceso a la Red Tor	11/19/18	11/21/18
Analisis y Observaciones	11/22/18	11/27/18
Revision y Analisis de Bitcoin	11/28/18	12/3/18
PEC4 - Memoria Tecnica	12/4/18	12/31/18
Observaciones y Concluciones	12/4/18	12/14/18
Trabajos Futuros	12/17/18	12/21/18
Revison y Entrega de Memoria Técnica	12/24/18	12/31/18

1.5. Estado del Arte

1.5.1. Significado de TOR y su Historia.

TOR proviene del acrónimo THE ONION ROUTER; este nombre se le da a la red de servidores, rúters o nodos que permite enrutar el tráfico de forma aleatoria con el fin de tratar de confundir el origen de la información. Con este concepto se puede referir que cualquier usuario que utilice TOR puede navegar, usar y generar servicios en el internet de manera oculta o anónima.

A continuación, se detalla un breve recorrido de la evolución del internet hacia la Deep Web:

Como se menciona en la introducción, El internet nace en 1970 a manos del gobierno de los Estados Unidos como el ARPANET.

Luego en 1990 se desarrolla el proyecto Onion Routing; la cual fue creada por el Laboratorio de Investigación Naval de los Estados Unidos, con el fin de proteger la información sensible y de inteligencia de Estados Unidos. Sus creadores fueron el matemático Paul Syverson, los científicos Michael G. Reed y David Goldshalag.

En el 2000, se crea el proyecto Freenet diseñada por Ian Clarke, el cual permite proporcionar un esquema de anonimato a través del uso de conexiones y espacio de almacenamiento de los ordenadores que componen la red de Freenet.

El 20 de enero del 2002 nace el proyecto de The Onion Routing o conocido como Tor Project; y es en el 2014 cuando se realiza el primer lanzamiento al público, con el cual el código fue liberado bajo licencia libre.

A partir de su publicación; Tor ha tenido varios usos, desde la navegación de internet, chat, mensajería, etc., todos con el objetivo de ocultar su identidad y con usos lícitos e ilícitos. En el 2013 el uso de Tor alcanzó los aproximadamente 4 millones de usuarios diarios. A continuación, se muestra análisis realizado por el Dr. Garetth Owen con respecto a los servicios usados por TOR en enero 2015 [14]

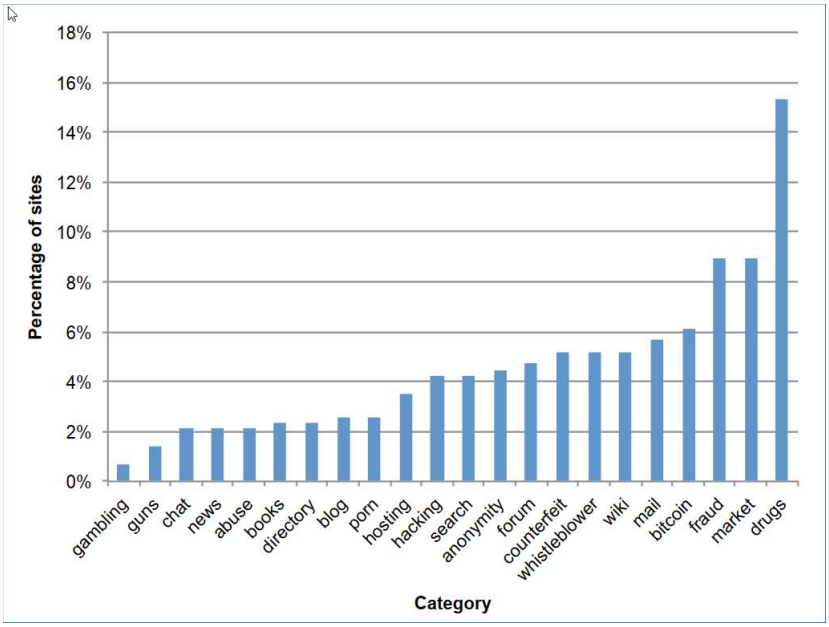


Ilustración 1 Servicios por Categoría utilizados en la Red TOR en enero del 2015 [14]

Como se observa en la Ilustración 1 Servicios por Categoría utilizados en la Red TOR en enero del 2015, el mayor porcentaje de servicios de sitios que utilizan el anonimato TOR es el de las drogas, algo que no es comúnmente encontrar en los sitios tradicionales como Google u otro buscador; estos sitios se los puede definir como la Deep Web.

La Deep Web no es más que los sitios web lícitos y no ilícitos que no son accesibles por el internet tradicional, o en su caso no están disponibles en cualquier buscador común. En el caso de sitios ilícitos tienen otro nombre y se los llama el Dark Web; sitios cuyo contenido son de naturaleza criminal; ejemplo: pedofilia, drogas, armas, etc.

El análisis realizado por una firma de abogados Cartwright King ha permitido determinar que el tamaño de la Deep web es mucho más amplio de lo que cualquier internauta puede imaginar. Solamente el 10% del internet le pertenece a la web tradicional (Facebook, Yahoo!, Google, etc.); mientras que el otro 90% le pertenece a la Deep Web.

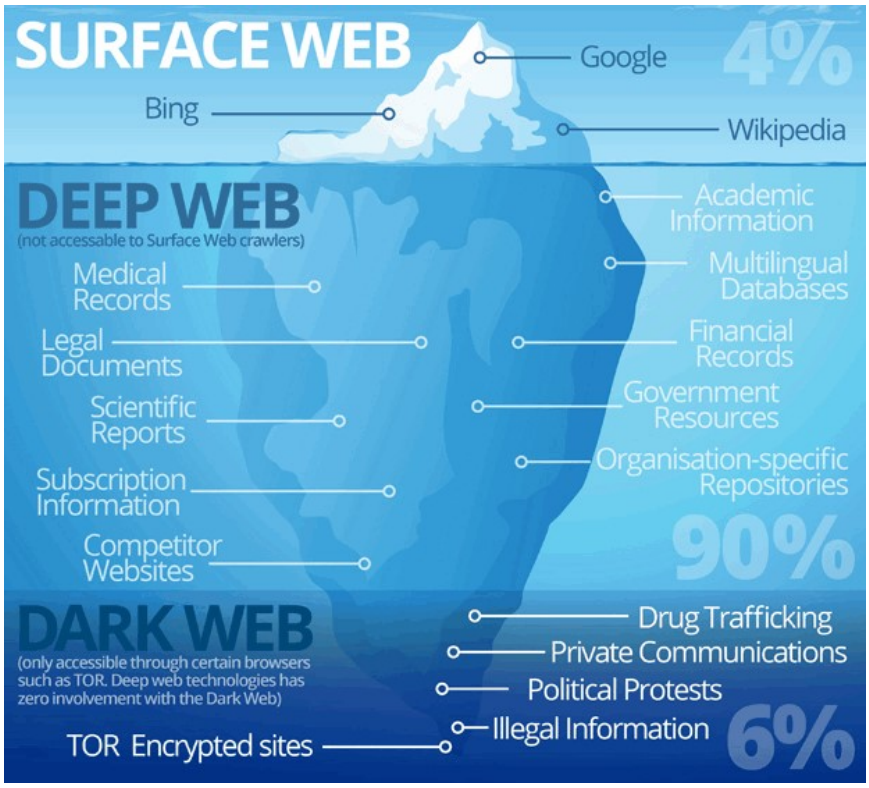


Ilustración 2 Representación de la Deep Web vs. Web Tradicional [fuente: https://nmas1.org/material/2017/11/09/darknet]

1.5.2. Proyecto Tor

Como se ha mencionado anteriormente, con el objetivo de anonimizar las conexiones de los usuarios de la red nació el Proyecto Tor [15]; el cual cuenta con varios proyectos a los que se mencionan a continuación:

Tor-Browser
Es un explorador (en su caso el proyecto ha usado Firefox) el cual tiene embebido el plugin de TOR con el objetivo de permitir la navegación anónima al usuario.

Nyx
Es un terminal para realizar el monitoreo del estado y recursos de Tor, el cual funciona mediante comandos CLI vía SSH.

Metrics Portal
Es un portal para análisis de la red TOR; dispone gráficos de uso de la herramienta, consumo de tráfico, con fines estadísticos.

Pluggable Transports (PT)
Este aplicativo transforma el tráfico entre el cliente y el bridge con el fin de que el tráfico se vea de manera inocente en vez de tráfico actual de Tor

Onionoo
Protocolo basado en Web que permite aprender sobre los relays y bridge de Tor.

Orbot
Es un proyecto que permite utilizar Tor en los dispositivos Android.

Shadow
Es un programa Open-Source que simula una red Tor.

Stem
Son librerías Phyton para aplicaciones y scripts que interactúan con Tor.

Tails
Es una distribución en Linux diseñada para preservar la privacidad y el anonimato.

TorBirdy
Es una extensión para Thunderbird que configura para realizar conexiones de la red Tor para envío de correos.

Txtorcon
Es una implementación basada en eventos Python y Twisted para el control del protocolo Tor.

OONI
Hace referencia al Open Observatory of Network Interference, el cual es una red de observación cuyo objetivo es recopilar datos utilizando software de código abierto para compartir observaciones y datos sobre diferentes tipos, métodos y cantidades de la manipulación de redes en el mundo.

El presente TFM mostraremos el funcionamiento de TOR mediante una simulación; para el cual utilizaremos la herramienta del proyecto Shadow; el mismo que ampliaremos su descripción en el siguiente capítulo.

2. Descripción técnica de Tor y su Funcionamiento

2.1. Enrutamiento de Cebolla (The Onion Router)

Como se ha referido en la introducción del presente texto; se menciona sobre la red TOR (the Onion Router) o enrutamiento de cebolla. Su nombre se debe al nivel de protección que este mantiene al momento enrutar el tráfico de inicio a fin. Para tener una idea más práctica del enrutamiento de cebolla se muestra el siguiente gráfico:

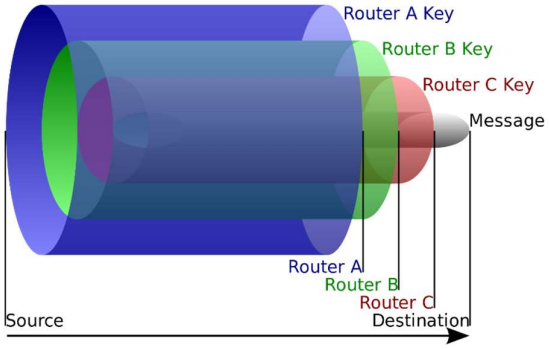


Ilustración 3 Representación 1 de los niveles de protección del circuito Tor. [Fuente: wikipedia.com]

Normalmente, el tráfico de red casi siempre puede ser conocido por el destinatario del mensaje debido a la comunicación y enrutamiento transparente que existe de inicio a fin. En las redes anonimizadoras esta información es oculta o confusa para el destinatario o para cualquier interceptor de tráfico, debido a que cada mensaje es reenviado por cada nodo de la red de comunicación de tal manera que el destinatario o intermediarios no conocen el origen de la información y solo el ultimo salto del encaminamiento; sin contar en la red TOR la información se transmite de manera encriptada por cada salto que realiza.

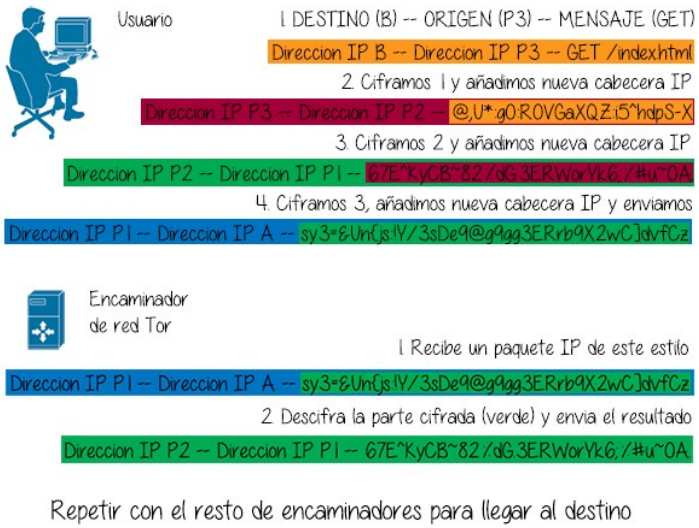


Ilustración 4 Representación 2 de los niveles de protección del circuito Tor [Fuente: http://carlosrdrz.es]

Si se pone en observación este comportamiento de enrutamiento desde un punto de vista más técnico se podría decir (como ejemplo) que la información viaja entre cada nodo de forma nateada, eso quiere decir que la información del ultimo nodo será expuesta como información de origen, ocultando lo que está detrás de este.

Ahora, no solo este es el único comportamiento y/o característica del enrutamiento cebolla. Adicional, por cada nodo por el que se encamine la información, esta es encriptada de tal manera la información es encapsulada y encriptada tantas veces como nodos y saltos existan en la conexión; formando así varias capas de encriptación y de seguridad haciendo casi imposible el descifrado de la información.

Debido a este tipo de seguridad y ocultamiento de información de origen, el enrutamiento forma varias capas de seguridad como se observa en la siguiente ilustración, el cual visto de frente da a parecer a una cebolla, por el cual toma el nombre de enrutamiento cebolla o The Onion Routing (TOR).

Ilustración 5 Representación de Capas similar a una Cebolla [Fuente: torproject.org]

Para añadir otra característica al sistema de enrutamiento cebolla, cada salto que realiza desde el origen a fin se lo nombra circuito de conexión. Este circuito de conexión es aleatorio, de tal manera que para el destinatario o intermediario la conexión siempre será cambiante y aleatoria.

2.2. Componentes de la red Tor

Con estas características brevemente mencionadas, abrimos paso a mencionar los componentes dentro del funcionamiento de la red TOR.

A continuación, en la Ilustración 6 Ejemplo de una Conexión Tor se muestra un gráfico de la creación de un circuito en la red Tor y sus componentes principales:

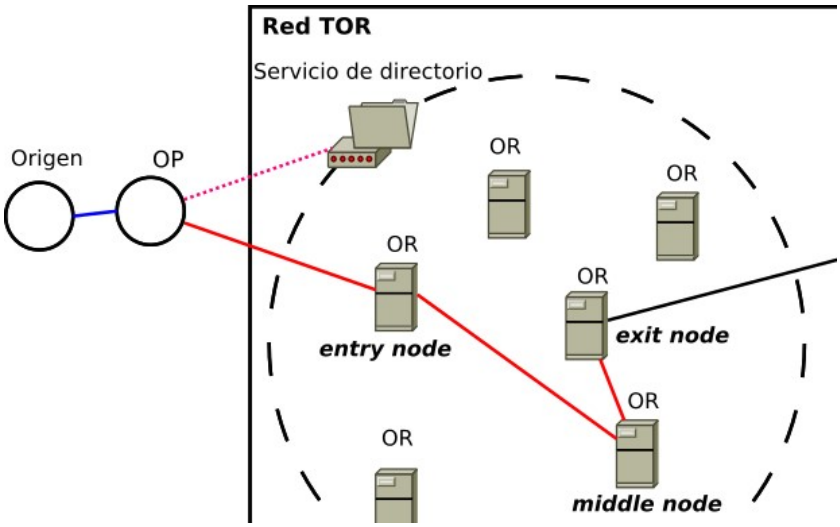


Ilustración 6 Ejemplo de una Conexión Tor

Cliente u Origen: Son los usuarios los cuales utilizan el servicio de la red Tor para mantener su identidad oculta. Los usuarios mantienen una conexión TCP vía Sockets para obtener los servicios respectivos.

Servicio Oculto: Es el nodo de la red Tor destino con algún servicio oculto (puede ser un servicio web). Estos servicios por lo general usan una dirección de dominio onion.

Onion Router: Los nodos Onion Router (OR) son los servicios de la red Tor corriendo en los clientes. Estos servicios no necesitan ejecutarse con privilegios especiales por lo que cualquier usuario puede ejecutarlo. Cada Onion Router mantiene una conexión cifrada con otros onion router de la red.

Onion Proxy: Los nodos Onion Proxy (OP) son los usuarios Tor los cuales establecen los circuitos de la red Tor por medio de los Onion Routers. Son los encargados de cargar los directorios y manejar la comunicación de las aplicaciones de los usuarios. Entre su característica se encuentra que los Onion Proxys aceptan conexiones TCP streams y los redireccionan a través de los circuitos.

2.2.1. Otros roles dentro de la Red Tor

Nodo de Entrada: Son los Onion Router de entrada al circuito de la red Tor; se lo denomina como el Onion Router 1 y es el primer salto de comunicación y creación del circuito.

Nodo de Salida: Son los Onion Router de salida de a la red Tor, eso quiere decir que es el último nodo del circuito; por lo que es el punto de comunicación con el servicio oculto. La salida de datos del Nodo de Salida normalmente pasa sin encriptación.

Nodo Medio: Son los Onion Router que forman el circuito de comunicación y que se encuentran entre el Nodo de Entrada y el Nodo de Salida.

Servicio de Directorio: es un directorio que publica una base de datos que asocia a cada Onion Router una serie de información. Este Servicio de Directorio se asimila como un Servidor DNS que a la vez permite disponer de información de la red a los usuarios finales.
Por medidas de seguridad existen OR principales a los que se denomina autoridades de directorio y otros servicios secundarios los cuales su función son de caches y backup.

2.2.2. Los Nodos y su Conocimiento de la Información del Usuario

A continuación, se detalla la información que es conocida por los nodos durante la creación de cualquier circuito dentro del encaminamiento cebolla.

Nodo de Entrada	
Conoce	• La IP del Usuario y su localización • La IP de los Nodos de Medio
No conoce	• La IP del Nodo de Salida • Mensajes de Nodo de Medio • Mensajes de Nodo de Salida
Nodo Medio	
Conoce	• La IP y Localización del Nodo Medio • La IP y Localización del Nodo de Salida
No conoce	• IP y Localización del Usuario Tor • Mensajes del Nodo de Salida • Mensajes del Nodo de Entrada
Nodo de Salida	

Conoce	• IP y localización del Nodo Medio
	• Contenido del mensaje del usuario Ej.:
	  ○ Fecha y Tiempo de la Transmisión
	  ○ Modelo de cantidad de tráfico enviado en x a z tiempo
	  ○ Alguien desea ver la ip 1.2.3.4
	  ○ Alguien desea conocer la ip de un DNS

No conoce	• IP y Localización del usuario Tor • IP y Localización del Nodo de entrada • Mensaje del Nodo de entrada • Mensaje para el Nodo de Medio

2.3. Funcionamiento de la red Tor

2.3.1. Mensajes y cifrado

Una vez que la comunicación cifrada TLS es establecida entre un par de nodos esos se paquetes de información las cuales se llaman células y se estructuran por 2 partes: cabecera y carga (Payload). La cabecera se compone de dos partes:

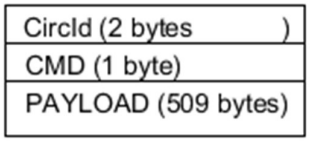


Ilustración 7 Célula del Mensaje Tor

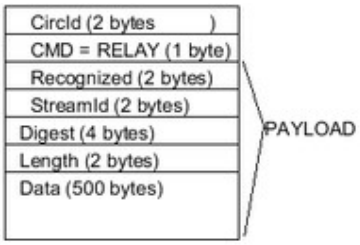


Ilustración 8 Célula Extendida del Mensaje Tor

CircID: es el ID del circuito, el cual identifica el circuito de la conexión. Este ID es distinto para cada OR y OP del circuito

CMD: es el parámetro de comando que establece el tipo de comunicación entre los nodos. Según el tipo de comando CMD hay dos tipos de células:

Células de Control: Estas células son las que permiten establecer el control de la comunicación entre los nodos adyacentes ya que pueden crear o destruir la comunicación; a continuación, se detalla los tipos de mensajes posibles:

- CREATE: para crear el circuito
- CREATED: para indicar que se ha cerrado el circuito
- DESTROY: destruir circuito
- CREATE_FAST: para crear un circuito reaprovechando operaciones de clave publica existentes)
- CREATED_FAST: para indicar que se ha creado el circuito de una manera rápida.

Células de transmisión: Estas células de transmisión son normalmente utilizadas en la comunicación entre el OP y el OR para identificar el flujo de datos durante la comunicación del circuito. Hay que observar que el OP es el que inicialmente controla el circuito creado por lo que al mismo tiempo la información de conexión debe ser establecida durante todo el flujo por estos comandos dentro de las células de transmisión; así mismo las células relay tienen cabeceras adicionales dentro del payload para pasar esta información en el circuito; entre los tipos de mensajes más comunes son: DATA, BEGIN, END, CONNECTED, EXTEND o EXTENDED.

Cifrado: Para establecer una comunicación segura dentro de la red Tor cada mensaje, paquete y/o comunicación entre los nodos va cifrada

mediante TLS/SSL3. Este intercambio de claves se los realiza mediante el protocolo criptográfico Diffie-Hellman.

Dentro los distintos claves de cifrado para la comunicación entre nodos de la red Tor se disponen los siguientes:

- Cifrado RSA de 1024-bits
- Cifrado de clave elíptica Curve25519
- Cifrado de clave pública Ed25519

2.3.2. Creación de circuito

Luego de haber descrito los componentes del enrutamiento Cebolla, detallamos el comportamiento de la creación de circuito para establecer la comunicación de un cliente y su host destino o servicio oculto.

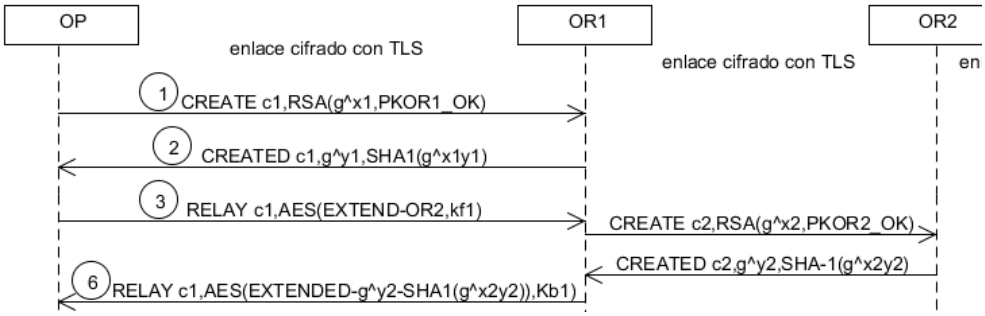


Ilustración 9 Proceso de la Creación del Circuito Tor

Pasos para el establecimiento de la comunicación Tor:

1. El usuario apertura el aplicativo Tor Browser o el plugin requerido para la comunicación con la red Tor

2. El aplicativo o plugin realiza la apertura y/o la comunicación por medio de sockets TCP al Onion Proxy (OP)

3. El OP inicia la negociación y construcción del circuito TOR con el resto de los nodos o Onion Router OR (OR1-OR2-…-ORn). Esta construcción se la realiza de manera telescópica. El OP escoge un Nodo de Salida según su política de salida.

4. Construcción del primer salto del circuito (Conexión OP con OR1):
    4.1. El OP abre una conexión TLS con el OR1
    4.2. El OP escoge el ID del circuito: circID
    4.3. OP realiza un intercambio de claves Deffie-Hellman con el nodo de entrada o OR1. De este proceso se derivan dos claves simétricas, una para cada sentido de la comunicación (forward-key[fk1] y backward-key[fk2]) entre OP a OR1 y OR1 a OP.

5. Construcción del segundo salto del circuito (Conexión OP con OR2 o ORn-1)
   Establecida la comunicación de OP con OR1 se procede a extender la comunicación del circuito.
   5.1. El OP envía un mensaje (Relay_extend) a OR1 para extender la comunicación.
   5.2. El OR1 procede a realizar el proceso de comunicación con el OR2. En este punto se intercambia y establece información mediante una estructura de datos como circID y la clave D-H.
   5.3. El OR1 envía a OP un mensaje (Reley_extended) que se ha establecido la comunicación entre OR1 y OR2. Adicional envía información D-H del OR2 para que se establezca la comunicación encriptada entre el OP y OR2 con nuevas claves simétricas (forward-key[fk2] y backward-key[fk3]). De esta manera el OR1 no conoce la información que recorre en el segundo salto del circuito establecido.
6. Se observa se ha concretado 2 saltos del circuito; el primero entre OP y OR1 y el segundo entre OP y OR2; este último se ha realizado dentro del canal de comunicación entre OP y OR1 como una capa de encriptación adicional (por este motivo se lo llama también enrutamiento cebolla).
   Este proceso se lo realiza n veces hasta llegar al Onion Router de salida establecido al inicio; asi se tendrían los siguientes saltos OP-OR1-OR2-….-ORn cada uno con sus respectivas claves simétricas hacia el Onion Proxy ([kf1][kb1] para OR1, [kf2][kb2] para OR2,…, [kfn][kbn2] para ORn)

7. Finalmente, cuando se ha establecido el circuito de la comunicación entre el OP y el ORn o Nodo de salida se puede iniciar la comunicación entre el cliente y el servidor destino. El Nodo de salida permite la comunicación con los servicios ocultos del sistema Tor y así mismo desencripta la información de tráfico por lo cual se puede indicar que la información no es encriptada a la salida de este punto.
   7.1. Para establecer la comunicación entre el cliente y el servidor de servicios oculto, el OP envía un mensaje hacia el Nodo de Salida con etiqueta Begin con la finalidad solicitar iniciar comunicación.
   7.2. El Nodo de Salida responde con mensaje al OP con etiqueta Connected de tal manera que se establezca la comunicación de datos.

Hay que resaltar los siguientes aspectos dentro del procedimiento de establecimiento de comunicación Tor.

- Por defecto son 3 nodos los que intervienen el establecimiento del circuito
- El establecimiento de los circuitos es aleatorio y depende del Onion Proxy y las políticas de salida establecidas en su configuración
- Solo se permite que el circuito de comunicación creada sea máximo de 10 minutos.

3. El Simulador Shadow

La simulación y emulación dentro de la investigación es necesaria para entender el entorno y ambiente que se está trabajando. Pero existe una gran diferencia entre estos dos tipos de experimentos:

- Simulación permite disponer de un modelo del comportamiento de la herramienta en tal, facilitando cambio de variables para tener un resultado aproximado y estimado al real.
- Emulación permite replicar el comportamiento o herramienta de estudio de tal manera que su análisis y resultados son más exacto, adicional como es una réplica se puede contar un estudio más detallado del comportamiento en cualquier estado del proceso.

Si se considera el concepto de Simulación y Emulación, en el internet existen varias herramientas que nos permiten visualizar en un aspecto aproximado o real el comportamiento de Tor.

Entre los varios simuladores y emuladores para entender la herramienta Tor se encuentran: Shadow, Chutney y TorPS.

Para el presente análisis y del comportamiento de la Tor y basándonos en las herramientas presentadas en el proyecto Tor, se procede a analizar la herramienta Shadow.

3.1. Estudio de la Herramienta Shadow

Básicamente Shadow permite simular como se comportaría la comunicación dentro de la red Tor bajo ciertos parámetros que se pueden cambiar a disposición como es el número de nodos, ancho de banda, latencia, entre otros.

Entre las principales características de Shadow tenemos:

- Crea un ambiente de simulación aislado en donde las máquinas virtuales pueden comunicarse de una u otro, pero no con el internet
- Nativamente ejecuta aplicaciones reales como Tor y Bitcoin
- Provee eficiencia, precisión y controlados experimentos.
- Modelos de topología de red, latencia y control de ancho de banda
- Se puede ejecutar sin super usuario o en una maquina Linux o en la nube.
- Shadow permite simular varias máquinas virtuales en tiempos virtuales.
- Simula procesos de CPU y de Red con retardos (delays)
- Permite ejecutar Redes Tor con modelos basado en tráfico por usuario con métricas Tor.

3.2. Funcionamiento de Shadow

El funcionamiento de Shadow se basa en correr una aplicación que se desea que ejecute dentro de la red Tor, pero para simular su funcionamiento, este es ejecutado en el ambiente Shadow; de esta manera se mantiene un ambiente controlado para la ejecución del aplicativo.

Como se mencionaba, Shadow permite controlar el ambiente de la red Tor como son los nodos, latencia entre nodos, ancho de banda, entre otros factores; todos estos parámetros son enviados a Shadow mediante un archivo en formato XML con la configuración de lo que se llama topología de red.

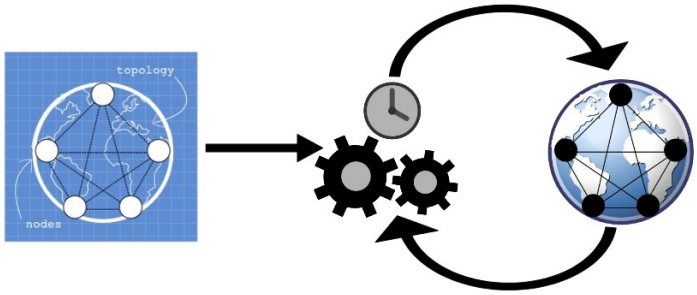


Ilustración 10 Gráfica representativa de Simulación Blueprint utilizada en Shadow [Fuente: [15]]

Luego que Shadow recibe los parámetros en XML; el simulador Shadow crea instancias de máquinas virtuales los cuales ejecutan las aplicaciones respectivas; estas aplicaciones envían los paquetes de datos y son recibidos por buffers de sockets creados dentro del simulador de Shadow, los cuales administran la información recibida. En este punto, Shadow puede administrar la cantidad de nodos y latencia configurada en el archivo de topología XML.

Durante el proceso, cada máquina virtual envía información a otros puntos o también llamada colas de evento discretos, a los cuales se puede configurar la latencia y limitar el ancho de banda; En este caso simularía un enrutamiento de red, que para la aplicación sería un enrutamiento aleatorio como se realiza en el proceso del enrutamiento cebolla.

El proceso de entrega de paquetes es realizado de igual manera hacia máquinas virtuales de destino que simularían también la latencia, control de ancho de banda y procesamiento.

A continuación, se adjunta un diagrama del flujo de paquetes utilizando Shadow por Steven J. Murdoch (v0.1 2013.06.14) [5]:

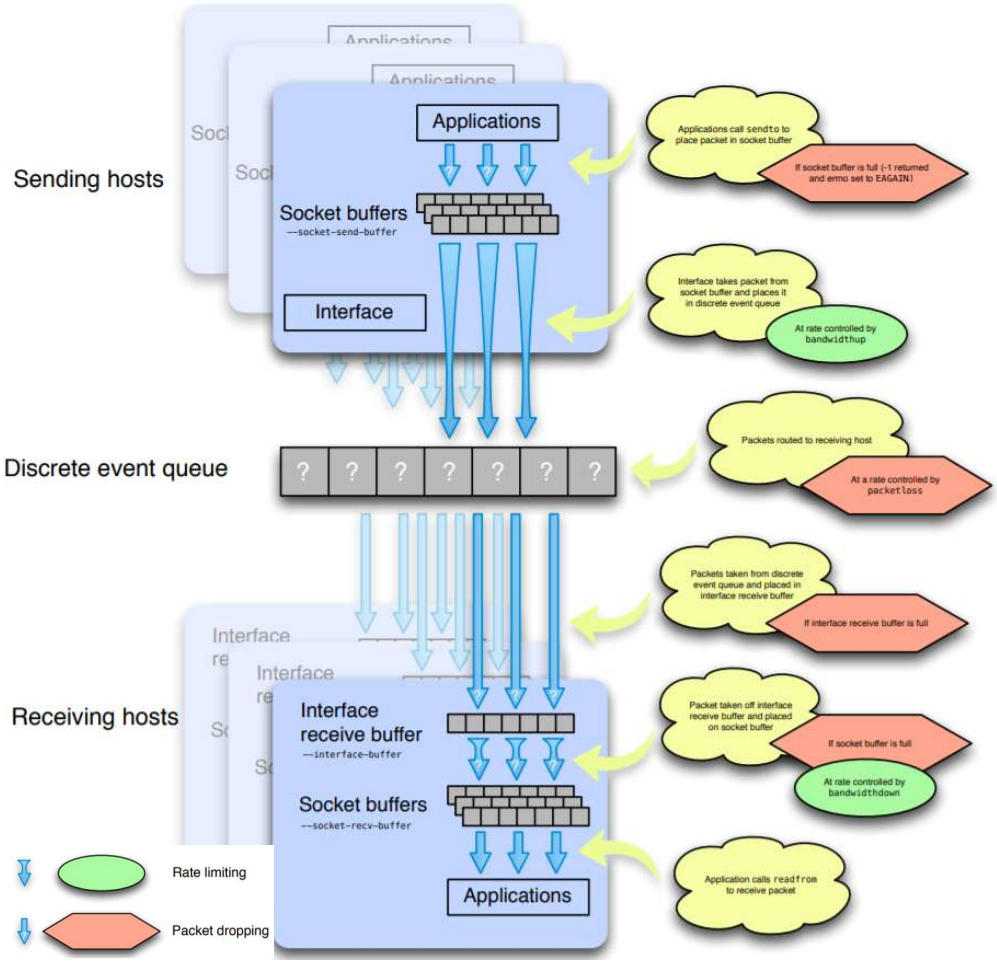


Ilustración 11 Diagrama del flujo de paquetes utilizando Shadow por Steven J. Murdoch (v0.1 2013.06.14) [Fuente: [5]]

3.3. Configuración de la Topología de Shadow.

Para realizar la simulación en Shadow se requiere configurar parámetros de control; estos parámetros son ingresados mediante un texto plano en formato XML como el siguiente ejemplo:

```
<shadow preload="~/.shadow/lib/libshadow-interpose.so" stoptime="1200">
  <topology><![CDATA[<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:
xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key attr.name="packetloss" attr.type="double" for="edge" id="d9" />
  <key attr.name="jitter" attr.type="double" for="edge" id="d8" />
  <key attr.name="latency" attr.type="double" for="edge" id="d7" />
  <key attr.name="type" attr.type="string" for="node" id="d5" />
  <key attr.name="bandwidthup" attr.type="int" for="node" id="d4" />
  <key attr.name="bandwidthdown" attr.type="int" for="node" id="d3" />
  <key attr.name="geocode" attr.type="string" for="node" id="d2" />
  <key attr.name="ip" attr.type="string" for="node" id="d1" />
  <key attr.name="packetloss" attr.type="double" for="node" id="d0" />
  <graph edgedefault="undirected">
    <node id="poi-1">
      <data key="d0">0.0</data>
```

```
      <data key="d1">0.0.0.0</data>
      <data key="d2">US</data>
      <data key="d3">17038</data>
      <data key="d4">2251</data>
      <data key="d5">net</data>
    </node>
    <edge source="poi-1" target="poi-1">
      <data key="d7">50.0</data>
      <data key="d8">0.0</data>
      <data key="d9">0.05</data>
    </edge>
  </graph>
</graphml>
]]></topology>
  <plugin id="tgen" path="~/.shadow/lib/libshadow-plugin-tgen.so"/>
  <host id="server" quantity="50">
    <process plugin="tgen" starttime="1" arguments="tgen.server.graphml.xml"/>
  </host>
  <host id="webclient" quantity="50">
    <process plugin="tgen" starttime="2" arguments="tgen.webclient.graphml.xml"/>
  </host>
  <host id="bulkclient" quantity="50">
    <process plugin="tgen" starttime="2" arguments="tgen.bulkclient.graphml.xml"/>
  </host>
</shadow>
```

Este archivo de configuración se llama shadow.config.xml

Dentro de estos parámetros encontramos 4 elementos:

- Shadow
  - Son las llamadas a los parámetros generales del entorno virtual como es la librería precargada “preload” y el tiempo de ejecución de las máquinas virtuales: “stoptime”
- Topology
  - Detalla parámetros de la topología de la red y su funcionamiento
    - Vértices (type.type): son los atributos para cada cliente, relay o servidor
    - Punto de interés (poi): representan la colección de routers de internet y representan los siguietes atributos: ip, geocode, bandwithup, bandwithdown, packetloss.
    - Edges: Representan los atributos de los paquetes que viajan en el enrutamiento como: latencia, jitter, packetloss.
- Plugin
  - Representa la librería cargada en la virtualización Shadow cuyos parámetros son ID (Nombre único representativo) y Path (Dirección del plugin con extensión *.so).
- Host
  - Representan un nodo o máquina virtual en la simulación. Se representa por un único ID y varios atributos como: *iphint*, *geocodehint*, *typehint*, *quantity*, *bandwidthdown*, *bandwidthup*, *interfacebuffer*, *socketrecvbuffer*, *socketsendbuffer*, *loglevel*, *heartbeatloglevel*, *heartbeatloginfo*, *heartbeatfrequency*, *cpufrequency*, *logpcap*, *pcapdir*. Como ejemplo tenemos

quantity el cual indica la cantidad de host utilizados en la simulación.

- Los hosts requieren el elemento process el cual indica los procesos o aplicaciones ejecutándose durante la simulación para ese elemento. Entre sus argumentos disponen los siguientes parámetros:
  - Plugin: Llamado del ID de plugin a ser usado.
  - Starttime: Tiempo en el cual inicia el proceso luego de haber iniciado la simulación
  - Arguments: atributo necesario para ejecutar el plugin en la simulación.

Tabla resumida de atributos y elementos de shadow:

Shadow.config.xml

Elementos	**Atributos**		**Tipo**
Shadow	Preload		String
	Stoptime		Integer
	Environment		String
Topology	Vertices	IP	String
		Geocode	String
		Bandwidthup	Integer
		Bandwitchdown	Integer
		Packetloss	Double
		Asn	Integer
	Edges	Latency	Double
		Jitter	Double
		Packetloss	Double
Plugin	ID		String
	Path		String
Host	ID		String
	Attributes	Iphint	String
		Geocondehit	String
		Typehit	String
		Quantity	Integer
		Bandwidthdown	Integer
		Bandwidthup	Integer
		Interfacebuffer	Integer
		Socketrecvbuffer	Integer
		Socketsendbuffer	Integer
		Loglevel	String
		Heartbeatloglevel	String
		Heartbeatloginfo	String
		Heartbeatfrequency	Integer
		Cpufrequency	Integer
		Logpcap	String
		Pcapdir	String
	Process	Plugin	String
		Starttime	Integer
		Arguments	String
		Preload	String
		Stoptime	Integer

3.4. Generación de Tráfico en Shadow.

Se ha observado como el archivo shadow.config.xml realiza el control del ambiente de simulación en Shadow por lo que ahora se mostrará cómo se controla el tráfico generado dentro de la simulación.

Tgen es el Shadow Traffic Generator o Generador de Tráfico de Shadow, es una aplicación en C que modela el comportamiento de tráfico utilizando un proceso de dependencia de acción representado por el estándar graphml.xml. Cada nodo Tgen toma un archivo con formato Graphml como parámetro y luego comienza a transferir datos desde y hacia otros nodos siguiendo una ruta a través de la traza de acción.

Formato de la Traza Acción-Dependencia:

Acciones

- Start (Requerido): este inicializa y es requerido para todos los archivos Tgen y solo una acción start es permitido. Los siguientes son atributos del Start:
    - Serverport: puerto local que ser abierto para escuchar por otras conexiones Tgen
    - Time: Tiempo de delay o retardo que tendrá el nodo Tgen antes de iniciar la ruta.
    - Socksproxy: un peer usado como proxy server el cual todas las conexiones a otros Tgen serán realizadas.
    - Timeout: El tiempo desde que comenzó la transferencia
    - Stallout: o también llamado tiempo muerto con el cual se descarta nuevas transferencias desde que se enviaron y recibieron los últimos bytes.
    - Heardbeat: Tiempo entre cual los mensajes son configurados o logeados con nivel de 'message'
    - Loglevel: nivel en el cual los mensajes serán filtrados, mostrados o logeados. Pueden ser: error, critical, message, info y debug.
    - Peers: Lista de conexiones usada para transmitir.

- Transfer (Opcional), los siguientes son atributos permitidos:
    - Type: tipo de transferencia: get "descarga" o put "carga"
    - Protocol: Protocolo utilizado en la transferencia (tcp)
    - Size: Cantidad de Datos a ser transferido
    - Stallout: o también llamado tiempo muerto con el cual se descarta nuevas transferencias desde que se enviaron y recibieron los últimos bytes.
    - Peers: Lista de conexiones usada para transmitir.

- Pausa (Opcional) permite pausar acciones. Atributos permitidos:
    - Time: tiempo en el que el Tgen será pausado antes de ser puesto en operación nuevamente.

- End (Optional) representa la finalización de las condiciones. Atributos permitidos:
    - Time: tiempo desde que el nodo comienza
    - Count: número de transferencias completadas
    - Size: cantidad de datos transferida por el nodo

3.5. Entorno de Pruebas con Shadow

Para el estudio de Tor mediante Shadow se creará de una máquina virtual en Linux y con compatibilidad con la última versión de Shadow:

Maquina Invitado: Core i7 / 8Gb Ram
SO: Windows 10
Virtualizador: Vmware Workstation 14
SO Invitado: Ubuntu 18.04.1 LTS / 2Gb Ram

Ilustración 12 Información de la Maquina Virtual de Simulación

3.5.1. Instalación del simulador Shadow

A continuación, se detallan los pasos para la instalación del Simulador de Shadow en un ambiente de Sistema Operativo Linux Distribución: Ubuntu 18.04.1 LTS.

Dependencias para la instalación:

- gcc, gcc-c++
- python 2
- glib (versión >= 2.32.0) (*)
- igraph (versión >= 0.5.4)
- cmake (versión >= 2.8.8)
- make
- xz-utils
- glibc debuginfo

Instalación de aplicativos y dependencias

```
sudo apt-get install -y gcc g++ python libglib2.0-0 libglib2.0-dev
libigraph0v5 libigraph0-dev cmake make xz-utils
sudo apt-get install libc-dbg
sudo apt-get install -y python-matplotlib python-numpy python-scipy
python-networkx python-lxml
sudo apt-get install -y git dstat screen htop
```

Instalación del aplicativo de Shadow

```
git clone https://github.com/shadow/shadow.git
cd shadow
./setup build --clean --debug --test
./setup install
./setup test
```

Configuración de la variable de dirección o Path para el Shadow

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

Confirmación de la instalación de Shadow

```
shadow --version
```

Mensaje de salida:

```
Shadow v1.13.0-11-g1190a345 2018-08-03 (built 2018-12-01) running
GLib v2.56.3 and IGraph v0.7.1
```

3.5.2. Inicialización de la primera prueba con Shadow.

Dentro del repositorio de Shadow se encuentra la carpeta de example con el cual será de guía para las siguientes pruebas y por el cual se realizará un análisis de su estructura.

La carpeta example contiene los siguientes archivos:

- shadow.config.xml
- tgen.bulkclient.graphml.xml
- tgen.server.graphml.xml
- tgen.webclient.graphml.xml

Los archivos tgen.*.graphml.xml son los Shadow traffic generator (Tgen) que serán utilizados generar el tráfico de la simulación.

tgen.bulkclient.graphml.xml

```
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key attr.name="count" attr.type="string" for="node" id="d5" />
  <key attr.name="size" attr.type="string" for="node" id="d4" />
  <key attr.name="type" attr.type="string" for="node" id="d3" />
  <key attr.name="protocol" attr.type="string" for="node" id="d2" />
  <key attr.name="serverport" attr.type="string" for="node" id="d1" />
  <key attr.name="peers" attr.type="string" for="node" id="d0" />
  <graph edgedefault="directed">
    <node id="start">
      <data key="d0">server1:30080,server2:30080,server3:30080,server4:30080,server5:30080</data>
      <data key="d1">30089</data>
    </node>
    <node id="transfer">
      <data key="d2">tcp</data>
      <data key="d3">get</data>
      <data key="d4">1 MiB</data>
    </node>
    <node id="end">
      <data key="d5">10</data>
    </node>
    <edge source="start" target="transfer" />
    <edge source="transfer" target="end" />
    <edge source="end" target="start" />
  </graph>
</graphml>
```

tgen.server.graphml.xml

```
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key attr.name="serverport" attr.type="string" for="node" id="d0" />
  <graph edgedefault="directed">
    <node id="start">
      <data key="d0">30080</data>
    </node>
  </graph>
</graphml>
```

tgen.webclient.graphml.xml

```
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key attr.name="count" attr.type="string" for="node" id="d6" />
  <key attr.name="size" attr.type="string" for="node" id="d5" />
  <key attr.name="type" attr.type="string" for="node" id="d4" />
  <key attr.name="protocol" attr.type="string" for="node" id="d3" />
  <key attr.name="time" attr.type="string" for="node" id="d2" />
  <key attr.name="serverport" attr.type="string" for="node" id="d1" />
  <key attr.name="peers" attr.type="string" for="node" id="d0" />
  <graph edgedefault="directed">
    <node id="start">
      <data key="d0">server1:30080,server2:30080,server3:30080,server4:30080,server5:30080</data>
      <data key="d1">30088</data>
    </node>
    <node id="pause">
      <data key="d2">
      1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,
      35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60</data>
    </node>
    <node id="transfer">
      <data key="d3">tcp</data>
      <data key="d4">get</data>
      <data key="d5">100 KiB</data>
    </node>
        <node id="end">
          <data key="d6">10</data>
        </node>
    <edge source="start" target="transfer" />
    <edge source="end" target="pause" />
    <edge source="pause" target="start" />
    <edge source="transfer" target="end" />
  </graph>
</graphml>
```

Como se observa cada archivo Tgen posea atributos que definen cada componente. Por ejemplo, se realizará envío de tramas tcp (id=d3) de tipo get (id=d4) con tamaño de 100KiB (id=d5); Adicional se realizará 10 peticiones por cada conexión entre cliente y servidor.

Se utilizará 5 Servidores (Server1, Server2, Server3, Server4, Server5) cuyo puerto de conexión es 30080

Finalmente se tiene el archivo shadow.config.xml con el cual se determinará la topología y comportamiento de toda la simulación. En este archivo hace llamado a las variables y plugin creados anteriormente como se detalla a continuación:

shadow.config.xml

```
1  <shadow preload="~/.shadow/lib/libshadow-interpose.so" stoptime="1200">
2    <topology><![CDATA[<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="
     http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
     http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
3    <key attr.name="packetloss" attr.type="double" for="edge" id="d9" />
4    <key attr.name="jitter" attr.type="double" for="edge" id="d8" />
5    <key attr.name="latency" attr.type="double" for="edge" id="d7" />
6    <key attr.name="type" attr.type="string" for="node" id="d5" />
7    <key attr.name="bandwidthup" attr.type="int" for="node" id="d4" />
8    <key attr.name="bandwidthdown" attr.type="int" for="node" id="d3" />
9    <key attr.name="geocode" attr.type="string" for="node" id="d2" />
10   <key attr.name="ip" attr.type="string" for="node" id="d1" />
11   <key attr.name="packetloss" attr.type="double" for="node" id="d0" />
12   <graph edgedefault="undirected">
13     <node id="poi-1">
14       <data key="d0">0.0</data>
15       <data key="d1">0.0.0.0</data>
16       <data key="d2">US</data>
17       <data key="d3">17038</data>
18       <data key="d4">2251</data>
19       <data key="d5">net</data>
20     </node>
21     <edge source="poi-1" target="poi-1">
22       <data key="d7">50.0</data>
23       <data key="d8">0.0</data>
24       <data key="d9">0.05</data>
25     </edge>
26   </graph>
27 </graphml>
   ]]></topology>
29   <plugin id="tgen" path="~/.shadow/lib/libshadow-plugin-tgen.so"/>
30   <host id="server" quantity="50">
31     <process plugin="tgen" starttime="1" arguments="tgen.server.graphml.xml"/>
32   </host>
33   <host id="webclient" quantity="50">
34     <process plugin="tgen" starttime="2" arguments="tgen.webclient.graphml.xml"/>
35   </host>
36   <host id="bulkclient" quantity="50">
37     <process plugin="tgen" starttime="2" arguments="tgen.bulkclient.graphml.xml"/>
38   </host>
39 </shadow>
```

De la línea 1 a la línea 27 hace referencia de los atributos generales de la simulación como son la ip, latencia, jitter, ancho de banda, etc. A partir de la línea 28 hace referencia de los elementos y plugins creados inicialmente y marcados con los siguientes id: server, webclient y bulkclient.

La topología hace referencia que serán creado 50 conexiones de server, webclient y bulkclient.

Para comenzar la simulación dentro de la carpeta example se inicializa con el siguiente comando:

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

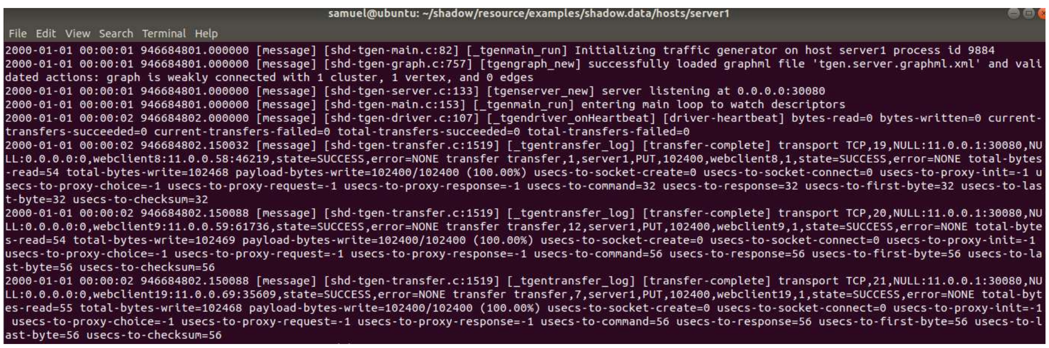
Los logs de la simulación son enviados a shadow.log mientras tanto se crea los datos generados por cada host intervenido durante la simulación en la carpeta shadow.data/hosts/; ambos archivos son de gran importancia para el análisis de la data generada por la simulación.

Salida generada luego de la simulación:

```
* Starting Shadow v1.13.0-11-g1190a345 2018-08-03 (built 2018-12-01)
with GLib v2.56.3 and IGraph v0.7.1
** Stopping Shadow, returning code 0 (success)
```

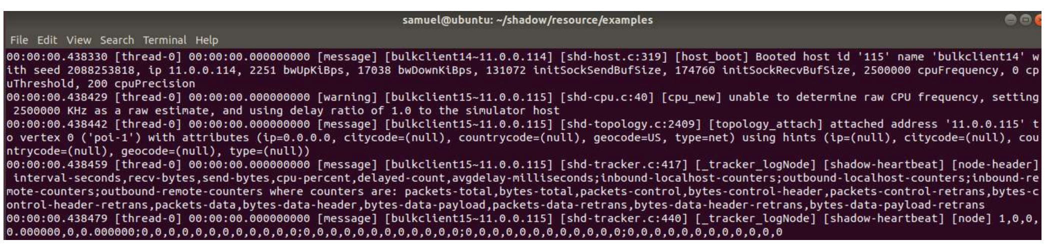
Información de hosts generados en la simulación shadow.data/hosts/:

Cada uno de estos archivos posee información de los procesos realizados durante la simulación como si fuese una conexión independiente. Estos archivos presentan el siguiente formato:



Este formato de log dispone de forma cronológica el proceso y estado de cada host que forma la simulación.

El mismo caso sucede con el archivo shadow.log el cual dispone de información más general de la simulación completa de inicio a fin:



Como se observa, el análisis de los logs es un trabajo muy complicado ya que existe mucha información por filtrar y ordenar, aunque ya la información venga distribuida cronológicamente. Para aquello, Shadow ha creado herramientas de análisis el cual permite realizar un estudio más práctico y estadístico de la simulación, utilizando el lenguaje de programación Python y en base a la información de logs general y por cada uno de los hosts creados durante la simulación.

Estas herramientas se encuentran en la siguiente dirección: ~/shadow/src/tools

Para realizar un análisis gráfico completo de la simulación realizada se procede con los siguientes comandos dentro de la carpeta example donde se generó los datos de la simulación.

Análisis de Shadow.log

```
samuel@ubuntu:~/shadow/resource/examples$ python
~/shadow/src/tools/parse-shadow.py --prefix example-results shadow.log

processing input from
/home/samuel/shadow/resource/examples/shadow.log...
done processing input: simulation ran for 0.0493207547222 hours and
consumed 0 GiB of RAM
dumping stats in /home/samuel/shadow/resource/examples/example-results
all done!
```

Análisis de Shadow.Data

```
samuel@ubuntu:~/shadow/resource/examples$ python
~/shadow/src/tools/parse-tgen.py --prefix example-results
shadow.data/hosts

processing input from 150 files...
done processing input: 1000 total successes, 0 total errors, 150 files
with names, 0 files without names
dumping stats in /home/samuel/shadow/resource/examples/example-results
all done!
```

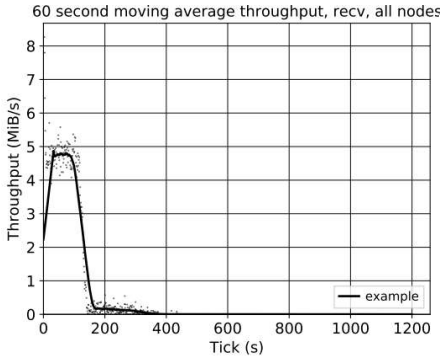
Generación de Graficas en PDF.

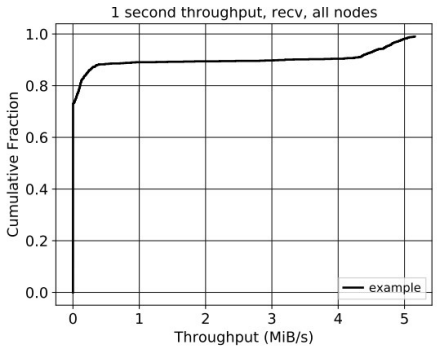
```
python ~/shadow/src/tools/plot-shadow.py --prefix "example" --data
example-results/ "example"
```

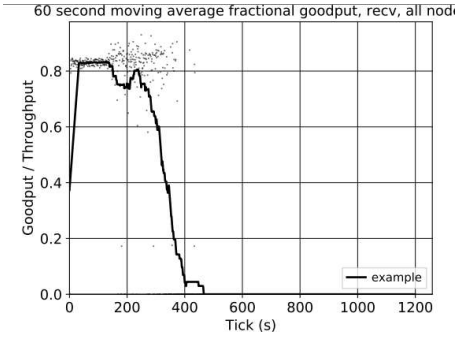
Finalmente se crea el archivo plot.shadow.results.pdf con 53 graficas, como:

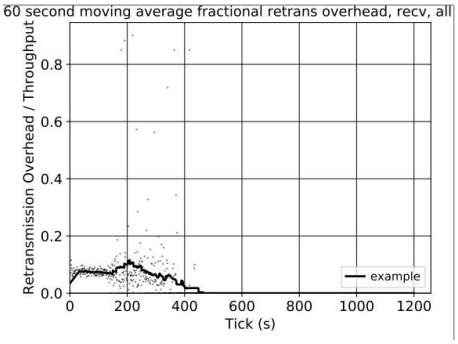
- 60 second movie average throghput, recv, all nodes
- 1 second throughput recv, all nodes
- 60 second moving fractional goodput, recv, all nodes
- 60 second moving avarage retrains overhead, send, all nodes
- Time to download first byte, all clients
- Etc..

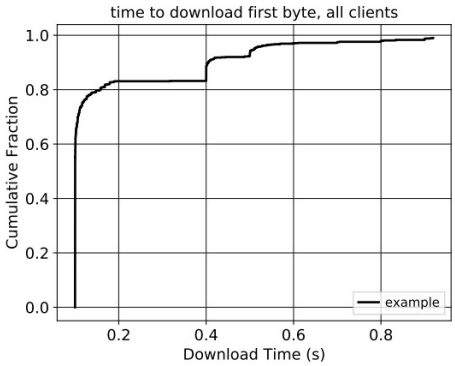
Gráficas obtenidas como resultado de la simulación de Shadow:

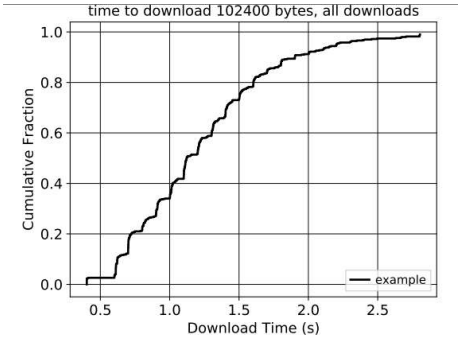

60 second moving average throughput, recv, all nodes
Throughput (MiB/s)
Tick (s)
example


1 second throughput, recv, all nodes
Cumulative Fraction
Throughput (MiB/s)
example


60 second moving average fractional goodput, recv, all nod
Goodput / Throughput
Tick (s)
example


60 second moving average fractional retrans overhead, recv, all
Retransmission Overhead / Throughput
Tick (s)
example


time to download first byte, all clients
Cumulative Fraction
Download Time (s)
example


time to download 102400 bytes, all downloads
Cumulative Fraction
Download Time (s)
example

3.5.3. Instalación de plugin para Shadow.

Los plugin de Shadow son librerías independientes que contienen aplicaciones que el usuario desea utilizar dentro de un ambiente de simulación de Shadow.

Cada Plugin de Shadow implementa una interfaz para comunicarse con Shadow a lo que se llama “Shadow Callbacks”,

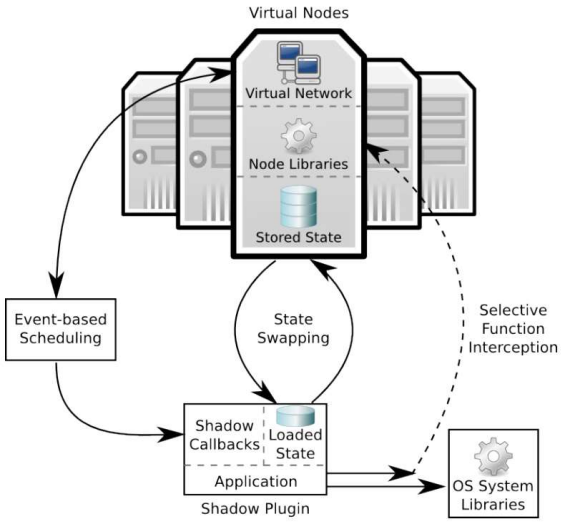


Ilustración 13 Arquitectura de Shadow [Fuente: [12]]

En la figura se aprecia la arquitectura de Shadow publicado por Rob Jansen en el paper “*Shadow: Running Tor in a Box for Accurate and Efficient Experimentation*” [12], esta muestra como cada Nodo Virtual (Virtual Nodes) se comunica con el Plugin de Shadow y dentro del Plugin se encuentra las aplicaciones y el Shadow Callbacks con el cual permite la comunicación con el simulador.

En los repositorios del proyecto Shadow se dispone de varios plugins como guía: shadow-plugin-Tor, shadow-plugin-extras, shadow-plugin-bitcoin, etc. Para la práctica se usará el shadow-plugin-Tor el cual permite el escenario de topología mínimo para la simulación de la red Tor con todos sus componentes.

A continuación, se detalla la instalación de shadow-plugin-Tor [4]:

1. Instalación de Dependencias

```
sudo apt-get -y install gcc automake autoconf zlib1g-dev liblzma5
liblzma-dev
```

2. Configuración e Instalación de shadow-plugin-Tor

```
git clone https://github.com/shadow/shadow-plugin-tor.git
cd shadow-plugin-tor
```

```
./setup dependencies
./setup build
./setup install
```

3.5.4. Ejecución de Tor plugins

Dentro de shadow-plugin-Tor se tiene el repositorio llamado shadowtor-minimal con el que cuenta con los plugins, aplicaciones y estructuras para la simulación del ambiente de la comunicación del ambiente Tor.

Ejercicio 1.

Para esta simulación, y con el objetivo de mostrar el funcionamiento de la red Tor se realizarán 3 simulaciones utilizando la comunicación de clientes Tor, equipos relays, equipos de salida y servicios ocultos; cada simulación tendrá distinta cantidad de conexiones con la finalidad de realizar una comparación y análisis de los 3 ambientes.

Simulación 1: 1 conexión Tor
Simulación 2: 10 conexiones Tor
Simulación 3: 50 conexiones Tor

A continuación, se detalla el ambiente de simulación con sus respectivas configuraciones:

Dentro de shadowtor-minimal se encuentran el archivo conf con los siguientes archivos:

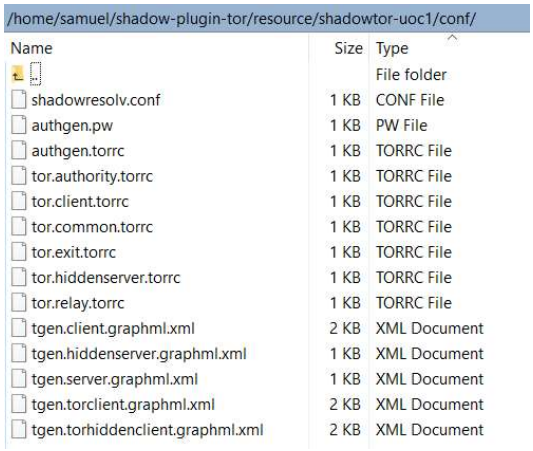


Ilustración 14 Archivos xml de shadowtor-minimal

Los archivos torrc poseen los atributos para cada componente de la simulación, Ejemplo:

Tor.client.torrc

```
1  ORPort 0
2  DirPort 0
3  ClientOnly 1
4  SocksPort 9000
5  SocksListenAddress 127.0.0.1
```

Tor.hiddenserver.torrc

```
ORPort 0
DirPort 0
ClientOnly 1
SocksPort 9000
SocksListenAddress 127.0.0.1
HiddenServiceDir shadow.data/hosts/hiddenserver/hs
HiddenServicePort 80 127.0.0.1:8080
```

Los archivos XML poseen la configuración para la generación de Tráfico entre componentes, Ejemplo:

Tgen.client.graphml.xml

```
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key attr.name="serverport" attr.type="string" for="node" id="d6" />
  <key attr.name="peers" attr.type="string" for="node" id="d5" />
  <key attr.name="type" attr.type="string" for="node" id="d4" />
  <key attr.name="protocol" attr.type="string" for="node" id="d3" />
  <key attr.name="size" attr.type="string" for="node" id="d2" />
  <key attr.name="count" attr.type="string" for="node" id="d1" />
  <key attr.name="time" attr.type="string" for="node" id="d0" />
  <graph edgedefault="directed">
    <node id="start">
      <data key="d5">fileserver:80</data>
      <data key="d6">8888</data>
      <data key="d0">60</data>
    </node>
    <node id="transfer">
      <data key="d3">tcp</data>
      <data key="d4">get</data>
      <data key="d2">1 MiB</data>
    </node>
    <node id="pause">
      <data key="d0">1</data>
    </node>
    <node id="end">
      <data key="d1">1</data>
      <data key="d2">100 MiB</data>
      <data key="d0">3600</data>
    </node>
    <edge source="start" target="transfer" />
    <edge source="transfer" target="end" />
    <edge source="end" target="pause" />
    <edge source="pause" target="start" />
  </graph>
</graphml>
```

Tgen.torclient.graphml.xml

```xml
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key attr.name="socksproxy" attr.type="string" for="node" id="d7" />
  <key attr.name="serverport" attr.type="string" for="node" id="d6" />
  <key attr.name="peers" attr.type="string" for="node" id="d5" />
  <key attr.name="type" attr.type="string" for="node" id="d4" />
  <key attr.name="protocol" attr.type="string" for="node" id="d3" />
  <key attr.name="size" attr.type="string" for="node" id="d2" />
  <key attr.name="count" attr.type="string" for="node" id="d1" />
  <key attr.name="time" attr.type="string" for="node" id="d0" />
  <graph edgedefault="directed">
    <node id="start">
      <data key="d5">fileserver:80</data>
      <data key="d6">8888</data>
      <data key="d7">localhost:9000</data>
      <data key="d0">60</data>
    </node>
    <node id="transfer">
      <data key="d3">tcp</data>
      <data key="d4">get</data>
      <data key="d2">1 MiB</data>
    </node>
    <node id="pause">
      <data key="d0">1</data>
    </node>
    <node id="end">
      <data key="d1">1</data>
      <data key="d2">100 MiB</data>
      <data key="d0">3600</data>
    </node>
    <edge source="start" target="transfer" />
    <edge source="transfer" target="end" />
    <edge source="end" target="pause" />
    <edge source="pause" target="start" />
  </graph>
</graphml>
```

Tgen.torhiddenclient.graphml.xml

```xml
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key attr.name="socksproxy" attr.type="string" for="node" id="d7" />
  <key attr.name="serverport" attr.type="string" for="node" id="d6" />
  <key attr.name="peers" attr.type="string" for="node" id="d5" />
  <key attr.name="type" attr.type="string" for="node" id="d4" />
  <key attr.name="protocol" attr.type="string" for="node" id="d3" />
  <key attr.name="size" attr.type="string" for="node" id="d2" />
  <key attr.name="count" attr.type="string" for="node" id="d1" />
  <key attr.name="time" attr.type="string" for="node" id="d0" />
  <graph edgedefault="directed">
    <node id="start">
      <data key="d5">hxsttdz4esasch5x.onion:80</data>
      <data key="d6">8888</data>
      <data key="d7">localhost:9000</data>
      <data key="d0">60</data>
    </node>
    <node id="transfer">
      <data key="d3">tcp</data>
      <data key="d4">get</data>
      <data key="d2">1 MiB</data>
    </node>
    <node id="pause">
      <data key="d0">1</data>
    </node>
    <node id="end">
      <data key="d1">1</data>
      <data key="d2">100 MiB</data>
      <data key="d0">3600</data>
    </node>
    <edge source="start" target="transfer" />
    <edge source="transfer" target="end" />
    <edge source="end" target="pause" />
    <edge source="pause" target="start" />
  </graph>
</graphml>
```

Finalmente se tiene el archivo de control de la simulación Shadow: shadow.config.xml

Para una mejor explicación se lo dividirá en partes:

Topology

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

Plugins

```
<plugin id="tgen" path="~/.shadow/lib/libshadow-plugin-tgen.so" />
<plugin id="tor" path="~/.shadow/lib/libshadow-plugin-tor.so" />
<plugin id="tor-preload" path="~/.shadow/lib/libshadow-preload-tor.so" />
<plugin id="torctl" path="~/.shadow/lib/libshadow-plugin-torctl.so" />
```

Services

```
<host id="fileserver" bandwidthdown="102400" bandwidthup="102400" >
    <process plugin="tgen" starttime="1" arguments="conf/tgen.server.graphml.xml" />
</host>

<host id="hiddenserver" bandwidthdown="102400" bandwidthup="102400" >
    <process plugin="tgen" starttime="1" arguments="conf/tgen.hiddenserver.graphml.xml" />
    <process plugin="tor" preload="tor-preload" starttime="900" arguments="--Address ${NODEID}
    --Nickname ${NODEID} --DataDirectory shadow.data/hosts/${NODEID} --GeoIPFile ~/.shadow/share/geoip
    --defaults-torrc conf/tor.common.torrc -f conf/tor.hiddenserver.torrc --BandwidthRate 1024000
    --BandwidthBurst 1024000" />
    <process plugin="torctl" starttime="901" arguments="localhost 9051
    STREAM,CIRC,CIRC_MINOR,ORCONN,BW,STREAM_BW,CIRC_BW,CONN_BW,BUILDTIMEOUT_SET,CLIENTS_SEEN,GUARD,CELL_
    STATS,TB_EMPTY,HS_DESC,HS_DESC_CONTENT"/>
</host>
```

Tor network Infrastructure

```
<!-- our Tor network infrastructure -->
<host id="4uthority" bandwidthdown="10240" bandwidthup="10240">
    <process plugin="tor" preload="tor-preload" starttime="1" arguments="--Address ${NODEID}
    --Nickname ${NODEID} --DataDirectory shadow.data/hosts/${NODEID} --GeoIPFile ~/.shadow/share/geoip
    --defaults-torrc conf/tor.common.torrc -f conf/tor.authority.torrc --BandwidthRate 1024000
    --BandwidthBurst 1024000" />
    <process plugin="torctl" starttime="2" arguments="localhost 9051
    STREAM,CIRC,CIRC_MINOR,ORCONN,BW,STREAM_BW,CIRC_BW,CONN_BW,BUILDTIMEOUT_SET,CLIENTS_SEEN,GUARD,CELL_
    STATS,TB_EMPTY,HS_DESC,HS_DESC_CONTENT"/>
</host>

<host id="exit" quantity="2" bandwidthdown="10240" bandwidthup="10240">
    <process plugin="tor" preload="tor-preload" starttime="60" arguments="--Address ${NODEID}
    --Nickname ${NODEID} --DataDirectory shadow.data/hosts/${NODEID} --GeoIPFile ~/.shadow/share/geoip
    --defaults-torrc conf/tor.common.torrc -f conf/tor.exit.torrc --BandwidthRate 1024000
    --BandwidthBurst 1024000" />
    <process plugin="torctl" starttime="61" arguments="localhost 9051
    STREAM,CIRC,CIRC_MINOR,ORCONN,BW,STREAM_BW,CIRC_BW,CONN_BW,BUILDTIMEOUT_SET,CLIENTS_SEEN,GUARD,CELL_
    STATS,TB_EMPTY,HS_DESC,HS_DESC_CONTENT"/>
</host>

<host id="relay" quantity="2" bandwidthdown="10240" bandwidthup="10240">
    <process plugin="tor" preload="tor-preload" starttime="60" arguments="--Address ${NODEID}
    --Nickname ${NODEID} --DataDirectory shadow.data/hosts/${NODEID} --GeoIPFile ~/.shadow/share/geoip
    --defaults-torrc conf/tor.common.torrc -f conf/tor.relay.torrc --BandwidthRate 1024000
    --BandwidthBurst 1024000" />
    <process plugin="torctl" starttime="61" arguments="localhost 9051
    STREAM,CIRC,CIRC_MINOR,ORCONN,BW,STREAM_BW,CIRC_BW,CONN_BW,BUILDTIMEOUT_SET,CLIENTS_SEEN,GUARD,CELL_
    STATS,TB_EMPTY,HS_DESC,HS_DESC_CONTENT"/>
</host>

<host id="bridge" iphint="100.0.0.1" bandwidthdown="10240" bandwidthup="10240">
    <process plugin="tor" preload="tor-preload" starttime="60" arguments="--Address ${NODEID}
    --Nickname ${NODEID} --DataDirectory shadow.data/hosts/${NODEID} --GeoIPFile ~/.shadow/share/geoip
    --defaults-torrc conf/tor.common.torrc -f conf/tor.relay.torrc --BandwidthRate 1024000
    --BandwidthBurst 1024000 --BridgeRelay 1" />
    <process plugin="torctl" starttime="61" arguments="localhost 9051
    STREAM,CIRC,CIRC_MINOR,ORCONN,BW,STREAM_BW,CIRC_BW,CONN_BW,BUILDTIMEOUT_SET,CLIENTS_SEEN,GUARD,CELL_
    STATS,TB_EMPTY,HS_DESC,HS_DESC_CONTENT"/>
</host>
```

Clients

```
<host id="client" >
    <process plugin="tgen" starttime="1200" arguments="conf/tgen.client.graphml.xml" />
</host>

<host id="torclient" >
    <process plugin="tor" preload="tor-preload" starttime="900" arguments="--Address ${NODEID}
    --Nickname ${NODEID} --DataDirectory shadow.data/hosts/${NODEID} --GeoIPFile ~/.shadow/share/geoip
    --defaults-torrc conf/tor.common.torrc -f conf/tor.client.torrc --BandwidthRate 1024000
    --BandwidthBurst 1024000" />
    <process plugin="torctl" starttime="901" arguments="localhost 9051
    STREAM,CIRC,CIRC_MINOR,ORCONN,BW,STREAM_BW,CIRC_BW,CONN_BW,BUILDTIMEOUT_SET,CLIENTS_SEEN,GUARD,CELL_
    STATS,TB_EMPTY,HS_DESC,HS_DESC_CONTENT"/>
    <process plugin="tgen" starttime="1200" arguments="conf/tgen.torclient.graphml.xml" />
</host>

<host id="torhiddenclient" >
    <process plugin="tor" preload="tor-preload" starttime="900" arguments="--Address ${NODEID}
    --Nickname ${NODEID} --DataDirectory shadow.data/hosts/${NODEID} --GeoIPFile ~/.shadow/share/geoip
    --defaults-torrc conf/tor.common.torrc -f conf/tor.client.torrc --BandwidthRate 1024000
    --BandwidthBurst 1024000" />
    <process plugin="torctl" starttime="901" arguments="localhost 9051
    STREAM,CIRC,CIRC_MINOR,ORCONN,BW,STREAM_BW,CIRC_BW,CONN_BW,BUILDTIMEOUT_SET,CLIENTS_SEEN,GUARD,CELL_
    STATS,TB_EMPTY,HS_DESC,HS_DESC_CONTENT"/>
    <process plugin="tgen" starttime="1200" arguments="conf/tgen.torhiddenclient.graphml.xml" />
</host>

<host id="torbridgeclient" >
    <process plugin="tor" preload="tor-preload" starttime="900" arguments="--Address ${NODEID}
    --Nickname ${NODEID} --DataDirectory shadow.data/hosts/${NODEID} --GeoIPFile ~/.shadow/share/geoip
    --defaults-torrc conf/tor.common.torrc -f conf/tor.client.torrc --BandwidthRate 1024000
    --BandwidthBurst 1024000 --UseBridges 1 --Bridge 100.0.0.1:9111" />
    <process plugin="torctl" starttime="901" arguments="localhost 9051
    STREAM,CIRC,CIRC_MINOR,ORCONN,BW,STREAM_BW,CIRC_BW,CONN_BW,BUILDTIMEOUT_SET,CLIENTS_SEEN,GUARD,CELL_
    STATS,TB_EMPTY,HS_DESC,HS_DESC_CONTENT"/>
    <process plugin="tgen" starttime="1200" arguments="conf/tgen.torclient.graphml.xml" />
</host>
```

Para realizar la simulación se realizarán cambios los archivos de generación de tráfico TGEN: client, torclient y torhiddenclient en el nodo id *end* y con parámetro d1 *count* de 1, 10 y 50 respectivamente.

```
<node id="end">
  <data key="d1">1</data>
  <data key="d2">100 MiB</data>
  <data key="d0">3600</data>
```

Gráficas comparativas de las Simulaciones 1, 2 y 3 realizadas.

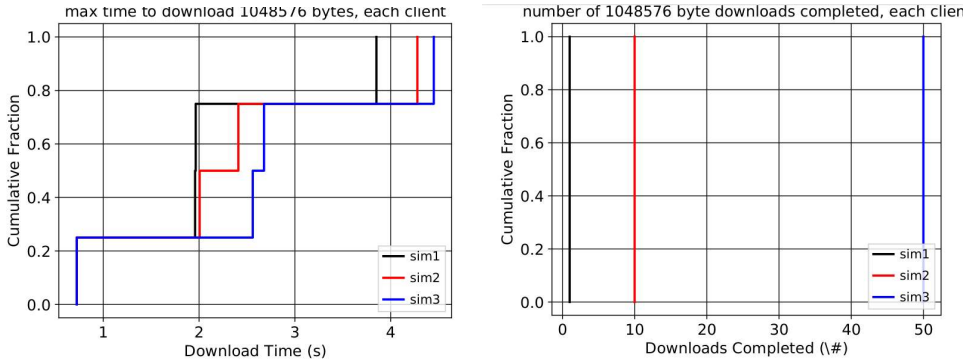


Ilustración 15 Fig. a. Resultado Ejercicio 1: Descargas Completadas

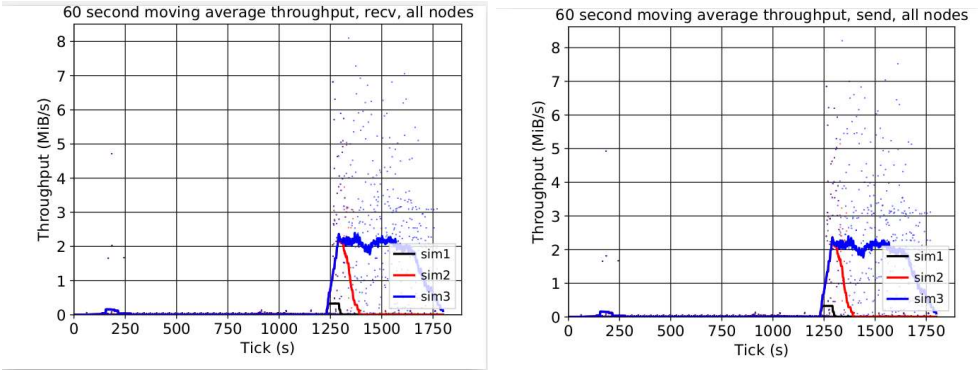


Ilustración 16 Fig. b.Resultado Ejercicio 1: 60seg Average Throghput

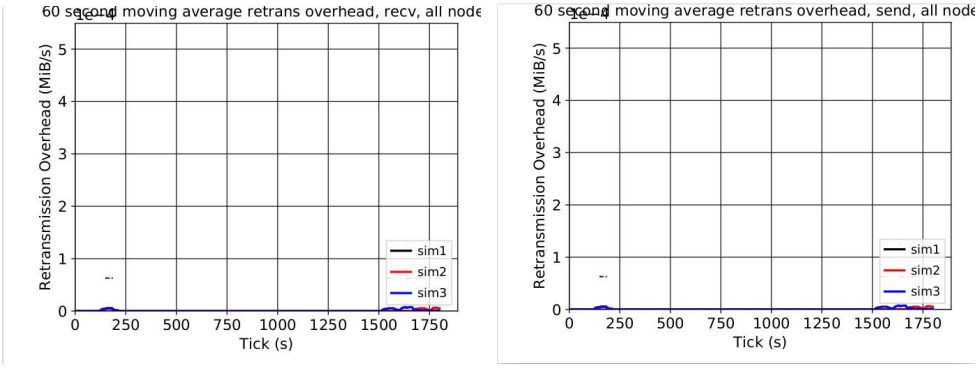


Ilustración 17 Fig. c. Resultado Ejercicio 1: 60seg. Retransmissions

Observaciones:

Como se observa, en general, se puede realizar una comparación estadística del comportamiento de las 3 simulaciones. Las tablas

muestran un comportamiento idóneo sin latencia y sin pérdida de paquetes, en el cual todas las conexiones se realizan, tomando en cuenta que en la primera simulación existe una sola conexión mientras que en la segunda simulación tiene 10 descargas y la tercera simulación tiene 50 descargas al servicio oculto (Hidden Service).

En la gráfica b se puede sacar que el máximo throuthput movido durante los 60 segundos de la simulación es de 2 MBps, adicional se no se observa o existe mínimo de retransmisiones durante la transmisión de datos; esto se debe a que se lo está realizando en un ambiente óptimo y controlado.

Ejercicio 2.

La siguiente simulación tendrá como objetivo simular un ambiente más real con pérdidas de paquete y luego lo compararemos con el similar sin pérdida de paquetes.

En la topología de la simulación (shadow.config.xml) se realizarán cambios en los parámetros de latencia y paquetes perdidos (d7 y d9 respectivamente)

```
<edge source="poi-1" target="poi-1">
        <data key="d7">100.0</data>
        <data key="d8">0.0</data>
        <data key="d9">0.8</data>
      </edge>
```

Resultados comparativos entre tráfico generado entre comunicación ideal vs. con latencia y perdida de paquetes:

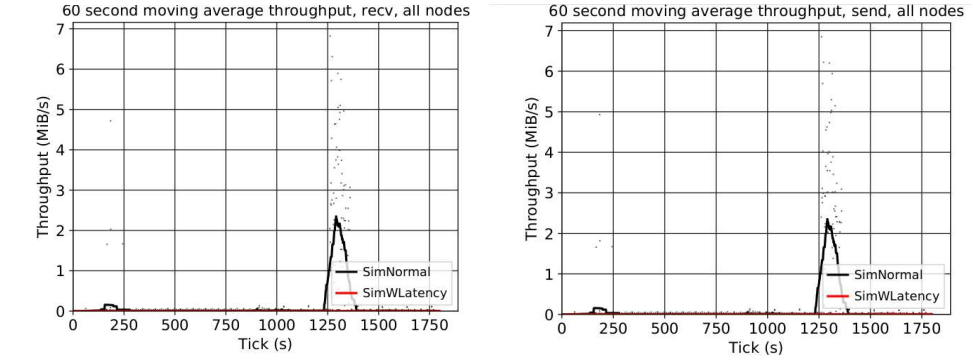


Ilustración 18 Fig. a. Resultado Ejercicio 2: 60seg Average Throghput

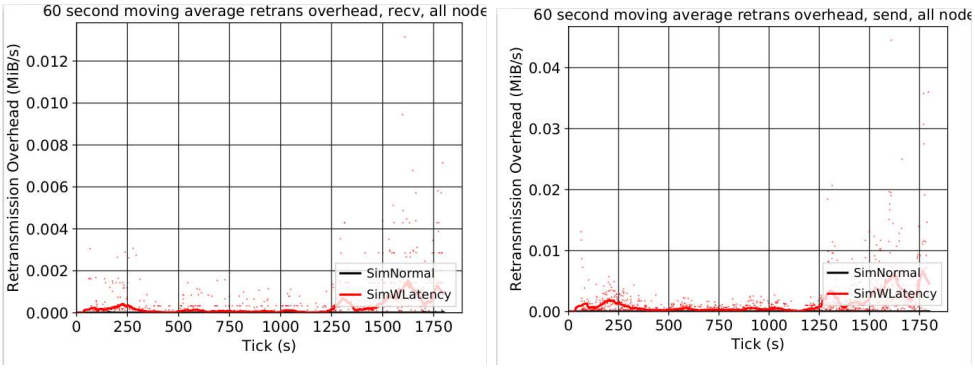


Ilustración 19 Fig. b. Resultado Ejercicio 2: 60seg. Retransmissions

En las tablas se observa los datos de Negro (SimNormal) que muestra la simulación de en la red Tor idónea y los datos de Rojo (SimWLatency) muestra la simulación Tor con latencia y perdida de paquetes.

En los gráficos de enviado y recibido de datos, se observa que en los primeros 60 segundos la simulación SimNormal tiene un pico de 2 MBps, mientras que el SimWLatency no se observa transmisión o transmisión mínima de datos. Adicional se aprecia en los diagramas de retransmisiones que la simulación SimWLatency presenta mucha latencia durante la simulación.

Con esta información inicial se evalúa el comportamiento del tráfico de la comunicación cliente Tor a un servicio oculto podría tener en ambientes idóneos o bajo ciertas circunstancias configuradas.

4. Cibermercados

En los apartados anteriores hemos intentado simular un ambiente de conexión Tor con el cual se ha realizado el estudio técnico del funcionamiento de la red y el uso del simulador Shadow. A continuación, se realizará el estudio de la Deepweb y los cibermercados; de como realmente son y funcionan; se ingresará a la Deepweb y analizará toda la información obtenida durante la navegación.

4.1. Buscadores y Direcciones Onion

Dentro del proyecto Tor se encuentra aplicaciones desarrolladas con las cuales se puede tener acceso a la red Tor de tal manera de llegar a los servicios ocultos. A continuación, detallaremos un par de las aplicaciones más conocidas y usados por los usuarios:

TorBrowser
TorBrowser es un proyecto con el cual permite la navegación web dentro de la red Tor mediante el navegador web Firefox. El Firefox en su versión ESR (Extended Support Release) es alterado con los plugin y configuraciones necesarios para la navegación segura; adicional este aplicativo posee varias características el cual permite la conexión con

proxy externos o puentes necesarios según la política de entrada deseada.

En varios casos el protocolo Tor es censurado por políticas locales empresariales mediante el administrador de red o en casos más extremos por el gobierno local como es el caso de China. Estos bloqueos pueden realizarse mediante el bloqueo de flujo de datos Tor que ciertos equipos especializados (Firewalls con módulos de detección de aplicaciones, flujo de datos o módulos DPI) poseen; otra forma de bloqueo es realizado mediante el bloqueo de la lista de Conexiones IP conocidas del directorio de Tor.

Cualquier que sea el caso, el proyecto Tor ha proporcionado herramientas para evadir estos bloqueos utilizando conexiones proxys a Puentes (Tor Bridges) durante la primera conexión de tal manera que se pueda ingresar a la red Tor sin restricción; estos puentes son utilizados mediante técnicas de transporte de ofuscación con la cual se intenta evadir los controles de los equipos administradores de redes; entre las diferentes técnicas de trasporte se encuentran obfs4, meek, Format- Transforming Encryption y ScambleSuit. Adicional a esto es posible solicitar al proyecto Tor listas de conexión nuevas en el caso de que las listas indicadas previamente se encuentren bloqueadas.

A continuación, se detalla una gráfica del proceso de instalación del TorBrowser y el uso de los Bridges:

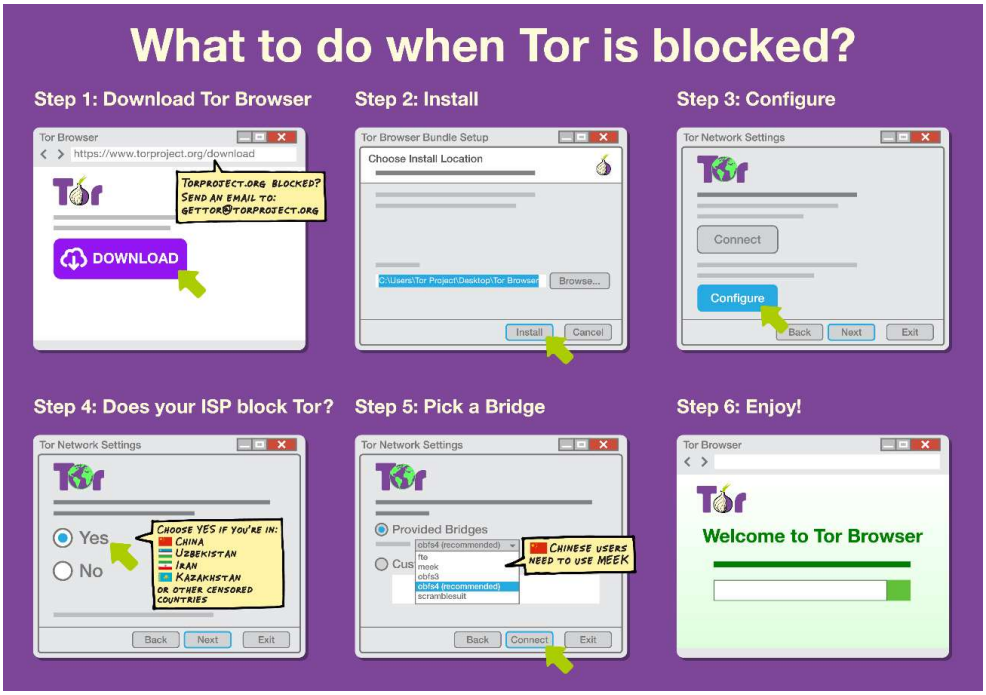


Ilustración 20 Instalación de TorBrowser [Fuente:[15]]

Tail OS

Tails significa "The Amnesic Incognito Live System"; esta es una distribución en Debian 6 que tiene Tor instalada entre sus aplicaciones. Como sistema operativo dispone de ciertos programas configurados para uso personal y de anonimato; entre estos aplicativos se encuentran: Firefox, Pidgin, KeepassX, Metadata Anonymistation Toolkit, etc.

Tails es un Live System o como se dice una Live Distro en el cual el Sistema Operativo se ejecuta en la memoria RAM del equipo, por tal razón cuando se apague o reinicie el PC o sistema Operativo Tails se eliminarán los datos generados durante el uso y así asegurar el anonimato.

Direcciones Onion

En la navegación habitual, cada servicio y pagina web se encuentra habilitado públicamente para cualquier usuario, utilizando cualquier explorador; así mismo, cada búsqueda realizada con Google u otro servicio de búsqueda se encuentra indexada, con el que es posible fiar aproximadamente el 100% de cualquier consulta realizada. Sin embargo, en la Deep Web hay que tener varias consideraciones para poder navegar o utilizar sus servicios:

1. Todos los dominios y nombres de servicios tienen la extensión. onion
2. Los servicios no son públicos y los nombres de dominios en la mayoría de las veces no reflejan el servicio ofrecido.

Ejemplo de sitio web onion:

Ilustración 21 Información del Sitio DuckDuckGo en Sitio Onion

3. Para acceder a los servicios web se requiere estar en la red Tor por lo que es necesario utilizar el TorBrowser.
4. Como los servicios de la red Tor no se encuentran indexados, no son fácilmente accesibles, muchos inclusive requieren autorización para acceder a sus servicios.

Para acceder a los servicios de la Deep Web es necesario utilizar algún repositorio o buscadores de listas de links onions, estas listas tendrán los sitios que podrán ser accesibles y otros sitios que se encuentren caídos debido a que el proveedor lo ha sacado de línea o fueron bloqueados.

Entre los principales buscadores en la red Tor tenemos:

Buscador	Sitio Web .Onion
DuckDuckGo	http://3g2upl4pq6kufc4m.onion/
The Abyss	http://nstmo7lvh4l32epo.onion/
Torgle	http://zw3crggtadila2sg.onion/torgle/index.html
Torch	http://xmh57jrzrnw6insl.onion/
Deep Search	http://xycpusearchon2mc.onion
Directorio Tor	http://4yiyky5kpplda4y.onion/
TorFind	http://ndj6p3asftxboa7j.onion/
Yacy	http://yacy2tp5a2dhywmx.onion
Not Evil	http://hss3uro2hsxfogfq.onion/
Onion	http://skunksworkedp2cg.onion/

4.2. Estudio de Servicios Ocultos y Comercio electrónico.

Con el objetivo de realizar un estudio más directo a los servicios ocultos de la Deep Web se creará un ambiente no controlado pero lo más seguro posible durante un acceso a la red Tor. La finalidad de este apartado es mostrar el ingreso a la red Tor y conocer un poco el manejo de los mercados y negocios que esta red mantiene.

Como se ha indicado en varias ocasiones, la navegación en los servicios de Tor puede ser oculta, pero en ciertas situaciones nuestra información se puede haber comprometida por el servicio o página web que se quiere acceder mediante algún track o malware que puede venir embebida o compartida.

Consideraciones para navegación dentro de la red Tor:

- En lo posible utilizar algún medio de VPN privado con la finalidad de ocultar el originen de la conexión y darle un nivel adicional de seguridad a la conexión.
- No ingresar a servicios con cuentas personales como correos, redes sociales o cuentas bancarias de tal manera que pueden ser robadas por vulnerabilidades del explorador o un malware o keylogger insertado durante la navegación.
- Permanecer anónimo durante cualquier sesión; esto quiere decir que no se deben ingresar datos personales o algún dato que permita alguna estafa.
- Establecer límites durante la navegación y estar seguro a los sitios a visitar teniendo en cuenta en todo momento que muchas de las paginas dentro de la red Tor son servicios con contenido ilegal.
- No habilitar o instalar complementos
- Validar que la conexión sea HTTPS
- No abrir documentos descargados en la sesión a la red Tor.

Para permanecer en el anonimato se ha tomado la consideración de crear un entorno de sistema operativo nuevo sin ningún tipo de datos guardados en el cual comprometa al usuario. Se recomienda utilizar el sistema operativo Tails del mismo proyecto Tor. Para la presente práctica se utilizará Tails dentro del hipervisor Virtual Box.

Como el objetivo del presente modulo no es observar los aspectos técnicos de la instalación de Tails, si no realizar un estudio de los Cibermercados, se adjuntará a los anexos el link del proyecto Tails con los pasos a seguir para la instalación.
Proyecto Tails: https://tails.boum.org/

4.2.1. Navegación dentro de la red Tor

Luego de realizar la ejecución de Tails o la instalación del TorBrowser en el sistema Operativo, procederemos a revisar el estado de conexión dentro de la red Tor para posterior realizar el estudio de los sitios web dentro de unos de los buscadores más conocidos, seguido se procederá a revisar cuales son algunos de los servicios ilegales ofrecidos y cuál es su comercio.

a. Estado de la conexión Tor

Para verificar el estado de conexión dentro de la red Tor se lo realizará mediante el chequeo de la IP de origen utilizando algún sitio web de servicio de geolocalización mediante IP como https://www.ip2location.com; adicional, para conocer si estamos dentro de la red Tor se lo realizará el chequeo mediante el servicio de Tor Check (**https://check.torproject.org/**).

Geolocalización mediante ip2location.com

Sin TorBrowser		Con TorBrowser	
Your IP Address 186.3.	ISP Clientes Netlife Guayaquil - Gepon	Your IP Address 94.230.208.147	ISP Nine Internet Solutions AG
Country Ecuador	Coordinates	Country Switzerland	Coordinates 47.5584, 7.57327
Region Guayas	Usage Type ISP	Region Basel-Stadt	Usage Type DCH
City Guayaquil	Domain telconet.net	City Basel	Domain nine.ch
ZIP Code 090	Elevation 8 m	ZIP Code 4003	Elevation 277 m

Tor Check

Sin TorBrowser	Con TorBrowser
Sorry. You are not using Tor. Your IP address appears to be: 186.3.176.203 If you are attempting to use a Tor client, please refer to the Tor website and specifically the instructions for configuring your Tor client.	Congratulations. This browser is configured to use Tor. Your IP address appears to be: 91.121.251.65 Please refer to the Tor website for further information about using Tor safely. You are now free to browse the Internet anonymously. For more information about this exit relay, see: Relay Search.

b. Resultados Onion en la búsqueda de servicios ocultos.

Utilizando el enlace: deepweblinks.org observamos que existen varios servicios según su clasificación.

- Hidden Service lists and search engines
- Marketplace Financial

- Marketplace Drugs
- Hosting
- Blogs
- Forums and Chans
- Email and Messaging
- Political
- Warez
- Erotic 18+
- Erotic Hard Candy
- Erotic Jailbait
- Non-English

Al analizar el sitio web de búsqueda TORCH, se observa publicidad de servicios como venta de drogas, tarjeta de créditos o cuentas de pago como PayPal, servicios sexuales, entre otros.

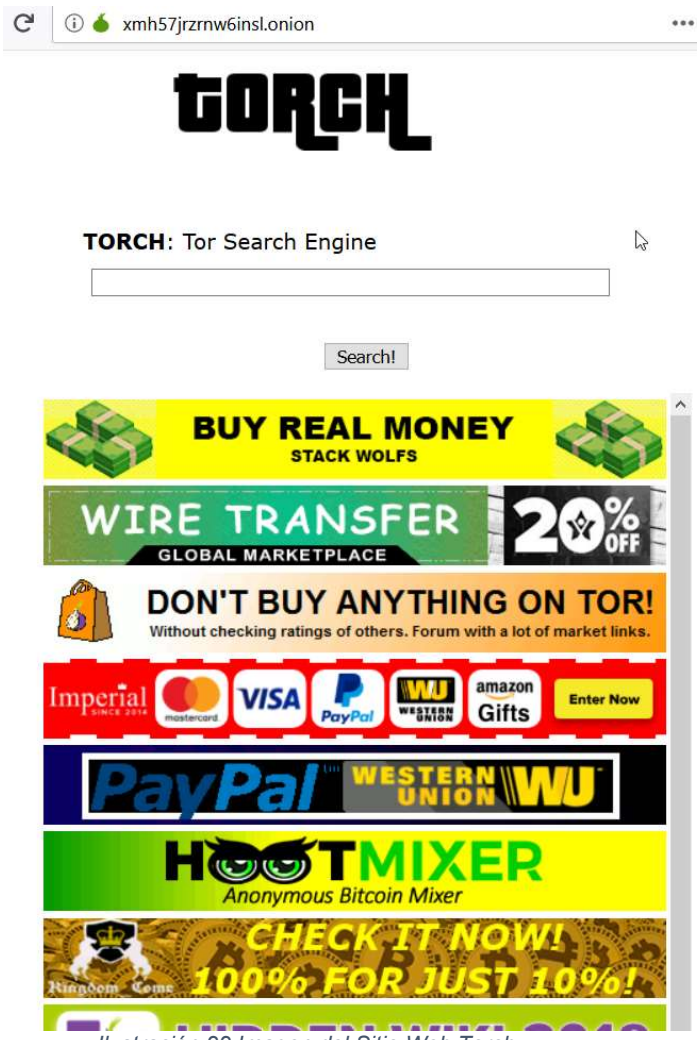
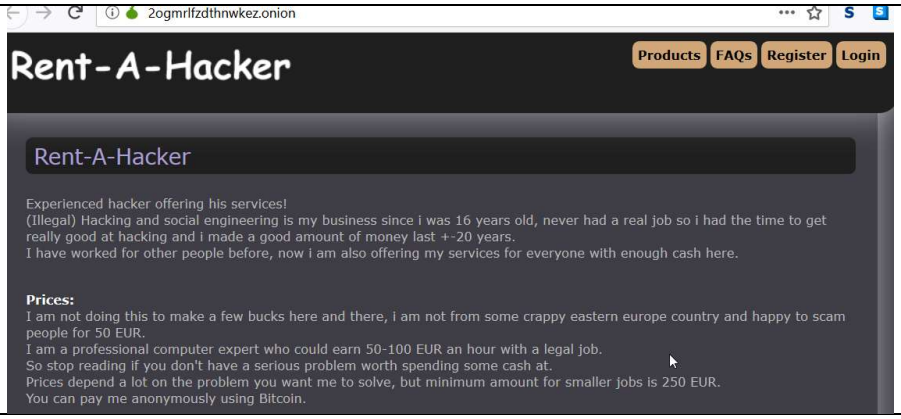


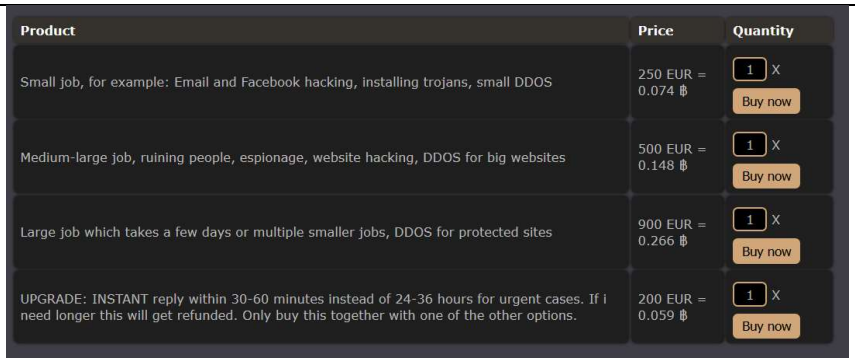
Ilustración 22 Imagen del Sitio Web Torch

Como referencia se tomará como ejemplo 3 sitios web categorizados como Servicios de Hacker, Venta de Equipos, Venta de Tarjetas de Crédito y Drogas:

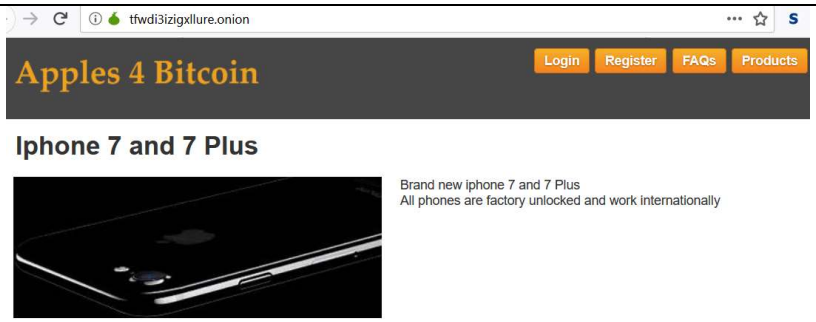
Rent-A-Hacker
Web: http://2ogmrlfzdthnwkez.onion/
Sitio Web:



Productos:



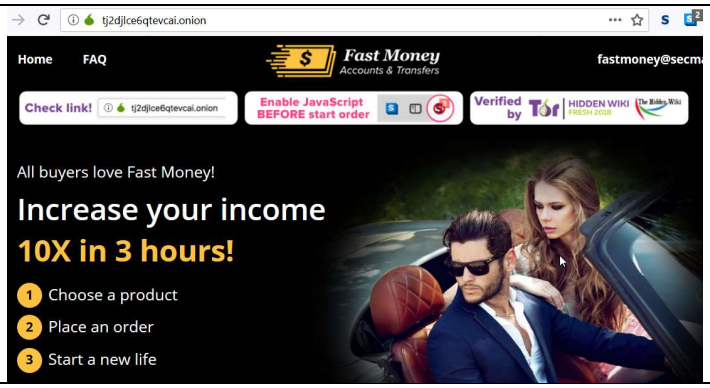
Apples 4 Bitcoin
Web: http://tfwdi3izigxllure.onion/
Sitio Web:



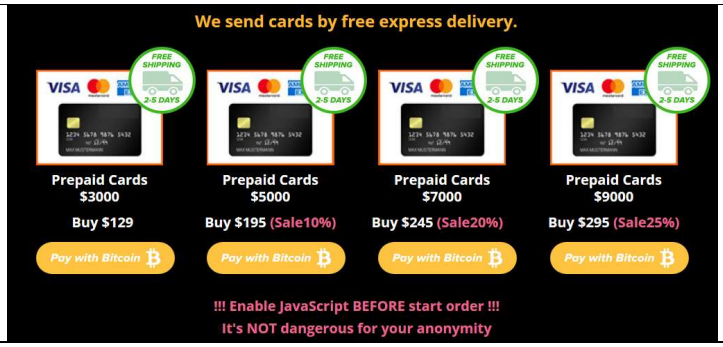
Productos:

Product	Price	Quantity
Iphone 7 128gb Jetblack	325 USD = 0.086 Ƀ	1 X Buy now
Iphone 7 plus 128gb Jetblack	400 USD = 0.106 Ƀ	1 X Buy now
Iphone 7 128gb Black	325 USD = 0.086 Ƀ	1 X Buy now
Iphone 7 plus 128gb Black	400 USD = 0.106 Ƀ	1 X Buy now

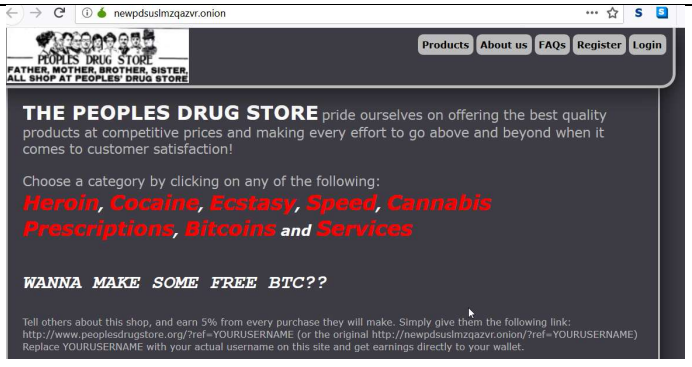
Fast Money
Web: http://tj2djlce6qtevcai.onion/
Sitio Web:



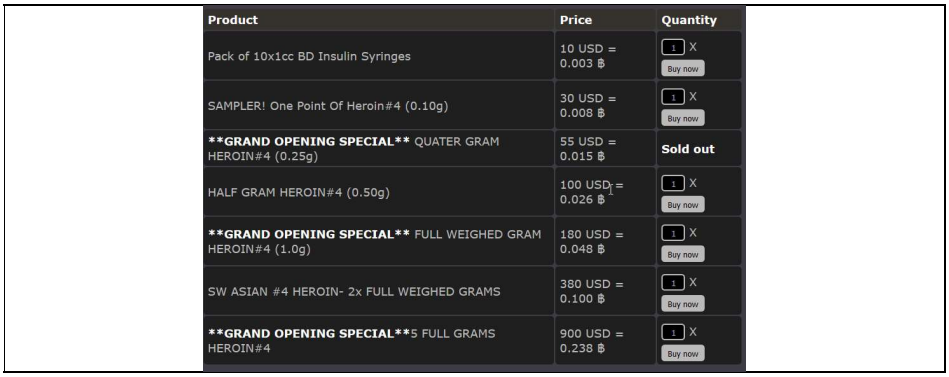
Productos:



The Peoples Drug Store
Web: http://newpdsuslmzqazvr.onion/
Sitio Web:



Productos:



Product	Price	Quantity
Pack of 10x1cc BD Insulin Syringes	10 USD = 0.003 ฿	1 X Buy now
SAMPLER! One Point Of Heroin#4 (0.10g)	30 USD = 0.008 ฿	1 X Buy now
****GRAND OPENING SPECIAL**** QUATER GRAM HEROIN#4 (0.25g)	55 USD = 0.015 ฿	**Sold out**
HALF GRAM HEROIN#4 (0.50g)	100 USD = 0.026 ฿	1 X Buy now
****GRAND OPENING SPECIAL**** FULL WEIGHED GRAM HEROIN#4 (1.0g)	180 USD = 0.048 ฿	1 X Buy now
SW ASIAN #4 HEROIN- 2x FULL WEIGHED GRAMS	380 USD = 0.100 ฿	1 X Buy now
****GRAND OPENING SPECIAL****5 FULL GRAMS HEROIN#4	900 USD = 0.238 ฿	1 X Buy now

Observaciones de los sitios visitados:

- Los sitios web ingresados como ejemplo, pertenecen a servicios ilegales, los cuales es muy común dentro de la red Tor por permitir la anonimización y evitar el rastreo de su propietario.
- Algo muy importante que se aprecia en todos los casos es el uso del BITCOIN [2] como moneda de pago para casi todos los servicios ilegales.
- Adicional, se pudo notar que ciertos sitios, como en el de venta de tarjetas de crédito, el administrador solicita previo al pago que se active el javascript en el navegador, el cual puede ser indicio a un robo de información al usuario, especialmente a usuarios que no conocen estos temas técnicos de seguridad.

No todos los sitios visitados son ilegales; existen casos como páginas de audio o foros en la cual los usuarios utilizan sus servicios y lo único que desean es permanecer anónimos.

4.3. Bitcoin

Entre los métodos de pago que se pudo observar de cada sitio del ejemplo anterior, utilizan y hacen referencia al Bitcoin como moneda de intercambio. Esto se debe a que el uso de Tarjeta electrónica y transferencia de dinero compromete directa o indirectamente a los ciberdelincuentes.

El bitcoin es por muchos considerado la nueva monedad digital, muy similar al dólar, euro u otra moneda del mercado. Pero técnicamente el bitcoin es un protocolo y red P2P el cual es utilizado como criptomoneda y sistema de pago.

El bitcoin posee características muy innovadoras, ya que debido a su tecnología es difícilmente falsificable.

Una de las principales características es que es descentralizado, eso quiere decir, que no requiere ningún intermediario para realizar las transacciones como bancos o gobierno; y esta es la razón por lo que es

muy utilizado por los cibercriminales ya que el registro de las transacciones es completamente anónimo. A continuación, una gráfica representando los dos tipos de transacciones: la habitual (ejemplo utilizando Visa) que requiere un ente bancario y la innovadora o con Bitcoin el cual son transacciones realizadas directamente entre usuarios.

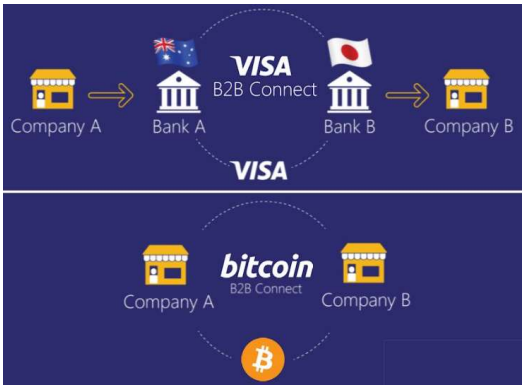


Ilustración 23 Comparativa de Forma de Pago Tradicional vs Bitcoin

Para entender el funcionamiento del Bitcoin se puede resumir a este como un libro de registro de transacciones; cada libro se lo llamará bloque donde se agrupan una cantidad limitada de transacciones bitcoin, adicional estos bloques están unidos cronológicamente a otros bloques de transacciones a lo que se llama cadena de bloques o Blokchain [1] [7]. Todos los bloques poseen características técnicas especiales que permite mantener la seguridad y la veracidad de los datos.

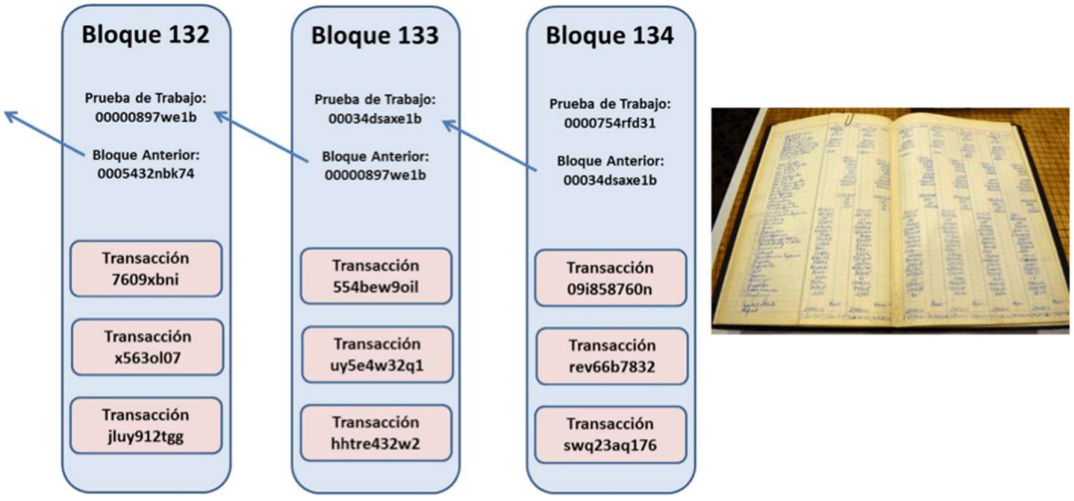


Ilustración 24 Estructura de Blockchain

Las partes generales de un bloque son:

- Hash del bloque anterior: es el hash generado por el bloque anterior
- Transacciones: Son todas las transacciones realizadas durante un tiempo determinado los cuales son registradas en el bloque

- Prueba del trabajo: es la respuesta con el cual se va a obtener un determinado hash. En este caso son las operaciones llevadas por los mineros con la finalidad de obtener un hash cuyos valores comiencen con 18 ceros y por la cual no existe alguna función para su cálculo.

A continuación, se detalla el hash de un bloque y sus transacciones contendidas en el mismo:

Ejemplo obtenido de Blockchain.com: *Bloque # 555853*
Link: https://www.blockchain.com/btc/block-height/555853

Información de Hashes:

Hashes	
Hash	00000000000000000023fd370ca75e76b803e90776ff87435161960149ce4a0a
Bloque Anterior	00000000000000000293eba21be1f4b401938e7785861e1a82707bfa5ec356c
Bloque(s) siguiente(s)	00000000000000000149e18da3aebbe10087a34867338ab828d58b68a699a58
Raíz de Merkle	ba5873060de08f76c1862c692a6fa65904ed70a49d0f63cda5e0c48c35f1ab7f

Ilustración 25 Hashes reales de un Bloque de BTC

Información del Bloque:

Resumen	
Número de Transacciones	2476
Total de salida	5,365.4551489 BTC
Volumen Estimado de la Transacción	513.62383466 BTC
Comisiones de la Transacción	0.04922199 BTC
Altura	555853 (Cadena principal)
Fecha y Hora	2018-12-28 05:22:39
Hora de Recepción	2018-12-28 05:22:39
Resuelto por	BTC.TOP
Dificultad	5,106,422,924,659.82
Bits	389488372
tamaño	1131.724 kB
Peso	3992.833 kWU
Versión	0x20000000
Mientras tanto	3730942809
Recompensa del Bloque	12.5 BTC

Ilustración 26 Tabla informativa de un Bloque BTC

Como se observa dentro de este bloque han existido 2476 transacciones y la recompensa para el minero es de 12.5 BTC

Las transacciones o Actas de transacciones se muestran de la siguiente manera:

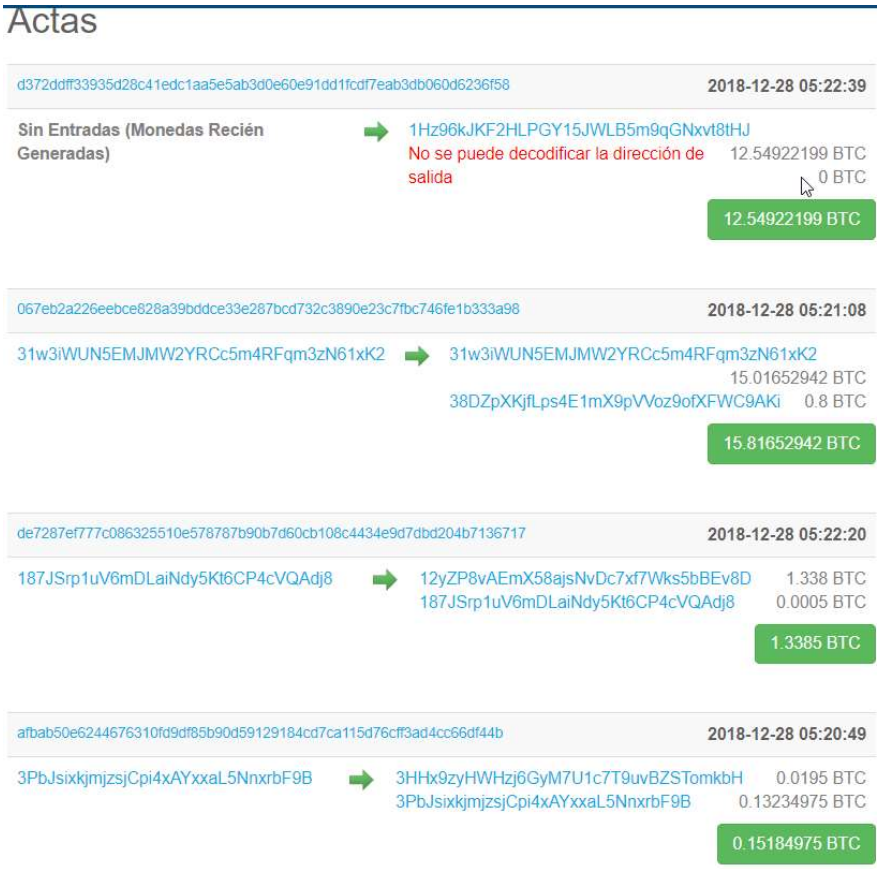


Ilustración 27 Actas de Transacciones grabadas en un Bloque de BTC

Nuevamente, si se observa en detalle cada una de las transacciones realizadas estas poseen información de las billeteras de bitcoin del que envía y del que recibe y los valores que son transferidos; pero en ninguna parte de la información se detalla los datos del dueño de las cuentas, por lo que el uso del bitcoin mantiene el anonimato de sus usuarios y al mismo tiempo prevalece su integridad y seguridad de información.

El bitcoin es muy usado dentro de la Deepwep y actualmente ha tenido mucha acogida en el comercio del mercado negro y el mercado legal o mejor dicho en la web normal. Instituciones bancarias y medios de pagos de confianza en tiendas en línea y físicas están acogiendo más este tipo de criptomoneda.

4.4. Estadísticas

Tor Metric

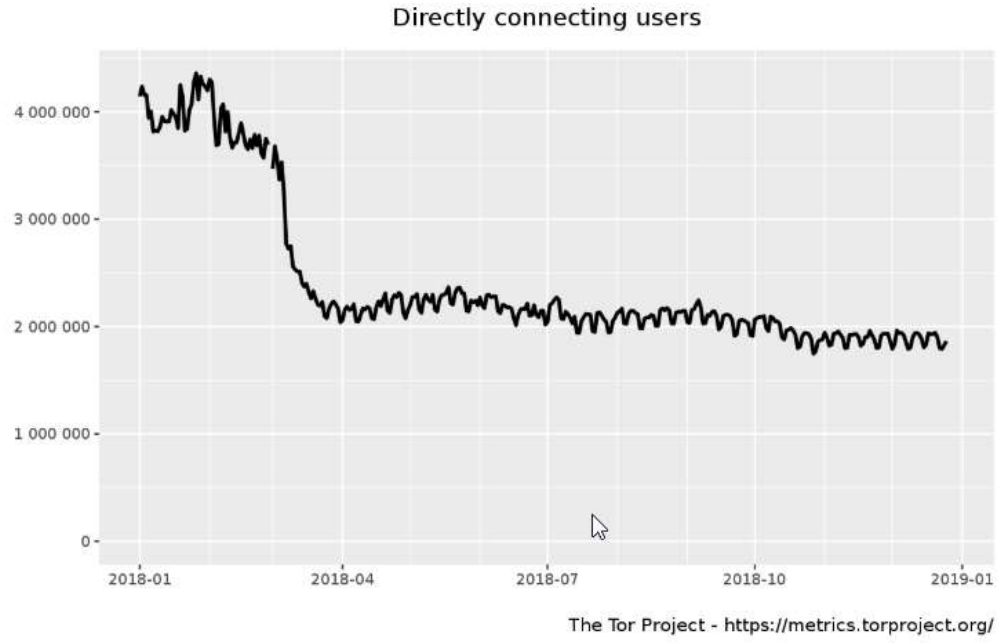
Para conocer el estado actual de la red Tor, el proyecto Tor dispone de la herramienta Tor Metric con el cual se puede observar los siguientes datos:

- Usuarios Conectados a la red Tor: Tipo de Conexión, Cantidad de Usuarios, Origen, etc.
- Servidores (Relays y Bridges)
- Trafico de la red Tor
- Capacidades de la red Tor

Entre los datos más relevantes podemos destacar los siguientes:

Usuarios Conectados

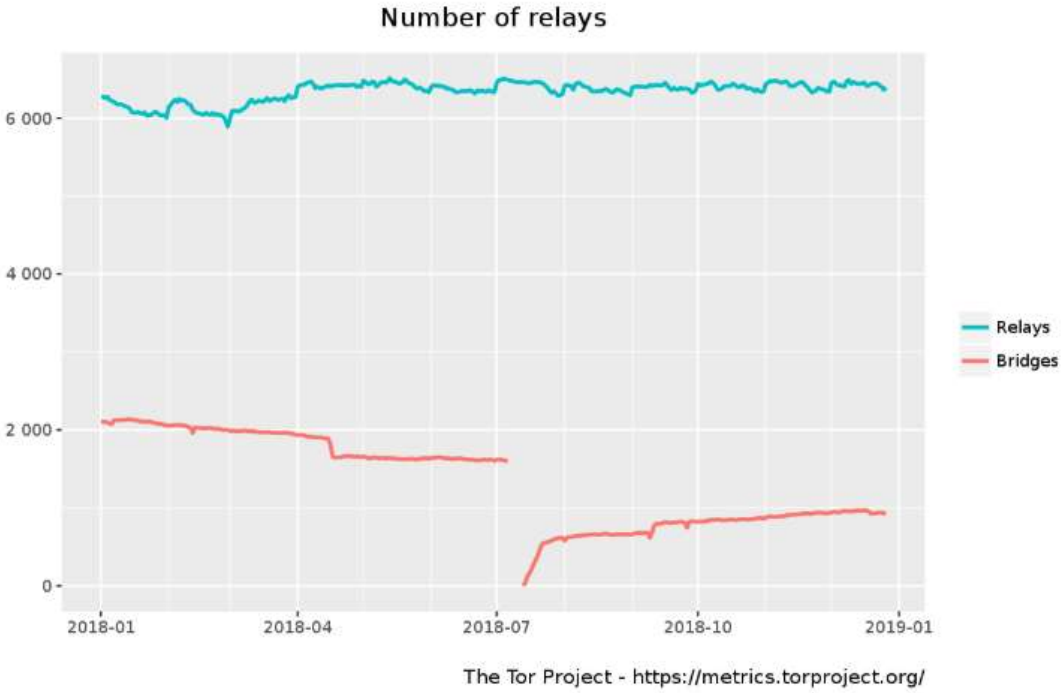
Rango de Muestra: 2018-01-01 al 2018-12-28

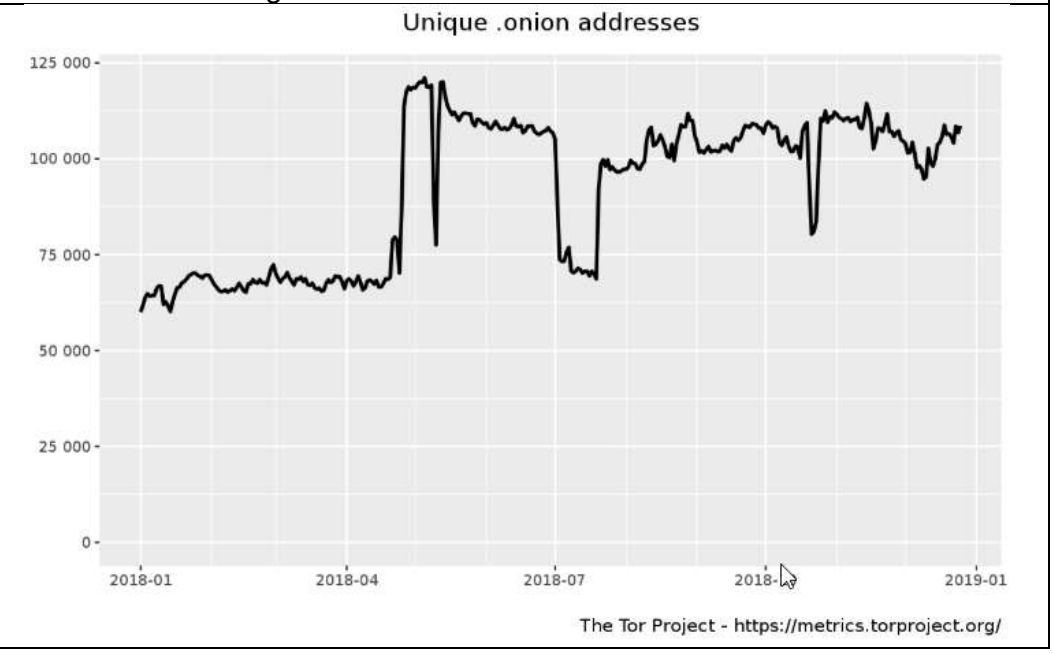


Media de los Principales Países con Usuarios Relays en la red Tor

Rango de Muestra: 2018-01-01 al 2018-12-28

Country	Mean daily users
Russia	10657 (17.04 %)
United States	5168 (8.26 %)
Iran	4940 (7.90 %)
Turkey	3633 (5.81 %)
India	2819 (4.51 %)
Indonesia	2636 (4.21 %)
Ukraine	1975 (3.16 %)
Brazil	1975 (3.16 %)
Vietnam	1922 (3.07 %)
Egypt	1915 (3.06 %)

Numero de Relays
Rango de Muestra: 2018-01-01 al 2018-12-28


Direcciones .Onion
Rango de Muestra: 2018-01-01 al 2018-12-28


Analizando la información recolectada desde el 1 de enero del 2018 al día de la toma de muestra 28 de diciembre del 2018, se puede apreciar que existió una media de aproximadamente 4 millones de usuarios conectados hasta inicios de marzo del 2018 luego del cual los usuarios conectados fueron de aproximadamente 2 millones.

También observamos que los principales usuarios de la red Tor son de Rusia con una media de 10657 usuarios conectados por día; y finalmente Tor Metrics registra una cantidad mayor a 100000 direcciones Onion.

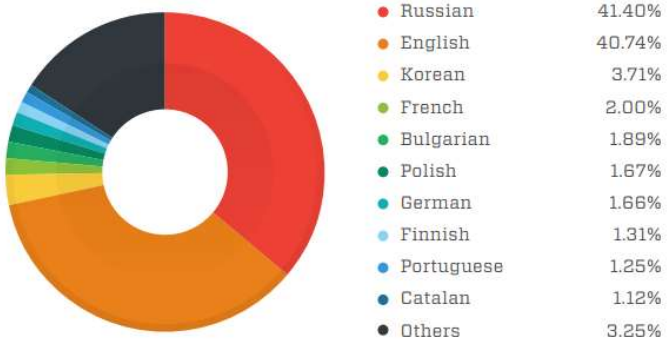
Con esta herramienta métrica (Tor Metrics) es posible tener una noción de la dimensión de la red Tor inclusive su estado actual.

DeWa

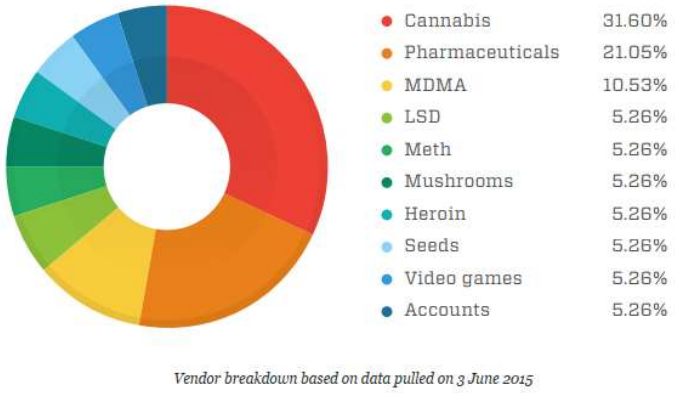
Por otra parte, el grupo de Investigación de amenazas de Trend Micro (FRT); ha realizado un estudio de más de dos años sobre el contenido de la Deep Web. La información referente a ese estudio se encuentra publicado en el artículo “Below the Surface: Exploring the Deep Web” [8].

El estudio se basó en la recolección de 576000 páginas web onion utilizando el Deep Web Analyser o DeWa. El cual consiste en extraer información de los sitios web, chequearla, indexarla y finalmente realizar una visualización cualitativa de sus resultados.

Entre las principales estadísticas mostradas por el DeWa se puede resumir las siguientes:

Distribución de Lenguaje
 *Most popular languages based on the number of URLs with content using them*
Muestra: 3454 dominios

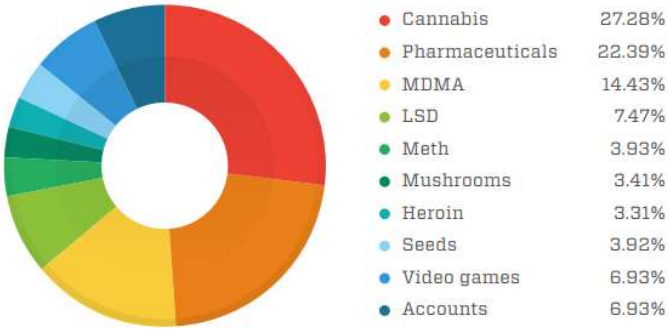
Venta según el tipo de Drogas



Vendor breakdown based on data pulled on 3 June 2015

Muestra: 15 Sitios de Comercio Electrónico

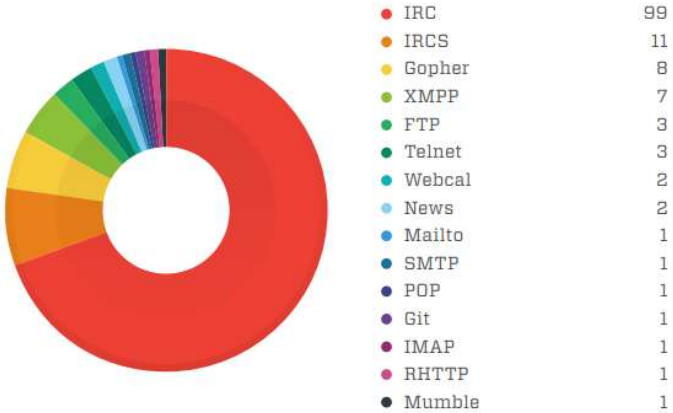
Compra Según el tipo de Drogas



Buyer breakdown based on data pulled on 3 June 2015

Muestra: 15 Sitios de Comercio Electrónico

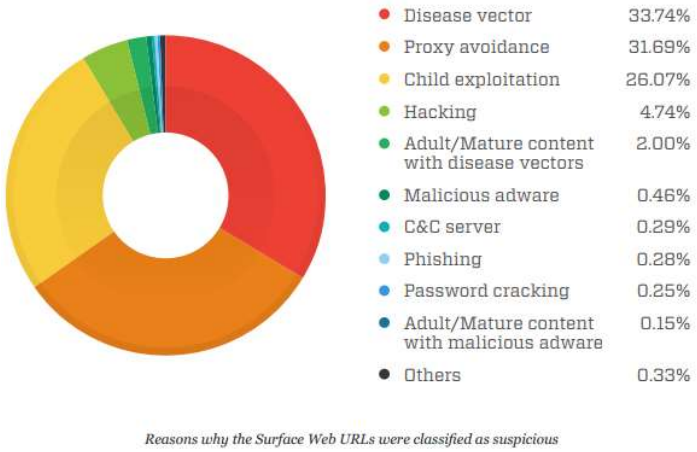
Protocolos Usados en la Deep Web in contar HTTP/HTTPS



Protocols found in the Deep Web apart from HTTP/HTTPS

Muestra: 22000 dominios

Clasificación de Paginas en la DeepWeb



Reasons why the Surface Web URLs were classified as suspicious

Muestra: 8707 paginas sospechosas

Con los datos analizados por Trend-Micro observamos nuevamente que predominan las paginas en ruso y probablemente con mayo acogida sitios ilegales de cibercriminales pertenecientes a este sector; y con similar porcentaje se tiene sitios en Ingles.

Se observa también que entre las principales drogas expendidas en la Deepweb se encuentran el Cannabis, Drogas sujetos a prescripción médica y la MDMA o más conocida como éxtasis. Si por ejemplo, comparamos esta información con la proporcionada por el ONU en su "Informe mundial sobre Drogas del 2017" [6], está acorde con los porcentajes de consumo:

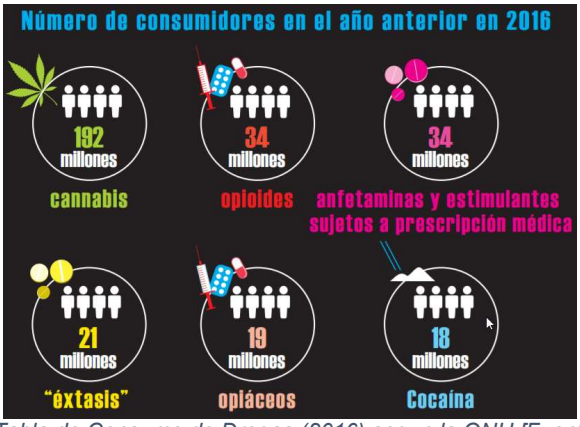


Ilustración 28 Tabla de Consumo de Drogas (2016) segun la ONU [Fuente: [6]]

En la última grafica estadística "Clasificación de Páginas en la Deepweb" se puede apreciar que tipo de cibermercados predominan en la red Tor. En primer lugar, se encuentra páginas con sospechas de virus, las cuales permiten descargas de software ilegalmente con posible contenido de virus y malware. Seguido estadísticamente, se observa que los usuarios utilizan la red para evitar bloqueos, estos pueden ser bloqueos geográficos o locales; y en tercer lugar y como dato asombroso existe un

aproximado de 26% de páginas cuyo cibermercado negro es el abuso infantil, a lo que se refiere directamente a pornografía infantil.

Todos estos datos de ciber-mercados son estadísticas que pretenden acercarse a la realidad. No se puede precisar cuántos negocios existen realmente debido al ambiente donde operan que es la red Tor.

5. Conclusiones

Durante el desarrollo de la presente TFM se realizó el estudio técnico de la red Tor y sus componentes; con el que se analiza en detalle el funcionamiento de su protocolo, creación de circuitos y demás módulos. Con estos detalles se concluye que la red Tor permite la anonimización de sus usuarios debido a sus capas de encriptación y tipo de enrutamiento que este realiza. Así mismo, se observa que existen puntos en el cual la conexión puede afectar su anonimización, como es el caso del nodo de salida, ya que toda la información se desencripta para llegar a su destino.

Adicional del análisis del funcionamiento del protocolo Tor; se realizó el estudio de la herramienta Shadow, el cual permitió simular un ambiente de la red Tor de manera controlada. Esta herramienta posee mucho potencial, no solo simulando ambientes de red Tor, si no que permite realizar y vincular aplicaciones para luego simular su comportamiento con muchos hosts y/o saltos de red, lo cual, dentro de una simulación, optimiza recursos innecesarios como los que crea el mismo Sistema Operativo. Lo negativo de la herramienta está en la complejidad de su uso, ya que este solo se administra a nivel de comando y archivos de configuración y solo se encuentra codificado a nivel de Linux.

Para el estudio de los Cibermercados se realiza un ingreso a la red Tor, en el cual se observa y analiza ciertas páginas web al azar. Como conclusión de este estudio y análisis se determina la ilegalidad de los negocios en que estos operan. Adicional se considera, relacionando y complementa este análisis con datos estadísticos de terceros.

Además, se determina lo peligroso de la operación e ingreso a páginas web onion, especialmente de las categorizadas con virus debido a que estas pueden infectar a los usuarios Tor y robar su información. Otras páginas se constató la posible estafa que los usuarios sin conocimiento técnicos pueden sufrir.

Se determina que es muy fácil acceder a la red Tor, utilizando las herramientas como TorBrowser que proporciona el proyecto Tor. Así una persona con conocimientos básicos de computación puede acceder a la red, sin tomar en consideración los vulnerable que puede estar.

Pero no todo en la red de anonimización Tor es negativo; existen categorías y paginas onion que permiten a las personas expresarse o proporcionar servicios como en la red normal como son los foros o wikis de interés. Esta clase de personas simplemente desean permanecer en anonimato sin vincular ningún dato personal.

Debido a que la red Tor es abierta y de libre acceso, además de ofrecer ventajas como anonimato, cualquier tipo de servicio se puede alojar en sus páginas y servidores; así no solamente se puede navegar a paginas onion sino también existen servicios de chat, correo, telnet, entre otros que permiten a los usuarios disponer de la misma amplitud de servicios que la web normal, pero en este caso de forma anónima. Debido a esta apertura de servicios la red Tor o Deepweb es tan amplia, esta es usada para cometer cibercrímenes como ataques de red a instituciones o disponer de cibernegocios como la venta de droga o pornografía infantil lo cual hace difícil su desanonimización.

Como no se puede indexar información de la red de anonimización Tor, no se tiene un valor exacto de cuanta información o servicios se encuentran dentro de la red; estimaciones de terceros indican una estadística de que más del 90% de la red pertenece a la Deepweb; lo que hace pensar en la infinidad de recursos y servicios que existen y no observamos.

Dentro del análisis de los cibermercados se observó el uso constante de la nueva criptomoneda Bitcoin. Como se explicó en el apartado anterior, este tipo de moneda permite realizar transacciones de forma anónima, lo cual, evita o dificulta el rastreo de las negociaciones dentro del Deepweb. Otro motivo por el cual se está usando este tipo de moneda, es la creciente acogida en el mercado y su alza de valor en el mercado de valores, lo cual hace atractivo para cualquier persona. Entre los usos que se le ha dado el Bitcoin en el mercado negro, es la compra y venta de armas, drogas, comercio de pornografía infantil, lavado de activos, entre otros.

5.1. Trabajos Futuros

Se ha tocado temas técnicos e informativos con respecto a aplicaciones referentes a Tor y su red; adicional se ha realizado un estudio e investigación sobre los cibermercados negros con los que se hace referencia a la ilegalidad de muchos negocios y métodos de pago como bitcoin.

Como trabajos futuros y con el objetivo de ahondar de manera técnica e informativa se propone los siguientes puntos:

- Estudio de otras herramientas y redes de anonimización como I2P, Freenet, Mixminion, Mixmaster, Free Haven, entre otras.
- Estudio más amplio de la herramienta Shadow. Como se observó en la práctica de la presente TFM, el objetivo del estudio de Shadow se basó en el entendimiento de la red Tor; aun así, Shadow es un simulador que permite crear ambientes de conexiones controladas para entender cómo funciona ciertas aplicaciones en ciertos ambientes. Como ejemplo se podría simular un ambiente de red de una criptomoneda como por ejemplo el bitcoin o simplemente envío de otros protocolos de red (sip, http, https ssh, etc.) o aplicaciones para observar su comportamiento cuando existe perdida de paquetes, jitter, etc.

- Legalidad y procesos contra los cibermercados. Se propone realizar el estudio de los marcos legales respecto a cierta categoría de delito o en forma general y cuáles son los entes que realizan e investigan estos delitos; hay que indicar que estos delitos en la mayoría de los casos son de índole internacional. Adicional cual debería ser el proceso de algún perito debería realizar como toma evidencias durante y después del delito realizado en la red Tor
- Estudio de las criptomonedas en general: tipos de criptomonedas, funcionamiento y usos en el mercado, trading y marco legal a nivel mundial.

6. Glosario

ARPAnet
fue una red de computadoras creada por encargo del Departamento de Defensa de los Estados Unidos (DOD) para utilizarla como medio de comunicación entre las diferentes instituciones académicas y estatales 5

Bitcoin
Bitcoin es un protocolo y red P2P que se utiliza como criptomoneda, sistema de pago y mercancía.....ii, 18, 45, 47, 48, 57

cifrado
es un procedimiento que utiliza un algoritmo de cifrado con cierta clave (clave de cifrado) para transformar un mensaje, sin atender a su estructura lingüística o significado, de tal forma que sea incomprensible o, al menos, difícil de comprender a toda persona que no tenga la clave secreta (clave de descifrado) del algoritmo 14, 16

DNS
Domain Name System o Sistema de Nombres de Dominio y además de apuntar los dominios al servidor correspondiente, nos servirá para traducir la dirección real, que es una relación numérica denominada IP, en el nombre del dominio 14

DPI
Deep Packet Inspection (DPI) o Inspección a fondo de los paquetes, es el acto de inspección realizado por cualquier equipo de red de paquetes que no sea punto final de comunicación, utilizando con algún propósito el contenido 40

Firewalls
Un cortafuegos (firewall) es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas. 40

Hacker
es alguien que descubre las debilidades de un computador o de una red informática, aunque el término puede aplicarse también a alguien con un conocimiento avanzado de computadoras y de redes informáticas 45

Hash
Los hash o funciones de resumen son algoritmos que consiguen crear a partir de una entrada (ya sea un texto, una contraseña o un archivo, por ejemplo) una salida alfanumérica de longitud normalmente fija que representa un resumen de toda la información que se le ha dado..... 48

IP
IP es un número que identifica, de manera lógica y jerárquica, a una Interfaz en red (elemento de comunicación/conexión) de un dispositivo14, 22, 40, 43

javascript
es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,3 basado en prototipos, imperativo, débilmente tipado y dinámico. 47

keylogger

7. Bibliografía

[1] “¿Qué es y cómo funciona el Bitcoin y la tecnología Blockchain? | España”. [En línea]. Disponible en: https://www.groupbtc.com/es/articulo/que-es-y-como-funciona-el-bitcoin-y-la-tecnologia-blockchain. [Consultado: 31-dic-2018].

[2] “Bitcoin - Dinero P2P de código abierto”. [En línea]. Disponible en: https://bitcoin.org/es/. [Consultado: 31-dic-2018].

[3] “Home | Bitcoin Block Explorer”. [En línea]. Disponible en: https://blockexplorer.com/. [Consultado: 31-dic-2018].

[4] “Shadow - The Shadow Simulator”. [En línea]. Disponible en: https://shadow.github.io/. [Consultado: 31-dic-2018].

[5] “tor-spec.txt - torspec - Tor’s protocol specifications”. [En línea]. Disponible en: https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt. [Consultado: 31-dic-2018].

[6] UNODC, “Informe Mundial Sobre las Drogas 2017”.

[7] Blockchain, “Blockchain”, 2000. [En línea]. Disponible en: https://www.blockchain.com.

[8] V. Ciancaglini, M. Balduzzi, R. Mcardle, y M. Rösler, “Exploring the Deep Web Contents”, 2013.

[9] R. Dingledine, “Tor: The Second-Generation Onion Router”.

[10] R. Dingledine, “Tor-Design”, p. 17, 2004.

[11] R. Dingledine y Roger Dingledine, pre-alpha: run an onion proxy now! 2002.

[12] R. Jansen y N. Hopper, “Shadow: Running Tor in a Box for Accurate and Efficient Experimentation”, Netw. Distrib. Syst. Secur. Symp., pp. 1–22, 2012.

[13] T. Mattfeldt, M. Drautz, y G. Mall, “Experimentelle Herzhypertrophie Durch Thyroxingabe”, Pathologe, vol. 10, núm. 4, pp. 206–211, 1989.

[14] G. Owen y N. Savage, “The Tor Dark Net”. 30-sep-2015.

[15] T. T. Proyect, “The Tor Proyect”, 2011. [En línea]. Disponible en: https://www.torproject.org/. [Consultado: 01-oct-2018].