Sistema de Detección de Ataques DDoS en Tor

TRABAJO DE FIN DE GRADO

Ignacio Gago Padreny

Directores:

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

Doble Grado en Ingeniería Informática y en Matemáticas

Facultad de Informática

Universidad Complutense de Madrid

Madrid, Junio de 2015

Agradecimientos

Quisiera agradecer a Luis Javier García Villalba y Ana Lucila Sandoval Orozco, Directores de este Trabajo Fin de Grado, el apoyo brindado.

Asimismo, quisiera agradecer la permanente ayuda de Jorge Maestre Vidal. Sin su inestimable ayuda el presente Trabajo Fin de Grado no hubiera sido posible.

Finalmente, mi más sincero agradecimiento al resto de miembros del Grupo GASS (Grupo de Análisis, Seguridad y Sistemas), Grupo de Investigación 910623 de la Universidad Complutense de Madrid adscrito al Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid, por las facilidades ofrecidas.

Abstract

To protect our privacy, Tor, a popular anonymity system, forwards traffic through multiple relays. This network has been the subject of numerous attacks trying to disclose user identities, being denial of service attacks one of the most widespread. Not only this attacks have been very popular on Tor, but also on the Internet. In the present work, an anomaly-based detection system is proposed for detecting such attacks. Traffic is analized without concerning user's privacy and from it some metrics are extracted which enable to model traffic as time series in order to find out anomalies. A particular challange has been testing the tool due to the dificulties of obtaining traffic from denial of service attacks on the Internet as well as on Tor, where no available datasets from the latter implicated the need for generating them.

Keywords

Anomalies, DDoS, Entropy, Predictive Models, Time Series, Tor.

Resumen

Para proteger la privacidad, Tor, un sistema anónimo muy popular, dirige el tráfico a través de mútiples *relays*. Esta red ha sido sujeto de numerosos ataques con la intención de desenmascarar las identidades de los usuarios, siendo los ataques de denegación de servicio unos de los más utilizados. No solo este tipo de ataques se han convertido en una amenaza en Tor, además han tenido gran importancia en Internet. En este trabajo, se propone un sistema basado en anomalías para detectar estos ataques. El tráfico se analiza sin comprometer la privacidad de los usuarios y de él se extraen datos que a través de ciertas métricas permiten modelizarlo a partir de series temporales para detectar anomalías. Realizar pruebas de la herramienta desarrollada ha supuesto un gran reto debido a las dificultades que supone obtener tráfico de ataques de denegación de servicio tanto en Internet como en Tor, donde la inexistencia de *datasets* de dominio público en esta última ha supuesto la necesidad de generarlos.

Palabras clave

Anomalías, DDoS, Entropía, Modelos Predictivos, Series Temporales.

El abajo firmante autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: "Sistema de Detección de Ataques DDoS en Tor", realizado durante el curso académico 2014-2015 bajo la dirección de Luis Javier García Villalba y Ana Lucila Sandoval Orozco en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Ignacio Gago Padreny

Índice General

Índice de Figuras

Capítulo 1

Introducción

En la última década se ha disparado el uso de las tecnologías de la información. Según el último informe de la Oficina Europea de Estadística (Eurostat)[1], en el año 2006 aproximadamente el 31 % de la población accedió diariamente a Internet. Esto contrasta con los datos publicados en el año 2014, donde lo hizo el 65 %. El principal motivo de este incremento es la rápida evolución de las formas de acceso, y la importancia que ha ganado la sociedad de la información. En consecuencia, cada día millones de empresas, organizaciones y sociedades procesan enormes cantidades de información, procedentes de muy diversas fuentes.

Una parte importante de esa información contiene datos sensibles, cuya visualización, modificación o divulgación atenta contra el derecho a la privacidad de los individuos dictado por la Declaración Universal de Derechos Humanos. Pero a pesar de los esfuerzos de concienciación, y las sanciones impuestas por las distintas agencias[2], siguen siendo un activo de especial interés para diversos colectivos, quienes frecuentemente hacen un uso indebido de ella.

Con el fin de salvaguardar la información privada, se han desarrollado diferentes herramientas, que han sido definidas por la Comisión Europea como Tecnologías que Mejoran la Privacidad o PETs[3]. De entre ellas cabe destacar el importante incremento del uso de la red anónima Tor, llegando a convertirse en la más utilizada en la actualidad.

Tor (abreviatura de *The Onion Router*) es un proyecto comunitario que sostiene una red de comunicaciones de baja latencia superpuesta sobre Internet, capaz de preservar el anonimato de los participantes, y la integridad de la información transmitida. En sus más de 10 años de existencia nunca ha sido comprometida, lo que la

convierte en una de las PETs preferidas a la hora de evadir la censura, y divulgar información de manera anónima.

Con el crecimiento de su popularidad, Tor ha sido víctima de una mayor cantidad de ciberataques. El objetivo de estas amenazas ha sido romper su privacidad y limitar su disponibilidad. Para lo primero se requiere una importante infraestructura y conocimientos avanzados. Sin embargo, existe una gran cantidad de herramientas de dominio público capaces de denegar su servicio, lo que hace las convierte en importantes amenazas.

En este trabajo se afronta el problema de la seguridad en Tor, haciendo especial hincapié en su disponibilidad, y en la lucha contra los ataques de denegación de servicio. Para ayudar a comprender mejor el esfuerzo realizado, cabe destacar que la comunidad investigadora apenas ha participado en esta área. Con el fin incentivar futuros esfuerzos, se ha recopilado una gran cantidad de información relacionada con su infraestructura, y los principales problemas que ha afrontado. Parte de ella pronto estará disponible en la web del proyecto Tor[4]. Asimismo se introduce una estrategia de detección de intentos de inutilización, capaz de operar con éxito a pesar de las limitaciones que ofrece su entorno de computación. A continuación se detalla una serie de conceptos previos, los objetivos fijados, y la estructura del resto del documento.

1.1. Conceptos previos

1.1.1. Internet

Internet es una red de redes que permite la interconexión descentralizada de computadoras a través de un conjunto de protocolos denominado TCP/IP. Tuvo sus orígenes en 1969, cuando una agencia del Departamento de Defensa de Estados Unidos comenzó a buscar alternativas ante una eventual guerra atómica que pudiera incomunicar a las personas. Internet se construyó orientada a alcanzar una gran amplitud (cantidad de datos que pueden ser transmitidos en un determinado tiempo) y escalabilidad (propiedad de aumentar la capacidad de trabajo o de tamaño de un sistema sin comprometer su funcionamiento y calidad).

Sin embargo, potenciar estas cualidades ha derivado en importantes problemas de seguridad[5]. Por ejemplo, el protocolo IP fue diseñado para permitir que los hosts

se conectasen fácilmente a una red, pasando por alto la verificación de los campos de la cabecera de los datagramas, donde se muestra información sensible, como las direcciones IP que identifican los extremos de la comunicación.

1.1.2. Seguridad de la información

La seguridad informática es la propiedad que establece que los recursos de los sistemas de información sean utilizados de la manera que previamente se haya decidido, y que tanto su accesibilidad, como su integridad, solo sean manipulables por entidades acreditadas, dentro de los límites de su autorización. En consecuencia, se denomina intrusión a cualquier acción que tenga como finalidad vulnerar la seguridad de un sistema. Si bien es cierto que todos los componentes de un sistema informático están expuestos a rupturas en su seguridad, son los datos y la información los objetos más tenidos en cuenta a la hora de desempeñar acciones defensivas.

A lo largo de los años se han postulado diferentes modelos de seguridad, siendo la popular tríada CIA (Confidencialidad, Integridad, Disponibilidad) la base de las nuevas aproximaciones, las cuales a menudo incorporan nuevas propiedades. A continuación se describe brevemente cada una de ellas:

- **Confidencialidad.** La confidencialidad se define como la cualidad que de un activo, de solo ser accedido por la entidad que posea autorización para ello.

- **Integridad.** La integridad es definida como la propiedad que posee un activo, de no ser modificado por entidades sin autorización.

- **Disponibilidad.** La disponibilidad es definida como la capacidad de un activo, de ser accesible y utilizable por los usuarios o procesos autorizados cuando lo requieran. También se refiere a la capacidad de que la información pueda ser recuperada en el momento que sea necesario.

La gran popularidad de este modelo ha llevado a que tradicionalmente, la seguridad de la información fuera definida como la suma de estas tres propiedades. Sin embargo, diferentes aproximaciones han añadido nuevas características, ganando concordancia con la evolución de las tecnologías de la información. A continuación son descritas dos de las más importantes:

- **Autenticación.** La autenticación se define como la capacidad de un activo de verificar su autoría y a quien pertenece.

- **No repudio.** El no repudio es definido como la propiedad de los procesos de comunicación, de verificar la identidad de sus dos extremos. A pesar de su similitud con la autenticación, tienen una finalidad diferente: si bien la autenticidad demuestra quien es el autor y cuál es el destinatario de un proceso de comunicación, el no repudio prueba que el autor fue quien envió la comunicación (en origen) y que el destinatario fue quien la recibió (en destino).

Para garantizar la seguridad de la información, se han propuesto diferentes herramientas. Tomando como eje de clasificación el tipo de acción realizada, estas pueden clasificarse como: preventivas, de detección o correctivas. Las primeras actúan antes de que la intrusión, y tienen como finalidad dificultar su desarrollo y reducir su impacto. La detección tiene como finalidad reconocer una amenaza que se está produciendo, o que está a punto de suceder. Finalmente, la corrección tiene por objetivo la mitigación del daño causado.

1.1.3. Tecnologías que mejoran la privacidad

Las tecnologías que mejoran la privacidad (del inglés, *Privacy-Enhancing Technologies* o PETs), son un conjunto de herramientas desarrolladas con la finalidad de garantizar la salvaguarda de la privacidad de los usuarios y entidades que participan en la sociedad de la información. En la actualidad no existe una definición aceptada de las PETs, y tampoco una clasificación. Sin embargo, cuando se hace referencia a PETs, se sobreentiende, entre otras cualidades, las siguientes funcionalidades: reducción del riesgo de comprometer la privacidad de los usuarios y su cumplimento legal, minimización de la información confidencial que preservan las diferentes organizaciones y la garantía de que los usuarios sean quienes controlen su información privada. Algunos ejemplos de PETs son las redes de comunicaciones anónimas (entre las que se encuentra Tor) o las aplicaciones provistas por organismos para que los usuarios gestionen su información en propiedad de terceras partes.

1.2. Objetivos del trabajo

El objetivo principal del trabajo realizado es el desarrollo de un mecanismo de defensa contra ataques de denegación de servicio (DoS y DDoS) dirigidos contra la red anónima Tor. Para ello deben satisfacerse los siguientes objetivos secundarios:

- El estudio en profundidad de la infraestructura de Tor y de los problemas de seguridad que conlleva.

- La investigación de las técnicas de denegación de servicio y sus contramedidas.
- El desarrollo de estrategias para la extracción, y la interpretación de las características del tráfico que se dirige hacia ellas.
- La construcción de métricas que permitan modelar el tráfico que fluye a través de Tor.
- La elaboración de modelos predictivos capaces de desenmascarar situaciones anómalas, en base a dicha información.
- La decisión de qué anomalías se corresponden con amenazas reales.
- La elaboración de una metodología de evaluación acorde a las características del sistema desarrollado.
- La verificación de la eficiencia de la propuesta.

La línea de estudio e investigación inicial requerida para este trabajo (ataques de denegación de servicio, el funcionamiento de la red Tor y el modelado de datos basado en métricas) ha sido realizado junto a José María Aguirre Martín, debido a la complejidad de dicho trabajo. A partir de este punto, en el presente trabajo se optó por elaborar un modelo predictivo preciso con una baja tasa de falsos positivos debido a la gran cantidad de tráfico a analizar, mientras que José María Aguirre Martín dio prioridad a la detección inmediata de los ataques de denegación de servicio, desarrollando un modelo predictivo rápido y con no demasiado coste computacional.

1.3. Estructura del documento

Además de la presente introducción, este documento se estructura de la siguiente manera:

- En el capítulo 2 se discuten los aspectos más relevantes de los ataques de denegación de servicio.
- En el capítulo 3 se describe la infraestructura Tor, haciendo especial hincapié en sus problemas de seguridad.
- En el capítulo 4 se explican las métricas basadas en la entropía y los modelos predictivos que fueron consideradas a lo largo del desarrollo de la propuesta.

- En el capítulo 5 se introduce el sistema de detección de ataques de denegación de servicio en la red Tor.
- En el capítulo 6 se detallan las características de la experimentación realizada y su metodología de evaluación.
- En el capítulo 7 se discuten los resultados obtenidos.
- Por último, en el capítulo 8 se presentan las conclusiones y propuestas de trabajo futuro.

Capítulo 2

Denegación de servicio

Los ataques de denegación de servicio se han convertido es una constante amenaza para la sociedad de la información. Según ha publicado recientemente la Agencia Europea de Seguridad de las Redes y de la Información (ENISA), entre los años 2013 y 2014 se observó su incremento en un 70 %[6] . Además han advertido de la actual tendencia a la ejecución de este tipo de intrusiones, para alcanzar diferentes objetivos, de aquellos para lo que fueron desarrollados. Entre ellos destaca el encubrimiento de otro tipo de acciones delictivas, tales como transferencias de dinero fraudulentas, o desanonimato[7].

En este capítulo se describen las principales características de esta amenaza, y se discuten los principales motivos que han impulsado su evolución y crecimiento. A continuación se presenta su clasificación, y los esfuerzos realizados por la comunidad investigadora para su mitigación. Finalmente se introducen las características de las metodologías de evaluación de los sistemas defensivos.

2.1. Ataques de denegación de servicio

Los ataque de denegación de servicio (del inglés *Denial of Service attacks*) o DoS, tienen como objetivo comprometer la disponibilidad de un activo o servicio mediante el agotamiento de sus recursos de cómputo. Cuando son originados desde distintas fuentes reciben el nombre de ataques de denegación de servicio distribuidos (del inglés *Distributed Denial of Service attacks*) o DDoS. Debido a su mayor capacidad de causar daño, estos últimos son los más frecuentes en la actualidad, y a menudo requieren del uso de redes de ordenadores zombis o botnets. Habitualmente, tanto los ataques DoS como los DDoS alcanzan sus objetivos mediante el envío de grandes cantidades de información, la cuales trata de ocupar la mayor parte del ancho de

banda de la red en la que se encuentra la víctima. Esto limita considerablemente el acceso a sus recursos.

Su modo de actuación generalmente comprende dos tipos de acciones. En primer lugar, el atacante puede inyectar paquetes de datos capaces de comprometer alguna vulnerabilidad de la víctima. Este es el caso de la intrusión popularmente conocida como "ping de la muerte". El "ping de la muerte" consiste en el envío de datagramas ICMP muy grandes, pero fragmentados en otros más pequeños, capaces de colapsar la capacidad de procesamiento de la víctima. Por otro lado, los ataques DDoS pueden tratar de inundar a la víctima mediante el envío de una gran cantidad de datos. Esta última acción requiere conocimientos menos avanzados para su ejecución, y su éxito a menudo depende de la cantidad de nodos infectados desde la que se ha originado. En [6] se describen en detalle ambos casos y se muestran otros muchos ejemplos. A continuación se discuten las principales motivaciones, aplicaciones de estas amenazas y su clasificación.

2.1.1. Crecimiento y motivaciones

El crecimiento de los ataques de denegación de servicio es atribuido a diferentes motivos. El primero de ellos es su relación con las *botnets*; estas son cada vez más grandes y difíciles de detectar, lo que incrementa el número de posibles focos de intrusión. Otra causa es el aumento de la cantidad de vulnerabilidades que permiten explotar elementos intermedios de red como reflectantes, y en mucho caso amplificadores, de los vectores de ataque. Los protocolos con más tendencia a ser comprometidos son DNS, NTP y SNMP.

Por otro lado, según la Oficina Europea de Policía (Europol)[8], la DDoS cada vez se relacionan más con el crimen organizado; en consecuencia, cada vez es más fácil su contratación para encubrir campañas de propagación de *malware* o *spam* desde el mercado negro. Esto ha llevado a la aparición de nuevas y sofisticadas estrategias para dificultar su detección, y al desarrollo de herramientas sencillas, que permiten su configuración y ejecución a pesar de no tener elevados conocimientos tecnológicos.

Tal y como se anuncia en [6], los autores de estas intrusiones son incentivados por diferentes causas. A continuación se enuncian las más repetidas:

- **Economía:** muchos individuos o empresas contratan este tipo de ataques con el objetivo de incrementar su poder adquisitivo, o reducir el de la competencia.

- **Venganza:** los ataques DDoS son frecuentes entre ex-empleados frustrados que tienen el objetivo de colapsar la empresa en la que trabajan.
- **Creencias:** existen grupos de individuos que llevan a cabo ataques basándose en sus creencias religiosas, sociales o políticas.
- **Experimentación:** gran cantidad de individuos interesados en aprender sobre este tipo de ataques, experimentan con ellos y los ejecutan para demostrar o mejorar sus habilidades.
- **Ciberguerra:** el cibercrimen es cada vez más frecuente. Muchas organizaciones aprovechan los ataques DDoS para bloquear departamentos ejecutivos, agencias civiles, organizaciones financieras, o infraestructuras de sus rivales.

2.1.2. Clasificación

A continuación se muestra una clasificación de los ataques de denegación de servicio, que tiene por eje, la característica que ensalza su capacidad de causar daño. Nótese que durante su elaboración únicamente han sido consideradas las acciones que por similitud, o por impacto, pueden llegar a tener algún tipo de relación con la red Tor. El resto quedan fuera del alcance del trabajo realizado.

En base a este criterio se han establecido tres conjuntos de ataques: aquellos que tienen su potencial en su capacidad de inundación, reflexión o amplificación. La taxonomía realizada no es disjunta. De este modo, el éxito de un ataque puede depender tanto de su capacidad de inundación, como de amplificación, siendo miembro de ambos grupos. A continuación se describe cada uno de ellos:

Denegación de servicio basada en inundación

La denegación de servicio basada en inundación trata de alcanzar sus objetivos por medio de la inyección de grandes volúmenes de tráfico. Dada su sencillez de ejecución, y la magnitud de su impacto, ha sido uno de los mayores temas de interés en la bibliografía. En la actualidad existen diferentes estrategias para conseguir una inundación eficaz, las cuales han sido diferenciadas en [9] como inundaciones de tasa alta y baja. Las primeras consisten en la emisión de grandes cantidades de tráfico de manera constante y uniforme. Se caracterizan por ser especialmente ruidosas, y por alcanzar buenos resultados rápidamente. Por otro lado, la inundación de tasa baja explota vulnerabilidades de los protocolos de red. Esto permite que el tráfico inyectado adopte patrones periódicos, que incrementan o decrementan su volumen

con el paso del tiempo. Es mucho menos ruidosa, pero su ejecución es más compleja.

Cuando los ataques de inundación actúan en la capa de red, aprovechan funcionalidades propias de sus protocolos, siendo TCP, UDP, ICMP y DNS los más explotados. En [10] son descritas algunas de sus variantes, siendo la más popular de ellas la denominada inundación SYN. Esta explota el protocolo TCP, y su negociación del inicio de sesión por medio del saludo a tres vías o *Handshake*. Para ello el atacante envía paquetes SYN con direcciones IP inexistentes o en desuso y cuando el servidor ubica la petición en la memoria, esperará a la confirmación del cliente. Mientras espera, dicha petición seguirá almacenada en la pila de la memoria. Como estas direcciones IP no son válidas, el servidor nunca recibirá la confirmación. De este modo, el ataque explota el hecho de que cada una de las conexiones "medio abiertas" ocupa un espacio de pila en la memoria, y que se mantendrán en ella hasta que expire tras vencer un cierto intervalo de tiempo. Con la pila llena, el servidor no puede tramitar nuevas peticiones, denegando el acceso a nuevos usuarios.

Por otro lado, la capa de aplicación ofrece nuevas posibilidades a los atacantes. En [11] se trata este tema en mayor profundidad, y se distinguen tres conjuntos de amenazas: las que se basan en inicios de sesión, envíos de petición y en respuestas lenta del servidor. De manera similar a la inundación SYN, el primer grupo trata de colapsar las colas que permiten el acceso de usuarios a los servicios web. Por otro lado, la inundación por peticiones consiste en el envío masivo de solicitudes (normalmente GET/POST) que el servidor deberá atender. Finalmente, los ataques de respuesta lenta se basan en intentar mantener las conexiones HTTP el mayor tiempo posible. Para esto, las peticiones son realizadas mediante el envío de datos lentamente, o bien procesando las respuestas con lentitud.

Denegación de servicio basada en reflexión

La inundación basada en reflexión surge de la necesidad de los atacantes, de ocultar el origen de la intrusión. A los ataques que integran este grupo se los denomina ataques de denegación de servicio distribuida y reflejada (del inglés *Distributed Reflection Denial of Service*) o DRDoS, y tienen en común que tratan de aprovechan vulnerabilidades en terceras partes para forzarlas a emitir el tráfico malicioso.

Un ejemplo de ataque de reflexión se encuentra en los conocidos ataques *smurf*. Los ataques *smurf* son una variante de la inundación SYN que aprovecha elementos intermedios de red para enmascarar su origen. En su ejecución, las direcciones de

origen de los paquetes son reemplazadas por la de la víctima. De esta manera, todas las máquinas intermedias responderán a ella tras recibir su solicitud.

Otro ejemplo es la amplificación mediante la explotación del protocolo de voz sobre IP o VoIP, y que funciona de la siguiente manera [12]: el protocolo VoIP opera bajo el protocolo SIP. Los servidores SIP necesitan acceso a Internet para aceptar las llamadas, y estas son tramitadas.

En ella Alice quiere hablar con Bob. Para ello envía un paquete al proxy SIP de Alice, que es quien se encarga de solicitar la dirección del proxy SIP de Bob. Por lo tanto el proxy SIP de Alice envía una invitación al proxy SIP de Bob. Cuando el proxy de Bob la recibe, la traslada a la dirección de registro de Bob. Cuando Bob acepta la llamada empieza la conversación. El ataque de denegación consiste en el envío de gran cantidad de invitaciones SIP con direcciones IP falsas, las cuales consumen una gran cantidad de recursos del servidor. Esto es debido a que entre sus tareas está la de distinguir las direcciones IP verdaderas de las falsas. Cuando el atacante inyecta tráfico, es posible que agote su capacidad de cómputo. Asimismo incrementa la carga de trabajo de los mecanismos encargados de gestionar las llamadas que recibe la víctima.

Denegación de servicio basada en amplificación

La inundación basada en amplificación consiste en realizar peticiones a terceras partes, con el objetivo de que las respuestas sean de mayor tamaño que el de las propias peticiones. Dichas peticiones llevan falsificada su dirección de retorno, de manera que las respuestas, en lugar de llegar al atacante llegan a la víctima. Se trata de una variante de los ataques basados en reflexión, pero con diferente motivación y consecuencias.

Uno de los elementos de red más aprovechados para lograr la amplificación son los servidores DNS. A su explotación con este fin se la denomina amplificación DNS. En [13] se discute este problema en detalle, y se señala como principal causante al hecho de que las consultas realizadas al servidor se realizan con datagramas que a menudo contienen menos información que las respuestas. En ocasiones son los propios atacantes quienes han insertado campos especialmente grandes en la información que almacena el servidor sobre dominios, que previamente, han sido comprometidos.

2.2. Estrategias defensivas

De manera tradicional, las distintas técnicas de defensa frente a intentos de denegación de servicio son clasificadas tomando como eje el momento del proceso de intrusión en que actúan. Por lo tanto se agrupan en cuatro tipos de medidas: prevención, detección, identificación del origen y mitigación. Todas ellas han sido muy tratadas en la bibliografía, y son resumidas a continuación.

2.2.1. Prevención

Los métodos de prevención son mecanismos de defensa que actúan antes de que el ataque suceda, e independientemente de su detección. Su objetivo es minimizar el daño recibido, y agrupan diferentes tecnologías, tales como modelos de análisis de riesgos, listas de accesos o protocolos de seguridad[11]. Si bien constituyen una primera línea de defensa, a menudo resultan insuficientes a la hora de tratar situaciones complejas.

2.2.2. Detección

La detección de los ataques es imprescindible para que actúen el resto de los procesos defensivos. Su eficacia se basa en la proporción de ataques reales que son detectados. Sin embargo, el hecho de que presenten un gran porcentaje de acierto no implica que sean de buena calidad. También deben ser capaces de afrontar otros problemas, como la emisión de tasas altas de falsos positivos (tráfico legítimo erróneamente etiquetado como malicioso), capacidad de procesamiento en tiempo real y distinción de ciertos fenómenos de red, tales como los conocidos *flash crowds*. Estos últimos son acumulaciones inesperadas de accesos a servidores o sistemas de forma legítima, y por usuarios autorizados, que habitualmente acarrean errores de detección. En [14] se profundiza más en ello.

En la identificación de DDoS son considerados los dos paradigmas de los sistemas de detección de intrusiones: reconocimiento de firmas y anomalías. El reconocimiento de firmas se basa en la identificación de patrones de ataques previamente conocidos. Esto reduce la emisión de falsos positivos, pero dificulta la detección de nuevas amenazas. Por este motivo, la mayor parte de la comunidad investigadora ha optado por el desarrollo de sistemas basados en el estudio de anomalías. Estos implican el modelado del comportamiento habitual y legítimo del entorno monitorizado, con el fin de identificar eventos que difieran considerablemente de las acciones legítimas.

En los últimos años ha aparecido una gran cantidad de publicaciones centradas en la detección de ataques de denegación de servicio basados en el reconocimiento de anomalías. Para ello han sido propuestas diferentes técnicas, tales como modelos probabilistas basados en Markov[15] , algoritmos genéticos[16], teoría del caos[17], análisis estadístico CUSUM con transformadas de ondícula[18], técnicas forenses basadas en visualización[19], lógica difusa[20] o estudio de las variaciones en la entropía [21][22].

2.2.3. Identificación del origen

En la etapa de identificación del origen, la víctima trata de desenmascarar la ruta del vector de ataque con el fin de señalar a su autor. Este proceso a menudo es muy complicado, ya que el atacante dispone de diferentes métodos para ocultar su rastro. Estos varían desde sencillos procesos de suplantación de identidad, hasta el atravesar tramos de redes anónimas (como redes *Fast-Flux* o la explotación del servicio Tor2web). En consecuencia, llegar hasta el extremo final es una situación idílica que muy pocas veces se consigue. Sin embargo, aproximar su ubicación permite la realización de un despliegue defensivo mucho más eficaz. En [23] se discuten estos problemas en mayor detalle.

La mayor parte de los métodos de identificación del origen se basan en el marcado de la ruta que siguen los paquetes. Algunos de ellos lo hacen añadiendo información en el propio datagrama, y otros, almacenando información acerca de su paso a través de dispositivos de red intermedios, como encaminadores o servidores de control. En [24] se presenta una buena recopilación de estos métodos y estudian sus diferencias.

2.2.4. Mitigación

Una vez detectada una amenaza debe procederse a su mitigación. En el caso de la denegación de servicio, la mitigación consiste en el despliegue de una serie de medidas que reduzcan el daño causado, y a ser posible, que restauren los servicios del sistema comprometido a la normalidad. Estas habitualmente consisten en el incremento de la reserva de recursos disponibles para la supresión de cuellos de botella[25], el aumento de la restricción en sistemas de autenticación por medio de puzles y señuelos, o la actualización de las listas de acceso y políticas de cifrado.

2.3. Esquemas de evaluación

A pesar del interés de la comunidad investigadora en esta área, no existe un consenso acerca de qué criterios deben establecerse. Tal y como se pone de manifiesto en [26], la mayor parte de las colecciones públicas de muestras de ataques carecen de validez por diferentes motivos, entre los que destacan su antigüedad, y falta de rigor en los procesos de captura. No obstante, las dos colecciones más utilizadas son KDDcup'99 y Caida'07. A continuación son brevemente descritas.

- **KDD'99.** Según [26], KDD'99 es una de las pocas colecciones que ofrece un etiquetado fiable. Tuvo su origen en el marco del concurso KDDcup del año 1999, e incluye parte de un conjunto de trazas de ataques publicados por la agencia norteamericana DARPA en el año 1998. En ella están presentes cuatro amenazas: denegación de servicio, enumeración y dos esquemas de escalada de privilegios. Sus muestras son caracterizadas por 41 parámetros diferentes, quedando el resto de su contenido anonimizado. Sus autores han etiquetado su estado actual como "comprometido", debido a su antigüedad, y a una serie de irregularidades encontradas en los procesos de captura. Sin embargo, a pesar de estos problemas sigue utilizándose, siendo una forma aceptada de comparar los resultados de las nuevas propuestas, con esquemas de detección clásicos.

- **CAIDA'07.** La colección CAIDA'07[27] contiene trazas de ataques de inundación (principalmente ICMP, SYN y HTTP) en formato .pcap, capturadas en Agosto del año 2007. Es el conjunto de muestras de referencia más actualizado, y el único apoyado por [26]. Pero tal y como subrayan sus autores, carece de ejemplos de tráfico legítimo. Estos fueron eliminados al finalizar el proceso de captura. En consecuencia, en muchos trabajos se combinan con la colección de capturas de tráfico legítimo CAIDA'08 [28], extraídas de los centros de procesamiento de datos Equinix de San José y Chicago (Estados Unidos) a finales del año 2007.

Adicionalmente, algunos autores han optado por generar sus propias muestras de ataques a partir de herramientas. Entre las más destacadas están D-ITG, Harpoon, Curl-loader o DDOSIM. Pero a pesar de que su uso permite adaptar los escenarios de pruebas a las necesidades del diseño, restan realismo a la experimentación. Además, el diseño de escenarios personalizados dificulta la comparativa de los resultados con otras propuestas.

Capítulo 3

La red Tor

La red Tor (abreviatura del inglés *The Onion Router*) es una red de comunicaciones distribuida de baja latencia y superpuesta sobre Internet, en la que el encaminamiento de los mensajes intercambiados entre los usuarios no revela su identidad. En la red Tor los mensajes viajan desde su origen a su destino, a través de una serie de encaminadores especiales, denominados "encaminadores de cebolla" (del inglés *onion routers*). Su desarrollo y mantenimiento es posible gracias a un conjunto de organizaciones e individuos que donan su ancho de banda y capacidad de procesamiento, y a una importante comunidad que lo respalda[4][30]. A continuación se describen sus rasgos más relevantes, entre los que se encuentra su motivación, arquitectura, funcionamiento, datagramas y vulnerabilidades más recientes.

3.1. Motivación del proyecto Tor

El objetivo principal del proyecto Tor es conseguir que Internet pueda usarse de forma anónima, de manera que el encaminamiento proteja la identidad de los usuarios. Es decir, persigue que no se pueda rastrear la información que envía un usuario para llegar hasta él. Para ello la red Tor cifra la información a su entrada y la descifra a su salida. De este modo el propietario del encaminador de salida puede ver toda la información cuando es descifrada antes de llegar a Internet.

3.2. Componentes

La red Tor está formada por una serie de nodos que se comunican entre sí mediante el protocolo de seguridad en la capa de transporte (del inglés *Transport Layer Security*) o TLS[31], que es predecesor del protocolo de capa segura (del inglés *Secure Sockets Layer*) o SSL[32]; ambos operan sobre TCP. En Tor se distinguen dos

componentes: nodos OR y OP. Estos son descritos a continuación:

- **Nodos OR.** Los nodos OR o simplemente OR funcionan como encaminadores y en algunos casos, como servidores de directorio. Los nodos OR mantienen una conexión TLS con cada uno de los otros OR. Las conexiones OR-OR no son nunca cerradas deliberadamente salvo cuando pasa cierto tiempo de inactividad.

- **Nodos OP.** Los nodos OP o simplemente OP tienen la finalidad de obtener información del servicio de directorio, establecer circuitos aleatorios a través de la red y manejar conexiones de aplicaciones del usuario. Funciona como software local con el que el usuario se comunica. Las conexiones OR-OP no son permanentes. Un OP debería cerrar una conexión a un OR si no hay circuitos ejecutándose sobre la conexión y ha vencido cierto temporizador.

3.3. El servidor de directorios

Un servicio de directorio o SD es una aplicación o un conjunto de aplicaciones que almacena y organiza la información sobre los usuarios de una red. En el caso de Tor, el servicio de directorio publica una base de datos que asocia a cada OR con cierta información. Esta información es accesible por todos los OR y por todos los usuarios finales, los cuales la emplean para tener un conocimiento de la red. En el caso de que se disponga de pocos servidores de directorio, es posible que se produzcan puntos de fallo. A efectos prácticos los servidores de directorios se comportan como grupos establecidos de ORs confiables.

3.4. Funcionamiento

La información circula a través de Tor de la siguiente manera:

1. A partir de la información obtenida de su configuración y del servicio de directorio, el OP decide un circuito por el que van a circular los paquetes. Por defecto el circuito tiene 3 nodos OR.

2. El OP negocia las claves de cifrado necesarias con cada OR del circuito con el fin de proteger sus datos en todo el camino, y antes de iniciar el proceso de transmisión. La obtención de las claves simétricas (una para cada sentido de comunicación: *forward key* (Kf), *backward key* (Kb)), se realiza a partir del protocolo de establecimiento de claves Diffie-Hellman.

3. A continuación el OP cifra el paquete con la clave del último OR del circuito.

4. Al paquete se añadirá una capa de cifrado por cada punto de paso del circuito.

5. Tras finalizar el cifrado, el OP envía el paquete resultante al primer nodo del circuito.

6. El primer OR quita 'su capa de la cebolla' y envía el paquete al siguiente nodo.

7. Al pasar por cada nodo intermedio, va perdiendo sus capas de cifrado. Esto permite que ningún OR puede hacerse con la imagen completa del circuito, ya que sólo conoce los OR/OP anterior y posterior.

8. Al alcanzar el nodo de salida, el mensaje queda descifrado por completo. Hay que tener en cuenta que este no es el mensaje original, este solo contiene la información necesaria para realizar la petición inicial y no se conoce quién originó dicha petición.

A continuación se muestra un ejemplo del algoritmo de Diffie-Hellman con el que Tor intercambia claves [33]. Dadas dos partes (Alice y Bob) que intentan establecer una clave secreta y un adversario (Mallory), su versión más sencilla es la siguiente:

1. Se establecen un número primo p y un generador $g \in \mathbb{Z}_p^*$. Estos son públicos, conocidos no solo por Alice y Bob, sino también por el adversario Mallory.

2. Alice escoge $a \in \mathbb{Z}_{p-1}$ al azar, calcula $A = g^a \mod p$, y envía A a Bob.

3. Bob escoge $b \in \mathbb{Z}_{p-1}$ al azar, calcula $B = g^b \mod p$, y envía B a Alice.

4. Tanto Alice como Bob pueden calcular la clave K a utilizar, siendo $K = g^{ab}$ mód p, sin conocer el valor escogido al azar del contrario, usando propiedades del grupo $\mathbb{Z}_p^*$.

3.5. Células

Una vez que se establece la conexión TLS, las entidades se comunican mediante paquetes de información estructurada llamadas células (*cells*).

El formato de las células es el siguiente:

- **circID.** Es el identificador de circuito y especifica el circuito al que se refiere la célula. Cada circuito tiene un CircId distinto para cada OR y OP del circuito.

- **CMD.** Indica el comando que especifica el significado de la célula. Atendiendo al tipo de comando (valor de CMD). Se consideran dos tipos de células: células de control y células de transmisión.

Dada su relevancia, a continuación se explican en detalle las características de las células de control y las células de transmisión.

- **Células de control.** Las células de control (en inglés *control cells*) son siempre interpretadas por el nodo que las recibe y permiten controlar la comunicación. Permiten la ejecución de los siguientes comandos:
  - CREATE: creación de un circuito.
  - CREATED: indica que se ha creado un circuito.
  - DESTROY: destrucción de un circuito.
  - CREATE_FAST: creación de un circuito reaprovechando operaciones de clave pública existentes.
  - CREATED_FAST: indica que se ha creado un circuito que se creo mediante el comando CREATE_FAST.
- **Células de transmisión.** Las células de transmisión son usadas en la comunicación entre el OP y cualquiera de los OR del circuito; normalmente el nodo de salida (*exit node*). Este tipo de células presentan campos que forman parte de la carga útil (*payload*) de la célula. Son los siguientes:
  - Relay command: indica el funcionamiento de la celda. Contiene los siguientes tres tipos:
    - *forward:* son enviados desde el OP origen del circuito.
    - *backward:* son enviados desde los OR del circuito al OP origen.
    - *both:* pueden funcionar como *forward* o como *backward*.
  - StreamID: es el identificador de flujo. De esta forma se permite que varios flujos puedan ser multiplexados en un solo circuito. Las células que afectan al circuito entero en lugar de a un streamID particular tienen este campo a 0 y son consideradas de control.
  - Digest: permite el control de integridad extremo a extremo (del inglés *end-to-end integrity checking*). Este campo es utilizado para determinar exactamente a que nodo del circuito va dirigida una célula.
  - Len: indica el número de bytes del campo DATA que contiene carga útil real. El resto del campo DATA estará rellenado con ceros.

- CMD: identifica el subcomando de la célula de transmisión. Algunos de estos subcomandos serían:
  - *Relay begin:* para abrir un nuevo flujo o *stream*.
  - *Relay data:* para enviar datos a través del flujo.
  - *Relay end:* para cerrar un flujo.
  - *Relay connected:* para notificar al OP de que *relay begin* concluyó con éxito.

La principal diferencia entre las células de control y las de transmisión, es que las primeras pueden ser leídas por cualquiera, mientras que las segundas solo por un nodo concreto. Por ejemplo cuando se envía una célula *DESTROY*, el OP la envía al primer OR, este la recibe, cierra todos los flujos y la transmite al siguiente OR. Así hasta llegar al final.

En las células *relay* el OP asigna el *digest* y después cifra la célula con cada una de las claves de los nodos OR. Como el *digest* está cifrado con distintos valores que han ido encapsulándose paso a paso, solo el nodo objetivo podrá descifrar su contenido, y por tanto, ejecutar la función que indique. Por otro lado, cuando un nodo OR recibe una célula debe verificar la validez de su código *digest*. De no ser así, será enviada al siguiente nodo del circuito.

3.6. Amenazas contra la red Tor

El crecimiento de la popularidad de Tor ha acarreado la aparición de una nueva generación de amenazas, capaces de explotar con mayor precisión algunas de sus vulnerabilidades. Entre ellas destacan cuatro tipos de ataques: *raptor*, *sniper*, *relay* y la adaptación de los métodos de denegación de servicio convencionales.

3.6.1. Ataque Raptor

El objetivo de los ataques Raptor es desanonimizar Tor[34]. Para este fin, hace uso de aspectos dinámicos de los protocolos de Internet, por ejemplo del protocolo BGP. El ataque Raptor está compuesto de tres ataques individuales que se unen para conseguir un efecto mayor.

En la primera etapa, se aprovecha la asimetría del enrutamiento de Internet, es decir, se toma ventaja del hecho de que el camino BGP de un elemento que envía información y de otro que la recibe, puede ser diferente que el camino BGP del

elemento que recibe al elemento que envía. Esto permite al atacante observar al menos una dirección de comunicación y realizar un análisis de tráfico. Este primer ataque resulta efectivo en los siguientes casos:

- Se dispone información de tráfico de las conexiones con sentido cliente-nodo de entrada y con sentido nodo de salida-servidor.
- Se dispone información de tráfico que fluye con sentido cliente-nodo de entrada y con sentido servidor-nodo de salida.
- Se dispone de información de tráfico con sentido nodo de entrada-cliente y con sentido nodo de salida-servidor.
- Se dispone de información sobre el tráfico con sentido nodo de entrada-cliente y con sentido servidor-nodo de salida.

En el segundo ataque, Raptor explota el hecho de que los caminos BGP cambian con el tiempo debido a fallos que se producen en los encaminadores. Estos cambios permiten a los atacantes observar tráfico adicional, permitiendo de esta manera desanonimizar a más usuarios de la red.

La tercera y última parte se basa en hacer uso de lo que se conoce como *BGP hijacks*, los cuales consisten en descubrir usuarios que hacen uso de determinados nodos de la red que están comprometidos.

3.6.2. Ataque Sniper

El ataque *sniper* es un tipo de DDoS que tiene como objetivo deshabilitar nodos de Tor arbitrarios[7]. Su éxito depende de dos aspectos clave de su funcionamiento. El primero de ellos es que una vez creado un circuito, sus nodos extremo (cliente y nodo de salida) controlan el tráfico mediante el uso de un contador de paquetes. Este es inicializado a 1000 y va disminuyendo a medida que se introducen células al circuito. Análogamente, otro contador es inicializado a 1000, pero esta vez su valor decrecerá a medida que se eliminen células. Cuando este contador llega a cero se vuelve a inicializar todo, de manera que nunca habrá más de 1000 células en un circuito. La otra asunción es que cuando un nodo destino deja de leer paquetes, el siguiente nodo del circuito almacena dichos paquetes.

El ataque *sniper* requiere disponer de dos nodos extremos comprometidos (un nodo cliente y otro de de salida). Procede de la siguiente manera:

1. El cliente comprometido crea un circuito delimitado por los nodos comprometidos. Dado que ambos son controlados por el atacante, puede prescindir de las limitaciones referentes a su máximo número de paquetes. Esto permitirá al nodo de salida generar gran cantidad de paquetes en forma de células.

2. Entonces el nodo cliente recibe la orden de dejar de leer paquetes. Esto hace que el siguiente nodo conserve una gran cantidad de paquetes sin procesar, quedando inhabilitado hasta que el sistema operativo cierre el proceso. Esto dificultará el acceso de nuevos usuarios.

3.6.3. Ataque Replay

El ataque *replay* parte de la situación en que los nodos entrada y de salida están comprometidos. Su objetivo es enlazar la comunicación entre el cliente y el nodo de entrada comprometido, con la comunicación entre el nodo de salida comprometido y el servidor. De esta manera se puede conocer qué cliente está accediendo a qué servidor[35]. Para llevar a cabo su objetivo, el encaminador de entrada identifica una célula del *stream* y la duplica (de ahí el nombre del ataque). Una vez duplicada, dicha célula avanza por el circuito y llega al nodo de salida. Al recibir dicha célula, genera un error debido al duplicado.

El error se produce debido a que cuando la célula es duplicada en el nodo de entrada, su descifrado en el segundo y tercer OR falla. Esto se debe a que el cifrado se realiza por medio de una implementación del AES, la cual se basa en un contador, el cual resulta afectado al duplicarse la célula. El cifrado de la célula original aumenta en uno el contador AES. El resto de nodos descifran correctamente la célula y aumentan también el contador.

Cuando el nodo de entrada cifra la célula duplicada, hace que el descifrado realizado en los siguientes nodos produzca una desincronización entre el cliente y los nodos. De esta manera será posible asegurar que los nodos comprometidos están en el mismo circuito y se desenmascarará al usuario que accede a cada servicio.

3.6.4. Denegación de servicio

Este ataque tiene el objetivo de controlar el nodo de entrada y el nodo de salida de un circuito con el fin de conocer qué cliente accede a qué servidor. En este contexto, se denominan circuitos comprometidos a aquellos que al menos tienen un extremo comprometido, y circuitos controlados a aquellos en los que ambos están

comprometidos.

Los servidores directorios de Tor asignan a cada nodo una bandera (*'Guard'* o *'Exit'*). En la creación de circuitos, solo los nodos con estas banderas desempeñarán estas labores, siendo el resto, nodos intermedios. Los nodos de entrada son elegidos de una lista de 3 posibles candidatos. De este modo, cada vez que un cliente crea un circuito, crea una lista de 3 nodos de entrada, y para su circuito elige uno de ellos. Si hay menos de 3 nodos en dicha lista, se añaden nuevos nodos. Un nodo es eliminado de ella solo si no se ha podido conectar a él durante un determinado periodo de tiempo. De esta manera la lista contendrá los nodos más seguros, reduciendo la probabilidad de que el cliente elija nodos comprometidos. El ataque de denegación de servicio se comporta de la siguiente manera: si el atacante controla solamente uno de los nodos de un circuito, utiliza un ataque DDoS para destruirlo. A continuación se reconstruirá y tendrá mayor probabilidad de controlar al menos dos nodos del nuevo circuito. Sin embargo romper todos los circuitos que el atacante no controla no es buena idea, pues determinados nodos podrían quedar marcados como "sospechosos". En ocasiones el atacante destruirá también circuitos comprometidos, pero no controlados, con el objetivo de pasar desapercibido[36].

Capítulo 4

Entropía y modelos predictivos en series temporales

En este capítulo se describen los aspectos más representativos de dos herramientas cuyo entendimiento, resulta imprescindible en la comprensión del sistema propuesto. Estas son la entropía y el análisis predictivo de series temporales. La primera cumple un papel esencial a la hora de extraer y modelar las características del tráfico que fluye a través de Tor. Por otro lado, la elaboración de pronósticos sobre series temporales facilita el reconocimiento de comportamientos inesperados en base a las observaciones realizadas ya que detrás de la mayor parte de estas anomalías, se esconden intentos de ataques de denegación de servicio.

4.1. Entropía

La entropía es un concepto usado originalmente en termodinámica, mecánica estadística y luego en teoría de la información. Se concibe como una medida del desorden o una medida de la incertidumbre, cuya información tiene que ver con cualquier proceso que permite acotar, reducir o eliminar la incertidumbre. Un ejemplo ilustrativo para entender el uso de la entropía es el siguiente:

"Cuando un vecino nos dice en el ascensor que las calles están mojadas, y sabemos que acaba de llover, estamos recibiendo información poco relevante, porque es lo habitual. Sin embargo, si el mismo vecino nos dice que las calles están mojadas, y sabemos que no ha llovido, aporta mucha más información (porque es de esperar que no rieguen las calles todos los días)."

En el ejemplo se observa claramente que el hecho de que suceda algo relevante o

no, depende de las observaciones previas. Esta es la diferencia que trata de expresar la entropía. A continuación se describen los orígenes de este concepto, su aplicación en la teoría de la información y la entropía de Rènyi.

4.1.1. Origen

Rudolf Clausius planteó por primera vez el concepto de entropía en el año 1865. Para ello se basó en el estudio de procesos termodinámicos curvilíneos reversibles, postulando la ecuación:

$$dS = \frac{\delta Q}{T}$$

donde δQ es la cantidad de calor absorbida en un proceso termodinámico concreto, y T es la temperatura absoluta. Esto puede interpretarse como la cantidad de calor intercambiada entre el sistema y el medio dependiente de su temperatura absoluta, que se produce cuanto en un proceso termodinámico reversible e isotérmico, se produce una transición de estados.

El concepto de entropía termodinámica resultó de inspiración en ciertas áreas de la estadística, lo que dio pie a la mecánica estadística. Una de las teorías termodinámicas estadísticas (concretamente, la de Maxwell-Boltzmann 1890-1900), define la relación entre ambos conceptos de la siguiente manera:

$$S = k \log \Omega$$

donde S es la entropía, k la constante de Boltzmann y Ω el número de microestados posibles para el sistema. Es importante destacar que esta ecuación ofrece por primera vez una definición absoluta de la entropía en un sistema, situación que era impensable únicamente bajo el contexto de la termodinámica.

Poco a poco la entropía como magnitud física, fue ganando el respaldo de la comunidad investigadora. Este proceso dio pie a diferentes interpretaciones, que con frecuencia entraban en conflicto.

En la actualidad, y desde un punto de vista estadístico, la entropía asociada a la variable aleatoria X es un número que depende directamente de la distribución de probabilidad de X, e indica cómo es de predictible el resultado del proceso sujeto a incertidumbre o experimento. Esto también puede interpretarse de manera matemática, de tal manera que cuanta más plana sea la distribución de probabilidad, más difícil será acertar cuál de las posibilidades se dará en cada instancia.

Nótese que se considera distribución plana a aquella cuyas probabilidades de X son similares. Por lo tanto, es poco plana cuando algunos valores de X son mucho más probables que otros (se dice que la función es más puntiaguda en los valores más probables). En una distribución de probabilidad plana (con alta entropía) es difícil poder predecir cuál es el próximo valor de X que va a presentarse, ya que todos los valores de X son igualmente probables.

4.1.2. Entropía de la información

La Entropía de la información, también conocida como entropía de Shannon fue desarrollada por C.E. Shannon en el año 1948[37]. Su objetivo es la medición del grado de incertidumbre de una fuente de información. Dado un conjunto de datos X, y un conjunto finito de símbolos $x_1 \dots x_n$ cuyas probabilidades de aparición son $p_1 \dots p_n$, la entropía de la información es expresada de la siguiente manera:

$$H(X) = \sum_i p(x_i) \log_2 p(x_i)$$

Nótese que se aplica el logaritmo en base 2 bajo la asunción de que la información a tratar es representada mediante código binario. Al cambiar el sistema de codificación, la base del logaritmo debe coincidir con la de la nueva representación.

El valor de la entropía de la información es mayor cuando X se asocia a una distribución uniforme. Su valor es 0 cuando una probabilidad p_i es 1, y el resto 0 (no hay incertidumbre). Para el resto de posibles distribuciones su valor se comprende entre 0 y $\log_2 n$, siendo este último el máximo alcanzable.

La entropía de la información ha sido frecuentemente aplicada en el área de la detección de ataques de denegación de servicio, siendo muy frecuente en la bibliografía. En [21] se demuestra que es una de las métricas menos dependientes de las características de la red, lo que hace que su uso sea especialmente recomendable para tratar el problema de la denegación de servicio. Sin embargo también advierten de que su popularización puede llevar a la aparición de ataques de "suplantación de entropía", basados en la inyección de tráfico con el fin de que sus variaciones pasen desapercibidas.

4.1.3. Entropía de Rènyi

Según la entropía de Shannon, el cálculo de su entropía espera la obtención de valores más altos cuando la variable de información es más alta. Análogamente, existe una tendencia a producir valores más bajos cuando dicha variable es más pequeña. Para cuantificar la aleatoriedad del sistema, A. Rènyi propuso una métrica para la entropía de orden α como generalización de la entropía de la información[38]. Dada la distribución de probabilidades $p_1 \dots p_n$, la entropía de Rènyi es definida como:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log_2(\sum_{i=1}^{n} p_i^\alpha)$$

donde $\alpha \in [0, 1)$. Al igual que en la entropía de Shannon, el máximo valor de se obtiene cuando todas las probabilidades p_i presentan el mismo valor. Las variaciones del orden α llevan a los diferentes casos particulares. Por ejemplo, cuando $\alpha = 1$ se considera la entropía de Shannon. El caso $\alpha = 2$ lleva a la entropía cuadrática de Rènyi o el caso $\alpha = \infty$ a la entropía mínima.

La entropía de Rènyi fue aplicada en [22] para evaluar la eficacia de diferentes detectores de ataques de denegación de servicio con métricas basadas en distintas entropías. Su estudio concluye en que los casos de orden elevado acarrean un nivel de restricción más alto. Esto se traduce en una mejor precisión reconociendo ataques, pero conlleva mayores tasas de falsos positivos.

4.2. Predicción en series temporales

Una serie temporal es una secuencia de datos, observaciones o valores, medidos en determinados momentos y ordenados cronológicamente. Los datos pueden estar espaciados a intervalos iguales (como la temperatura en un observatorio meteorológico en días sucesivos al mediodía) o desiguales (como el peso de una persona en sucesivas mediciones en el consultorio médico, la farmacia, etc.). Para el análisis de las series temporales se usan métodos que ayudan a interpretarlas y que permiten extraer información representativa sobre las relaciones subyacentes entre los datos de la serie o de diversas series y que permiten en diferente medida y con distinta confianza extrapolar o interpolar los datos y así predecir el comportamiento de la serie en momentos no observados; sean en el futuro (extrapolación pronóstica), en el pasado (extrapolación retrógrada) o en momentos intermedios (interpolación). Estos métodos se basan en encontrar el proceso estocástico que originó dicha serie temporal. Formalmente, un proceso estocástico es una aplicación tal que:

$$X : \Omega \times T \longrightarrow S$$
$$(\omega, t) \longrightarrow X(\omega, t)$$

El análisis clásico de las series temporales se basa en la suposición de que los valores que toma la variable de observación es la consecuencia de cuatro componentes, cuya actuación conjunta da como resultado los valores medidos. A continuación se describen dichos componentes:

- **Tendencia.** La tendencia indica la marcha general y persistente del fenómeno observado. De este modo refleja su evolución a largo plazo.
- **Variación estacional.** La variación estacional es el movimiento periódico de corto plazo. Se trata de una componente causal debida a la influencia de ciertos fenómenos que se repiten de manera periódica, y que recoge las oscilaciones que se producen en esos períodos de repetición.
- **Variación cíclica.** La variación cíclica muestra patrones que se dan en relación a la tendencia.
- **Ruido.** El ruido, de carácter errático, también denominada residuo, no muestra ninguna regularidad y es impredecible, debido a fenómenos de carácter ocasional. Muchos métodos de predicción se basan en modelizar todos los componentes mostrando que el único componente que queda sin explicar es justamente ruido.

Tomando como eje la relación entre sus componentes, las series temporales habitualmente se clasifican en aditivas, multiplicativas o mixtas. A continuación se describe cada uno de estos grupos:

- **Aditivas.** El conjunto de series aditivas reúne aquellas que se componen sumando la tendencia T_t, estacionalidad E_t, variación cíclica C_t y ruido $\mathcal{E}_t$. Se expresan de la siguiente manera:

$$X_t = T_t + E_t + C_t + \mathcal{E}_t$$

- **Multiplicativas.** Las series multiplicativas son aquellas compuestas por el producto de la tendencia T_t, estacionalidad E_t, variación cíclica C_t y ruido $\mathcal{E}_t$. Se expresan de la siguiente manera:

$$X_t = T_t \cdot E_t \cdot C_t \cdot \mathcal{E}_t$$

- **Mixtas.** Las series mixtas son aquellas se componen combinando sumas y productos de la tendencia T_t, estacionalidad E_t, variación cíclica C_t y ruido $\mathcal{E}_t$. Existen varias alternativas, siendo algunas de ellas:

$$X_t = T_t + E_t \cdot C_t \cdot \mathcal{E}_t$$
$$X_t = T_t + E_t \cdot \mathcal{E}_t$$
$$X_t = T_t \cdot E_t \cdot C_t + \mathcal{E}_t$$

La versión aditiva asume que los efectos estacionales son constantes y no dependen del nivel medio de la serie. Por el contrario, la versión multiplicativa supone que las componentes estacionales varían en función del nivel medio local desestacionalizado. Dicho de otro modo, las fluctuaciones estacionales crecen (o decrecen) proporcionalmente con el crecimiento (o decrecimiento) del nivel medio de la serie.

4.2.1. Métodos de predicción

Uno de los usos más extendidos de las series de datos temporales es su análisis predictivo. A continuación se explica uno de los métodos más populares y precisos, el método ARIMA.

La familia de modelos ARIMA (abreviado del ingés *Auto Regressive Integrated Moving Average*), también conocidos como modelos de Box-Jenkins, juegan un papel importante en el campo de las series temporales[39]. Son capaces de recoger la tendencia y la estacionalidad de los datos pero, a diferencia del método de Holt-Winters y de los modelos estructurales, no se basan en la descomposición de las series en tales factores. Para comenzar, deben aclararse algunos términos o conceptos que se usarán a la hora de definir los modelos ARIMA.

Un proceso es **estacionario en sentido estricto** si el comportamiento de una colección de variables aleatorias solo depende de su posición relativa, no del instante t. Al ser esta condición muy restrictiva se suele asumir una relajación: la **estacionariedad débil**. Se dice que un proceso es estacionario en sentido débil si:

$$E[X_t] = \mu < +\infty \qquad \forall t$$
$$V[X_t] = \gamma_0 < +\infty \qquad \forall t$$
$$Cov[X_t, X_{t+k}] = \gamma_k \qquad \forall t, \forall k$$

Este tipo de covarianza se llama autocovarianza y los coeficientes $\{\gamma_0, \gamma_1, \dots\}$ cons-

tituyen la función de autocovarianza. A menudo, la función de autocovarianza se estandariza, dando lugar a la función de autocorrelación $\{\rho_0, \rho_1, \dots\}$.

$$\rho_k = \frac{\gamma_k}{\gamma_0} \qquad k = 0, 1 \dots$$

donde $\gamma_0 = Cov[X_t, X_t] = Var[X_t]$. En procesos estacionarios ρ_k mide la correlación entre X_t e X_{t+k}.

Por otro lado, el **operador retardo** B, aplica un desfase en el periodo de la serie temporal. Se expresa como:

$$BX_t = X_{t-1}$$

y al aplicarse sucesivamente s veces, desfasa s periodos su valor:

$$B^s X_t = X_{t-s}$$

Finalmente, el **operador diferencia** de orden 1 se define como:

$$\Delta X_t = (1 - B)X_t$$

Generalizándo para el orden d:

$$\Delta^d X_t = (1 - B)^d X_t$$

El método de ARIMA es una combinación de los modelos autoregresivos (AR), modelos de medias móviles (MA) y métodos que eliminan la tendencia de una serie temporal. A continuación describe cada uno de ellos.

Modelos autorregresivos (AR)

Explican el comportamiento de la serie temporal en base a su propio pasado. Es una regresión de la variable en sí misma (autorregresión). En un modelo $AR(p)$ se tiene que:

$$X_t = \mu + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + a_t$$

donde a_t es un proceso de ruido blanco (ruido con media cero).

La **condición de estacionariedad** se exige sobre la parte autorregresiva del modelo y es una condición necesaria para el ajuste de modelos ARMA. Establece

que el polinomio

$$1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p$$

debe tener sus raíces fuera del círculo unidad.

Modelo de medias móviles (MA)

Explica el comportamiento de la serie temporal en base al pasado del proceso de error (media móvil de la serie de errores). Permite al modelo "aprender de sus errores". En un modelo $MA(q)$ se tiene que:

$$X_t = \mu - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} + a_t$$

donde a_t es un proceso de ruido blanco.

A diferencia de los procesos AR, los procesos MA son siempre estacionarios. Sin embargo, existe otra condición importante denominada **condición de invertibilidad**, condición que cumplen los procesos AR pero no los MA. Para que un proceso MA la cumpla, es necesario que el polinomio

$$1 - \theta_1 z - \theta_2 z^2 - \cdots - \theta_q z^q$$

tenga sus raíces fuera del círculo unidad. Intuitivamente, la condición de invertibilidad dice que datos más alejados en el tiempo tienen menos peso en la predicción que datos recientes.

Modelo autorregresivo de medias móviles (ARMA)

Los modelos AR se pueden combinar con los MA para formar una familia de modelos de series temporales más general y, por consiguiente, más útil. La expresión general de un modelo estacionario $ARMA(p, q)$ es:

$$X_t = \mu + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

donde a_t es un proceso de ruido blanco. Esta expresión es equivalente a:

$$(1 - \phi_1 B - \cdots - \phi_p B^p) X_t = \mu + (1 - \theta_1 B - \cdots - \theta_q B^q) a_t$$

que también suele encontrarse como:

$$\Phi_p(B) X_t = \mu + \Theta(B) a_t$$

Para que las estimaciones de los parámetros de un modelo ARMA tengan las propiedades estadísticas adecuadas, es necesario que la serie muestral sea estacionaria. Pero desafortunadamente, es muy común encontrar series no estacionarias a las que no se les puede ajustar directamente un modelo ARMA. Para solucionar este inconveniente existen procedimientos para su transformación.

- Si el proceso no es estacionario en varianza es recomendable la aplicación de logaritmos.
- Si el proceso no es estacionario en media, se puede hacer que lo sea a través del operador diferencia. Este operador se aplica sobre la propia serie (no sobre el proceso residual).

Los procesos **integrados** son aquellos que precisan de la realización del operador diferencia para ser estacionarios. Un proceso integrado de orden d verifica:

$$(1 - B)^d X_t = a_t$$

donde a_t es un proceso de ruido blanco. Decimos que si X_t es un proceso $ARIMA(p, d, q)$ entonces $(1 - B)^d X_t$ es un $ARMA(p, q)$.

Capítulo 5

Fortalecimiento de Tor frente a DDoS

En este capítulo se propone una estrategia de detección de ataques de denegación de servicio en la red Tor. Para alcanzar este objetivo, la aproximación realizada trata de identificar las amenazas DDoS dirigidas contra sus nodos OR. La detección de estos ataques permitirá el despliegue de contramedidas, y de esta forma fortalecer su resistencia.

En base a las características de Tor, es asumible que el contenido único de las trazas de tráfico a analizar son los encabezados de las células de control y de transimisión. Adicionalmente, y de manera previa a la fase de desarrollo, han sido asumidas las siguientes características sobre los ataques de denegación de servicio.

1. Los ataques DDoS en Tor se corresponden con la clasificación de ataques de inundación. Actualmente no se conocen ataques basados en reflexión o amplificación. De hecho la ejecución de estos resulta especialmente compleja dadas las características de la red.

2. Los ataques DDoS en Tor presentan patrones de tasa alta y tasa baja, por ser éstas las dos únicas maneras viables de ejecutar con éxito amenazas basadas en inundación[9].

3. El objetivo principal de los ataques DDoS es agotar el ancho de banda de la infraestructura. Sin embargo, también pueden ser aplicados para alcanzar fines más sutiles, tal y como sucede en [35][36].

A continuación se describen la arquitectura, y los diferentes procesos involucrados en la detección de ataques: modelado de tráfico, análisis de la información y toma de decisiones.

5.1. Arquitectura

La arquitectura de la propuesta está representada en 5.1. En ella destacan tres bloques de procesamiento de información: monitorización, modelado y análisis.

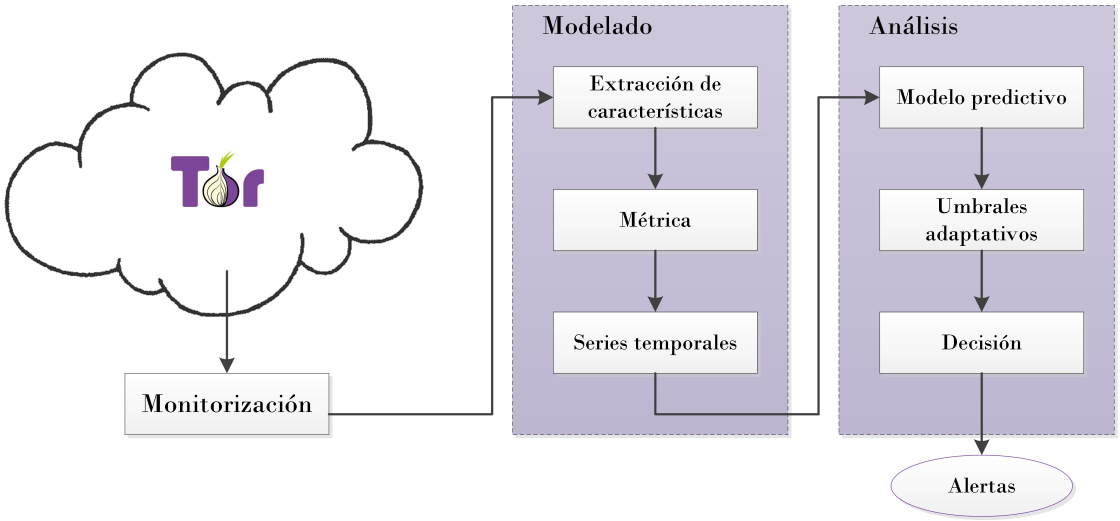

Figura 5.1: Arquitectura del sistema de detección de DDoS en Tor

En la etapa de monitorización se observa el tráfico que fluye a través del sensor. Asumiendo que será desplegado en nodos OR, el tráfico de interés es el entrante y el saliente. El proceso de modelado es llevado a cabo en tres pasos: en primer lugar, se extraen las características más importantes. Como se mostrará en la siguiente sección, este es un punto especialmente sensible, dado el alto grado de restricción que ofrece Tor. A partir de dichas características es posible la construcción de métricas, las cuales son alineadas en el tiempo formando series temporales. Al concluir esta etapa, las series temporales modelan las características del tráfico monitorizado en los últimos periodos de observación.

La detección de ataques se basa en la identificación de anomalías en las series temporales construidas a partir de las métricas. Con este fin, la etapa de análisis construye modelos predictivos capaces de pronosticar el valor que alcanzarán las métricas en futuras observaciones. Cuando se produce error en alguna predicción, se emite una alerta. Sólo entonces el operador podrá iniciar las acciones de mitigación correspondientes.

5.2. Modelado del tráfico

El modelado del tráfico se realiza en tres etapas: extracción de la información, métrica y series temporales. A continuación se describe cada una de ellas.

5.2.1. Extracción de la información

La mayor parte de las propuestas de la bibliografía que comparten el objetivo de reconocer ataques DDoS basan sus métodos de análisis en el estudio de las características de las conexiones observadas. Se trata de una metodología exportada de otras áreas de la detección de intrusiones, centrada en el estudio de flujos de información, también conocidos como *Flows*, y cuyo uso fue estandarizado por el IETF en el protocolo IP, adquiriendo el nombre flujos de tráfico IP o *IP Flows*. En [40] se profundiza en su aplicación en otras áreas de la seguridad de la información. Dados los buenos resultados obtenidos en trabajos previos, la información extraída en esta aproximación trata de adaptar el concepto de *IP Flow* a las limitaciones de la red Tor.

Los *IP Flows* están construidos por una dirección IP origen, una dirección destino, y el número de datagramas que durante un intervalo de tiempo de observación, han sido enviados entre ellos. Pero las características que hacen de Tor una red anónima eficaz, impiden la visualización de estos valores. En su lugar, en este trabajo se propone por primera vez el concepto de *Flow* sobre entorno Tor, a lo que se ha denominado *Tor Flow*.

Los *Tor Flows* son calculados en base a la conexión TLS y al *circid* del circuito al que pertenece cada célula monitorizada. Esto es debido a que todo nodo OR establece una conexión TLS con los demás nodos de la red Tor. Para cada conexión TLS, el *circid* de la célula entrante determina de qué circuito llega dicha célula. De esta manera, a pesar de que los *Tor Flows* no aportan tanta información como los *IP Flows*, sí que permiten distinguir el origen del tráfico, y si éste sigue un mismo circuito en común, a pesar de que no se conozcan sus siguientes saltos en el circuito. Nótese que el uso de Tor Flows no pone en riesgo la privacidad ofrecida por la red, ya que en ningún momento se facilita el desenmascaramiento de ninguno de los extremos finales.

Formalmente, sea T el conjunto de los identificadores de las conexiones TLS y sea C el conjunto de los *circid* en un momento dado. Un *Tor Flow* queda definido

como $f_{ij} = \{(t_i, c_j)|t_i \in T, c_j \in C\}$.

5.2.2. Métrica

La métrica que aplica el sistema propuesto es la adaptación de la entropía de Shannon, a la medición de la incertidumbre de la cantidad y el tipo de *Tor Flows* que fluyen a través del sensor. La decisión del uso de dicha entropía parte del estudio publicado en [22], dónde el uso del factor de ajuste α sobre la entropía de Rènyi ha demostrado que en valores más bajos, los sensores se comportan de manera menos restrictiva. El valor eficaz más bajo fue $\alpha = 1$, que corresponde con el caso particular de la entropía de Shannon. Con esto se pretende reducir el problema de las altas tasas de falsos positivos, típico de los sensores basados en el reconocimiento de anomalías.

A partir de la información extraída es posible conocer la cantidad de células que componen cada *Tor Flow*, en los periodos de observación. A partir de ello es posible hallar su probabilidad de aparición en dicha observación. Esta viene dada por la expresión:

$$p_{ij}(t_i, c_j) = \frac{N_{ij}(t_i, c_j)}{\sum_i \sum_j N_{ij}(t_i, c_j)}$$

donde (t_i, c_j) representa el *Tor Flow* f_{ij} y $N_{ij}(t_i, c_j)$ representa el número de células relativas al *Tor Flow* f_{ij}. A partir de esto es posible el cálculo de la entropía:

$$H(F) = -\sum_{i,j} p_{ij}(t_i, c_j) \log_2 p_{ij}(t_i, c_j)$$

5.2.3. Series temporales

Con el fin de facilitar la identificación de las variaciones en la entropía, éstas son tratadas como una serie temporal univariante de N observaciones a lo largo del tiempo, expresada de la siguiente manera:

$$H_\alpha(X) = H_\alpha(X)_t : t \in 1, \ldots, N$$

5.3. Análisis de la información

La información es analizada en tres etapas: elaboración de modelos predictivos y predicción, generación de umbrales adaptativos y toma de decisiones. A continuación se describe cada una de ellas.

5.3.1. Modelos predictivos

El componente encargado del análisis de la información procesada tiene como parámetro de entrada, la serie temporal generada a partir de las métricas extraídas. A partir de ella se construye un modelo predictivo ARIMA que permitirá predecir las futuras variaciones de la entropía. El modelo se construye de la siguiente forma[41]:

1. Identificar el polinomio diferenciador $\delta(d) = (1 - B)^d$ que contiene las raíces unidad.

2. Minimizar el Criterio de Información Bayesiano (BIC) dado por

$$BIC_{p,q} = \ln(\sigma^2_{p,q}) + (p + q)\frac{\ln(N - d)}{N - d}$$

siendo N el número de observaciones y

$$\sigma^2_{p,q} = \frac{1}{N}\sum_{t=p}^{n}(X_t - \sum_{i=1}^{p}\Phi_i^{(p,q)}X_{t-i} + \sum_{k=1}^{q}\Theta_k^{(p,q)}a_{t-k})$$

5.3.2. Umbrales adaptativos

Para facilitar la toma de decisiones, en esta etapa se construyen dos umbrales adaptativos. El primero limita las cotas superiores del intervalo de predicción, mientras que el segundo limita las cotas inferiores. En realidad, estos umbrales adaptativos son los extremos del intervalo de confianza de grado $1 - \alpha$, donde $\alpha \in (0, 1)$, que es calculado a partir de una distribución normal obtenida a partir de la serie original y de la serie de los errores.

5.3.3. Toma de decisiones

La toma de decisiones tiene en consideración la entropía de cada periodo de observación, y los intervalos de predicción construidos a partir del modelo ARIMA. Si la entropía excede alguno de estos umbrales, las observaciones son consideradas anómalas, y se emitirá una alerta.

Capítulo 6

Experimentación

En este capítulo se describe la experimentación realizada. Para facilitar su comprensión ha sido dividido en tres secciones. En la primera sección se explica la implementación del sistema de detección. A continuación se detallan las características de los conjuntos de muestras considerados. Finalmente, se introduce la metodología de evaluación, haciendo hincapié en las distintas pruebas realizadas para validar la herramienta.

6.1. Implementación

El sistema de detección desplegado en la experimentación, distingue dos etapas de procesamiento: modelado y análisis. En la primera de ellas se extraen las características del entorno protegido, las cuales varían en función del tipo de tráfico. Las pruebas realizadas requieren del tratamiento de tráfico TCP/IP y Tor. Para el primer caso, los datos necesarios son las direcciones IP (origen y destino), y los puertos (origen y destino) de cada datagrama. Esto permite la construcción de *Flows*. Sin embargo, para la red Tor los datos a tratar son su *circid* y la conexión TLS, facilitando la definición de *Tor Flows*.

Una vez obtenidas las características se procede a determinar la métrica, en este caso la entropía. Este proceso se realiza en el lenguaje de programación C++. Para ello se leen los datos de tráfico y se crean los distintos *Flows*. Se define como observación, a cada conjunto de paquetes de tamaño n capturados de manera consecutiva. Tras analizar n paquetes, se calcula su entropía (el valor asignado por defecto a n es 1000). Este proceso se realiza sucesivamente hasta alcanzar una cantidad considerable de observaciones (el valor asignado por defecto es 80). A partir de estos valores se genera una serie temporal y comienza la segunda fase, el análisis de estos datos.

El análisis de la serie temporal involucra la construcción de un modelo ARIMA, capaz de pronosticar la entropía de la siguiente observación. Esta segunda fase está desarrollada en el lenguaje de programación Python. Las principales funciones implementadas se comentan a continuación:

- **Init.** La función *Init* genera un modelo ARIMA asociado a la serie temporal de observaciones. De acuerdo a los datos introducidos, calcula los parámetros p, d, q.
- **Forecast.** La función *Forecast* construye el intervalo de predicción de un modelo ARIMA.
- **Update.** La función *Update* actualiza la serie temporal con una nueva observación.
- **Remodel.** La función *Remodel* recalcula los parámetros p, d, q.
- **Summary.** La función *Summary* devuelve los errores de predicción cometidos a lo largo del análisis.

En este punto se dispone de dos procesos aislados. En primer lugar, el algoritmo de la entropía en C++, que analiza tráfico y calcula sus valores. Por otro lado, el método de ARIMA en Python que, dada una serie temporal inicial, predice un intervalo de confianza donde debería encontrarse el siguiente valor de la serie. Para enlazar ambos procesos se ha dispuesto de varias técnicas: variables compartidas, *pipes* y *sockets*. Tras el estudio de cuál era el método más conveniente, se optó por el uso de *sockets*. Esto es debido a que el módulo en Python estaba escrito de forma que se podía usar como una API. En 6.1 se puede ver la arquitectura de la solución propuesta.

Para establecer la comunicación se ha desarrollado un servidor en Python (server.py) y una API en C (client.c) con las funciones necesarias. La comunicación se basa en el envío de mensajes, compuestos de un comando y de la carga útil necesaria relacionada con dicho comando. Todos los mensajes se confirman con un ACK para asegurar la sincronización entre las dos partes.

El hecho de utilizar un cliente y un servidor escritos en lenguajes de programación diferentes, supone que la información intercambiada entre ambos debe tener la misma representación en ambos lenguajes. Es por esto por lo que los mensajes del

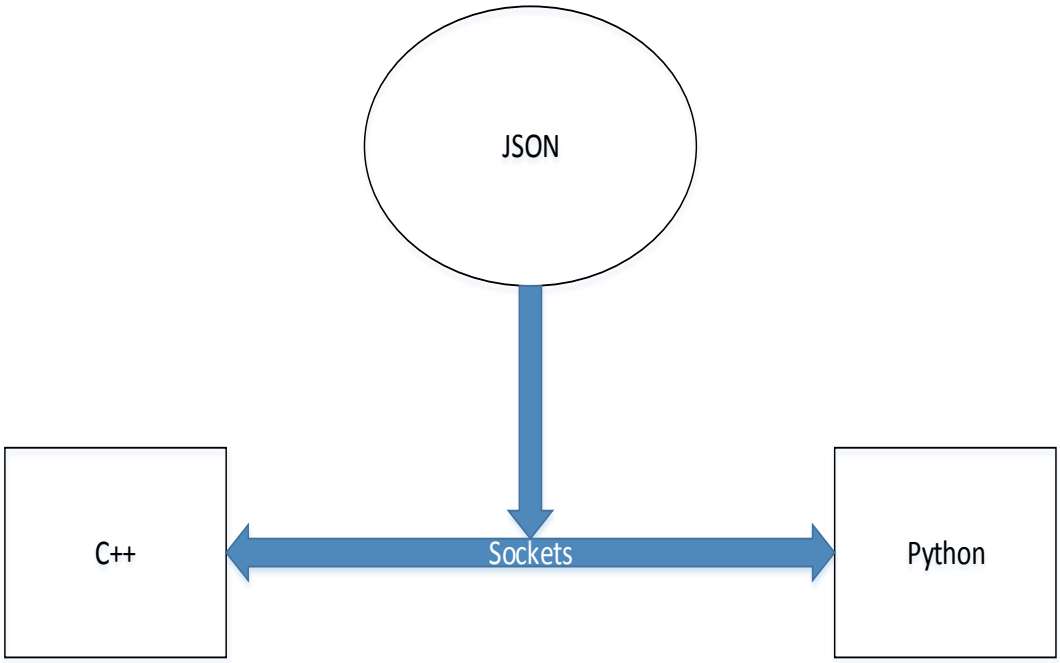

Figura 6.1: Comunicación entre los distintos módulos

protocolo propio han sido encapsulados con JSON (del inglés *JavaScript Object Notation*)[42], un formato ligero para el intercambio de datos, que usa lenguaje legible por humanos para transmitir objectos que consisten de pares atributo-valor.

6.2. Colección de muestras

A pesar de que el sistema desarrollado es capaz de analizar tráfico en tiempo real, evaluarlo correctamente requiere del uso de colecciones de muestras previamente etiquetadas. A continuación se describen los conjuntos de muestras con que se ha experimentado, agrupados en función de su entorno de captura.

6.2.1. Capturas TCP/IP

Actualmente existe una gran cantidad de colecciones públicas de tráfico TCP/IP. Su uso es frecuente en la bibliografía, ya que permite comparar los resultados obtenidos con propuestas previas. En la experimentación realizada se ha aplicado la colección CAIDA'07[27], ya que se han asumido las indicaciones de [26]. Sin embargo, y tal y como indican sus autores, el contenido de tráfico legítimo de CAIDA'07 ha sido prácticamente eliminado de sus muestras. Para suplir esta carencia, en la expe-

rimentación se han aplicado muestras de las capturas pasivas de tráfico CAIDA'14 [29], recopiladas entre los años 2013 y 2014. Tanto las trazas de tráfico legítimo como las de ataques son tomadas en el mismo equipo y en la misma red en años diferentes. Además, ambas vienen dadas en formato .pcap, el cual es entendido por programas como *Wireshark* o *tcpdump*. Para poder interpretar dichos datos primeramente es necesario su parseo utilizando la librería libpcap, generando ficheros de texto entendibles por la herramienta.

6.2.2. Capturas Tor

Para la obtención de capturas de tráfico en Tor, se ha utiliza la herramienta Chutney. El proyecto Chutney parte de la necesidad de emular y configurar una red privada con Tor en la que capturar tráfico no tenga implicaciones éticas, permitiendo crear varios escenarios en los que es capaz de levantar autoridades de directorio, *relays*, clientes, *bridges* y cualquier elemento adicional que conforma la red de Tor. Se trata de una herramienta muy reciente, y con escasa documentación, a la cual se ha contribuido a lo largo de este trabajo. El estado actual del proyecto puede consultarse en el repositorio [43].

A partir de Chutney se han configurado dos redes:

- Una red con 4 directorios de autoridad, 50 nodos cliente, 30 nodos *relay* (los cuales pueden ser nodos de entrada o nodos de salida) y 20 nodos intermedios.
- Una red con 1 directorio de autoridad, 25 nodos cliente, 15 nodos *relay* y 7 nodos intermedios.

Para generar tráfico se ha seguido el siguiente proceso:

- En el fichero *chaneltls.c* que forma parte del código fuente de Tor se añade un fragmento de código para generar *logs* de forma que cada vez que un nodo de la red procesa una célula, esto quede registrado, y por lo tanto se guarda su *circid* y la conexión TLS de la célula. De esta manera, tras generar tráfico obtenemos un *log* para cada nodo de la red con las células que ha procesado.
- Cada nodo cliente de la red escucha por un puerto determinado. Para generar tráfico en abundancia se crea un *script* que mediante el protocolo SOCKS5 hace peticiones a los nodos clientes de Tor para acceder a un servidor web. De esta manera, al tener que realizar una petición a un servidor web, cada nodo cliente debe crear un circuito con un nodo de entrada, otro intermedio

y otro de salida, y enviar células a través del circuito para llevar a cabo la petición y recoger la respuesta del servidor. De este modo se consigue crear tráfico legítimo en la red Tor.

Para producir denegación de servicio a un nodo de Tor se han hecho dos variaciones del ataque *replay*[35]:

- En el primero, se ha modificado el código fuente de Tor para crear un nodo malicioso que duplique las células *relay* que recibe.
- En el segundo se ha modificado el primero para que además de duplicar la célula (que causa el cierre del circuito) envíe esa misma célula un cierto número de veces más.

Este proceso ha permitido la obtención de un conjunto de *datasets* con tráfico legítimo y tráfico atacante en una red local de Tor.

6.3. Metodología de evaluación

Con objetivo de evaluar adecuadamente el funcionamiento de la herramienta, se han realizado diversos experimentos. Tanto en el caso de la red TCP/IP como en la red Tor, consisten en analizar ficheros que contienen tráfico legítimo seguido de tráfico atacante.

Concretamente para verificar la eficacia de la herramienta en la red TCP/IP se han utilizado 24 trazas de tráfico legítimo del año 2013, 24 trazas de tráfico legítimo del 2014 y 16 trazas de ataques en CAIDA'07. En total se llevaron a cabo 200 combinaciones distintas de tráfico legítimo-malicioso, que fueron analizadas por el sistema propuesto.

Para comprobar la eficacia de la herramienta en la red Tor local generada por Chutney se han utilizado 54 trazas de tráfico legítimo seguidas de tráfico de ataque (enlazadas según la funcionalidad del componente en cuestión) de la primera de las topologías mencionadas anteriormente y 22 trazas de la segunda topología obtenidas siguiendo el mismo método.

Los puntos de especial interés del proceso de evaluación son la tasa de acierto y tasa de falsos positivos del sistema. La primera determina la frecuencia con que los ataques DDoS son identificados por el detector. La tasa de falsos positivos indica

la frecuencia con que el tráfico legítimo es erróneamente etiquetado como malicioso. Ambos valores son importantes, y su resultado depende del nivel de restricción de la herramienta. De este modo, cuando el sistema de detección se comporta de manera restrictiva, tiende a detectar más ataques, pero también a bloquear el flujo de una mayor cantidad de tráfico legítimo por error. A medida que disminuye la restricción, el sistema se vuelve más permisivo. Por lo tanto, menos tráfico legítimo es confundido con malicioso, pero existe un mayor riesgo de que los ataques pasen inadvertidos.

Capítulo 7

Resultados

7.1. Resultados obtenidos con CAIDA'07

Tras analizar las 200 combinaciones de tráfico legítimo seguido de tráfico de ataque con la herramienta se observa:

- Una tasa de aciertos del 97 %. En la mayoría de los casos el sistema de detección identifica que se produce una anomalía.

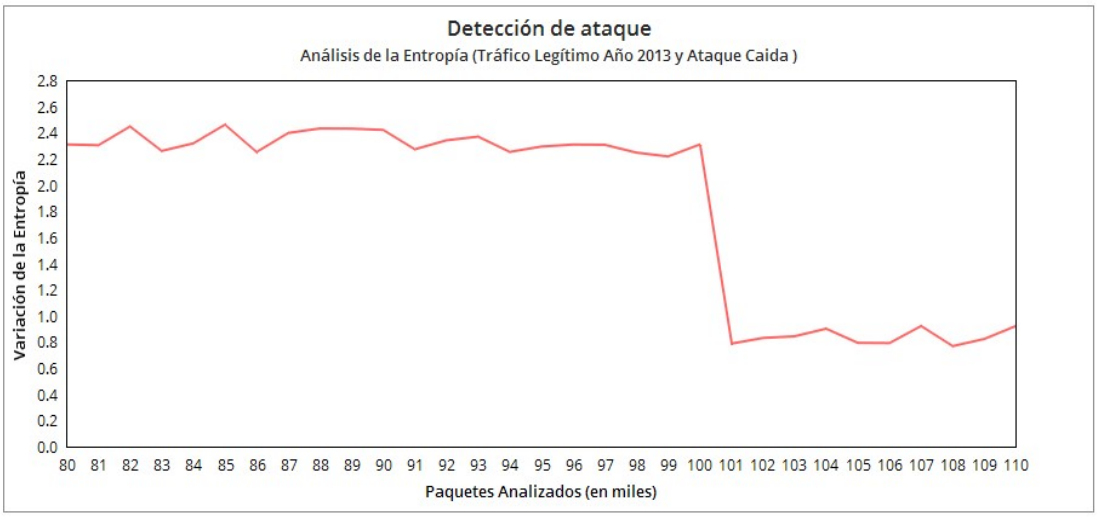

Figura 7.1: Tráfico legítimo 2013 junto al de CAIDA'07

Como se puede ver en 7.1 y 7.2 se produce un cambio brusco en la entropía cuando se han analizado 101000 paquetes, que es justo cuando se ha analizado por primera vez tráfico atacante.

- Una tasa de falsos positivos del 2 %, debido a variaciones bruscas en el tráfico legítimo que ocasionan que la herramienta lo detecte y emita una alerta. En 7.3

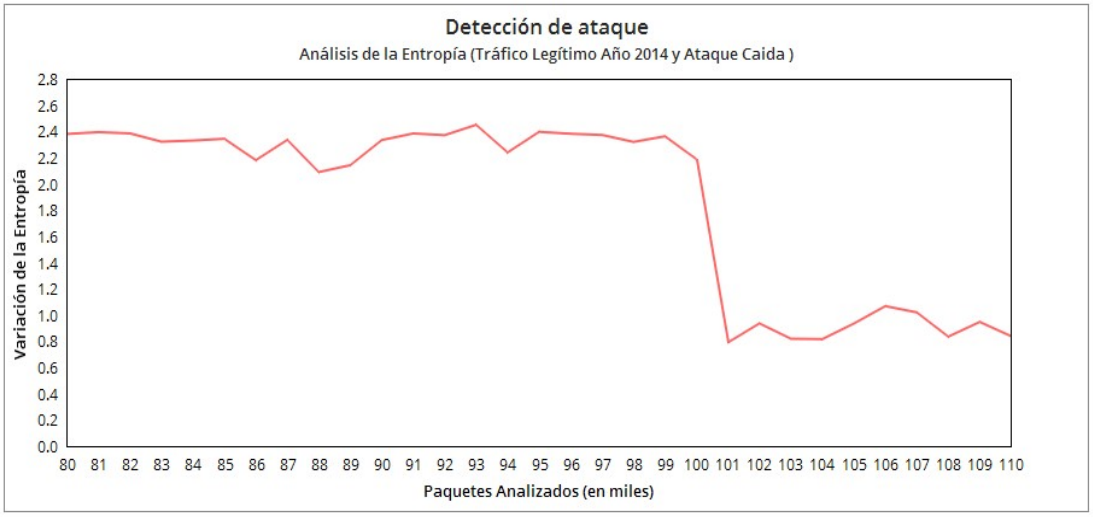

Figura 7.2: Tráfico legítimo 2014 junto al de CAIDA'07

se puede ver este cambio, a pesar de no ser tan brusco como en 7.1 o 7.2. Hay que tener en cuenta que al combinar un mismo fichero de tráfico legítimo con varios de tráfico de ataque, en caso de producirse un falso positivo se produciría en todas sus combinaciones. Debido a esto a pesar de haberse realizado 200 pruebas, para los falsos positivos solo tienen sentido 48 de ellas.

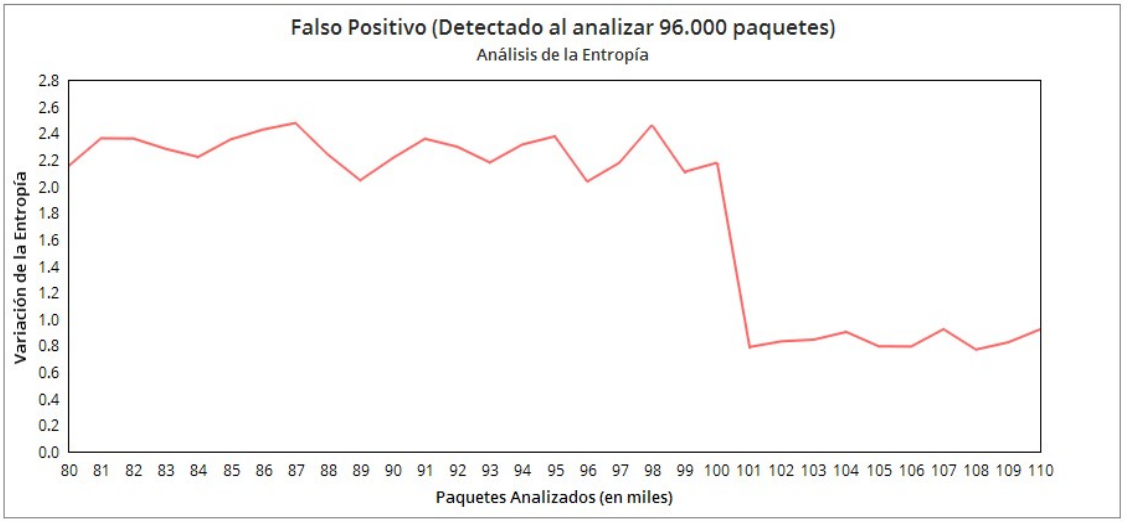

Figura 7.3: Falso positivo

7.2. Resultados obtenidos en Tor

Tras analizar los 76 *datasets* que contienen tráfico legítimo seguido de tráfico atacante se observa:

- Una tasa de aciertos del 98 %.
- Una tasa de fallos del 5 %.

Cabe destacar que en este caso las anomalías detectadas se deben a un incremento de la entropía y no debido a su disminución como cabría esperar en un ataque de denegación de servicio. Esto se debe a que el ataque *replay* al romper circuitos provoca que nuevos circuitos tengan que crearse y en consecuencia se registre en los *logs* más variedad de *circids*, llevando a un consiguiente aumento en la entropía.

También cabe mencionar que los valores de entropía obtenidos en el tráfico legítimo son bastante cercanos a cero (ver 7.4), debido a que se necesita un número muy grande de clientes en comparación con el número de *relays* para la creación de numerosos circuitos. En la red Tor se tiene actualmente 1 *relay* por cada 212,76 clientes[44], lo que hace inviable realizar pruebas siguiendo esta proporción en Chutney.

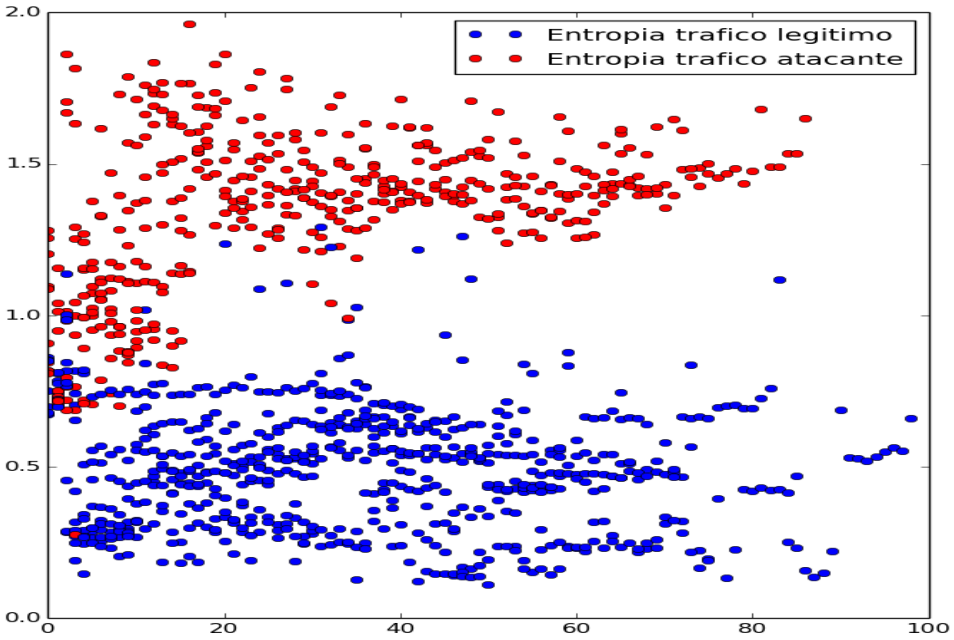

Figura 7.4: Entropía con tráfico legítimo y tráfico atacante

Capítulo 8

Conclusiones y Trabajo Futuro

8.1. Conclusiones

En este trabajo se ha desarrollado un sistema de detección capaz de reconocer ataques de denegación de servicio, tanto en redes TCP/IP convencionales, como en la red Tor. Por lo tanto, se ha cumplido el objetivo principal del proyecto.

El sistema propuesto combina métodos de elaboración de métricas basados en el grado de incertidumbre del tráfico que fluye a través de la red monitorizada, con la construcción de modelos predictivos capaces de pronosticar su futuro valor. A la hora de tomar decisiones, se ha considerado la diferencia de la última observación, respecto a intervalos de predicción. En el caso de que estos sean superados, se considera un comportamiento impredecible, y por lo tanto anómalo, situación que desencadena la emisión de una alerta.

La experimentación realizada ha considerado pruebas sobre la colección de muestras pública CAIDA'07, respaldada por la comunidad investigadora en gran cantidad de publicaciones. Por otro lado, se ha experimentado directamente sobre la infraestructura de Tor, por medio de la herramienta de simulación Chutney. Esto ha permitido recopilar capturas de tráfico legítimo y malicioso, para su posterior aplicación en los procesos de evaluación. En ambos escenarios de pruebas, los resultados obtenidos han arrojado una excelente precisión, con una alta tasa de acierto, y una baja tasa de falsos positivos.

Para la elaboración de este trabajo se ha llevado a cabo el estudio de las principales características de los ataques DDoS y sus contramedidas. Esto ha implicado revisar las publicaciones más relevantes de los últimos años relacionadas con esta

área. Asimismo, se han investigado las características de la infraestructura Tor. Cabe destacar que esto último ha resultado ser una labor especialmente compleja, debido a la escasa documentación disponible, y a que Tor es una red anónima, situación que dificulta el rastreo de información. Asimismo, y en el marco de la experimentación realizada, se ha trabajado directamente sobre el código fuente del proyecto. Debido a esto, hemos estado en contacto directo con su comunidad de desarrollo, participando activamente en listas de correo, foros y canal IRC. Esto ha llevado a la oportunidad de establecer un contacto continuo con Nick Mathewson, uno de los principales creadores del proyecto Tor, quién ha resuelto personalmente algunas de las dudas que se han planteado durante su transcurso.

8.2. Trabajo futuro

El trabajo realizado deja abierta una gran cantidad de futuras líneas de investigación. Las primeras de ellas están relacionadas con la propia estrategia de detección: sería de interés valorar el comportamiento del sistema al aplicar otro tipo de métricas, modelos predictivos o algoritmos de inicialización.

Por otro lado, y de cara a mejorar su integración en Tor, sería conveniente el estudio de estrategias de rastreo de ataques o estrategias de mitigación, dentro de dicha infraestructura. Ambas proponen interesantes desafíos, relacionados tanto con alcanzar una buena eficacia, como en preservar la privacidad de los extremos de las comunicaciones.

Finalmente, y en relación con la experimentación, sería de interés ampliar el conjunto de pruebas realizadas. Existen diferentes aspectos relacionados con el rendimiento, o la calidad de servicio que no han sido evaluados. Además, sería importante estudiar la capacidad de detección del sistema frente a diferentes tipos de ataques de denegación de servicio, e incluso técnicas de evasión.

Bibliografía

[1] Eurostat (2015). "Information society statistics - households and individuals 2014". Available: http://ec.europa.eu/eurostat/statistics-explained/index.php/Information_society_statistics_-_households_and_individuals

[2] ENISA (2015). "Privacy and Data Protection by Design". Available: https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/privacy-and-data-protection-by-design

[3] European Commission. "Press release: Privacy Enhancing Technologies(PETs)". May 2, 2007.

[4] Tor Project (2015). Available: https://www.torproject.org

[5] J.A. Cowley, F.L. Greitzer, B. Woods, "Effect of network infrastructure factors on information system risk judgments", *Computers & Security*, Vol. 52, pp. 142-158, July 2015.

[6] T. Peng, C. Leckie, K. Ramamohanarao. "Survey of network-based defense mechanisms countering the DoS and DDoS problems", *ACM Computing Surveys*, Vol. 39 (1), no. 3, pp. 1-42, 2007.

[7] R. Jansen, F. Tschorsch, A. Johnson, B. Scheuermann, "The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network", *in Proc. of the 18th Symposium on Network and Distributed System Security (NDSS)*, San Diego, Ca, US, August 2014.

[8] European Police (2015), "The Internet Organised Crime Threat Assessment (iOCTA)". Available: https://www.europol.europa.eu

[9] W. Wei, F. Chen, Y. Xia, G. Jin. "A rank correlation based detection against distributed reflection DoS attacks", *IEEE Communications Letters*, Vol. 17 (1), pp. 173-175, January 2013.

[10] C. Douligeris, A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art", *Computer Networks*, Vol. 44 (5), pp. 643–666, April 2004.

[11] S. T. Zargar, J. Joshi, D. Tipper. "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks", *IEEE Communications Surveys & Tutorials*, Vol. 15 (4), pp. 2046-2069, March 2013.

[12] H. Sengar, H. Wang, D. Wijesekera, S. Jajodia. "Detecting VoIP Floods Using the Hellinger Distance", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19 (6), pp. 794-805, June 2008.

[13] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, S. Gritzalis. "DNS amplification attack revisited", *Computers & Security*, Vol. 39, part B, pp. 475-485, November 2013.

[14] W. Zhou, W. Jia, S. Wen, Y. Xiang, W. Zhou. "Detection and defense of application-layer DDoS attacks in backbone web traffic", *Future Generation Computer Systems*, vol. 38, pp. 36-46, January 2014.

[15] S. Shin, S. Lee, H. Kim, S. Kim. "Advanced probabilistic approach for network intrusion forecasting and detection", *Expert Systems with Applications*, Vol. 40, no. 1, pp. 315-322, 2013.

[16] S.M. Lee, D.S. Kim, J.H. Lee, J.S. Park. "Detection of DDoS attacks using optimized traffic matrix", *Computers & Mathematics with Applications*, Vol. 63, no. 2, pp. 501-510, September 2012.

[17] Y. Chen, X. Ma, X. Wu. "DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory", *IEEE Communications Letters*, Vol. 17, no. 5, pp. 1052-1054, May 2013.

[18] C. Callegari, S. Giordano, M. Pagano, T. Pepe. "Wave-cusum: improving cusum performance in network anomaly detection by means of wavelet analysis", *Computers & Security*, Vol. 31, no. 5, pp. 727-7J5, July 2012.

[19] Y. Cai, R.M. Franco, M. García-Herranz. "Visual latency-based interactive visualization for digital forensics", *Journal of Computational Science*, Vol. 1, no. 2, pp. 115-120, June 2010.

[20] P.A.R. Kumar, S. Selvakumar. "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems", *Computer Communications*, Vol. 36, no. 3, pp. 303-19, February 2013.

[21] I. Ozcelik, R.R. Brooks. "Deceiving entropy based DoS detection", *Computers & Security*, Vol. 48, no. 1, pp. 234-245, February 2015.

[22] M.H. Bhuyan, D. K. Bhattacharyya, J.K. Kalita. "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection", *Pattern Recognition Letters*, Vol. 51, no. 1, pp. 1-7, January 2015.

[23] A.R. Kiremire, M.R. Brust, V.V. Phoha. "Using network motifs to investigate the influence of network topology on PPM-based IP traceback schemes", *Computer Networks*, Vol. 72 (1), pp. 14-32, October 2014.

[24] N.M. Alenezi, M.J. Reed. "Uniform DoS traceback", *Computers & Security*, Vol. 45 (1), pp. 17-26, September 2014.

[25] S. Khanna, S.S. Venkatesh, O. Fatemieh, F. Khan, C.A. Gunter. "Adaptive selective verification: an efficient adaptive countermeasure to thwart DoS attacks", *IEEE/ACM Transactions on Netwowking*, Vol. 20 (3), pp. 715–728, June 2012.

[26] S. Bhatia, D. Schmidt, G. Mohay, A. Tickle. "A framework for generating realistic traffic for Distributed Denial-of-Service attacks and Flash Events", *Computers & Security*, Vol. 40, no. 1, pp. 95-107, February 2014.

[27] The CAIDA UCSD (2015), "DDoS Attack 2007 Dataset". Available: http://www.caida.org/data/passive/ddos-20070804_dataset.xml

[28] The CAIDA UCSD (2015), "Anonymized Internet Traces 2008". Available: http://www.caida.org/data/passive/passive_2008_dataset.xml

[29] The CAIDA UCSD Anonymized Internet Traces 2014 (2015), Available: http://www.caida.org/data/passive/passive_2014_dataset.xml

[30] R. Dingledine, N. Mathewson, P. Syverson. "Tor: the second-generation onion router", *in Proc. of the 13th conference on USENIX Security Symposium*, San Diego, CA, US, Vol. 13, August 2004.

[31] T. Dierks, E. Rescorla. "The Transport Layer Security (TLS) Protocol", *IETF RFC 5248*, August 2008.

[32] A. Freier, P. Karlton. "The Secure Sockets Layer (SSL) Protocol Version 3.0", *IETF RFC 6101*, August 2011.

[33] W. Diffie, M. Hellman. "New directions in cryptography", *IEEE Transactions on Information Theory*, Vol. 22 (6), pp. 644-654, November 1976.

[34] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, p. Mittal. "RAPTOR: Routing Attacks on Privacy in Tor", *in Proc. of the 24thconference on USENIX Security Symposium*, Washington, DC, US, August 2015.

[35] R. Pries, W. Yu, X. Fu, W. Zhao. "A New Replay Attack Against Anonymous Communication Networks", *in Proc. of the IEEE International Conference on Communications (ICC'08)*, Beijing, Chine, pp. 1578-1582, May 2008.

[36] [9] N. Danner, S. Defabbia-Kane, D. krizanc, M. Liberatore. "Effectiveness and detection of denial-of-service attacks in Tor", *ACM Transactions on Information and System Security (TISSEC)*, Vol. 15 (3), pp. 11-25, November 2012.

[37] C.E. Shannon. "A mathematical theory of communication", *Bell system technical journal*, Vol. 27, pp.397-423, 1948.

[38] A. Rènyi. "On measures of entropy and information", *in Proc. of the 4th Berkeley symposium on mathematical statistics and probability*, Berkeley, CA, US, Vol. 1, 547-561, June 1961.

[39] G.E.P. Box, G.M. Jenkins. "Time Series Analysis: Forecasting and Control", *Holden Dayr*, San Francisco, California, 1976.

[40] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, B. Stiller. "An overview of IP flow-based intrusion detection", *IEEE Communications Surveys & Tutorials*, Vol. 12(3), pp. 343-356, July 2010.

[41] A. Maravall, D. Pérez. "Applying and interpreting model-based seasonal adjustment", *The Euro-Area Industrial Production Series*, N. 1116, 2011.

[42] JSON (2015). Available: json.org

[43] Chutney (2015). Available: https://gitweb.torproject.org/chutney.git

[44] Tor Metrics (2015). Available: https://metrics.torproject.org/