Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado
Facultad de Ingeniería de Sistemas e Informática
Unidad de Posgrado

Contribuciones para la Detección de Ataques Distribuidos de Denegación de Servicio (DDoS) en la Capa de Aplicación

TESIS

Para optar el Grado Académico de Doctor en Ingeniería de Sistemas e Informática

AUTOR

Silvia Jeaneth BRAVO MULLO

ASESOR

Dr. David Santos MAURICIO SÁNCHEZ

Lima, Perú

2019

Referencia bibliográfica

Mauricio, S. (2019). *Contribuciones para la Detección de Ataques Distribuidos de Denegación de Servicio (DDoS) en la Capa de Aplicación*. [Tesis de doctorado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Silvia Jeaneth Bravo Mullo
Tipo de documento de identidad	DNI
Número de documento de identidad	EC/0502437122
URL de ORCID	0000-0002-0682-5260
Datos de asesor	
Nombres y apellidos	David Santos Mauricio Sánchez
Tipo de documento de identidad	DNI
Número de documento de identidad	PE/06445495
URL de ORCID	0000-0001-9262-626X
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Hugo Froilán Vega Huerta
Tipo de documento	DNI
Número de documento de identidad	PE/06147737
Miembro del jurado 1	
Nombres y apellidos	Erik Alex Papa Quiroz
Tipo de documento	DNI
Número de documento de identidad	PE/10451642
Miembro del jurado 2	
Nombres y apellidos	Augusto Ernesto Bernuy Alva
Tipo de documento	DNI
Número de documento de identidad	PE/10321499
Datos de investigación	
Línea de investigación	Inteligencia Artificial

Grupo de investigación	No Aplica
Agencia de financiamiento	Sin financiamiento.
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Surco Calle: Av. Óscar R. Benavides 5737, Callao 07006, Perú Latitud: -11.77453 Longitud: -76.98543
Año o rango de años en que se realizó la investigación	Marzo 2015 – Junio 2019 2015 - 2019
URL de disciplinas OCDE	Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04 Communication engineering and systems https://purl.org/pe-repo/ocde/ford#2.02.00

Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE DOCTOR EN INGENIERÍA DE SISTEMAS E INFORMÁTICA

En la Ciudad Universitaria, a los cinco (05) días del mes de junio del 2019, siendo las 19:10 *horas, se reunieron en el Auditorio de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos, el Jurado de Tesis conformado por los siguientes docentes:*

Dr. Hugo Froilán Vega Huerta (Presidente)
Dr. Erik Alex Papa Quiroz (Miembro)
Dr. Augusto Ernesto Bernuy Alva (Miembro)
Dr. David Santos Mauricio Sánchez (Asesor)

Se inició la Sustentación invitando a la candidata a Doctor ***Silvia Jeaneth Bravo Mullo****, para que realizara la exposición oral y pública de la tesis para optar el Grado Académico de Doctor en Ingeniería de Sistemas e Informática, siendo la Tesis intitulada:*

"Contribuciones para la Detección de Ataques Distribuidos de Denegación de Servicio (DDoS) en la Capa de Aplicación"

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por la graduanda; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

18 Muy Bueno

Por tanto el Presidente del Jurado, de acuerdo al Reglamento General de Estudios de Posgrado, otorga a la Magister ***Silvia Jeaneth Bravo Mullo*** *el Grado Académico de Doctor en Ingeniería de Sistemas e Informática.*

Siendo las 20:10 *horas, el Presidente del Jurado de Tesis da por concluido el acto académico de Sustentación de Tesis.*

Dr. Hugo Froilán Vega Huerta
(Presidente)

Dr. Erik Alex Papa Quiroz
(Miembro)

Dr. Augusto Ernesto Bernuy Alva
(Miembro)

Dr. David Santos Mauricio Sánchez
(Asesor)

Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado

Unidad de Posgrado

INFORME DE EVALUACION DE ORIGINALIDAD

Nº 004-UPG-VDIP-FISI-2019

1. *Autoridad Académica que emite el Informe de Originalidad:*	*Directora(e) de la Unidad de Posgrado*
2. *Apellidos y Nombres de la autoridad académica:*	*Mamani Rodriguez Zoraida Emperatriz*
3. *Operador del programa informático de similitudes:*	*Mamani Rodriguez Zoraida Emperatriz*
4. *Documento evaluado:*	*Tesis para Posgrado* *Titulo: "Contribuciones para la Detección de Ataques Distribuidos de Denegación de Servicio (DDoS) en la Capa de Aplicación"*
5. *Autor del documento:*	*Silvia Jeaneth Bravo Mullo*
6. *Fecha de recepción de documento*	*20/02/2019*
7. *Fecha de aplicación del programa detector de similitudes:*	*21/02/2019*
8. *Software utilizado:*	*Turnitin*
9. *Configuración del programa detector de similitudes:*	*i. Excluye textos entrecomillados: SI* *ii. Excluye biografías: SI* *iii. Excluye cadenas menores a 40 palabras: SI* *iv. Otro criterio (especificar): NO*
10. *Porcentaje de similitudes según programa detector de similitudes*	*Diez por ciento (10%)*
11. *Fuentes originales de las similitudes encontradas*	*Se adjuntan en tres (03) fojas al presente informe*
12. *Observaciones:*	*Ninguna*
13. *Calificación de originalidad* *i. Documento cumple criterios de originalidad, sin observaciones.* *ii. Documento cumple criterior de originaldiad, con observaciones.* *iii. Documento no cumple criterios de originalidad.*	*Documento cumple criterio de originaldiad, sin observaciones.*
14. *Fecha del Informe:*	*21/02/2019*

Mg. Zoraida Mamani Rodriguez
Directora (e) de la UPG-FISI

AGRADECIMIENTO

Agradezco al Dr. David Mauricio por haber guiado mi trabajo doctoral durante este tiempo. Además, agradezco al Dr. Ángel Hernández por sus valiosos aportes para la conclusión exitosa de este trabajo. Especialmente, agradezco a mi familia por apoyarme en mis estudios doctorales, pero, sobre todo por estar conmigo en los momentos más difíciles, gracias por no abandonarme. Finalmente, agradezco a la Universidad Nacional Mayor de San Marcos y su excelentísimo personal docente.

Mg. Silvia Jeaneth Bravo Mullo

Doctorando

DEDICATORIA

- A Dios, a mi padre y mi hermana que están a su lado
- A mi familia

Mg. Silvia Jeaneth Bravo Mullo

Doctorando

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

Los ataques distribuidos de denegación de servicio (DDoS) son uno de los mayores problemas que enfrenta Internet. Por lo tanto, la detección de los ataques DDoS es de gran importancia para los especialistas en seguridad informática. Con el fin de comprender su funcionamiento, en este trabajo se analizaron seis aspectos sobre la detección de ataques DDoS: técnicas, variables, herramientas, ubicación de implementación, punto en el tiempo y precisión de detección. Este análisis permitió realizar una contribución útil al diseño de una estrategia adecuada para neutralizar estos ataques.

En los últimos años, estos ataques se han dirigido hacia la capa de aplicación. Este fenómeno se debe principalmente a la gran cantidad de herramientas para la generación de este tipo de ataque. Por ello, además, en este trabajo se propone una alternativa de detección basada en el dinamismo del usuario web. Para esto, se evaluaron las características del dinamismo del usuario extraídas de las funciones del mouse y del teclado.

Finalmente, el presente trabajo propone un enfoque de detección de bajo costo que consta de dos pasos: primero, las características del usuario se extraen en tiempo real mientras se navega por la aplicación web; en segundo lugar, cada característica extraída es utilizada por un algoritmo de orden (O1) para diferenciar a un usuario real de un ataque DDoS. Los resultados de las pruebas con las herramientas de ataque LOIC, OWASP y GoldenEye muestran que el método propuesto tiene una eficacia de detección del 100% y que las características del dinamismo del usuario de la web permiten diferenciar entre un usuario real y un robot.

Keywords: Ataques distribuidos de denegación de servicio, DDoS, método de detección, capa de aplicación, dinamismo del usuario

ABSTRACT

Distributed denial of service (DDoS) attacks are one of the biggest problems facing the Internet. Therefore, the detection of DDoS attacks is of great importance for computer security specialists. In order to understand its operation, in this work we analyzed six aspects about the detection of DDoS attacks: techniques, variables, tools, implementation location, point in time and detection precision. This analysis allowed us to make a useful contribution to the design of an adequate strategy to neutralize these attacks.

In recent years, these attacks have been directed towards the application layer. This phenomenon is mainly due to the large number of tools for the generation of this type of attack. Therefore, in addition, this work proposes an alternative detection based on the dynamism of the web user. For this, the characteristics of the dynamism of the user extracted from the mouse and keyboard functions were evaluated.

Finally, this paper proposes a low-cost detection approach that consists of two steps: first, the user's characteristics are extracted in real time while browsing the web application; second, each extracted feature is used by an order algorithm (O1) to differentiate a real user from a DDoS attack. The results of the tests with the attack tools LOIC, OWASP and GoldenEye show that the proposed method has a detection efficiency of 100% and that the characteristics of the dynamism of the user of the web allow to differentiate between a real user and a robot.

Keywords: Distributed denial of service attacks, DDoS, detection method, application layer, user dynamism

CAPÍTULO I. INTRODUCCIÓN

1.1 Situación problémica

Denegación de Servicio (DoS) son ataques informáticos considerados una amenaza para la seguridad de Internet y se han convertido en un problema que ha sido estudiado desde su aparición en el año 1980 (Zargar et al., 2013). Este tipo de ataques constituyen acciones ilegítimas por medio de las cuales un atacante interrumpe los recursos o servicios de un sistema (Waguih, 2013).

Posteriormente, en el año 1999 se informó sobre la aparición de un ataque DoS más sofisticado denominado ataques distribuidos de denegación de servicio (DDoS) (Criscuolo, 2000). Este ataque involucra a dos o más ordenadores, que se pueden encontrar localizados en diversas partes del mundo, y son ejecutados por el atacante (Ni et al., 2014). Algunos autores, tales como Zargar et al. (2013) y Choi et al. (2014), coinciden en que el principal problema en la detección de este ataque está en no poder diferenciar los flujos legítimos de los flujos de ataque, lo que origina altas tasas de falsos positivos y negativos en los métodos de detección empleados.

Zargar et al. (2013) clasifica a los ataques DDoS de acuerdo al nivel de protocolo al desde el cual se dirige el ataque, siendo estos a nivel de red y a nivel de aplicación. En el nivel de red los atacantes se enfocan en interrumpir la conectividad del usuario legítimo al agotar el ancho de banda de la red de la víctima. Mientras que los ataques

enfocados a la capa de aplicación se centran en interrumpir los servicios legítimos del usuario al agotar los recursos del servidor, por ejemplo, sockets, CPU, memoria, etc.

La detección de ataques DDoS es uno de los mayores problemas que enfrenta la arquitectura de seguridad de la red. Por lo tanto, se ha convertido en un importante factor de estudio en el campo de la seguridad informática. Un ataque DDoS ocurre cuando un atacante coordina sus ataques usando varias máquinas, llamadas zombies, hacia un objetivo o servidor específico. El objetivo del atacante es realizar solicitudes masivas a la máquina víctima para saturarlo y dejar de atender las solicitudes de usuarios reales.

Para contrarrestar este tipo de ataque, se han propuesto varios mecanismos de detección, tanto a nivel de red como a nivel de aplicación tal y como se puede observar en la Tabla 1. La tasa de detección más alta obtenida hasta la fecha es del 99,99% y se ha logrado mediante la implementación de un método de nivel de red (Jia et al., 2017). El conjunto de datos utilizado en ese trabajo es KDD CUP 1999, del cual se extrajeron registros de conexión entre ataques DDoS y usuarios reales. Por otro lado, en los métodos implementados a nivel de capa de aplicación, la mejor tasa de detección obtenida es 98.31% (Johnson et al., 2016), para la implementación del método propuesto se empleó el cálculo de entropía.

Tabla 1. **Mecanismos en la capa de red y aplicación**

Level	Reference
Network	Al-Duwairi et al. (2006); Al-Wang et al. (2014); Anurekha et al. (2012); Beak et al. (2007); Chen et al. (2005); Chen et al. (2006); Chen et al. (2008); Chen et al. (2007a); Chen et al. (2007b); Chen et al. (2013a); Chen et al. (2013b); Chonka et al. (2009); Doron et al. (2011); Duwairi et al. (2013); François et al. (2012); Kang et al. (2013); Kang et al. (2014); Kim et al. (2006); Kulkarni et al. (2006); Kumar et al. (2011); Kumar et al. (2013), Lee et al. (2005), Lee et al.

	(2008), Lee et al. (2012), Li et al. (2005); Liu et al. (2011); Lu et al. (2009); Luo et al. (2013); Luo et al. (2014); Ma et al. (2014); Meenakshi et al. (2007); Mirkovic et al. (2005); Rahmani et al. (2012); Seo et al. (2013); Spyridopoulos et al. (2013); Yan et al. (2009); Sachdeva et al. (2014); Udhayan et al. (2013); Varalakshmi et al. (2013); Wang et al. (2007); Wang et al. (2012); Wang et al. (2014); Wu et al. (2013); Xiang et al. (2011); Xiao et al. (2006); Xiao et al. (2015); Yaar et al. (2005); Yau et al. (2005); Zhang et al. (2012); Zhenwei et al. (2011)
Application	Dick et al. (2016); Giralte et al. (2013); Huang et al. (2014); Johnson Singh et al. (2016); Ranjan et al. (2009); Saravanan et al. (2016); Xie et al. (2009); Zhou et al. (2014); Zolotukhin et al. (2016)

Fuente. Autor

Los mecanismos de detección, en su mayoría, enfocan sus esfuerzos en la capa de red. Sin embargo, actualmente el mayor número de ataques se han dirigido a la capa de aplicación, porque son fáciles de ejecutar debido a la gran cantidad de software existente (Xiang et al., 2011), Wu et al., 2015) y más difíciles de detectar porque son solicitudes ilegítimas que se camuflan como solicitudes de usuarios reales.

Actualmente, todos los métodos de detección de ataques en la capa de aplicación se basan en características de la solicitud, su eficiencia depende de ellos. Sin embargo, ningún método de detección contempla la interacción del usuario con el sistema, la misma puede diferenciar entre un ser humano y un robot (Wu et al., 2013)).

Verisign en su reporte del último trimestre del año 2018, menciona las principales industrias objetivo de este tipo de ataque. Siendo los servicios de instituciones financieras las más afectadas con un 43% del total de ataques. Los servicios IT, Cloud y SAAS son los segundos más afectados con el 37%. Finalmente, las industrias de entretenimiento y media son las siguientes más afectadas con el 20% (Verisign, 2018).

En el estudio anual Riesgos de seguridad de TI realizado por Kaspersky Lab, en abril de 2017, más de 5.200 pequeñas, medianas y grandes empresas en 29 países recibieron problemas de seguridad e incidentes de seguridad cibernética. Este estudio muestra que los costos de los ataques DDoS están aumentando significativamente. Estas empresas ahora pueden esperar enfrentar costos de 123,000 dólares estadounidenses por ataque. En el caso de las grandes empresas, un ataque DDoS puede incluso golpearlas con un daño financiero de 2,3 millones de dólares en promedio. Sin embargo, a pesar de estos altos riesgos financieros, el estudio muestra que solo el 19% de las empresas solicitadas tienen una solución DDoS disponible (Bender, 2018).

Las violaciones de datos pueden tener efectos dañinos duraderos para cualquier negocio, independientemente de su industria. Un informe de 2017 realizado por el Ponemon Institute e IBM reveló que el costo total promedio de una violación de datos en los Estados Unidos alcanzó un récord de $ 7.35 millones, un aumento del 5 por ciento respecto al año anterior (IBM, 2017).

Por lo tanto, la presente investigación se centra en nuevas características basadas en la interacción del usuario con el sistema para la detección de ataques DDoS. Así mismo, se propone un método simple y de bajo costo basado en las características del dinamismo del usuario para el proceso de detección de ataques DDoS en la capa de aplicación. Para ello, se emplean las pulsaciones de teclas, la dinámica del ratón y la interacción con la interfaz gráfica de usuario (GUI) (Abramson et al., 2013)) para la identificación de usuarios reales.

El método propuesto ha sido validado en un caso de estudio para evaluar su eficiencia. Para esto, se implementó un algoritmo que detecta la interacción del usuario y el sistema. Fue probado en un sistema web en tiempo real. El sistema tiene una arquitectura de tres capas para implementar la interfaz de usuario y el algoritmo de detección. Los ataques se generaron utilizando las herramientas LOIC, OWASP y GoldeEye para provocar ataques de inundación. Los resultados muestran un 100% de eficiencia en la detección de ataques DDoS.

1.2 Formulación del problema

El desafío principal en la detección de ataques DDoS en la capa de aplicacióm está en diferenciar las solicitudes reales de los ataques DDoS. Este desafío se basa en que actualmente los atacantes simulan las características de solicitudes reales. Para lograr esta simulación los atacantes envían solicitudes de baja frecuencia en determinados intervalos de tiempo emulando la solicitud de un usuario, lo cual provoca que el ataque pueda eludir los mecanismos de detección (Ranjan et al., 2006).

Otro desafío en el problema de la detección de ataques DDoS en la capa de aplicación es distinguir estos ataques de las grandes solicitudes de tráfico legítimo que se producen en sitios web populares cuando miles de solicitudes acceden a los servidores web en períodos de tiempo relativamente corto. Las solicitudes masivas son bastante similares con los ataques DDoS en términos de anomalía de red y fenómeno de tráfico. Incluso pueden causar que el sitio web o el objetivo ralentice su servicio para los usuarios o incluso que se cierre temporalmente debido al aumento significativo del tráfico (Bulajoul, 2013).

Además, dado que la mitigación del daño de un ataque DDoS depende de su detección oportuna, se supone que el proceso de detección tiene lugar en un modo en tiempo real. Por esta razón, la construcción del modelo normal de comportamiento del usuario y la detección de una actividad anómala requieren cantidades considerables de memoria y recursos informáticos. Por lo tanto, el problema de la gestión adecuada de estos recursos es uno de los desafíos importantes a la hora de diseñar un sistema de detección de ataques DDoS (Zolotukhin, 2016).

1.2.1. Problema General

¿Cómo se puede diferenciar un usuario real de un ataque DDoS para detectar este tipo de ataque de manera eficiente?

1.2.2. Problemas específicos

- ¿Qué características permiten diferenciar un usuario real de un ataque DDoS?
- ¿Cómo se debería capturar las características necesarias para la detección de ataques DDoS en tiempo real?
- ¿Mediante que mecanismo se deberían implementar las características establecidas para la detección de ataques DDoS con el fin de lograr mayor eficiencia de detección?
- ¿Cómo debería estar construido el entorno donde se va a implementar el mecanismo de detección de ataques DDoS?

1.3. Motivación

Entre uno de los aspecto más importantes en la detección de ataques DDoS en la capa de aplicación son las características empleadas (Zargar et al., 2013). Debido a que los métodos de detección de ataques utilizan características definidas, su eficiencia depende de ellas. Sin embargo, ningún método de detección contempla el dinamismo del usuario con el sistema (Chen et al., 2016). Los mecanismos de detección de ataques DDoS en la capa de aplicación emplean características del flujo de datos, los mismos no permiten diferenciar claramente un ataque DDoS y un usuario real. Debido a la falta de criterios concretos para esta diferenciación, las características basadas en el flujo de datos conllevan tasas considerables de falsos positivos y negativos (Ranjan et al., 2006).

Por lo antes mencionado, estos mismos mecanismos tienen problemas de detección al producirse una gran cantidad de usuarios reales ingresando al sistema. Este evento

denominado también como flash crowd (Oumokinou et al., 2009), puede ser detectado como un ataque DDoS lo cual provocaría que los usuarios legítimos tengan acceso a los servicios del sistema. Todas estas falencias en los sistemas de detección actuales evidencian la necesidad de proponer nuevas características basadas en el dinamismo del usuario, las cuales permiten diferenciar entre un ser humano y un robot (Dickinson et al., 2013). En esta investigación se emplean estas nuevas características basadas en el dinamismo del usuario con el sistema, específicamente su interacción con los periféricos.

1.4 Justificación teórica

El presente trabajo tiene su sustento en los siguientes aspectos:

- Las estadísticas de las organizaciones dedicadas a contrarrestar los ataques DDoS, como Verisign, Netscout Arbor y Akamai Community, coinciden en que la cantidad y el tamaño de dichos ataques DDoS van en aumento. Verisign también notó un aumento en el tamaño promedio de ataques de 7.6 Gbps, 850% más que en el tercer trimestre de 2017 (Verisign, 2017).
- Del mismo modo, los datos de Netscout Arbor muestran que los proveedores de servicio que experimentaron más de 500 ataques por mes aumentaron a 17% del 15% en 2016 (Netscout Arbor, 2017). Además, los datos recopilados por Akamai Community, los ataques DDoS aumentaron significativamente a partir del cuarto trimestre de 2017, con un aumento del 115% en el tercer trimestre de 2017. También mostraron un aumento del 115% en los ataques de nivel de aplicación en el trimestre (Akamai Community, 2017).
- El costo de los ataques DDoS varían en función de la industria, el tamaño de la empresa, el presupuesto operativo de seguridad. Entonces, para algunos negocios

financieros y basados en la web, los ataques DDoS pueden generar millones de dólares en daños por hora. La cantidad promedio de tiempo de inactividad después de un ataque DDoS es de 54 minutos y el costo promedio de cada minuto de tiempo de inactividad es de $22,000. Sin embargo, el costo puede variar desde $1 a $100,000 por minuto de tiempo de inactividad (World Economic Forum, 2018).

- Una encuesta exhaustiva de seguridad de más de 370 administradores de redes y seguridad de más de 14 industrias informó que los encuestados experimentaron en promedio 4.5 ataques DDoS por año y una duración promedio de ataque de 8.7 horas. La Tabla 2 calcula el costo de tres años de los escenarios descritos en el presente documento utilizando la información provista por la encuesta. En este escenario los ingresos de la compañía a 12 meses fueron de $ 35,000,000 de 152,174 clientes en línea con una compra promedio de $ 230 por cliente.

Tabla 2. **Costo estimado de ataque DDoS**

	Minorista en línea - Escenario Ataque DDoS
Costo de incidente simple (8 horas)	$98,72
Costo estimado de tres años	Costo de incidente simple x 13.5 = $1,332,770
Costo estimado mensualmente	$36,743

Fuente. NSFOCUS

En la Tabla 2 se observa el costo producido por un ataque DDoS a una compañía promedio que maneja su negocio a través de internet. El costo estimado mensual es de $36,743 lo cual indica una pérdida significativa en los activos de cualquier compañía.

1.5. Justificación práctica

El presente trabajo presenta un método de detección de ataques DDoS eficiente y de bajo costo. El mismo emplea las ocho características del dinamismo del usuario web, las cuales resultan novedosas, debido a que ningún otro trabajo similar las ha propuesto. Estás características revelan de manera única la dinámica entre el usuario y el sistema, de tal forma que permite diferenciar una solicitud realizada por un humano en comparación a un ataque.

Además, se propone un algoritmo eficiente para capturar las características del dinamismo del usuario web. Este algoritmo extrae información cada vez que el usuario realiza una acción en el sistema en tiempo real. Luego la información recogida será analizada por un algoritmo de clasificación que se ejecuta cuando se realiza una solicitud al sistema, haciendo que se pueda realizar la detección en tiempo real.

Adicionalmente, se propone un algoritmo eficaz de clasificación que emplea las características del dinamismo del usuario web para la detección de ataques DDoS. Este algoritmo es de bajo costo computacional en comparación con la eficiencia que presenta. La detección se realiza de manera rápida debido a que analiza cada solicitud que contiene la información recolectada en el algoritmo de captura de características.

Finalmente, se presenta un método de detección de ataques DDoS ágil que permite identificar un usuario real de un ataque. Para ello, se emplean ocho nuevas características del dinamismo del usuario web y dos algoritmos de alto rendimiento y bajo costo que captura las características y las clasifica respectivamente. Los resultados de las simulaciones realizadas muestran una eficiencia del 100% para detectar y mitigar ataques DDoS.

1.6. Objetivo

1.6.1 Objetivo General

Contribuir con la detección de ataques DDoS en la capa de aplicación mediante la evaluación de nuevas características del dinamismo del usuario en un método de detección inteligente, eficiente y ágil.

1.6.2 Objetivo Específicos

O1: Analizar las características y formas de mitigación de los métodos de predicción y detección de ataques DDoS en la capa de aplicación para conocer sobre su funcionamiento.

O2: Determinar nuevas características que serán empleadas por el método de detección propuesto empleando algoritmos de evaluación.

O3: Diseñar un mecanismo ágil que permita detectar ataques DDoS en la capa de aplicación empleando las características del dinamismo del usuario.

1.7. Hipótesis

Un método de detección basado en las características del dinamismo del usuario web permitirá detectar ataques DDoS en forma eficiente y ágil.

1.7.1. Hipótesis específicas

- Las características del dinamismo del usuario web permiten identificar un usuario real de un ataque DDoS
- Las características del usuario real pueden ser capturadas mediante un algoritmo en tiempo real permitiendo la eficiencia del mismo
- Mediante el empleo de un algoritmo de clasificación se podrá implementar las características del dinamismo del usuario web para detectar ataques DDoS

1.8. Identificación de variables

1.8.1. Variable independiente

Método de detección basado en las características del dinamismo del usuario web

1.8.2. Variable dependiente

Detección de ataques DDoS

1.9. Operacionalización de variables

Tabla 3. **Operacionalización de variables**

Variable	Tipo de variable	Definición conceptual	Dimensión	Indicador	Descripción del indicador
Método de detección basado en las características del dinamismo del usuario	Independiente	Procesos secuenciales para la detección de ataques distribuidos de denegación de servicio en la capa de aplicación donde se extraen las características de la interacción entre usuario y sistema	Tecnología	Método de detección	Técnicas informáticas para la detección de ataques distribuidos de denegación de servicio
			Análisis de datos	Dinamismo del usuario	Extracción y análisis de datos (características) para su empleo en el método de detección
Detección de ataques DDoS	Dependiente	Proceso para diferenciar un usuario real de ataque DDoS	Seguridad informática	Detección de ataques	Aporte para la seguridad de la información evitando intrusiones no autorizadas

Fuente. Autor

1.10 Propuesta

La presente investigación propone un método de detección de ataques DDoS basado en las características del dinamismo del usuario. Para ello, se plantean ocho nuevas características del dinamismo del usuario. Además, se plantean dos algoritmo de clasificación en tiempo real, el primero para la extracción de las características y el segundo para la clasificación de ataques y usuarios reales.

Figura 1. **Esquema del método de detección de ataques DDoS en la capa de aplicación**

La Figura 1 muestra el esquema del método de detección de ataques DDoS en la capa de aplicación propuesto en esta investigación. En la misma se observan cuatro pasos principales que intervienen en el proceso. El primer paso es la interface, la misma

permite determinar la interacción del usuario con el sistema, al momento que el usuario interactúa con ella. La segunda es la extracción, la cual permite ir registrando las acciones realizadas por el usuario en un banco de información hasta que el usuario realica una solicitud al sistema. La tercera es la detección, misma que determinará si un usuario es real o un ataque mediante la verificación del dinamismo registrado por el usuario. Por último, la mitigación corta las solicitudes generadas por usuarios identificados como robots en el paso anterior, permitiendo que únicamente aquellos usuarios que hayan tenido un registro de dinámica con el sistema continúen realizando solicitudes.

1.11 Organización de la tesis

La presente tesis esta organizada en seis capítulos, y que se sintetizan a seguir:

- En el Capítulo dos se observa un background sobre ataques DoS y sus tipos, definimos el ataque DDoS y su clasificación. Además se observan elementos que intervienen en la ejecución de este tipo de ataque.
- El Capítulo tres trata sobre el estado del arte de los métodos de detección de ataques DDoS.
- En el Capítulo cuatro se presentan las características empleadas en la detección de ataques DDOS.
- En el Capítulo cinco se muestra la propuesta de método de detección de ataques DDoS basado en las características del dinamismo del usuario web en la capa de aplicación
- El Capítulo seis muestra las conclusiones y trabajos futuros

CAPÍTULO II. MARCO TEÓRICO

2.1. Marco Filosófico o epistemológico de la investigación

2.1.1. Denegación de servicio (DoS)

Los ataques de denegación de servicio (DoS) de Internet funcionan inundando algunos recursos limitados en Internet, impidiendo así que los usuarios legítimos accedan a ese recurso. Los objetivos incluyen el ancho de banda de los enlaces de acceso y otros cuellos de botella de red, y también los recursos informáticos y de memoria en servidores y clientes. Si bien esto podría solucionarse fácilmente mediante la creación de un software más sólido, prácticamente todos los dispositivos conectados a Internet tienen cierta vulnerabilidad a un ataque, tal como se muestra en la Figura 2.

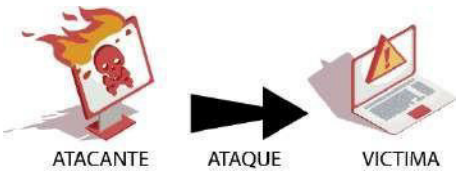

Figura 2. **Ataque de denegación de servicio**

2.1.2. Ataque Distribuido de denegación de servicio (DDoS)

Los ataques DDoS se han convertido en un gran problema para los usuarios de sistemas informáticos conectados a Internet. Los atacantes DDoS secuestran sistemas de víctimas secundarias usándolos para librar un ataque coordinado a gran escala contra los principales sistemas de víctimas. A medida que se desarrollan nuevas contramedidas para prevenir o mitigar los ataques DDoS, los atacantes desarrollan constantemente nuevos métodos para eludir estas nuevas contramedidas.

Como se mencionó anteriormente, las botnets son los mecanismos dominantes que facilitan los ataques de inundación DDoS en las redes o aplicaciones de las computadoras. La mayoría de los ataques de inundación DDoS de capa de aplicación más recientes y más problemáticos han empleado botnets.

De acuerdo con Peng et al. (2015), hay dos razones principales que hacen que el desarrollo de un mecanismo de defensa DDoS sea aún más desafiante cuando los atacantes emplean zombis para lanzar ataques de inundación DDoS. En primer lugar, una gran cantidad de zombis involucrados en el ataque facilita a los atacantes para hacer que los ataques sean más grandes en escala y más destruptivos. En segundo lugar, las direcciones IP de los zombis generalmente son falsificadas bajo el control del atacante, lo que hace que sea muy difícil rastrear el tráfico de ataque incluso a los zombis, así se muestra en la Figura 3.

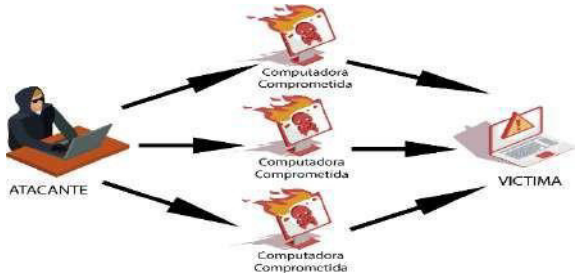

Figura 3. **Ataque distribuido de denegación de servicio (DDoS)**

2.1.3 Elementos empleados para ejecutar un ataque DDoS

El atacante con el fin de llevar a cabo un ataque adecuado, emplea los siguientes elementos:

a) Atacante real: La persona encargada de llevar a cabo el ataque

b) Control total: El programa de control maestro funciona como interfaz entre el atacante real y los zombies. Además instruye a los zombies a atacar a la víctima.

c) Esclavos o zombies: Son sistemas comprometidos y estos sistemas son responsables de generar el tráfico. Para ello, tienen instalado un software específico que está siendo controlado por un atacante real a través del control maestro.

d) Victima: Máquina atacada

2.1.4. Causas para darse ataques ddos

Los ataques DDoS pueden distribuirse de manera amplia y fácil en Internet, debido, a la disponibilidad de herramientas de ataque y la potencia de estas herramientas para generar tráfico de ataque (Gupta et al., 2010). Varias razones que crean oportunidades para que los atacantes usen herramientas de ataque fácilmente y lancen un ataque exitoso son:

a) La seguridad de Internet es altamente interdependiente: la susceptibilidad de los ataques DDoS depende de la seguridad global de Internet en lugar de la seguridad de la víctima.

b) Los recursos de Internet son limitados: Internet tiene recursos limitados que pueden ser consumidos por un número limitado de usuarios.

c) La responsabilidad no se impone: con mecanismos como la suplantación de identidad, el autor puede ocultar su identidad real y, por lo tanto, no se puede juzgar la fuente real del ataque.

d) El control se distribuye: dado que Internet se distribuye por una red se ejecuta según las políticas y regulaciones particulares definidas, es casi imposible implementar un determinado mecanismo de seguridad global y, además, debido a problemas de privacidad, a veces es casi imposible investigar el comportamiento de la red cruzada.

e) Núcleo simple y borde complejo: uno de los principios de diseño es que Internet debe mantener las redes centrales simples e introducir cualquier complejidad en los hosts finales (Mirkovic et al., 2003; Peng et al., 2007). Por lo tanto, los enrutadores centrales no realizan las comprobaciones de autenticación necesarias. El vacío de las comprobaciones de autenticación a nivel de red fomenta los intentos no autorizados no deseados.

f) Enrutamiento de múltiples rutas: el enrutamiento de múltiples rutas dificulta la autenticación, por lo tanto, puede fomentar actividades no autorizadas. El enrutador intermedio enruta el paquete IP desde el origen hasta el destino y no tiene forma de saber si el paquete IP que está reenviando es el paquete legítimo o falsificado (Peng et al., 2007).

2.1.5. Motivación DDoS

Un ataque DDOS suele estar motivado por varias razones. Se han clasificado estos ataques DDoS según la motivación de los atacantes en siete categorías principales:

a) Por factores financieros: los ataques lanzados para obtener ganancias financieras son a menudo, los más peligrosos y difíciles de detener. Estas son las principales preocupaciones de las corporaciones.

b) Rendimiento lento de la red: el atacante lanza un ataque para bloquear los recursos del sistema víctima, lo que ralentiza el rendimiento del sistema y la red.

c) Venganza: los atacantes de este tipo normalmente tienen habilidades técnicas más bajas y son individuos frustrados, como respuesta a una injusticia percibida.

d) Por motivos ideológicos: Los atacantes en esta categoría están inspirados en sus creencias ideológicas para atacar a sus objetivos. Esta categoría es actualmente uno de los principales incentivos para que los atacantes lancen ataques DDoS.

e) Desafío intelectual: en este caso, ataca los sistemas específicos para experimentar y aprende cómo lanzar varios ataques. Por lo general, son jóvenes entusiastas de la piratería que quieren mostrar sus competencias.

f) Indisponibilidad del servicio. En este atacante se sobrecargan los servicios ofrecidos por el sistema víctima a través de tráfico no deseado o falso.

g) Guerra cibernética: los atacantes de esta clase normalmente pertenecen a las organizaciones militares o terroristas de un país y tienen motivaciones políticas para atacar a una amplia gama de secciones críticas de otro país.

2.2. Antecedentes de investigación

2.2.1. Prevención contra ataques DDOS

Para prevenir un ataque DDoS se requiere de la implementación de un conjunto de defensas, prácticas y configuraciones antes, durante y después del ataque, con el objetivo de reducir el impacto de tal ataque. A continuación se describen posibles categorías que se pueden mencionar para clasificar la mayoría de los enfoques de prevención DDoS.

2.2.1.1 Sobre aprovisionamiento. Este enfoque se basa en la prevenir un ataque en un sitio web mediante la preparación anticipada de mucho más tráfico del que se esperaría durante el funcionamiento normal del sistema. En su momento fue un método

utilizado regularmente, sin embargo, ahora con el aumento en el tamaño de los ataques DDoS, esto claramente no es sostenible (Ardor Network, 2017).

2.2.1.2 Modificación de los algoritmos de programación. El objetivo de este enfoque es favorecer el tráfico legítimo sobre el tráfico malicioso. Ranjan et al. (2006) propone un mecanismo para asignar tráfico sospechoso. Este contexto evalúa el valor de sospecha para su decisión mediante programación. El mecanismo de asignación de sospecha utiliza las sesiones de llegadas y perfiles de carga de trabajo como entrada. En particular, las pruebas han demostrado que las políticas conscientes de sospecha reducen el impacto en el tiempo de respuesta de un ataque DDoS con políticas adecuadas de sospecha.

2.2.1.3 Protección de recursos. Para proteger los recursos de un sistema, es necesario hacer un seguimiento de los mismos (ancho de banda de la red, tiempo de procesador) (Mirkovic, 2005). Estos recursos se protegen mediante mecanismos que permiten la configuración del tráfico, la programación y el evitar congestiones. Para ello se establecen prioridades de la clase de tráfico y de esta forma garantizar un servicio eficaz a los usuarios legítimos. Por lo general, el objetivo es controlar métricas como el tipo de paquete, el retraso y la fluctuación de los mismos. Sin embargo, resulta infructuoso al momento que los atacantes ataquen a los clientes de clase prioritaria, que es donde más perjudican estas amenazas al uso de la red.

2.2.2. Tipos de ataques DDoS

2.2.2.1 Ataques a nivel de capa de red. Estos ataques DDoS consumen recursos como el ancho de banda de la red o el equipo al enviar un alto volumen de paquetes. Estas máquinas no están disponibles para transacciones válidas y pueden fallar durante la carga. La forma más común de ataque de este tipo de ataque es mediante el envío de una gran cantidad de paquetes TCP, UDP o ICMP aparentemente legítimos se dirigen a un destino específico. Para dificultar aún más la detección, estos ataques también pueden falsificar la dirección de origen, es decir, falsear la dirección IP que supuestamente generó la solicitud para evitar la identificación.

2.2.2.2 Ataques DDoS a nivel de capa de aplicación. El poder de ataque se puede amplificar forzando al objetivo a ejecutar operaciones de alto riesgo. Estos ataques pueden consumir todo el ancho de banda corporativo disponible y llenar las redes con tráfico ilegítimo. Los protocolos de enrutamiento también pueden verse afectados y los servicios se interrumpen al restablecer los protocolos de enrutamiento o al ofrecer datos que dañan la operación del servidor.

2.2.2.3 Ataques DDoS de inundación HTTP. Un ataque que bombardea servidores web con solicitudes HTTP se llama un ataque de inundación HTTP. Estos ataques de inundación HTTP son comunes en la mayoría de los programas de software Botnet. Los atacantes envían una solicitud HTTP y luego formulan las solicitudes HTTP de diferentes maneras para maximizar el poder de ataque o para evitar la detección. Un atacante, por ejemplo, puede manipular la Botnet para enviar solicitudes HTTP para descargar un archivo grande desde el destino. El archivo es leído por el destino desde el disco duro, se almacena en la memoria y finalmente se carga, que se envían de vuelta al Botnet. Por lo tanto, una simple solicitud HTTP puede consumir significativamente recursos en la CPU, la memoria, los dispositivos de entrada / salida y el enlace de Internet saliente, como la Figura 4.

2.2.3. Arquitectura de la ejecución y detección de ataques DDoS en la capa de aplicación

Los ataques DDoS en la capa de aplicación se caracterizan por el envío masivo de solicitudes, lo que provoca limitaciones en el acceso a los servicios web de los usuarios legítimos. La Figura 4 muestra, la transaccionalidad del sistema, observamos las solicitudes realizadas por el usuario o atacante al servidor web. En el proceso de detección de este tipo de ataques, es necesario extraer las características de las solicitudes enviadas al servidor. Para esto, se utilizan algoritmos o procedimientos que filtran información sobre características tales como las mediciones de distancia (Gavrilis & Dermatas, 2005; Nguyen & Choi, 2008) proporcionadas por los flujos de solicitud (Wang et al., 2008). Una vez que se obtienen las características, se utilizan algoritmos o criterios de clasificación para detectar ataques. Los algoritmos de aprendizaje automático se usan comúnmente en la clasificación de usuarios reales y

ataques DDoS (Xiang & Zhou, 2005). También hay criterios de clasificación basados en técnicas de computación suave y su enfoque hidrológico (Zargar et al., 2013). Finalmente, cuando se detecta un ataque DDoS, estos serán descartados del conjunto de solicitudes, mientras que las solicitudes de los usuarios reales ingresan al servidor web para obtener una respuesta.

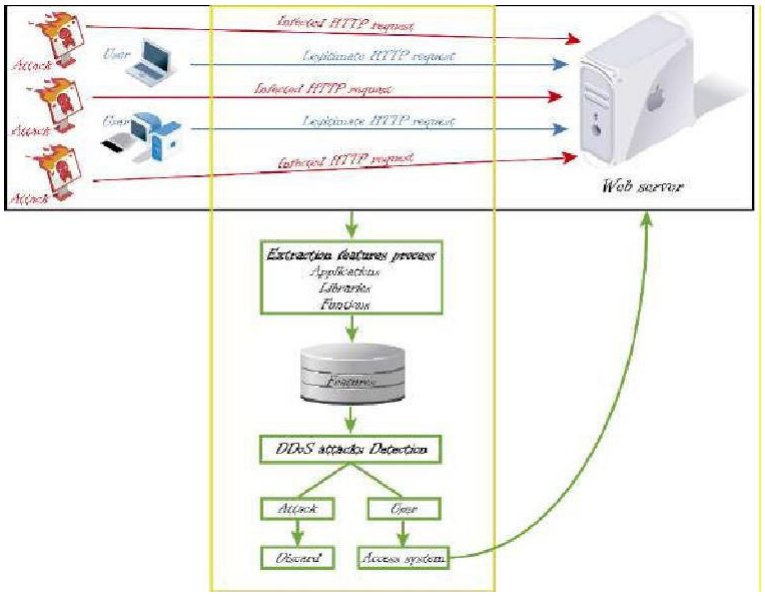

Figura 4. **Ejecución y detección de ataques DDoS en la capa de aplicación**

2.2.4. Clasificación de ataques basada en grado de automaticación

Cuando se produce un ataque DDoS, el mismo puede realizarse de forma manual o automatizada. Según el grado de automatización, puede haber tres tipos de ataques DDoS:

a) Manual: Un atacante busca manualmente máquinas remotas en busca de vulnerabilidades. Una vez localizada la máquina adecuada, instala el código de ataque y luego ordena el inicio del ataque.

b) Semiautomático: En los ataques semiautomáticos, la red DDOS se compone de máquinas controladas (maestras) y máquinas esclavos o zombis. Las fases de reclutamiento, explotación e infección son automatizadas. En la fase de ataque, el atacante especifica el tipo de ataque, el inicio, la duración y la víctima mediante el controlador a los esclavos, que envían paquetes a la víctima.

c) Ataque DDoS automático: Se automatiza la fase de ataque además de las fases de reclutamiento, explotación e infección, y evitan la necesidad de comunicación entre los atacantes y las máquinas infectadas. La hora de inicio del ataque, el tipo de ataque, la duración y la víctima están preprogramados en el código de ataque. Los mecanismos de despliegue de esta clase de ataque ofrecen una exposición mínima al atacante, ya que solo participa un solo comando al inicio del proceso de reclutamiento. La especificación de ataque codificada sugiere un uso único de la red DDOS, o la naturaleza inflexible del sistema. Sin embargo, los mecanismos de propagación generalmente dejan abierta una puerta trasera a la máquina comprometida, lo que permite un fácil acceso y modificación en el futuro del código de ataque.

2.3. Bases teóricas

2.3.1. Capa de aplicación

De acuerdo al modelo OSI, la capa de aplicación viene a ser la interfaz de usuario. La capa de aplicación OSI es responsable de mostrar los datos e imágenes al usuario en un formato reconocible por el ser humano y de interactuar con la capa de presentación

que se encuentra debajo. En una implementación, las capas de aplicación y presentación se combinan con frecuencia (Lee, 2013).

2.3.2. Modelo OSI

El modelo OSI (ISO / IEC 7498-1) es un modelo conceptual que se caracteriza y estandariza las funciones internas de un sistema de comunicación. El mismo está dividido en capas de abstracción. El modelo es un producto del proyecto de Interconexión de Sistemas Abiertos en la Organización Internacional de Normalización (ISO). El modelo agrupa funciones de comunicación similares en una de las siete capas lógicas.

2.3.3. Herramientas de ataque

Las herramientas de ataque se encargan de lanzar malware hacia el sistema. Estos ataques son ejecutados sin el conocimiento del propietario del sistema. Por lo general, utilizan una estructura en capas donde el atacante usa un programa cliente para conectarse a los sistemas comprometidos que a su vez facilitan el ataque DDoS. (Dittrich, 1999).

2.3.4. Ataques a la capa de aplicación

Un ataque DDoS se enfoca en la de la capa de aplicación principalmente para ejecutar interrupciones de transacciones y acceso a bases de datos (Higgins, 2013). Es importante mencionar que requiere menos recursos que los ataques de la capa de red. Un ataque DDoS puede simular tráfico legítimo, excepto que se dirige a funciones específicas de la aplicación. El ataque a la capa de aplicación puede interrumpir servicios como la recuperación de información o las funciones de búsqueda en un sitio web (Ginovsky, 2014).

2.3.5. Técnica de defensa

Las respuestas defensivas a los ataques de denegación de servicio generalmente involucran el uso de una combinación de herramientas de detección de ataques, clasificación de tráfico y respuesta, con el objetivo de bloquear el tráfico que identifican como ilegítimo y permitir el tráfico que identifican como legítimo (Loukas, 2010).

2.3.6. Extorsión DDoS

En 2015, las redes de bots DDoS crecieron en importancia, apuntando a las instituciones financieras (Cloudbric, 2015). Los extorsionistas cibernéticos suelen comenzar con un ataque de bajo nivel y una advertencia de que se llevará a cabo un ataque más grande si no se paga un rescate en Bitcoin (Solon, 2015). Los expertos en seguridad recomiendan sitios web específicos para no pagar el rescate. Los atacantes tienden a entrar en un extenso esquema de extorsión una vez que reconocen que el objetivo está listo para pagar (Greenberg, 2015).

CAPÍTULO III. ESTADO DEL ARTE

Los ataques informáticos, como el ataque de denegación de servicio (DoS), son una amenaza para la seguridad de Internet y han planteado un problema desde su inicio en 1980 (Zargar et al., 2013). Tales ataques constituyen en accesos ilegales a través de las cuales un atacante interrumpe los recursos o servicios de un sistema (Chen et al., 2008), y afecta el acceso a la red, las cuentas en línea, el correo electrónico y los recursos de computadora (Jain & Singh, 2012).

Más tarde, apareció un tipo más sofisticado de ataque DoS llamado ataques distribuidos de denegación de servicio (DDoS). Este ataque involucra dos o más computadoras, que pueden estar ubicadas en varias partes del mundo, y son ejecutadas por el mismo atacante (Stevanovic & Vlajic, 2013). Los primeros informes de este tipo de ataque aparecieron en 1999 (Criscuolo, 2000). Autores como Zargar et al. (2013) y Choi et al. (2014), coinciden en que el principal problema para detectar este tipo de ataque es no poder diferenciar los flujos legítimos de los flujos de ataque, lo que resulta en altas tasas de falsos positivos y negativos en los métodos de detección empleados.

Estos ataques pueden orientarse a la capa de red (protocolos, hubs, switch) y a la capa de aplicación (sistema, CPU, recursos), este último ha aumentado en los últimos años debido a su fácil ejecución y difícil detección, por lo tanto, los esfuerzos de los mecanismos de detección se centran en este tipo de ataque.

Los ataques DDoS dirigidos a la capa de aplicación se consideran sofisticados porque imitan las solicitudes de usuarios reales, por lo que es más difícil detectarlos. Los métodos consideran la información de las solicitudes de los usuarios y cierta lógica que permite relacionarlos con un ataque o un usuario. La lógica está dada por técnicas

tales como redes neuronales, algoritmos genéticos, máquinas de vectores de soporte y modelos estadísticos que en general consumen recursos considerables.

El consumo excesivo de recursos significa que el proceso de detección es más lento y más aún con grandes cantidades de información. La lentitud del proceso afecta al sistema y provoca la saturación del ancho de banda y el consumo de los recursos del servidor. Además, las técnicas mencionadas tienen un tiempo de espera antes de la detección, para saber si se trata de un ataque, lo que afecta la productividad de los servicios. La tarea más difícil que tienen los métodos de detección es diferenciar una solicitud para identificarla como un usuario real o un ataque. Brosso et al. (2010) introduce las características del dinamismo del usuario, indicando que provienen de la interacción entre el usuario y el sistema. Oikonomou et al. (2009) menciona que las características del dinamismo del usuario permiten diferenciar un robot de un usuario real.

3.1. Métodología para la revisión

La revisión sistemática que se llevó a cabo en esta investigación se basa en el modelo propuesto por (Kitchenham, 2004), que se divide en tres fases:

Planificación de la revisión: se plantean preguntas sobre los objetivos de la investigación y la revisión.

Realización de la revisión: en esta etapa, el plan se ejecuta y los estudios principales que siguen los criterios de inclusión y exclusión seleccionados se mencionan o se descartan.

Informe de la revisión: en esta etapa se muestran los resultados de la revisión estadística y el análisis presentado en las secciones III y IV, respectivamente.

3.1.1. Planificación de la revisión

Para llevar a cabo la revisión de la literatura sobre la detección de ataques DDoS, se plantearon las siguientes preguntas de investigación:

P1: ¿Cuáles son las técnicas utilizadas para la detección?

P2: ¿Cuáles son las variables utilizadas?

P3: ¿Cuáles son las herramientas utilizadas?

P4: ¿Dónde se implementan?

P5: ¿En qué momento antes de los ataques se debe activar el mecanismo de detección?

P6: ¿Con qué proporción de precisión detectan las técnicas un ataque DDoS?

Las respuestas a las preguntas de investigación anteriores se encontraron en las siguientes fuentes de datos: DOAJ (Directorio de Open Access Journal), IEEE Xplore, Science Direct y Springer. Para encontrar artículos científicos publicados en revistas con un factor de impacto de SJR (Scimago Journal and Country Rank), en el período comprendido entre 2005 y 2017, se realizó el siguiente procedimiento de búsqueda (Tabla 3), teniendo en cuenta el título, el resumen y las palabras clave.

Tabla 4. **Cadena fuente para búsqueda**

Fuente	Cadena
DOAJ	distributed denial of service or ddos; 2005-2017

Fuente. Autor

IEEE Xplore	((distributed denial of service) OR ddos) and refined by Year: 2005-2017
Science Direct	pub-date ¿= 2005 and (distributed denial of service) and ddos
Springer	”distributed denial of service” or ”ddos”’within 2005 - 2017

Además, estos términos están adaptados para coincidir con las preguntas de investigación y las necesidades individuales del motor de búsqueda. A los resultados de las búsquedas de las fuentes de información se aplicaron los criterios de inclusión y exclusión que se muestran en la Tabla 5.

Tabla 5. **Criterios de inclusión y exclusión**

Criterios de inclusión	Criterios de exclusión
Modelos, métodos y técnicas para detectar ataques DDoS	Detección presenta propuestas que no incluyen los resultados experimentales
Variables propuestas en los ataques de detección	Mecanismos de detección de presente en botnets generals
Componentes propuestos que conforman el mecanismo	Libros, actas, carteles, tesis, talleres
Herramientas propuestas en los mecanismos de detección	Presentado en su flujo de ataque de seguimiento
Responde directamente a las preguntas de investigación	Envíe contribuciones dirigidas a entornos de computación en la nube, redes P2P, MANET, áreas locales inalámbricas,

	centros de datos, redes de alta velocidad y servidores DNS

Fuente. Autor

3.1.2. Elaboración de la revisión

Los resultados de búsqueda obtenidos, de acuerdo con la estrategia propuesta, fueron sometidos a un proceso de selección, de acuerdo con los criterios de inclusión y exclusión establecidos. Fue necesario realizar una revisión preliminar de su contenido para determinar su relevancia para el presente estudio y determinar si estos trabajos se aplican a la detección de ataques DDoS. La mayoría de los artículos se descartaron porque correspondían a otro tema en estudio, como encuestas, taxonomía y redes de bots. El proceso implementado y los resultados obtenidos en cada etapa se muestran en la Figura 5. Posteriormente, se procedió a analizar los artículos para responder las preguntas de investigación.

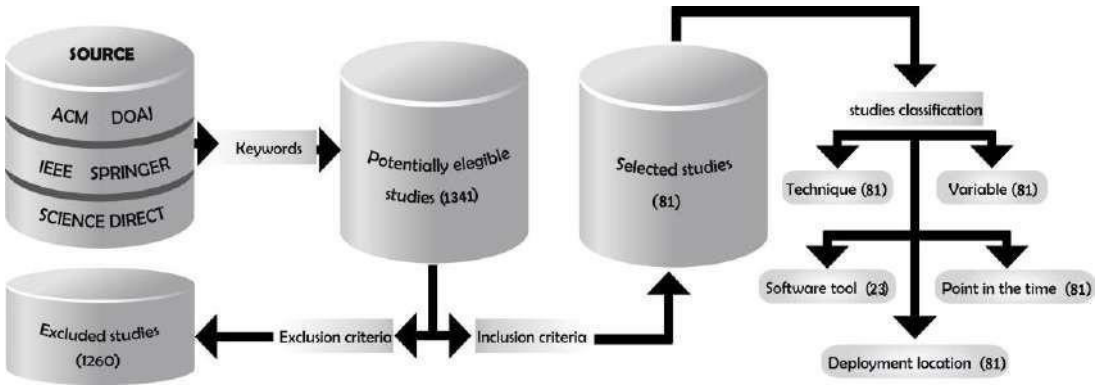

Figura 5. **Proceso de revisión de la literatura**

Los resultados de la búsqueda realizada mostraron un total de 1341 artículos. De estos, se seleccionaron 81, que cumplían con los criterios de inclusión y exclusión establecidos, como puede verse en la Tabla 5.

Tabla 6. **Fuente de artículos seleccionado**

Fuente	**Estudios potencialmente elegibles**	**Estudios seleccionados (Journal article)**
DOAJ	158	20
IEEE Xplore	80	21
Science Direct	843	30
Springer	260	10
Total	1341	81

Fuente. Autor

3.2 Resultados de la búsqueda

3.2.1 Tendencias temporales de las publicaciones.

La Figura 6 muestra la tendencia temporal de las publicaciones sobre la detección de ataques DDoS, seleccionados de la metodología, por la fase de la muestra de revisión.

En él, puede ver el aumento en el número de publicaciones en los últimos 13 años. La tendencia en el número de artículos publicados refleja la importancia que la comunidad científica le ha otorgado a este tema de estudio.

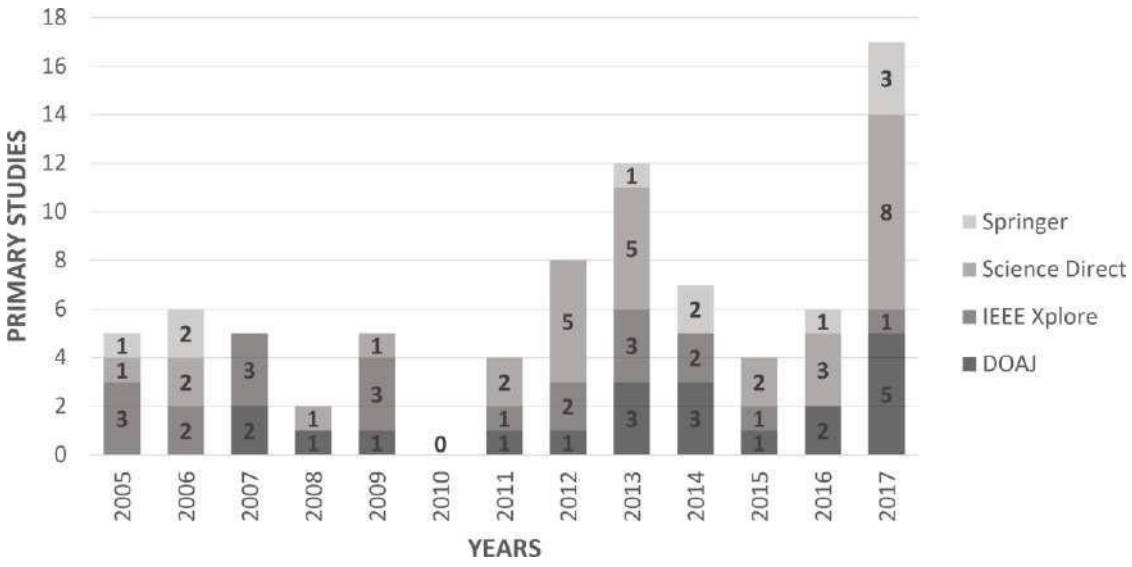

Figura 6. **Tendencia temporal de artículos seleccionados**

3.2.2. Fuentes de datos

Los resultados de la búsqueda muestran que la mayor cantidad de artículos se obtuvieron de las bases de datos de Science Direct y IEEE Xplore. Además, las fuentes de referencia que proporcionaron información sobre el tema fueron Springer y DOAJ, como se puede ver en la Tabla 6.

3.2.3. Aspectos

Se eligieron los siguientes elementos para la detección de ataques DDoS: técnicas, variables, herramientas, ubicación de implementación, punto en el tiempo y precisión de detección. La Tabla 7 muestra estos aspectos junto con sus respectivas definiciones:

Tabla 7. **Definición de aspectos**

Aspectos	Definición
Técnica	Se refiere al conjunto de procedimientos o recursos utilizados en una actividad en particular. En este trabajo consideramos las técnicas empleadas por los mecanismos de detección.
Variables	Se define como el carácter que se mide en diferentes individuos u objetos. Necesidad de conocer las características que utiliza el mecanismo para detectar ataques DDoS.
Herramientas de software	Estos son programas informáticos que ayudan al especialista en el proceso de diseño y desarrollo de software o documentación (Nyakundi, 2015).
Ubicación de la implementación	Se refiere a la ubicación donde se debe implementar el mecanismo de detección, es decir, en la fuente, en la red, en el destino o en un híbrido de los anteriores (Zargar et al., 2013).
Punto en el tiempo	Se define como el momento en que el mecanismo de detección debe activarse cuando se da un ataque (Zargar et al., 2013).
Precisión / tasa de detección	El valor global de todas las instancias correctamente clasificadas, es decir, tanto verdaderos positivos como verdaderos negativos (Tawdar et al., 2017).

Fuente. Autor

Tabla 8. **Aspectos de la detección de ataques DDoS**

Fuente	Técnicas	Variables	Herramientas	Lugar de detección	Punto en el tiempo
DOAJ	Beak et al. (2007); Meenakshi et al. (2007); Chen et al. (2008); Yan et al. (2009); Liu et al. (2011); Tiruchengode et al. (2012); Al-Duwairi et al. (2013); Chen et al. (2013); Udhayan et al. (2013); Huang et al. (2014); Sachdeva et al. (2014); Wang et al. (2014); Saleh et al. (2015); Johnson et al. (2016); Cepheli et al. (2016); Zhou et al. (2017); Gu et al. (2017); Mirvazir (2017); Jia et al. (2017); Peraković et al. (2017)	Beak et al. (2007); Meenakshi et al. (2007); Chen et al. (2008); Yan et al. (2009); Liu et al. (2011); Tiruchengode et al. (2012); Al-Duwairi et al. (2013); Chen et al. (2013); Udhayan et al. (2013); Huang et al. (2014); Sachdeva et al. (2014); Wang et al. (2014); Saleh et al. (2015); Johnson et al. (2016); Cepheli et al. (2016); Zhou et al. (2017); Gu et al. (2017); Mirvazir (2017); Jia et al. (2017); Peraković et al. (2017)	Beak et al. (2007); Meenakshi et al. (2007); Al-Duwairi et al. (2013); Zhou et al. (2017); Jia et al. (2017)	Beak et al. (2007); Meenakshi et al. (2007); Chen et al. (2008); Yan et al. (2009); Liu et al. (2011); Tiruchengode et al. (2012); Al-Duwairi et al. (2013); Chen et al. (2013); Udhayan et al. (2013); Huang et al. (2014); Sachdeva et al. (2014); Wang et al. (2014); Saleh et al. (2015); Johnson et al. (2016); Cepheli et al. (2016); Zhou et al. (2017); Gu et al. (2017); Mirvazir (2017); Jia et al. (2017); Peraković et al. (2017)	Beak et al. (2007); Meenakshi et al. (2007); Chen et al. (2008); Yan et al. (2009); Liu et al. (2011); Tiruchengode et al. (2012); Al-Duwairi et al. (2013); Chen et al. (2013); Udhayan et al. (2013); Huang et al. (2014); Sachdeva et al. (2014); Wang et al. (2014); Saleh et al. (2015); Johnson et al. (2016); Cepheli et al. (2016); Zhou et al. (2017); Gu et al. (2017); Mirvazir (2017); Jia et al. (2017); Peraković et al. (2017)
IEEE Xplore	Chen et al. (2005); Mirkovic et al. (2005); Yau et al. (2005); Kim et al. (2006); Yaar et al. (2006); Chen et al. (2007); Chen and Park et al. (2007); Wang et al. (2007); Chonka et al. (2009); Ranjan et al. (2009); Xie et al. (2009); Xiang et al. (2001); François et al. 2012; Yu et al. (2012); Chen et al. (2013); Luo et al. (2013); Wu et al. (2013); Luo et al. (2014); Ma et al. (2014); Wu et al. (2014); Luo et al. (2017);	Chen et al. (2005); Mirkovic et al. (2005); Yau et al. (2005); Kim et al. (2006); Yaar et al. (2006); Chen et al. (2007); Chen and Park et al. (2007); Wang et al. (2007); Chonka et al. (2009); Ranjan et al. (2009); Xie et al. (2009); Xiang et al. (2001); François et al. 2012; Yu et al. (2012); Chen et al. (2013); Luo et al. (2013); Wu et al. (2013); Luo et al. (2014); Ma et al. (2014); Wu et al. (2014); Luo et al. (2017);	Yau et al. (2005); Chen et al. (2007); Xie et al. (2009)	Chen et al. (2005); Mirkovic et al. (2005); Yau et al. (2005); Kim et al. (2006); Yaar et al. (2006); Chen et al. (2007); Chen and Park et al. (2007); Wang et al. (2007); Chonka et al. (2009); Ranjan et al. (2009); Xie et al. (2009); Xiang et al. (2001); François et al. 2012; Yu et al. (2012); Chen et al. (2013); Luo et al. (2013); Wu et al. (2013); Luo et al. (2014); Ma et al. (2014); Wu et al. (2014); Luo et al. (2017);	Chen et al. (2005); Mirkovic et al. (2005); Yau et al. (2005); Kim et al. (2006); Yaar et al. (2006); Chen et al. (2007); Chen and Park et al. (2007); Wang et al. (2007); Chonka et al. (2009); Ranjan et al. (2009); Xie et al. (2009); Xiang et al. (2001); François et al. 2012; Yu et al. (2012); Chen et al. (2013); Luo et al. (2013); Wu et al. (2013); Luo et al. (2014); Ma et al. (2014); Wu et al. (2014); Luo et al. (2017);
Science Direct	Lee et al. (2005); Al-Duwairi et al. (2006); Chen et al. (2006); Lee et al. (2008); Lu et al. (2009); Doron et al.	Lee et al. (2005); Al-Duwairi et al. (2006); Chen et al. (2006); Lee et al. (2008); Lu et al. (2009); Doron et al.	Lee et al. (2008); Lu et al. (2009); Doron et al. (2011); Kumar et al.	Lee et al. (2005); Al-Duwairi et al. (2006); Chen et al. (2006); Lee et al. (2008); Lu et al. (2009); Doron et al.	Lee et al. (2005); Al-Duwairi et al. (2006); Chen et al. (2006); Lee et al. (2008); Lu et al. (2009); Doron et al.

	(2011); Kumar et al. (2011); Lee et al. (2012); Rahmani et al. (2012); Shiaeles et al. (2012); Wang et al. (2012); Zhang et al. (2012); Giralte et al. (2013); Kumar et al. (2013); Seo et al. (2013); Spyridopoulos et al. (2013); Varalakshmi et al. (2013); Xiao et al. (2015); Malialis et al. (2015); Kalkan et al. (2016), Sachdeva et al. (2016); Saied et al. (2016); Jazi et al. (2017); Mirvaziri (2017); Sreeram et al. (2017); Nunes et al. (2017); Prasad et al. (2017); Behal et al. (2017); Singh et al. (2017); Hoque et al. (2017)	(2011); Kumar et al. (2011); Lee et al. (2012); Rahmani et al. (2012); Shiaeles et al. (2012); Wang et al. (2012); Zhang et al. (2012); Giralte et al. (2013); Kumar et al. (2013); Seo et al. (2013); Spyridopoulos et al. (2013); Varalakshmi et al. (2013); Xiao et al. (2015); Malialis et al. (2015); Kalkan et al. (2016), Sachdeva et al. (2016); Saied et al. (2016); Jazi et al. (2017); Mirvaziri (2017); Sreeram et al. (2017); Nunes et al. (2017); Prasad et al. (2017); Behal et al. (2017); Singh et al. (2017); Hoque et al. (2017)	(2011); Kumar et al. (2013); Seo et al. (2013); Spyridopoulos et al. (2013); Varalakshmi et al. (2013); Xiao et al. (2015); Sachdeva et al. (2016); Nunes et al. (2017); Hoque et al. (2017)	(2011); Kumar et al. (2011); Lee et al. (2012); Rahmani et al. (2012); Shiaeles et al. (2012); Wang et al. (2012); Zhang et al. (2012); Giralte et al. (2013); Kumar et al. (2013); Seo et al. (2013); Spyridopoulos et al. (2013); Varalakshmi et al. (2013); Xiao et al. (2015); Malialis et al. (2015); Kalkan et al. (2016), Sachdeva et al. (2016); Saied et al. (2016); Jazi et al. (2017); Mirvaziri (2017); Sreeram et al. (2017); Nunes et al. (2017); Prasad et al. (2017); Behal et al. (2017); Singh et al. (2017); Hoque et al. (2017)	(2011); Kumar et al. (2011); Lee et al. (2012); Rahmani et al. (2012); Shiaeles et al. (2012); Wang et al. (2012); Zhang et al. (2012); Giralte et al. (2013); Kumar et al. (2013); Seo et al. (2013); Spyridopoulos et al. (2013); Varalakshmi et al. (2013); Xiao et al. (2015); Malialis et al. (2015); Kalkan et al. (2016), Sachdeva et al. (2016); Saied et al. (2016); Jazi et al. (2017); Mirvaziri (2017); Sreeram et al. (2017); Nunes et al. (2017); Prasad et al. (2017); Behal et al. (2017); Singh et al. (2017); Hoque et al. (2017)
Springer	Li et al. (2005); Kulkarni et al. (2006); Xiao et al. (2006); Kang et al. (2013); Kang et al. (2014); Zhou et al. (2014); Dick et al. (2016); Prasad et al. (2017); Boro et al. (2017); Merouane et al. (2017)	Li et al. (2005); Kulkarni et al. (2006); Xiao et al. (2006); Kang et al. (2013); Kang et al. (2014); Zhou et al. (2014); Dick et al. (2016); Prasad et al. (2017); Boro et al. (2017); Merouane et al. (2017)	Li et al (2005); Zhou et al. (2014); Merouane et al. (2017)	Li et al. (2005); Kulkarni et al. (2006); Xiao et al. (2006); Kang et al. (2013); Kang et al. (2014); Zhou et al. (2014); Dick et al. (2016); Prasad et al. (2017); Boro et al. (2017); Merouane et al. (2017)	Li et al. (2005); Kulkarni et al. (2006); Xiao et al. (2006); Kang et al. (2013); Kang et al. (2014); Zhou et al. (2014); Dick et al. (2016); Prasad et al. (2017); Boro et al. (2017); Merouane et al. (2017)
Total	81	81	23	81	81

Fuente: Autor

La Tabla 8 muestra la distribución de los 81 estudios seleccionados, de acuerdo con los aspectos identificados para detectar ataques DDoS como se definió anteriormente. Se puede observar que el 100% de los artículos seleccionados presentaron al menos una técnica para detectar ataques. Y solo el 28% menciona la herramienta utilizada para implementar la técnica para detectar un ataque DDoS.

3.3 Análisis y discusión

El análisis de la información recopilada en la sección anterior se realizó sobre la base de las preguntas de investigación planteadas en la Sección 3.3.1. Los resultados se presentan en tablas que contienen la descripción del aspecto que se analizará, junto con los nombres de los autores que los utilizaron.

3.3.1. P1: ¿Cuáles son las técnicas utilizadas en la detección?

Las técnicas utilizadas en la detección de ataques DDoS se muestran y describen en la Tabla 9. Como se puede apreciar en esta tabla, se han propuesto diferentes técnicas 48. Los aspectos de cada técnica se discuten a continuación.

Tabla 9. **Técnicas usadas para la detección de ataques DDoS**

Id	Técnica	Descripción

T1	Bagging	Representante de los métodos paralelos de aprendizaje en conjunto. Emplea muestreo aleatorio en el conjunto de datos de muestreo. El algoritmo se enfoca principalmente en la disminución de la varianza.
T2	Algoritmo Bat	El algoritmo de murciélago usa el comportamiento de determinación de ubicación basado en eco de los murciélagos para resolver problemas de optimización de objetivo único y de objetivo múltiple.
T3	Filtro Bloom	El filtro Bloom es un tipo de estructura de datos hash que ahorra espacio. Se usa un filtro Bloom modificado para construir una tabla hash que pueda registrar paquetes de control TCP de tres vías a un costo de almacenamiento limitado.
T4	Change aggregation tree (CAT)	Este mecanismo CAT está diseñado para su uso a nivel de enrutador para detectar cambios abruptos en los flujos de tráfico. Cuando se lanza un ataque DDoS, los enrutadores observan cambios en la distribución temporal temporal de los volúmenes de tráfico.
T5	Análisis Cluster	El análisis consiste en agrupar datos para que los objetos en un grupo dado sean similares entre sí y diferentes de los de otros grupos. Al utilizar el análisis de conglomerados, podemos separar el tráfico normal y cada fase de los conglomerados que forman DDoS tienen diferencias entre ellos. Atacar en grupos con particiones si las variables involucradas
T6	Congestion Participation Rate (CPR)	Tasa de participación en la congestión (RCP) para identificar los flujos de LDDoS midiendo la intención de los flujos de red para controlar la red. Según nuestro conocimiento, es la primera métrica que es capaz de reconocer los flujos de LDDoS al cuantificar la intención de cada flujo de congestionar la red.
T7	Análisis de Correlación	La correlación se utiliza para describir la similitud de diferentes flujos. Sin embargo, en algunos casos, puede indicar una correlación cero. Aunque los dos flujos están completamente correlacionados, hay una diferencia de fase.
T8	Mecanismo de Conteo	Asigna un valor continuo en contraposición a una medida binaria a cada sesión del cliente, y el programador utiliza estos valores para determinar si y cuándo programar las solicitudes de una sesión.

T9	Búsqueda Cuckoo	Técnica estimulada por el acto parásito de algunas aves cuco. Las especies de tipo Cuco no pueden completar su ciclo de reproducción sin el huésped adecuado.
T10	Algoritmo Cusum	Un procedimiento de suma acumulativa no paramétrica (CUSUM, por sus siglas en inglés) comúnmente utilizado para la detección de una amplia gama de cambios posibles y generalmente se ve favorecido por su simplicidad y baja sobrecarga computacional.
T11	Entropia	Las entropías generalizadas de Renyi son una familia de medidas que caracterizan la distribución de una variable aleatoria. La entropía shanon se ha utilizado para conceptualizar la entropía de la dirección de origen y la entropía del cluster de tráfico.
T12	Firewall	La función de firewall da al defensor la opción de establecer el valor que es el umbral por encima del cual se eliminan todos los paquetes de un flujo.
T13	Lógica difusa	Estimador difuso en el paquete medio entre los tiempos de llegada. Interpreta bien las reglas, pero sufre la desventaja de no poder adquirir las reglas automáticamente.
T14	Algoritmos genéticos	Un algoritmo genético es una búsqueda heurística que imita el proceso de evolución natural. Los algoritmos genéticos pertenecen a la clase más amplia de algoritmos evolutivos (EA), que generan soluciones a los problemas de optimización utilizando herencia, mutación, selección y técnicas de cruce inspiradas en la evolución natural.
T15	Posición estratégica Google	La idea principal de JUST-Google es permitir que los enrutadores de borde de los ISP permitan que el tráfico que se origina en fuentes aprobadas por Google y destinadas a una víctima dentro de ese ISP pase mientras filtra todo el tráfico destinado a la misma víctima. Un algoritmo HsMM que describe el proceso estocástico que varía con el tiempo y supervisa los ataques App-DDoS que ocurren durante un evento de multitud repentina.

T16	Modelo Hidden semi-Markov (HsMM)	Un algoritmo HsMM que describe el proceso estocástico que varía con el tiempo y supervisa los ataques App-DDoS que se producen durante un evento de multitud de flash.
T17	Filtro Hop-Count	La dirección IP de origen sirve como el índice en la tabla para recuperar el conteo correcto de saltos para esta dirección IP. Si la cuenta de saltos calculada coincide con la cuenta de saltos almacenada, el paquete se ha autenticado.
T18	Información de distancia	Una métrica utilizada para detectar ataques DDoS de baja velocidad al medir la diferencia entre el tráfico legítimo y el tráfico de ataque.
T19	Información de divergencia	Calcula las distancias entre las mediciones de probabilidad independientemente de los parámetros y detecta al atacante y descarta los paquetes del adversario durante un período de tiempo fijo de manera organizada.
T20	Tasa de desviación conjunta (JDR)	Es una nueva métrica para describir la tasa de desviación de los estados de tráfico de la red. JDR es una combinación de las desviaciones de todas las funciones múltiples en NetWork Traffic State (NTS).
T21	Vecinos cercanos	El algoritmo vecino más cercano a k es un método que predice clases de flujo basadas en los ejemplos de entrenamiento más cercanos a k en el espacio de características. Un flujo se clasifica por el voto mayoritario de sus vecinos y k es un entero positivo, generalmente pequeño.
T22	Complejidad Kolmogorov	La complejidad de Kolmogorov establece que la medida de complejidad conjunta de las cadenas aleatorias es inferior a la suma de las complejidades de las cadenas individuales cuando las cadenas presentan una cierta correlación.
T23	Servicio de mapeo	Un proveedor de servicios registra los enlaces de su (s) nombre (s) de dominio a las direcciones IP en el sistema de nombres de dominio (DNS). Cuando un cliente desea obtener un servicio del proveedor de servicios, su computadora primero consulta el nombre de dominio

		y luego envía una solicitud al servidor que usa la dirección IP devuelta.
T24	Modelos matemáticos	Modelo matemático para estimar el efecto de ataque de este tipo sigiloso de DDoS. Al capturar originalmente el comportamiento de ajuste de una víctima en la ventana de congestión de TCP, nuestro modelo puede evaluar exhaustivamente la configuración) y el efecto del ataque en el entorno de red.
T25	Aplicaciones Multiagente	Un agente se crea como una agregación de capacidades, y dichas capacidades se seleccionan de acuerdo con las acciones primitivas que proporciona un mecanismo.
T26	Redes neuronales	Una red neuronal consiste en procesar elementos llamados neuronas. Estas redes neuronales están diseñadas para aprender un nuevo patrón, una nueva asociación y nuevas dependencias funcionales. La ventaja de una red neuronal es una mejor capacidad de generalización.
T27	Estrategia de minimización de costos de Neyman Pearson	La teoría de Neyman Pearson (NeP), donde se desconoce el conocimiento previo de la distribución de datos. La hipótesis NeP es útil en situaciones donde diferentes tipos de error tienen diferentes consecuencias.
T28	Red superpuesta	Mantiene anillos virtuales o escudos de protección alrededor de clientes registrados. Un anillo se compone de un conjunto de IPS que se encuentran a la misma distancia (número de saltos) del cliente.
T29	Filtrado de paquetes	La clasificación de paquetes y el esquema de filtrado se implementarán en los enrutadores de borde de la red ISP que contiene el sistema de destino, y se deben activar después de que se detecte un ataque DDoS basado en el TCP.
T30	Marcado de paquetes	Un esquema que le permite a una víctima DDoS filtrar paquetes de ataque por paquete con una alta precisión después de que solo se hayan recibido unos pocos paquetes de ataque.
T31	Identificadores de ruta	Utilizado en negociado entre dominios vecinos como objetos de enrutamiento interdominio.

T32	Pushback	Los comandos pueden contener algunas solicitudes de límite de frecuencia, de modo que, cuando un enrutador ascendente recibe el comando, limitará el tráfico a la víctima y no causará congestión cerca de la víctima.
T33	Puzzles	Captura correlaciones temporales complejas en múltiples escalas de tiempo con una complejidad computacional muy baja.
T34	Modelo de colas	Lleva información sobre las características del tráfico y las propiedades de congestión.
T35	Random Forrest	La selección de características aleatorias se introduce aún más en el proceso de capacitación para el bosque aleatorio.
T36	Filtro de tasa límite	El enrutador congestionado comienza con un límite de velocidad local, y luego empuja progresivamente el límite de velocidad a algunos enrutadores vecinos y más lejos, formando un árbol de límite de velocidad dinámico, que puede ser costoso de mantener.
T37	Relación de flujo colectivo (RCF)	Responsable de clasificar un flujo como legítimo, sospechoso o flujo de ataque basado en la información de paquetes obtenida del módulo de monitoreo y la carga actual en una cola de salida.
T38	Resilient Back Propagation (RBP)	Se encontró que el algoritmo RBP funcionaba mejor. Un solo clasificador comete errores en diferentes muestras de entrenamiento. Así, al crear un conjunto de clasificadores y combinar sus salidas, se puede reducir el error total y se puede aumentar la precisión de detección.
T39	Regulación de enrutador	Contribuye a la comprensión fundamental de la limitación del enrutador como un mecanismo contra los ataques DDoS. En particular, un modelo teórico de control útil para comprender un sistema es el comportamiento bajo una variedad de parámetros y condiciones de operación.
T40	Protocolo de información de	RIP (protocolo de información de enrutamiento), un protocolo representativo de IGP (protocolo de pasarela interior). RIP, que funciona mediante el intercambio de tablas entre enrutadores, opera

	enrutamiento (RIP)	dentro de AS (un sistema autónomo). RIP se utiliza como protocolo de enrutamiento en el interior de AS.
T41	Diferencia-ción semántica del tráfico	La diferenciación semántica del tráfico tiene dos ventajas principales sobre los enfoques de diferenciación por paquete y por usuario: 1) Detecta fácilmente el tráfico de ataque generado aleatoriamente (con o sin suplantación de identidad), ya que dicho tráfico crea estructuras de corta duración sin semántica superior. 2) Detecta fácilmente las estructuras que participan en comunicaciones de una sola vía, enviando agresivamente el tráfico a una parte que no responde.
T42	Enfoques basados en firma	Perfiles que describen las características de una seguridad de red conocida según los requisitos de seguridad de los objetos de red en una red.
T43	Matriz de secuencia especial	SSM es una matriz de expansión dinámica. Usados en bModel, se producen dinámicamente y también hacen que el diámetro de la matriz crezca dinámicamente.
T44	Análisis espectral	El análisis de espectros se puede aplicar tanto al tráfico de entrenamiento como a los flujos de tráfico entrante en el tesbed. Es efectiva detectando ataques DDoS en el nivel de flujo de tráfico y cortando los flujos maliciosos a un nivel de flujo refinado.
T45	Métodos estadísticos	La idea clave es priorizar un paquete basado en una puntuación que estima su legitimidad dados los valores de atributo que lleva.
T46	Vector de soporte de descripción de datos (SVDD)	Un método de detección de anomalías que utiliza datos sin etiquetar para encontrar un modelo para instancias inusuales.
T47	Estadísticas TCP/IP y HTTP	Se calculan los siguientes valores estadísticos para cada usuario entrante: número de solicitudes de obtención, desviación estándar de obtención, media de flujos por usuario, desviación de flujos estándar por usuario y desviación de publicaciones estándar, flujos por minuto por usuario, solicitud por minuto por usuario y así sucesivamente.

T48	Análisis Wavelet	Captura una correlación temporal compleja en múltiples escalas de tiempo con una complejidad computacional muy baja.

Fuente. Autor

La Tabla 9 muestra los resultados de 81 estudios que presentan técnicas de detección de DDoS. Se puede observar que la técnica de red neuronal es la más utilizada, ya que se ha aplicado en siete estudios. Esta técnica es la más utilizada debido a su capacidad computacional y lógica para identificar anomalías entre los flujos de entrada, su fuente de IP y su simplicidad. Doce técnicas también han sido utilizadas por varios autores que trabajan en combinación.

3.2.2. P2: ¿Cuáles son las variables utilizadas en la detección?

En los estudios analizados, se identificaron un total de 28 variables para la detección de ataques DDoS, como se puede ver en la Tabla 10. Esta tabla también proporciona una descripción de las variables que se han identificado.

Tabla 10. **Técnicas usadas para la detección de ataques DDoS**

Id	Característica	Descripción
V1	Consumo de ancho de banda absoluto	Esta característica representa el ancho de banda promedio consumido por las solicitudes encontradas en el intervalo de tiempo absoluto definido. Esta característica también se considera significativa ya que la estimación del consumo de ancho de banda es crítica en la evaluación de la carga.

V2	Cuenta de acceso absoluto a la página	Esta característica representa el número promedio de solicitudes en un intervalo de tiempo absoluto definido. Esta característica también es crítica entre las características consideradas, ya que el recuento de acceso a la página junto con el intervalo de sesión absoluto.
V3	Tiempo de acceso absoluto a la página	Esta característica representa el tiempo promedio empleado en cada solicitud de página en un intervalo de tiempo absoluto definido. El motivo para considerar esta característica es que la carga de solicitudes con un tiempo de acceso mínimo de cada página es sospechosa.
V4	Cuenta de sesión absoluta	Esta característica representa el número promedio de sesiones encontradas en un intervalo de tiempo absoluto definido. Esta característica se considera ya que la carga en cualquier servidor web de destino se estima por el número de sesiones en un intervalo de tiempo determinado.
V5	Intervalo de sesión absoluto	Esta característica representa el tiempo promedio de procesamiento de cada sesión en un intervalo de tiempo absoluto definido. Esta función es crítica, ya que el tiempo de la sesión indica el tiempo empleado por una fuente en el servidor web de destino con una intención de uso justo o un ataque.
V6	Intervalo de tiempo absoluto	Esto denota el tiempo absoluto tomado por el conjunto de sesiones iniciadas en un umbral de tiempo determinado. Esta característica se considera significativa, ya que el flujo HTTP es acumulativo de sesiones múltiples y flujo de paquetes diversificado. Las características exploradas aún más para el intervalo de tiempo absoluto definido.
V7	Número ACK	El número de ACK enviado por el terminal receptor es el último número de secuencia cuando la comunicación fue exitosa.
V8	Tasas de clicks de objetos web	Estimación del porcentaje de clics de los anuncios disponibles para una consulta de búsqueda determinada. Cuanto más interactivo sea, mayor será la tasa de clics.

V9	Relación de diversidad de fuentes	Esta característica representa el número promedio de fuentes divergentes que inician las sesiones en un intervalo de tiempo absoluto definido. La carga de solicitudes de fuentes eminentes es tolerable, por lo tanto, esta característica se considera significativa.
V10	Dirección IP	La única dirección de origen IP válida para los paquetes que se originan en la PC es la asignada por el ISP (ya sea asignada de forma estática o dinámica).
V11	Tráfico de red	Inicios de sesión remotos y transferencias de archivos
V12	Número de conexiones	Características de comportamiento de una conexión en términos de número, tipo de diversos elementos de datos con respecto al tiempo. Estas características se utilizan para determinar las propiedades estadísticas, como la desviación estándar y la varianza.
V13	Número de ICMP	Número de paquetes de respuesta de eco ICMP de la misma fuente.
V14	Número de paquetes	Paquetes transmitidos o recibidos sin errores.
V15	Número de solicitudes	Las solicitudes de ventanas abiertas actualmente y si el número de solicitudes de una ventana de tiempo abierta es viable.
V16	Número de UDP	Número de paquetes de eco UDP a un puerto especificado
V17	Número de usuarios	Conjunto de usuarios reales accediendo a un servidor.
V18	Paquetes	Paquetes que llevan información de la ruta. El nodo víctima puede defenderse del ataque DDoS filtrando los paquetes que transmiten a través de / desde un nodo atacante.
V19	Puertos	El puerto de E / S determina qué puertos de servicio se están utilizando.
V20	Protocolo	Protocolos de Internet (IP), ahora existe un estándar sobre cómo las computadoras de propósito general, como las computadoras personales, las estaciones de trabajo y los servidores pueden intercambiar datos a través del sistema telefónico.

V21	Tasa de paquetes	Esta característica se calcula en los paquetes enviados desde un remitente en particular.
V22	Ratio de paquetes entrantes SMTP	Es más probable que un host que no tenga conexión entrante sea un spammer que uno que tenga tráfico SMTP entrante
V23	Ratio de paquetes salientes SMTP	Muestra la serie de tiempo del tráfico SMTP saliente para un host que se sabe que ha enviado correo no deseado
V24	Solicitudes de sesión	Sesiones entre horas de llegada entre sesiones consecutivas.
V25	Flujos TCP	Flujos con una gran cantidad de datos para enviar, como transferencias FTP
V26	Tasa de trafico	Definido como el número total de bits recibidos durante un cierto intervalo de tiempo.
V27	Tipo de paquetes	Fundamentalmente, todas las redes tienen esencialmente dos tipos de paquetes. Paquetes de datos que pertenecen a usuarios y transportan usuarios o tráfico de aplicaciones. Los paquetes de control pertenecen a la red y se usan para construir y operar dinámicamente la red
V28	Varianza de tiempo	Variación de la diferencia de tiempo entre dos paquetes consecutivos

Fuente. Autor

La Tabla 10 muestra las variables utilizadas en las técnicas de detección y se resumen los nombres de los autores. Se puede observar que las variables utilizadas con mayor frecuencia son paquetes y direcciones IP. Paquetes porque de esta variable se pueden extraer características como la fuente IP, el peso, la velocidad, entre otras. La dirección IP porque esta variable se utiliza para identificar la dirección de origen para cortar el

tráfico que se envía desde ellos. Es importante destacar que doce estudios emplean una combinación de dos variables o más para detectar ataques.

Tabla 11. **Técnicas, estudios y variables utilizadas por los mecanismos de detección de ataques DDoS**

Técnica	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28
T1											Jia et al. (2017)														Jia et al. 2017			
T2		Sreeram et al. (2017)			Sreeram et al. (2017)	Sreeram et al. (2017)											Sreeram et al. (2017)	Sreeram et al. (2017)										
T3																		Xiao et al. (2006)										
T4										Chen et al. (2007)									Chen et al. (2007)	Chen et al. (2007)								
T5										Lee et al. (2008); Gu et al. (2017)				Lee et al. (2008)					Lee et al. (2008)								Lee et al. 2008	
T6											Zhang et al. (2012)																	
T7										Hoque et al. (2017)				Yu et al. (2012)	Zhou et al. (2014)										Hoque et al. 2017			
T8										Liu et al. (2011)								Liu et al. (2011)						Ranjan et al. (2009)				
T9	Prasad et al. (2017)	Prasad et al. (2017)	Prasad et al. (2017)	Prasad et al. (2017)	Prasad et al. (2017)	Prasad et al. (2017)			Prasad et al. (2017)	Udhayan et al. (2013); Xiang et al. (2011)																		
T10						Jazi et al. (2017)						Jazi et al. (2017)			Jazi et al. (2017)						Jazi et al. (2017)							
T11					Sachdeva et al. (2016)					Xiang et al. (2011); Saleh et al. (2015); Sachdeva et al.(2016)	Yan et al. (2009)							Sachdeva et al. (2014); Behalet al. (2017)											
T12										Chen et al. (2008)								Spyridopoulos et al. (2013); Kang et al. (2013)										
T13						Giralte et al. (2013)						Kumar et al. (2013);	Kumar et al. (2013);			Kumar et al. (2013)						Kumar et al. (2013)	Kumar et al. (2013)				Kumar et al. 2013	
T14										Lee et al. (2012); Johnson et al (2016); Singh et al. (2017)		Kumar et al. (2013)	Kumar et al. (2013)		Singh et al. (2017)	Kumar et al. (2013)			Singh et al. (2017)	Singh et al. (2017)		Kumar et al. (2013)	Kumar et al. (2013)				Kumar et al. 2013	
T15										Al-Duwairi et al. (2013)																		
T16								Huang et al. (2014)							Xie et al. (2009)		Xie et al. (2009)											
T17										Wang et al (2007)																		
T18										Ma et al. (2014)	Cepheli et al. (2016)							Zhou et al. (2017)										
T19										Rahmani et al. (2012); Boro et al. (2017)				Varalakshmi et al. (2013)														
T20										Wang et al. (2012)								Wang et al. (2012)	Wang et al. (2012)	Wang et al. (2012)					Wang et al. 2012			
T21																		Xiao et al. (2015); Jia et al. (2017)							Jia et al. 2017			
T22																		Kulkarni et al. (2006)										

T23			Luo et al. (2013)				Luo et al. (2013)												
T24																	Luo et al. 2014		
T25			Nunes et al. (2017)																
T26			Johnson et al. (2016); Peraković et al. (2017); Saied et al. (2016); Singh et al. (2017)	Chen et al. (2013); Wu et al. (2013)	Kumar et al. (2013)	Kumar et al. (2013)	Chonka et al. (2009)		Kumar et al. (2013)	Saied et al. (2016)	Singh et al. (2017); Saied et al. (2016)	Peraković et al. (2017) Singh et al. (2017)	Perak ović et al. (2017)	Kumar et al. (2013)	Kumar et al. (2013)	Singh et al. (2017)			Kumar et al. 2013
T27				Kumar et al. (2011)															
T28								Xiang et al. (2011)											
T29		Al-Duwairi et al. (2006)	Lee et al. (2005); Al-Duwairi et al. (2006)							Meenakshi et al. (2007); Wang et al. (2014); Seo et al. (2013); Beak et al. (2007);									
T30										Yaar et al. (2006); Chen and Park et al. (2007); Seo et al. (2013)									
T31			Luo et al. (2017)																
T32										Chen and Park et al. (2007)									
T33																Wu et al. (2015)			
T34			Doron et al. (2011)							Lu et al. (2009)									
T35										Jia et al. (2017)							Jia et al. 2017		
T36										Chen et al. (2005)									
T37										Tiruchengode et al. (2012)									
T38				Kumar et al. (2011)															
T39										Malialis et al. (2015)								Yau et al. 2005	
T40					Kang et al. (2014)														
T41			Mirkovic et al. (2005)																
T42			Chen et al. (2013); Merouane et al. (2017)	Cepheli et al. (2016)							Chen et al. (2013); Merouane et al. (2017)	Chen et al. (2013)							
T43								Zhou et al. (2014)											
T44			Chen et al. (2006)								Chen et al. (2006);	Chen et al. (2006);							
T45			Mirvaziri et al. (2017); Kalkan et al. (2016)							Kim et al. (2006); Kalkan et al. (2016);	Kalkan et al. (2016)	Kalkan et al. (2016)					Kalkan et al. 2016		
T46					Dick et al. (2016)					Dick et al. (2016)		Dick et al. (2016)							
T47	Giralte et al. (2013)						Giralte et al. (2013)	Giralte et al. (2013)											
T48				Li et al. (2005)															

Fuente. Autor

3.2.3. P3: ¿Cuáles son las herramientas que se utilizan para la implementación de las técnicas?

Las nueve herramientas que emplean las técnicas de detección y los autores que las han utilizado se resumen en la Tabla 12. En dicha tabla se puede observar que Matlab y Network simulator son las herramientas más utilizadas, debido a que ellas presentan una amplia funcionalidad en el análisis de los datos. Por otro lado, las herramientas menos utilizadas son CRF++ toolkits, Globus toolkit, LIBSVM toolkits, Preset resiliense simulator, SSFNet simulator y Weka.

3.2.4. P4: ¿Dónde se implementan las técnicas de detección?

Las técnicas de detección de ataques DDoS se pueden implementar en cuatro ubicaciones: *origen, destino, red* e *híbrido*. *Fuente* se refiere a la fuente del ataque, mientras que el *destino* es el objetivo del ataque. La *red* es el lugar donde circula el tráfico de información e *híbrido* significa que la detección se realiza en múltiples lugares y generalmente existe cooperación entre los puntos de implementación. La Tabla 13 muestra los cuatro sitios de implementación junto con los autores que los utilizan.

En la Tabla 13 se puede ver que la *red* es donde se han implementado la mayoría de las técnicas de detección, es decir, aproximadamente el 58% de la cantidad total. Esto se debe a que la *red* es el lugar desde donde se pueden extraer los datos de tráfico. Se utilizan varias características o variables para la detección. Por el contrario, la *fuente* es donde se implementan la menor cantidad de técnicas, ya que su implementación requiere un alto grado de cooperación entre las redes de datos.

Tabla 12. **Técnicas y herramientas usadas para la detección de ataques DDoS**

Techniques	CRF++ toolkits	Globus Toolkit	LIBSVM toolkits	Matlab	Network simulator	SAS Enterpriser Miner	SSFNet simulator	Tstat	Weka	Preset resiliense simulator
T5				Gu et al. (2017)		Lee et al. (2008)			Gu et al. (2017)	
T6					Zhang et al. (2012)					
T7				Hoque et al. (2017) Zhou et al. (2014)				Varalakshmi et al. (2013)		
T11					Sachdeva et al. (2014)					
T12				Spyridopoulos et al. (2013)	Spyridopoulos et al. (2013)					
T13				Kumar et al. (2013)						
T14				Kumar et al. (2013)						
T16				Xie et al. (2009)						
T19		Varalakshmi et al. (2013)								
T21								Xiao et al. (2015)		
T25										Nunes et al. (2017)
T26				Peraković et al. (2017) Kumar et al. (2013)						
T27				Kumar et al. (2011)						
T29					Meenakshi et al. (2007)					

				Seo et al. (2013)		
T30				Seo et al. (2013)	Beak et al. (2007)	
T33			Lu et al. (2009)		Doron et al. (2011)	
T37			Kumar et al. (2011)			
T38				Yau et al. (2005)		
T41	Chen et al. (2013)	Chen et al. (2013)				Chen et al. (2013)
T42			Zhou et al. (2014)	Merouane et al. (2017)		
T43				Chen et al. (2006)		

Fuente. Autor

Tabla 13. **Ubicaciones de despliegue donde se implementan mecanismos de detección**

Lugar de despliege	Estudios	Total
Fuente	Mirkovic et al. (2005); Chen et al. (2013); Luo et al. (2017)	3
Destino	Huang et al. (2014); Saleh et al. (2015); Johnson et al. (2016); Mirvaziri et al. (2017); Jia et al. (2017); Ranjan et al. (2009); Xie et al. (2009); Al-Duwairi et al. (2006); Lee et al. (2008); Doron et al. (2011); Giralte et al. (2013); Xiao et al. (2015); Jazi et al. (2017); Sreeram et al. (2017); Nunes et al. (2017); Prasad et al. (2017); Singh et al. (2017); Kang et al. (2013); Dick et al. (2016); Prasad et al. (2017); Boro et al. (2017)	21
Red	Beak et al. (2007); Meenakshi et al. (2007); Yan et al. (2009); Liu et al. (2011); Al-Duwairi et al. (2013); Chen et al. (2013); Udhayan et al. (2013); Sachdeva et al. (2014); Wang et al. (2014); Cepheli et al. (2016); Zhou et al. (2017); Gu et al. (2017); Peraković et al. (2017); Chen et al. (2005); Yau et al. (2005); Kim et al. (2006); Yaar et al. (2006); Chen et al. (2007); Wang et al. (2007); Chonka et al. (2009); Xiang et al. (2011); François et al. (2012); Yu et al. (2012); Luo et al. (2013); Wu et al. (2013); Luo et al. (2014); Ma et al. (2014); Lee et al. (2005); Chen et al. (2006); Lu et al. (2009); Kumar et al. (2011); Lee et al. (2012); Shiaeles et al. (2012); Zhang et al. (2012); Kumar et al. (2013); Seo et al. (2013); Spyridopoulos et al. (2013); Varalakshmi et al. (2013); Malialis et al. (2015); Kalkan et al. (2016); Sachdeva et al. (2016); Saied et al. (2016); Behal et al. (2017); Hoque et al. (2017); Li et al. (2005); Kulkarni et al. (2006); Kang et al. (2014)	47
Híbrido	Chen et al. (2008); Tiruchengode et al. (2012); Chen et al. (2007); Wu et al. (2015); Rahmani et al. (2012); Wang et al. (2012); Mirvaziri et al. (2017); Xiao et al. (2006); Zhou et al. (2014); Merouane et al. (2017)	10
Total		81

Fuente. Autor

3.2.5. P5: ¿En qué momento del tiempo se debe activar el mecanismo de detección en un ataque?

El mecanismo de detección puede actuar contra un posible ataque DDoS *antes, durante* y *después* (Zargar et al., 2013). El punto en el tiempo *antes* se refiere a la prevención del ataque antes de que ocurra, mientras que *durante* se refiere al momento en que se realiza el ataque; y finalmente, *después* se refiere a cuándo ocurre el ataque en el destino y, por lo tanto, puede considerarse como mitigación. La Tabla 14 muestra los puntos en el tiempo en que las técnicas de detección pueden actuar junto con los autores que las emplean en cada ubicación.

Tabla 14. **Ubicaciones de despliegue de mecanismos de detección**

Punto en el tiempo	Estudios	Total
Antes	Tiruchengode (2012), Mirkovic et al. (2005), Chen and Park et al. (2007), Chen et al. (2013), Luo et al. (2017), Xiao et al. (2006), Merouane (2017).	7
Durante	Beak et al. (2007), Meenakshi et al. (2007), Chen et al. (2008), Yan et al. (2009), Liu et al. (2011), Al-Duwairi et al. (2013), Chen et al. (2013), Udhayan et al. (2013), Huang et al. (2014), Sachdeva et al. (2014), Wang et al. (2014), Saleh et al. (2015), Johnson et al. (2016), Cepheli et al. (2016), Zhou et al. (2017), Gu et al. (2017), Mirvaziri et al. (2017), Jia et al. (2017), Peraković et al. (2017), Chen et al. (2005), Yau et al. (2005), Kim et al. (2006), Yaar et al. (2006), Chen et al. (2007), Wang et al. (2007), Chonka et al. (2009), Ranjan et al. (2009), Xie et al. (2009), Xiang et al. (2011), François et al. (2012), Yu et al. (2012), Luo et al. (2013), Wu et al. (2013), Luo et al. (2014), Ma et al. (2014), Lee et al. (2005), Al-Duwairi et al. (2006), Chen et al. (2006), Lee et al. (2008), Lu et al. (2009), Doron et al. (2011), Kumar et al. (2011), Lee et al. (2012), Rahmani et al. (2012), Shiaeles et al. (2012), Wang et al. (2012), Zhang et al. (2012), Giralte et al. (2013), Kumar et al. (2013), Seo et al. (2013),	71

	Spyridopoulos et al. (2013), Varalakshmi et al. (2013), Xiao et al. (2013), Kalkan et al. (2016), Sachdeva et al. (2016), Saied et al. (2016), Jazi et al. (2017), Sreeram et al. (2017), Nunes et al. (2017), Prasad et al. (2017), Behal et al. (2017), Singh et al. (2017), Hoque et al. (2017), Li et al (2005), Kulkarni et al. (2006), Kang et al. (2013), Kang et al. (2014), Zhou et al. (2014), Dick et al. (2016), Prasad et al. (2017), Boro et al. (2017)	
Después	Wu et al. (2015), Malialis et al. (2015), Mirvaziri et al. (2017)	3
Total		81

Fuente: Autor

En la Tabla 14 se puede ver que *durante* es el lugar donde se han implementado la mayoría de las técnicas de detección, ya que la detección en ese lugar se ejecuta en tiempo real, cuando el flujo de ataque ha llegado. En contraste, *después* es el momento en el que las técnicas de detección se implementan menos porque el mecanismo realiza un proceso de mitigación después de detectar el ataque.

3.2.6. P6: ¿Cuál es la precisión con la que las técnicas detectan un ataque DDoS?

En este trabajo de investigación solo se consideraron los estudios que tuvieron una detección o una tasa de precisión mayor o igual al 98% y donde se realizaron pruebas con conjuntos de datos que consisten en flujos reales y DDoS. La tasa de detección se calcula mediante la siguiente ecuación: la detección de TP es igual a TP / (TP + FN), y la precisión corresponde a (TP + TN) / (TP + TN + FP + FN) donde, TP = número de positivos verdaderos, TN = número de negativos verdaderos, FP = número de falsos positivos y FN = cantidad de falsos negativos. La precisión con la que las técnicas detectan un ataque DDoS se muestra en la Tabla 15.

Tabla 15. **Mejores tasas de los mecanismos para detectar ataques DDoS**

Estudios	**Tasa de detección (%)**	**Dataset**
Jia et al. (2017)	99.99	Knowledge Discovery and Data mining (KDD) Cup 1999 dataset
Hoque et al. (2017)	99.67	CAIDA, TUIDS and DARPA
Kumar et al. (2011)	99.4	CAIDA 2007, DARPA 2009, BONESI-generated
Johnson et al. (2016)	98.31	KDD Cup1999

Fuente: Autor

En la Tabla 15 se puede ver que el mecanismo de detección con la mayor precisión fue logrado por Jia et al. (2017). La tasa de detección fue del 99,9%. La alta tasa de detección de este mecanismo emplea una combinación novedosa de tres técnicas (bosque aleatorio, vecinos más cercanos y bagged), la implementación de este mecanismo se da a nivel de red, de modo que la detección se produce durante el ataque como resultado de lo cual su impacto es mitigado. Hoque et al. (2017) propuso un nuevo mecanismo que emplea la técnica de correlación. Se utiliza la herramienta Matlab, se mantienen el lugar de implementación (nivel de red) y la detección durante el ataque llega al 99.67%.

CAPÍTULO IV. CARACTERÍSTICAS DEL DINAMISMO DEL USUARIO PARA DETECTAR ATAQUES DDOS EN LA CAPA DE APLICACIÓN

4.1. Característica del comportamiento del usuario.

4.1.1. Características propuestas

El dinamismo del usuario es la interacción del usuario con el sistema y, a través de él, es posible conocer el comportamiento de un usuario y su diferencia con otros (Ghezzi et al., 2014). La autenticación de un usuario a través de su comportamiento ha sido una tarea estudiada desde el punto de vista de la seguridad de la información (Stevanovic et al., 2014). Por lo tanto, para evitar el acceso de usuarios no autorizados, varias investigaciones (Ghezzi et al., 2014, Stevanovic et al., 2014, Urban 2015, Abramson et al., 2013, Kim et al., 2014, Shen et al., 2013) han centrado sus esfuerzos en un proceso llamado comportamiento biométrico. Dentro de este proceso se encuentran: el uso de pulsaciones de teclas, la dinámica del mouse y la interacción con la interfaz gráfica de usuario (GUI) (Stevanovic et al., 2014) para la identificación de usuarios.

La Tabla 16 muestra 24 características que permiten detectar el dinamismo del usuario y diferenciarlo de otro. Estas características se dividen en dos grupos, estos grupos surgen de la interacción del mouse o el teclado y el usuario. En este documento, se

evalúan dos características (movimiento del mouse y clic derecho), porque en el conjunto de datos utilizado para la evaluación, estas características están presentes.

El movimiento del ratón y el clic derecho permiten identificar inequívocamente a un usuario real de un robot. En el caso del movimiento del mouse, un usuario real mueve este periférico para navegar a través del entorno web (Salmeron-Majadas et al., 2014). Si bien el clic con el botón derecho es un evento especial que permite el acceso a submenús desplegables, aunque no es un evento que se usa con regularidad, también identifica la dinámica del usuario y el entorno (Shen et al., 2013). Por otro lado, los robots son generados por software especializado para realizar el mayor número de solicitudes a un sistema (Zargar et al., 2013), sin el uso de ningún periférico. Vale la pena mencionar que las características presentadas en la Tabla 16, a pesar de ser utilizadas en el proceso biométrico para identificar a un usuario de otro, no se han demostrado en la diferenciación de usuarios reales y robots.

Tabla 16. **Características del usuario de acuerdo a la literatura**

Id	Características del mouse	Referencias
M1	Un solo click	Shen et al. (2013)
M2	Doble click	
M3	Desplazamiento del movimiento	
M4	Curva de velocidad contra el tiempo	
M5	Curva de aceleración contra el tiempo	
M6	Tiempo	Salmeron-Majadas et al. (2014)
M7	Movimiento	
M8	Botón izquierdo o derecho presionado o soltado	
M9	Coordenadas de un evento	
M10	Coordenadas de posición del ratón	Graepel et al. (2010)
M11	Trayectoria del ratón	

M12	Ángulo del camino en varias direcciones	
M13	Curvatura y su derivada	
M14	Movimiento del ratón	
M15	Velocidades angulares	
M16	Aceleración tangencial y tirón	Gamboa et al. (2003)
M17	Coordenada de movimiento del ratón	
M18	Ángulo del movimiento	
M19	Tiempo del movimiento	
M20	Tiempo de clics del mouse	
	Características del teclado	
T1	Número de eventos de pulsación de tecla	Shen et al. (2013)
T2	Tiempo promedio entre eventos de pulsación de tecla	
T3	Tiempo promedio por toque	
T4	Número de veces que una tecla dada ha sido presionada	

Fuente: Autor

4.1.2. Captura de características

La Tabla 16 describe las características del ratón que se pueden capturar y las técnicas utilizadas para tales fines. Estas características se pueden capturar utilizando software desarrollado en lenguajes de programación que incorporan bibliotecas o funciones especiales para este (Shen et al., 2013, Salmeron-Majadas et al., 2014, Graepel et al. 2010, Gamboa et al., 2003).

Tabla 17. **Extracción de las características del mouse de acuerdo a la literatura**

Id	Método de extracción	Referencias

M1	Windows application (written in C#)	Shen et al. (2013)
M2		
M3		
M4		
M5		
M6	Java (kSquared.de library)	Salmeron-Majadas et al. (2014)
M7		
M8		
M9		
M10	ND	Graepel et al. (2010)
M11		
M12		
M13		
M14		
M15		
M16		
M17	Java applet and javascript	Gamboa et al. (2003)
M18		
M19		
M20		

Fuente: Autor

4.1.3. Algoritmo de clasificación

La Figura 7 muestra el algoritmo de clasificación que permite la identificación de ataques DDoS mediante funciones del mouse. Las características propuestas permiten saber si hay un ataque o no, el proceso consiste en verificar si la solicitud de servicio

incluye al menos una de las características propuestas, que se considera un usuario humano, de lo contrario se considera un robot. El algoritmo calcula la tasa de precisión de los ataques DDoS verificando el número de ataques encontrados por el algoritmo entre los números de ataques reales en el conjunto de datos.

```
Input: Dataset
begin
Query right click, mouse movement, request URL
Loop Dataset
if request URL is active
   if right click is active or mouse movement is active
      add user;
   else
      add attack;
Query abnormal URL
   accuracy is equal request – abnormal_URL;
end
output accuracy;
```

Figura 7. **Algoritmo de clasificación para usuarios reales y robots**

4.2. Experimentos numéricos

4.2.1 Criterios de detección

La Figura 8 muestra la arquitectura del entorno de validación utilizado para la construcción del algoritmo de clasificación de usuarios reales y ataques DDoS. En él, consideramos el conjunto de datos de entrada dado por Lichman 2013, y que se analiza

en la sección 4.2. También se observó el uso del administrador de base de datos MySQL para la extracción de las características que se usaron en la validación, a fin de crear un nuevo conjunto de datos con las características seleccionadas. Entra en la aplicación creada en Java para el proceso de clasificación. Cabe señalar que el algoritmo de clasificación, el mismo mencionado en la sección 4.1.3, permite la evaluación de las dos características de interacción para la detección de ataques informáticos, que son: movimiento del mouse y clic derecho. Finalmente, se generan informes de resultados, en los que se muestra el número total de ataques DDoS y usuarios reales encontrados, así como el tiempo total empleado en la ejecución de todo el proceso.

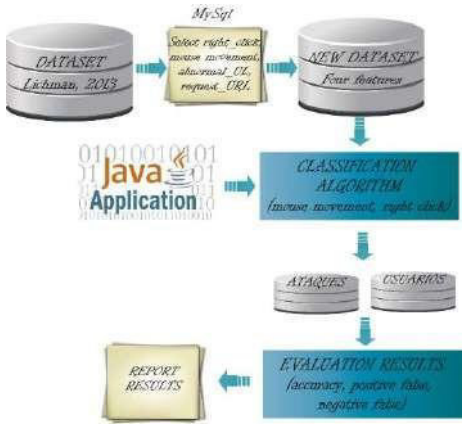

Figura 8. **Arquitectura del entorno de validación**

4.2.2. Conjunto de datos

El conjunto de datos utilizado en este trabajo para el proceso de validación del algoritmo de clasificación fue creado por Lichman 2013. Contiene 11055, de los cuales 9096 son usuarios reales y el resto son ataques DDoS. Se seleccionó este conjunto de datos porque informa las características del mouse que se evaluará. Además, este conjunto de datos contiene 31 atributos de los cuales se extrajeron cuatro para realizar la validación (clic derecho, movimiento del mouse, URL anormal y URL de solicitud).

Cabe señalar que, a través de la función de solicitud de URL, se sabe si se realizó una solicitud al sistema o no. Por otro lado, la URL anormal permite identificar las solicitudes que son ataques informáticos.

4.2.3. Extracción de características

La Figura 9 muestra el algoritmo general que extrae las características propuestas en este trabajo. Para ello, una solicitud activa se identifica en el set con los datos para identificar a continuación las variables propuestas. Las características son extraídas por consultas SQL a la base de datos. Después de ejecutar las consultas, todos los registros se obtienen cuando se solicita un servicio o recurso para su posterior análisis e informe de los resultados.

```
Input: Request = active
begin
Open database
Query = Select mouse_movement,
right_click from dataset
Execute Query
end
```

Figura 9. **Algoritmo para la extracción de características**

4.2.4. Resultados

El algoritmo utilizado para implementar los criterios de clasificación se creó en Java versión 1.8.0 usando NetBeans IDE 8.2. Las pruebas se desarrollaron en una máquina cuyo procesador es Intel (R) Core (TM) i7 CPU 2.60 GHz, 8 GB de RAM, con sistema operativo Windows 10. La Tabla 18 muestra la tasa de detección de ataques obtenida usando las dos características del mouse, esto es 100%, tanto para el número de usuarios reales como para el número de ataques DDoS. Este resultado muestra que con

el uso de software diseñado para la detección de ataques y el uso de las dos características del dinamismo del usuario, se alcanza la tasa de precisión más alta. Vale la pena mencionar que el tiempo utilizado por la aplicación para realizar la clasificación fue de 50 milisegundos. Cabe mencionar que en este trabajo es difícil identificar falsos positivos y negativos, porque se usa un conjunto de datos con datos exactos, donde se observa la interacción del usuario real en las solicitudes realizadas. Por lo tanto, cuando se realiza una solicitud, esto se hace a través de la interacción con el mouse, de lo contrario es un ataque DDoS. Sin embargo, se puede decir que con el uso de más características y medios de entrada de datos, podría haber casos de falsos positivos y negativos. Estos porcentajes muestran la importancia de estas características para la detección de este tipo de ataque informático.

Tabla 18. **Tasa de detección de las características del mouse propuestas**

Usuarios	**Datos reales**	**Criterios de detección**	**Tasa de precisión (%)**	**Tiempo de ejecución (mil)**
Real users	9096	9096	100	90
DDoS attacks	1959	1959	100	

Fuente: Autor

4.2.5. Discusión

Los resultados obtenidos en las pruebas realizadas muestran que todos los ataques DDoS no tienen las características del mouse y del clic derecho, por lo que su detección es del 100%. Las características evaluadas (movimiento del mouse y clic derecho) muestran el dinamismo del usuario. Por lo tanto, estas características permiten diferenciar una solicitud real de un ataque informático. Usan un bajo costo para la

detección de un ataque contra otras características propuestas en la literatura, ya que el algoritmo utilizado consume pocos recursos debido a la simplicidad del código programado. Estas características también le permiten detectar comportamientos de usuario que otras características no. Por ejemplo, operaciones de mouse que no se habían propuesto en otros trabajos dirigidos a detectar ataques DDOS. Cabe mencionar que existen otras características del dinamismo del usuario que pueden ser consideradas para la identificación de usuarios reales y robots (teclado). Sin embargo, con el uso de más funciones y medios de entrada de datos, aparecerían casos de falsos positivos y negativos. También se debe tener en cuenta que con el avance en los mecanismos de detección de ataques, los atacantes encuentran nuevas alternativas para eludir los mecanismos que se proponen. Por lo tanto, en el futuro, los atacantes podrían falsificar las variables que miden las características del comportamiento del usuario, simulando los datos de entrada e identificando a un robot como un usuario real.

CAPÍTULO V. MÉTODO ÁGIL PARA DETECTAR ATAQUES DDOS EN LA CAPA DE APLICACIÓN SEGÚN EL DINAMISMO DEL USUARIO

5.1 Método propuesto

En este trabajo presentamos un método de detección de bajo costo que permite detectar ataques DDoS orientados a la capa de aplicación. Para ello, utiliza características del dinamismo del usuario extraído en tiempo real. Estas características muestran la interacción del usuario con el sistema.

5.1.1. Arquitectura del método de detección

La Figura 10 muestra la arquitectura utilizada para la implementación del método de detección de ataques DDoS. En el mismo se observa la entrada de las solicitudes provenientes de internet a la interfaz de la aplicación web. Las solicitudes realizadas generan un banco de datos donde se registran las conexiones establecidas y los procesos realizados. El banco de datos generado en la capa de aplicación es analizado por un detector de interacción. En el nivel de aplicación, los procesos que genera el usuario se registran (enlaces, recursos, formularios, etc.). El detector registra la actividad entre el usuario y el mouse y los periféricos del teclado. Las características se extraen en tiempo real mediante programación en PHP y Javascript. Estas características se almacenan hasta que el usuario ejecuta la siguiente solicitud. Tanto la solicitud como las características del usuario se envían al algoritmo de detección de la Figura 10 para su evaluación. Como se indicó en la sección anterior, este algoritmo

es responsable de determinar la existencia de solicitudes e interacciones con el sistema, tomando una decisión entre un usuario real o un ataque informático.

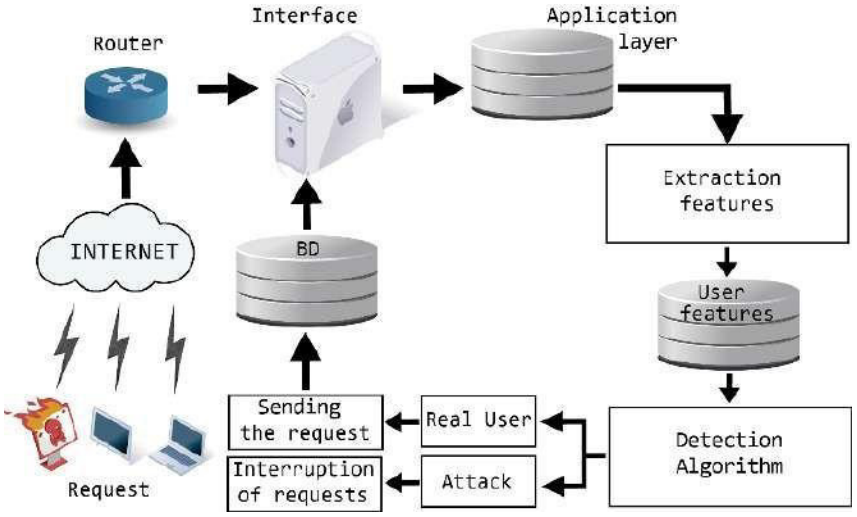

Figura 10. **Arquitectura del método de detección**

5.1.2. Dinamismo del usuario

El presente trabajo considera las características del dinamismo del usuario en el sistema informático. El dinamismo del usuario surge cuando el usuario interactúa con el sistema. Leiva et al. (2011), menciona que la dinámica del usuario es el interés de los usuarios y sus preferencias. El modelo de solicitudes de usuario y las respuestas del servidor proporcionan un conocimiento limitado sobre el comportamiento del usuario. Para una mejor compresión es mejor moverse hacia el lado del cliente. Para hacer esto, recopile información como mover el mouse, hacer clic, difuminar o

cambiar el tamaño. Kundu et al. (2012), mencionan que una alternativa para predecir la próxima página web que abrirá un usuario, proviene del dinamismo del movimiento del mouse con la dirección que toma en la interfaz gráfica. Sznur et al. (2015), proponen una técnica para identificar usuarios agrupando dinámicas de pulsación de teclas. Kulkarni et al. (2012), utilizaron la dinámica del pulso, que utiliza el ritmo y la forma en que un individuo escribe los caracteres en el teclado, se utiliza como biométrica de comportamiento. Los ritmos de pulsación de un usuario, en términos de tiempo, se miden para desarrollar una plantilla biométrica única del patrón de escritura del usuario para una futura autenticación. Bravo et al. (2017) evaluaron las características del mouse para identificar usuarios reales de ataques DDOS. Verificar el dinamismo del mouse proporciona características únicas para identificar este tipo de ataque.

5.1.3. Características del dinamismo del usuario.

Las características del comportamiento del usuario se extraen de los procesos entre los periféricos utilizados y la interacción con el sistema. En este trabajo, el dinamismo del usuario se observa a través de las transacciones que se realizan con el mouse y los periféricos del teclado. La Tabla 18 muestra las características de usuario que se extraen y utilizan en el método de detección de ataque DDoS propuesto. Vale la pena mencionar que estas características se extraen utilizando las funciones de PHP y Javascript en tiempo real.

Tabla 19. **Características del dinamismo del usuario**

Id	Features	Descripción
f1	Movimiento del mouse	El mouse se mueve a una ubicación en la pantalla para realizar una acción.

f2	Click del mouse	Cuando un usuario presiona y suelta un botón del mouse y hay cinco tipos de eventos de clic que se registran: clic izquierdo, clic derecho y doble clic izquierdo.
f3	Punto culminante del mouse	Esta acción comienza con un clic / retención izquierdo del ratón para comenzar el resaltado y termina con la liberación del mouse.
f4	Arrastre del ratón	Cuando un objeto es arrastrado y soltado. Esta acción comienza con un clic / retención izquierdo del mouse y termina con la liberación del mouse.
f5	Caída del ratón	
f6	Desplazamiento del ratón	La rueda o el desplazamiento del ratón es un evento en el que el movimiento de la rueda o el desplazamiento tiene un efecto de red hacia arriba o hacia abajo. El efecto resultante se basa en los movimientos de rueda o desplazamiento consecutivos.
f7	Rueda del ratón	
f8	Pulsaciones de teclas	Esto sucede cuando un usuario presiona una tecla y desliza el dispositivo táctil (dedo o lápiz)

Fuente: Autor

5.1.4. Algoritmo de validación para las características de detección

La Figura 11 muestra el algoritmo de clasificación que permite la identificación de ataques DDoS mediante las funciones del mouse. Las características propuestas permiten saber si hay un ataque o no, el proceso consiste en verificar si la solicitud de servicio incluye al menos una de las características propuestas, que se considera un usuario humano, de lo contrario se considera un robot. El algoritmo calcula la tasa de precisión de los ataques DDoS verificando el número de ataques encontrados por el algoritmo entre los números de ataques reales en el conjunto de datos.

```
1.  Input dataset
2.  begin
3.  Query right click, mouse movement, request URL
4.  Loop Dataset
5.  if request URL is active
6.      if right click is active or mouse movement is active
7.          add user
8.      else
9.          add attack
10. Query abnormal URL
11.     accuracy is equal request – abnormal_URL
12. end
13. output accuracy
```

Figura 11. **Algoritmo de detección de ataques DDoS**

5.1.5. Arquitectura del método de detección.

La Figura 12 muestra el algoritmo utilizado para capturar las características del dinamismo del usuario. El algoritmo funciona cada vez que el usuario realiza una operación con el mouse o el teclado y su interacción con la interfaz gráfica. Cuando un usuario interactúa con el mouse y el teclado, se registra mediante una función de Javascript. Las características capturadas se almacenan en un registro para ser enviado a continuación para ser verificado por el algoritmo de detección. Cuando un usuario solicita un servicio, el usuario se ve obligado a usar un periférico para realizar la solicitud. La captura de las características del usuario consiste en tomar las pulsaciones que el usuario está realizando con los periféricos. Cuando un usuario interactúa con un periférico, es registrado por una función de Javascript en un banco de datos o registro.

```
1. begin catch
2. when interaction then
3. id = operation
4. if request then
5. submit id
6. end  catch
```

Figura 8. **Algoritmo de captura de características (ACF)**

5.1.6. Algoritmo de detección y mitigación

La idea principal del algoritmo es verificar si la solicitud realizada al sistema presenta alguna de las características del usuario web para diferenciar a un usuario real de un ataque informático en tiempo real.

```
1. Input request
2. begin verification
3. for i equal 1 to n
4. if fi stores true then id stored true
5. end verification
6. begin send each request
7. when id stores true then execute query request
8. when id stores false then execute message
9. end send
```

Figura 13. **Algoritmo de detección de ataques DDoS (ADDA)**

En la Figura 13 se presenta el algoritmo de detección de ataques propuesto, utiliza las características del dinamismo del usuario web que se muestran en la Tabla 18 y verifica si están activas o no. Para este propósito, 1) se realiza una solicitud al sistema, 2) se inicia la verificación de las características capturadas, 3) se utiliza un bucle repetitivo

que va de una a la totalidad de las características utilizadas en este trabajo, 4) si las analizadas características que han sido activadas, las activaciones realizadas se almacenarán en otra variable, 5) y se completará la verificación de las características del dinamismo del usuario. 6) Cuando se realiza una solicitud, se deben verificar las características del dinamismo del usuario. 7) Cuando la variable que almacena la verificación está activa, se realiza la solicitud. 8) Cuando la variable que almacena la verificación está inactiva, se envía un mensaje que debe ser respondido por el usuario, de lo contrario no se dará la solicitud. Este último paso es mitigar el algoritmo de que las solicitudes de los atacantes se envían al servidor.

5.2. Experimentos numéricos

5.2.1. Diseño experimental

Para validar el algoritmo propuesto, se consideraron los servicios web de un hotel en la ciudad de Detroit en los Estados Unidos. Recibe 3100 solicitudes semanalmente. El hotel tiene un servicio de restaurante y la información está en un servidor dedicado que utiliza Linux CentOS, procesador de 4 núcleos, memoria de 8 GB, espacio en disco de 1 TB. La Figura 14 muestra el entorno de validación incorporado que permite la incorporación de tres niveles para la detección de ataques DDoS. El primer nivel involucra la interfaz del usuario, donde el usuario interactúa con el sistema realizando solicitudes de enlaces, videos, gráficos, etc. En el segundo nivel se ubican las funciones que cargan información a todas las aplicaciones del sistema. En este nivel se encuentran las funciones que realizan la llamada al algoritmo ACF. Finalmente, en el tercer nivel está el algoritmo ADDA. La respuesta del algoritmo tiene dos salidas, ejecutar la solicitud realizada por el usuario o enviar un mensaje de verificación.

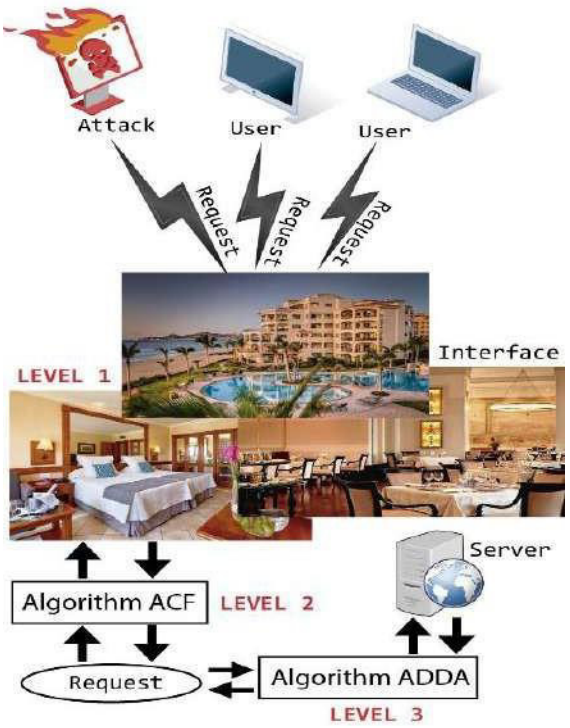

Figura 14. **Entorno de validación para la detección de ataques DDoS**

El servidor utilizado en este trabajo ha sido sometido a una serie de ataques simulados para verificar la eficiencia del método propuesto. Los resultados obtenidos se han extraído utilizando las mismas herramientas de ataque para su posterior análisis.

5.2.2. Simulación de ataques

Para generar ataques DDoS, se utilizaron el software LOIC (Low Orbit Ion Canon) (Zhang et al., 2017), OWASP DOS HTTP POST (Arafat et al., 2015) y GoldenEye HTTP (Jazi et al., 2017). Cabe mencionar que estas herramientas fueron seleccionadas porque son las más utilizadas para la generación de este tipo de ataques, debido a su simplicidad y efectividad (Jazi et al., 2017). Para hacer esto, se realizaron varios ataques con cada herramienta hacia el servidor del hotel (víctima), a fin de evaluar la tasa de ataque necesaria para sobrecargar el servidor. En cada ataque, se obtuvieron

los valores de sobrecarga, que luego se evaluarían utilizando el algoritmo de detección propuesto. La Tabla 20 muestra las herramientas que se utilizaron para simular el ataque al sistema web, la cantidad de soledades generadas y el tiempo que tomó el sistema para sobrecargarse.

Tabla 20. **Evaluación sin nigún método**

Herramienta de ataque	**Numbero de solicitudes**	**Tiempo de sobrecarga del sistemas (min)**
LOIC	4800	2.15
OWASP DOS HTTP POST	4000	1.30
GoldenEye HTTP Denial of Service Tool	5300	3.20

Fuente: Autor

Los resultados de la Tabla 20 muestran que los ataques informáticos generados por el software LOIC, OWASP y GoldenEye usan aproximadamente dos minutos para sobrecargar el sistema, lo que provoca la inaccesibilidad de los recursos y servicios para usuarios reales. También se observa que el número de solicitudes utilizadas para sobrecargar el sistema varía entre 4000 y 5300.

5.2.3. Resultados

La Tabla 21 muestra los resultados obtenidos utilizando el método de detección propuesto. Muestra que el 100% de los ataques generados por las herramientas han sido detectados de manera efectiva. El tiempo empleado en la detección fue en promedio de 60 milisegundos y se utilizaron las mismas cantidades de solicitudes de

simulación para generar el ataque. Vale la pena mencionar que no existen conjuntos de datos relacionados con los ataques DDoS para las pruebas. Además, los trabajos con la tasa de detección más alta en la capa de aplicación (Zolotukhin et al., 2016) y (Saravanan et al. 2016) no muestran las herramientas que se utilizaron para evaluar los métodos propuestos.

Tabla 21. **Evaluación con el método propuesto**

Herramientas de ataque	**Número de solicitudes**	**Tiempo de detección (mil)**	**Tasa de detección (%)**
LOIC	4800	60	100
OWASP DOS HTTP POST	4000	58	100
GoldenEye HTTP Denial of Service Tool	5300	63	100

Fuente: Autor

La Tabla 21 muestra que el mecanismo de detección desarrollado mediante el uso de las funciones de dinamismo del usuario web es efectivo con una tasa de detección del 100% para las tres herramientas de generación de ataques. Además, el tiempo empleado es de unos 60 milisegundos. Estos resultados muestran la efectividad del método de detección a través de la interacción del usuario con el sistema a través de los periféricos utilizados. Cabe mencionar que con la mejora de los mecanismos de detección, los atacantes también mejoran sus estrategias de ataque, por lo que no se descarta la posibilidad de que los valores de entrada de las características de usuario evaluadas en este trabajo puedan ser suplantados.

5.2.4. Resultados frente a otras propuestas de detección de ataques DDoS

La Tabla 22 muestra los resultados obtenidos por el método de detección propuesto en este trabajo frente a los métodos estudiados en el Capítulo III correspondiente al estado del arte. En la misma se observan cuatro estudios con mayor tasa de detección llegando a porcentajes desde el 98.31% hasta el 99.99%, también se muestran las características que se emplearon en cada una de ellas.

Tabla 22. **Resultados frente a otras propuestas de detección de ataques DDoS**

Estudios	**Tasa de detección (%)**	**Características empleadas**
Jia et al. (2017)	99.99	Tráfico de red Paquetes Flujos TCP
Hoque et al. (2017)	99.67	Dirección IP Flujos TCP
Kumar et al. (2011)	99.4	Tráfico de red
Johnson et al. (2016)	98.31	Dirección IP
Método inteligente	100.0	Movimiento del mouse Click del mouse Punto culminante del mouse Arrastre del ratón Caída del ratón Desplazamiento del ratón Rueda del ratón Pulsaciones de teclas

Fuente: Autor

La Tabla 22 muestra que el método ágil propuesto alcanza el 100% de tasa de detección en relación al método propuesto por Jia et al. (2017) con 99.99%. Este último, además se observa emplea tres características en método de detección frente al método propuesto que emplea ocho características del dinamismo del usuario web. También se observa que todos los autores utilizan características extraídas de la red de datos. El método propuesto analiza la dinámica del usuario en tiempo real, verificando las acciones realizadas luego de cada solicitud realizada por el usuario.

5.2.5. Discusión

Los resultados obtenidos en las pruebas aplicadas en el método inteligente propuesto, muestran que todos los ataques DDoS se han detectado en un 100%. Las características empleadas en el método de detección permiten identificar el dinamismo del usuario real. Por lo tanto, estas características son empleadas en un método de detección inteligente que detecta estas características para ser analizadas cuando se realiza una petición al servidor web. Los algoritmos de extracción y clasificación se encuentran en tres capas, las cuales permiten consumir pocos recursos debido a la simplicidad del código programado. Los ataques DDoS, una vez detectados, el método propuesto impide que se sigan procesando solicitudes de ese usuario. Sin embargo, se pueden plantear otras alternativas de mitigación, por ejemplo el uso de puzles para comprobar la legitimidad del usuario.

CAPÍTULO VI. CONCLUSIONES Y TRABAJOS FUTUROS

6.1 Conclusiones

La revisión sistemática de la literatura presentada en este estudio ha identificado los principales aspectos involucrados en los mecanismos de detección de ataques DDoS, centrándose en las técnicas, variables y herramientas, en lugar de en el punto de detección en el tiempo o la precisión de la detección. Un análisis de los resultados ha proporcionado respuestas a las cinco preguntas secundarias de investigación. Identificamos cuarenta y ocho técnicas que se utilizan en la detección de ataques DDoS. También se identificaron un total de veintiocho variables y fue evidente que las herramientas más utilizadas son Matlab y el simulador de red, debido a las funcionalidades y ventajas del procesamiento de la información. El lugar más utilizado para la implementación de un mecanismo es la Red, porque los flujos de datos se analizan antes de que lleguen al servidor. El punto más utilizado en el tiempo para el despliegue de una técnica es durante, ya que la detección se realiza en tiempo real durante el ataque. El mecanismo más efectivo para lograr una alta tasa de detección es el propuesto por Jia et. Al (2017), que alcanzó una presión del 99,9%.

Se han introducido 8 nuevas características basadas en el comportamiento del usuario web. Se extraen de la transaccionalidad del usuario con el sistema en tiempo real, por lo que son características económicas computacionalmente hablando debido a su fácil obtención. Con el fin de evaluar la utilidad de las variables propuestas, se realizaron pruebas utilizando un conjunto de datos de 11055 solicitudes entre usuarios reales y ataques. El conjunto de datos utilizado en las pruebas contiene dos de las 8 variables propuestas en este documento para la detección de ataques en la capa de aplicación.

La evaluación de las dos variables (movimiento del mouse y clic con el botón derecho), utilizando un software diseñado en Java, logró alcanzar el 100% de eficiencia en la diferenciación de usuarios reales y robots. Por lo tanto, las variables de clic derecho y movimiento del mouse se identifican como características del dinamismo del usuario, las mismas pueden considerarse para su implementación en los mecanismos de detección de ataques DDoS.

Se presenta un mecanismo de detección ágil (orden 1) y efectivo basado en las características del dinamismo del usuario web para la detección de ataques DDoS en la capa de aplicación. Este mecanismo emplea 8 nuevas características de comportamiento del usuario que no se han utilizado en ningún otro trabajo similar. El método para detectar ataques DDoS utilizando las características del comportamiento del usuario tiene una efectividad del 100% en la detección. Este resultado muestra la influencia de las características que identifican a un usuario al interactuar con el sistema. Las pruebas en una plataforma en tiempo real y la aplicación de las herramientas de ataque LOIC, OWASP y GoldenEye permiten evaluar el algoritmo en un entorno de ataque simulado. Estas simulaciones permitieron verificar que el algoritmo alcanza un resultado óptimo al procesar grandes cantidades de solicitudes.

6.2. Trabajos futuros

Entre los trabajos futuros que se desprenden del presente trabajo de investigación están la detección de ataques masivos generados por medio de agentes inteligentes. Este tipo de ataque podría llevar un proceso de detección más amplio debido a que un agente puede realizar solicitudes en forma independiente como un usuario real, lo cual pondría a prueba el método descrito en este trabajo. Otro trabajo que puede realizarse

a futuro está la incorporación del método propuesto en otros mecanismos tales como teléfonos móviles y/o tablets.

BIBLIOGRAFÍA

Arafat, M. Y., Alam, M. M. & Alam, M. F. (2015). A Practical Approach and Mitigation Techniques on Application Layer DDoS Attack in Web Server. International Journal of Computer Applications, 131(1). DOI: 10.5120/ijca2015907209

Abramson, M., & Aha, D. W. (2013). User Authentication from Web Browsing Behavior. In FLAIRS conference, (pp. 268-273). DOI: 10.1.1.975.4819

Akamai Community. (2017). State of the Internet/Security Q4 2017 Report. 28 pages. WEF-GRR18 www3.weforum.org/docs/WEF_ GRR18_Report.pdf

Al-Duwairi, B. & Manimaran, G. (2006). Distributed packet pairing for reflector based DDoS attack mitigation. Computer communications, vol. 29, no 12, p. 2269-2280. DOI: 10.1016/j.comcom.2006.03.007

Al-Duwairi, B., Al-Qudah, Z. & Govindarasu, M. (2013). A novel scheme for mitigating botnet-based DDoS attacks. Journal of Networks, vol. 8, no 2, p. 297. DOI: 10.1.1.369.3328.

Beak, C., Chaudhry, J. A., Lee, K., Park, S. & Kim, M. (2007). A novel packet marketing method in DDoS attack detection. American Journal of Applied Sciences, 2007, vol. 4, no 10, p. 741-745. DOI: 10.1.1.129.3079.

Behal, S. & Kumar, K. (2017). Detection of DDoS attacks and flash events using novel information theory metrics. Computer Networks, vol. 116, p. 96-110. DOI: 10.1016/j.comnet.2017.02.015

Bender, N. (2018).Two Sides of DDoS Attacks: The Largest Attack of All Time and Focus on SMEs. Dotmagazine.

Brosso, I., La Neve, A., Bressan, G. & Ruggiero, W. V. (2010). A continuous authentication system based on user behavior analysis. Availability, Reliability and Security. ARES'10 International Conference on IEEE. DOI: 10.1109/ARES.2010.63

Bulajoul, W., James, A., & Pannu, M. (2013). Network intrusion detection systems in high-speed traffic in computer networks. Proceedings of the 10th IEEE International Conference one-Business Engineering (ICEBE), pp 168–175. DOI: 10.1109/ICEBE.2013.26

Cepheli, Ö., Büyükçorak, S. & Karabulut Kurt, G. (2016). Hybrid intrusion detection system for ddos attacks. Journal of Electrical and Computer Engineering, vol. 2016. DOI: 10.1155/2016/1075648

Chen, R., Park, J. M. & Marchany, R. (2007). A divide-and-conquer strategy for thwarting distributed denial-of-service attacks. IEEE Transactions on Parallel and Distributed Systems, vol. 18, no 5, p. 577-588. DOI: 10.1109/TPDS.2007.1014

Chen, S. & Song, Q. (2005). Perimeter-based defense against high bandwidth DDoS attacks. IEEE Transactions on Parallel & Distributed Systems, no 6, p. 526-537. DOI: 10.1109/TPDS.2005.74

Chen, S. W., Wu, J. X., Ye, X. L. & Guo, T. (2013). Distributed denial of service attacks detection method based on conditional random fields. Journal of Networks, vol. 8, no 4, p. 858. DOI: 10.1.1.369.6378

Chen, Y. & Hwang, K. (2006). Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. Journal of Parallel and Distributed Computing, vol. 66, no 9, p. 1137-1151. DOI: 10.1016/j.jpdc.2006.04.007

Chen, Y., Das, S. & Dhar, P. (2008). Detecting and Preventing IP-spoofed Distributed DoS Attacks. IJ Network Security, vol. 7, no 1, p. 69-80. DOI: 10.1109/ARTCom.2009.167

Chen, Y., Hwang, K. & Ku, W. S. (2007). Collaborative detection of DDoS attacks over multiple network domains. IEEE Transactions on Parallel & Distributed Systems, no 12, p. 1649-1662. DOI: 10.1109/TPDS.2007.1111

Chen, Y., Ma, X. & Wu, X. (2013). DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory. IEEE Communications Letters, vol. 17, no 5, p. 1052-1054. DOI: 10.1109/LCOMM.2013.031913.130066

Choi, J., Choi, C., Ko, B., Kim, P. (2014). A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. Soft Computing, vol. 18, no 9, p. 1697-1703. DOI: 10.1007/s00500-014-1250-8

Chonka, A., Singh, J. & Zhou W. (2009). Chaos theory based detection against network mimicking DDoS attacks. IEEE Communication Letters, vol. 13, no 9, p. 717-719. DOI: 10.1109/LCOMM.2009.090615

Cloudbric (2015). Who's Behind DDoS Attacks and How Can You Protect Your Website? Retrieved 15 September 2015.

Criscuolo, P. J. (2000). Distributed denial of service, tribe flood network 2000, and stacheldraht CIAC-2319, Department of Energy Computer Incident Advisory Capability (CIAC). UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory.

Dickinson, S., & Pizlo, Z. (2013). Shape perception in Human & Computer Vision. New York, NY: Springer.

Dittrich, D. (1999). The stacheldraht distributed denial of service attack tool. University of Washington. Retrieved 2013-12-11.

Doron, E. & Wool, A. (2011) Wda: A web farm distributed denial of service attack attenuator. Computer Networks, vol. 55, no 5, p. 1037-1051. DOI: https://doi.org/10.1016/j.comnet.2010.05.001

François, J., Aib, I. & Boutaba, R. (2012). FireCol: a collaborative protection network for the detection of flooding DDoS attacks. IEEE/ACM Transactions on Networking (TON), vol. 20, no 6, p. 1828-1841. DOI: 10.1109/TNET.2012.2194508

Gamboa, H., & Fred, A. L. (2003). An Identity Authentication System Based on Human Computer Interaction Behaviour. In PRIS (pp. 46-55).

Gavrilis, D., & Dermatas, E. (2005). Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. Computer Networks, 48(2), 235-245. DOI: 10.1016/j.comnet.2004.08.014

Ghezzi, C., Pezzè, M., Sama, M., & Tamburrelli, G. (2014). Mining behavior models from user-intensive web applications. In Proceedings of the 36th International Confer-ence on Software Engineering (pp. 277-287). ACM. DOI: 10.1145/2568225.2568234

Ginovsky, J. (2014). What you should know about worsening DDoS attacks. ABA Banking Journal. Archived from the original on 2014-02-09.

Giralte, L. C., Conde, C., De Diego, I. M. & Cabello, E. (2013). Detecting denial of service by modelling web-server behaviour. Computers & Electrical Engineering, vol. 39, no 7, p. 2252-2262. DOI: 10.1016/j.compeleceng.2012.07.004

Graepel, T., Candela, J. Q., Borchert, T., & Herbrich, R. (2010). Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. Omnipress.

Greenberg, A. (2015). Akamai warns of increased activity from DDoS extortion group. SC Magazine. Retrieved 15 September 2015.

Gu, Y., Wang, Y., Yang, Z., Xiong, F. & Gao, Y. (2017). Multiple-Features-Based Semisupervised Clustering DDoS Detection Method. Mathematical Problems in Engineering, vol. 2017. DOI: 10.1155/2017/5202836

Gupta, B., Joshi, R., and Misra, M. (2017). Distributed Denial of Service Prevention Techniques. International Journal of Computer and Electrical Engineering, Vol. 2, no. 2, pp. 268-276.

Higgins, K. J. (2013). DDoS Attack Used 'Headless' Browser In 150-Hour Siege. Dark Reading.

Hoque, N., Kashyap, H. & Bhattacharyya, D. K. (2017). Real-time DDoS attack detection using FPGA. Computer Communications, vol. 110, p. 48-58. DOI: 10.1016/j.comcom.2017.05.015

Huang, C., Wang, J., Wu, G. & Chen, J. (2014). Mining Web User Behaviors to Detect Application Layer DDoS Attacks. JSW, vol. 9, no 4, p. 985-990. DOI: 10.4304/jsw.9.4.985-990

IBM. (2017). 2017 Ponemon Cost of Data Breach Study. Retrieved from https://www.ibm.com/security/data-breach

Jain, A., & Singh, A. K. (2012). Distributed denial of service (ddos) attacks-classification and implications. Journal of Information and Operations Management, 2012, vol. 3, no 1, p. 136.

Jazi, H. H., Gonzalez, H., Stakhanova. N. & Ghorbani, A. A. (2017). Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. Computer Networks, vol. 121, p. 25-36. DOI: 10.1016/j.comnet.2017.03.018

Jia, B., Huang, X., Liu, R., Ma, Y. (2017). A DDoS attack detection method based on hybrid heterogeneous multiclassifier ensemble learning. Journal of Electrical and Computer Engineering, vol. 2017. DOI: 10.1155/2017/4975343

Johnson Singh, K., Thongam. K. & De, T. (2016). Entropy-Based Application Layer DDoS Attack Detection Using Artificial Neural Networks. Entropy, vol. 18, no 10, p. 350. DOI: 10.3390/e18100350

Kalkan, K. & Alagöz, F. (2016). A distributed filtering mechanism against DDoS attacks: ScoreForCore. Computer Networks, vol. 108, p. 199-209. DOI: 10.1016/j.comnet.2016.08.023

Kim, Y., & Kim, I. (2014). Involvers' Behavior-based Modeling in Cyber Targeted Attack. Proceedings of SECURWARE.

Kim, Y., Lau, W. C., Chuah, M. C. & Chao, H. J. (2006). PacketScore: a statistics based packet filtering scheme against distributed denial-of-service attacks. IEEE transactions on dependable and secure computing, vol. 3, no 2, p. 141-155. DOI: 10.1109/TDSC.2006.25

Kitchenham, B. (2004). Procedures for performing systematic reviews. Keele, UK, Keele University, vol. 33, no 2004, p. 1-26.

Kumar, P. A. R. & Selvakumar, S. (2011). Distributed denial of service attack detection using an ensemble of neural classifier. Computer Communications, vol. 34, no 11, p. 1328-1341. DOI: 10.1016/j.comcom.2011.01.012

Kumar, P. A. R. & Selvakumar, S. (2013). Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. Computer Communications, vol. 36, no 3, p. 303-319. DOI: 10.1016/j.comcom.2012.09.010

Lee, N. (2013). Counterterrorism and Cybersecurity: Total Information Awareness. Springer. ISBN 9781461472056.

Lee, F. Y. & Shieh, S. (2005). Defending against spoofed DDoS attacks with path fingerprint. Computers & Security, vol. 24, no 7, p. 571-586. DOI: 10.1016/j.cose.2005.03.005

Lee, K., Kim, J., Kwon, K. H., Han, Y. & Kim S. (2008). DDoS attack detection method using cluster analysis. Expert systems with applications, vol. 34, no 3, p. 1659-1665. DOI: 10.1016/j.eswa.2007.01.040

Lee, S. M., Kim, D. S., Lee, J. H. & Park, J. S. (2012). Detection of DDoS attacks using optimized traffic matrix. Computers & Mathematics with Applications, vol. 63, no 2, p. 501-510. DOI: 10.1016/j.camwa.2011.08.020

Li, L. & Lee, G. (2005). DDoS attack detection and wavelets. Telecommunication Systems, vol. 28, no 3-4, p. 435-451. DOI: 10.1007/s11235-004-5581-0

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu /ml]. Irvine, CA: University of California, School of Information and Computer Science.

Liu, H., Sun, Y., Kim, M. S. (2011). A Scalable DDoS Detection Framework with Victim Pinpoint Capability. JCM, vol. 6, no 9, p. 660-670. DOI:10.4304/jcm.6.9.660-670

Lu, W. Z., Gu, W. X. & Yu, S. Z. (2009). One-way queuing delay measurement and its application on detecting DDoS attack. Journal of Network and Computer Applications, vol. 32, no 2, p. 367-376.

Luo, H., Chen, Z., Li, J. & Vasilakos, A. V. (2017). Preventing distributed denial-of-service flooding attacks with dynamic path identifiers. IEEE Transactions on

Information Forensics and Security, vol. 12, no 8, p. 1801-1815. DOI: 10.1109/TIFS.2017.2688414

Luo, H., Lin, Y., Zhang, H. & Zukerman, M. (2013). Preventing DDoS attacks by identifier/locator separation. IEEE network, vol. 27, no 6, p. 60-65. DOI: 10.1109/MNET.2013.6678928

Luo, J., Yang, X., Wang, J., Xu, J., Sun, J. & Long, K. (2014). On a Mathematical Model for Low-Rate Shrew DDoS. IEEE Trans. Information Forensics and Security, vol. 9, no 7, p. 1069-1083. DOI: 10.1109/TIFS.2014.2321034

Loukas, G., Oke, G. (2010). Protection against Denial of Service Attacks: A Survey. Comput. J. 53 (7): 1020–1037. doi:10.1093/comjnl/bxp078

Ma, X. & Chen, Y. (2014). DDoS detection method based on chaos analysis of network traffic entropy. IEEE Communications Letters, vol. 18, no 1, p. 114-117. DOI: 10.1109/LCOMM.2013.112613.132275

Malialis, K. & Kudenko, D. (2015). Distributed response to network intrusions using multiagent reinforcement learning. Engineering Applications of Artificial Intelligence, vol. 41, p. 270-284. DOI: 10.1016/j.engappai.2015.01.013

Meenakshi, S. & Srivatsa, S. K. (2007). A distributed framework with less false positive ratio against distributed denial of service attack. Information Technology Journal, vol. 6, no 8, p. 1139-1145. DOI: 10.3923/itj.2007.1139.1145

Mirkovic, J. (2003). D-WARD: source end defense against distributed denial of service attacks, Ph.D. thesis, University of California.

Mirkovic, J., & Reiher, P. (2005). D-WARD: a source-end defense against flooding denial-of-service attacks. IEEE transactions on Dependable and Secure Computing, vol. 2, no 3, p. 216-232. DOI: 10.1109/TDSC.2005.35

Mirvaziri, H. (2017). A new method to reduce the effects of HTTP-Get Flood attack. Future Computing and Informatics Journal, vol. 2, no 2, p. 87-93. DOI: 10.1016/j.fcij.2017.07.003

Netscout Arbor. (2017). Insight into the global threat landscape. 93 pages. 2017.

Nguyen, H. V., & Choi, Y. (2008). Proactive detection of DDoS attacks using k-NN classifier in an Anti-DDoS Framework. International Journal of Computer System Sci-ence and Engineering, 247-252.

Ni, T., Gu, X. & Wang, H. (2014). Detecting DDoS Attacks Against DNS Servers Using Time Series Analysis. Indonesian Journal of Electrical Engineering and Computer Science, vol. 12, no 1, p. 753-761. DOI:10.11591/telkomnika.v12i1.3355

Nunes, I., Schardong, F. & Schaeffer-Filho, A. (2017). BDI2DoS: an application using collaborating BDI agents to combat DDoS attacks. Journal of Network and Computer Applications, vol. 84, p. 14-24. DOI: 10.1016/j.jnca.2017.01.035

Nyakundi, E. (2015). Using support vector machines in anomaly intrusion detection. Tesis Doctoral.

Oikonomou, G. & Mirkovic, J. (2009). Modeling human behavior for defense against flash-crowd attacks. Communications. ICC'09. IEEE International Conference on IEEE, 2009. DOI: 10.1109/ICC.2009.5199191

Peng, T., Leckie, C., and Ramamohanarao, K. (2007). Survey of network based defense mechanisms countering the DoS and DDoS problems, Computer Journal of ACM Computing Surveys, vol. 39, no. 1, pp. 123-128. DOI: 10.1145/1216370.1216373

Peraković, D., Periša, M., Cvitić, I. & Husnjak, S. (2017). Model for detection and classification of DDoS traffic based on artificial neural network. Telfor Journal, vol. 9, no 1, p. 26. DOI: 10.5937/telfor1701026P

Prasad, K. M., Reddy, A. R. M. & Rao, K. V. (2017). BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web. Journal of King Saud University-Computer and Information Sciences. DOI: 10.1016/j.jksuci.2017.07.004

Rahmani, H., Sahli, N. & Kamoun, F. (2012). DDoS flooding attack detection scheme based on F-divergence. Computer Communications, vol. 35, no 11, p. 1380-1391. DOI: 10.1016/j.comcom.2012.04.002

Ranjan, S., Swaminathan, R., Uysal, M., & Knightly, E. W. (2006). DDoS-Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection. In INFOCOM. DOI: 10.1109/INFOCOM.2006.127

Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A. & Knightly, E. (2009). DDoS-shield: DDoS-resilient scheduling to counter application layer attacks. IEEE/ACM Transactions on networking, vol. 17, no 1, p. 26-39. DOI: 10.1109/TNET.2008.926503

Sachdeva, M. & Kumar, K. (2014). A traffic cluster entropy based approach to distinguish DDoS attacks from flash event using DETER testbed. ISRN Communications and Networking, vol. 2014. DOI: 10.1155/2014/259831

Sachdeva, M., Kumar, K. & Singh, G. (2016). A comprehensive approach to discriminate DDoS attacks from flash events. Journal of Information Security and Applications, vol. 26, p. 8-22. DOI: 10.1016/j.jisa.2015.11.001

Saied, A., Overill, R. E. & Radzik, T. (2016). Detection of known and unknown DDoS attacks using Artificial Neural Networks. Neurocomputing, vol. 172, p. 385-393. DOI: 10.1016/j.neucom.2015.04.101

Saleh, M. A. & Abdul Manaf, A. (2015). A novel protective framework for defeating HTTP-based denial of service and distributed denial of service attacks. The Scientific World Journal, vol. 2015. DOI: 10.1155/2015/238230

Salmeron-Majadas, S., Santos, O. C., & Boticario, J. G. (2014). An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective mod-eling in an educational context. Procedia Computer Science, 35, 691-700. DOI: https://doi.org/10.1016/j.procs.2014.08.151

Saravanan R., Shanmuganathan S. and Palanichamy Y., Behavior-based detection of application layer distributed denial of service attacks during flash events. Turkish Journal of Electrical Engineering & Computer Sciences, 24(2), 510-523, 2016. DOI: 10.3906/elk-1308-188

Seo, D., Lee, H. & Perrig, A. (2013). APFS: adaptive probabilistic filter scheduling against distributed denial-of-service attacks. Computers & Security, vol. 39, p. 366-385. DOI: 10.1016/j.cose.2013.09.002

Shen, C., Cai, Z., Guan, X., Du, Y., & Maxion, R. A. (2013). User authentication through mouse dynamics. IEEE Transactions on Information Forensics and Security, 8(1), 16-30. DOI: 10.1109/TIFS.2012.2223677

Shiaeles, S. N., Katos, V., Karakos, A. S. & Papadopoulos, B. K. (2012). Real time DDoS detection using fuzzy estimators. Computers & security, vol. 31, no 6, p. 782-790. DOI: 10.1016/j.cose.2012.06.002

Singh, K. J. & De, T. (2017). MLP-GA based algorithm to detect application layer DDoS attack. Journal of Information Security and Applications, vol. 36, p. 145-153. DOI: 10.1016/j.jisa.2017.09.004

Solon, O. (2015). Cyber-Extortionists Targeting the Financial Sector Are Demanding Bitcoin Ransoms. Bloomberg. Retrieved 15 September 2015.

Spyridopoulos, T., Karanikas, G., Tryfonas, T. & Oikonomou, G. (2013). A game theoretic defence framework against DoS/DDoS cyber attacks. Computers & Security, vol. 38, p. 39-50. DOI: 10.1016/j.cose.2013.03.014

Sreeram, I. & Vuppala, V. P. K. (2017). HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm. Applied Computing and Informatics. DOI: 10.1016/j.aci.2017.10.003

Stevanovic, D., & Vlajic, N. (2014). Application-layer DDoS in dynamic Web-domains: Building defenses against next-generation attack behavior. In Communica-tions and Network Security (CNS), 2014 IEEE Conference on (pp. 490-491). DOI: 10.1109/CNS.2014.6997519

Tawdar, A. P., Bewoor, M. S., & Patil, S. H. (2017). Incremental Approach of Neural Network in Back Propagation Algorithms for Web Data Mining. IAES International Journal of Artificial Intelligence, 6(2), 74. DOI: http://doi.org/10.11591/ijai.v6.i2.pp74-78

Tiruchengode, N. (2012). Dynamic approach to defend against distributed denial of service attacks using an adaptive spin lock rate control mechanism. Journal of Computer Science, vol. 8, no 5, p. 632-636. DOI : 10.3844/jcssp.2012.632.636

Udhayan, J. & Babu, M. R. (2013). Deteriorating distributed denial of service attack by recovering zombies using penalty scheme. Journal of Computer Science, vol. 9, no 11, p. 1618. DOI: 10.3844/jcssp.2013.1618.1625

Urban, R. J. (2015). Detection of exit behavior of an Internet user. U.S. Patent Application No 14/829,409.

Varalakshmi, P. & Selvi, S. T. (2013). Thwarting DDoS attacks in grid using information divergence. Future Generation Computer Systems, vol. 29, no 1, p. 429-441. DOI: 10.1016/j.future.2011.10.012

Verisign. (2017). Verisign Distributed Denial of Service Trends Report Q4. 11 pages.

Verisign. (2018). Verisign Distributed Denial of Service Trends Report Q3. 11 pages.

Waguih, H. (2013). A data mining approach for the detection of denial of service attack. IAES International Journal of Artificial Intelligence, vol. 2, no 2, p. 99.

Wang, D., Chang, G., Feng, X., & Guo, R. (2008). Research on the detection of distributed denial of service attacks based on the characteristics of IP flow. In IFIP International Conference on Network and Parallel Computing (pp. 86-93). Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-88140-7_8

Wang, F., Wang, H., Wang, X. & Su, J. (2012). A new multistage approach to detect subtle DDoS attacks. Mathematical and Computer Modelling, vol. 55, no 1-2, p. 198-213. DOI: 10.1016/j.mcm.2011.02.025

Wang, H., Jin, C. & Shin, K. G. (2007). Defense against spoofed IP traffic using hop-count filtering. IEEE/ACM Transactions on Networking (ToN), vol. 15, no 1, p. 40-53. DOI: 10.1109/TNET.2006.890133

Wang, Y. & Sun, R. (2014). An IP-traceback-based packet filtering scheme for eliminating DDoS attacks. Journal of Networks, vol. 9, no 4, p. 874. DOI: DOI: 10.4304/jnw.9.4.874-881

Wu, X. & Chen, Y. (2013). Validation of chaos hypothesis in NADA and improved DDoS detection algorithm. IEEE Communications Letters, vol. 17, no 12, p. 2396-2399. DOI: 10.1109/LCOMM.2013.102913.130932

Wu, Y., Zhao, Z., Bao, F. & Deng, R. H. (2015). Software puzzle: A countermeasure to resource-inflated denial-of-service attacks. IEEE Transactions on Information forensics and security, vol. 10, no 1, p. 168-177. DOI: 10.1109/TIFS.2014.2366293

Xiang, Y., & Zhou, W. (2005). Mark-aided distributed filtering by using neural network for DDoS defense. In GLOBECOM'05: IEEE Global Telecommunications Conference, 28 November-2 December 2005 St. Louis, Missouri, USA, discovery past and future (pp. 1701-1705). IEEE Globecom. DOI: 10.1109/GLOCOM.2005.1577940

Xiang, Y., Li, K. & Zhou, W. (2011). Low-rate DDoS attacks detection and traceback by using new information metrics. IEEE transactions on information forensics and security, vol. 6, no 2, p. 426-437. DOI: 10.1109/TIFS.2011.2107320

Xiao, P., Qu, W., Qi, H. & Li, Z. (2015). Detecting DDoS attacks against data center with correlation analysis. Computer Communications, vol. 67, p. 66-74. DOI: 10.1016/j.comcom.2015.06.012

Xie, Y. & Yu, S. Z. (2009). Monitoring the application-layer DDoS attacks for popular websites. IEEE/ACM Transactions on Networking (TON), vol. 17, no 1, p. 15-25. DOI: 10.1109/TNET.2008.925628

Yaar, A., Perrig, A. & Song, D. (2006). StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense. IEEE Journal on Selected Areas in Communications, vol. 24, no 10, p. 1853-1863. DOI: 10.1109/JSAC.2006.877138

Yan, R. & Zheng, Q. (2009). Using Renyi cross entropy to analyze traffic matrix and detect DDoS attacks. Information Technology Journal, vol. 8, no 8, p. 1180-1188.

Yau, D. K., Lui, J., Liang, F. & Yam, Y. (2005). Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. IEEE/ACM Transactions on Networking (TON), vol. 13, no 1, p. 29-42. DOI: 10.1109/IWQoS.2002.1006572

Yu, S., Zhou, W., Jia, W., Guo, S., Xiang, Y. & Tang, F. (2012). Discriminating DDoS attacks from flash crowds using flow correlation coefficient. IEEE Transactions on Parallel and Distributed Systems, vol. 23, no 6, p. 1073-1080. DOI: 10.1109/TPDS.2011.262

Zargar, S., Joshi, J. & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE communications surveys & tutorials, vol. 15, no 4, p. 2046-2069. DOI: 10.1109/TPDS.2011.262

Zhang, C., Cai, Z., Chen, W., Luo, X. & Yin, J. (2012). Flow level detection and filtering of low-rate DDoS. Computer Networks, vol. 56, no 15, p. 3417-3431. DOI: 10.1016/j.comnet.2012.07.003

Zhang, L. Y., Ming, Q. I. A. N. & Chi, Y. B. (2017). DDoS Attack Detection Using Sliding Window Method. DEStech Transactions on Computer Science and Engineering. DOI: 10.1016/j.comnet.2012.07.003

Zhou, L., Liao, M., Yuan, C. & Zhang, H. (2017). Low-Rate DDoS Attack Detection Using Expectation of Packet Size. Security and Communication Networks, vol. 2017. DOI: 10.1155/2017/3691629

Zolotukhin, M., Kokkonen, T., Hämäläinen, T., & Siltanen, J. (2016). On Application Layer DDoS Attack Detection in High-Speed Encrypted Networks.

ANEXO 1

Indonesian Journal of Electrical Engineering and Computer Science
Vol. 14, No. 1, April 2019, pp. 155~168
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v14.i1.pp155-168

Systematic review of aspects of DDoS attacks detection

Silvia Bravo[1], David Mauricio[2]
[1]Faculty of Engineering and Applied Sciences, Technical University of Cotopaxi, Latacunga, Ecuador
[2]Department of Computer Science, National University of San Marcos, Lima, Peru

Article Info

Article history:

Received Jul 7, 2018
Revised Oct 10, 2018
Accepted Nov 25, 2018

Keywords:

Attack detection
DDoS
Distributed denial of service

ABSTRACT

Distributed Denial of Service attacks (DDoS) are one of the biggest problems facing the Internet. To eliminate this type of attack, the number of which has increased in the period under study, various methods of defense have been proposed. However a detection mechanism that is able to completely counteract the attacks has not yet been found. Therefore, detection and defense against DDoS attacks is of great importance for specialists engaged in computer security. This paper presents a systematic review of the scientific literature on methods of detecting DDoS attacks. From the literature the main aspects related to detection have been formulated. Six aspects for analysis in this investigation were identified: techniques, variables, tools, deployment location, point in time and detection accuracy. It was found that each technique used for the detection of attacks exploits certain characteristics of the network traffic, user requests and specific tools. Finally, it managed to identify the mechanisms that have the highest detection accuracy, such as the datasets they use. It has been concluded that an adequate analysis of the above aspects of detection of DDoS attacks can make a useful contribution to designing an appropriate strategy for neutralizing the attacks.

Corresponding Author:

Silvia Bravo,
Faculty of Engineering and Applied Sciences,
Technical University of Cotopaxi,
Latacunga 050150, Ecuador.
Email: silvia.bravom@utc.edu.ec

1. INTRODUCTION

Computer attacks, such as denial of service (DoS), are a threat to Internet security and have posed a problem since its appearance in 1980 [1]. These attacks are illegal actions through which an attacker interrupts the resources or services of a system [2] and affects access to the network, online accounts, email and computer resources [3].

Later, a more sophisticated type of DoS attack called Distributed Denial of Service (DDoS) appeared. This attack involves two or more computers, which can be located in different parts of the world, and are executed by the same attacker [4]. The first reports of this type of attack appeared in 1999 [5]. In [6] they states that the main problem to detect this type of attack is to differentiate the legitimate flows from the attack flows, which results in high rates of false positives and negatives in the detection methods used. Therefore, the research topic of detection of DDoS attack has generated great interest in the scientific community. Likewise, [7] they suggest a classification of this type of attacks, according to the layer in which they are executed, these are network layer and application layer.

Several investigations focus their efforts on the review of aspects that intervene in the detection of DDoS attacks. In [7] they conducted an review of the literature on attacks and defense mechanisms with an analysis of prevention, detection and response. On the other hand, [8] they published a review article describing the characteristics of the mechanisms by means of which the network detection mechanism and

the reaction to an attack are activated. In [9] they presented an investigation of DDoS attacks, detection methods and tools used in wired networks. Although these works analyze the detection mechanisms, they are limited to an analysis at the network layer level and the depth application layer is not considered where the attacks have a considerable impact in recent years, such as show several studies [10]-[12]. In addition, these works do not consider the aspects that characterize the detection of DDoS attacks for a possible improvement of it.

Therefore, this paper presents the aspects that characterize the detection of DDoS attacks, these aspects are techniques, variables and tools used, as well as where the detection was implemented and at what point in time. For the aforementioned, the main objective of this document is to carry out a systematic review of the literature to analyze these aspects of detection of DDoS attacks. For this, six research questions have been raised and presented in Section 2. These questions have helped to identify, evaluate and interpret the main relevant issues related to the topic. The present work is organized according to the following structure. In Section 2 explains the methodology used. Section 3 performs an analysis and discussion of the results. Finally, Section 4 presents the conclusions drawn from this study.

2. RESEARCH METHOD

The systematic review for carrying out this research is based on the model proposed by [13], which is divided into three phases:

Planning the review: questions are raised as to the goals of the research and review.

Conducting the review: in this stage the plan is executed and major studies following the inclusion and exclusion criteria selected are referred to or discarded.

Reporting the review: at this stage the results of the statistical review and analysis presented in sections III, are shown.

2.1. Planning the review

To carry out the literature review on the detection of DDoS attacks the following research questions were raised:

Q1: What are the techniques used for detection?

Q2: What are the variables used?

Q3: What are the tools used?

Q4: Where are they implemented?

Q5: At what point in time before the attacks must the detection mechanism be activated?

Q6: With what ratio of precision do the techniques detect a DDoS attack?

Answers to the above research questions, were found in the following data sources: DOAJ (Directory of Open Access Journal), IEEE Xplore, Science Direct and Springer. To find scientific articles published in journals with an impact factor of SJR (Scimago Journal and Country Rank), in the period between 2005 to 2017, the following search procedure was undertaken, as shown in Table 1, taking into account the title, abstract and keywords.

In addition, these terms are adapted to match the research questions and individual needs of the search engine. To the results of searches from various sources of information the criteria for inclusion and exclusion shown in Table 2 were applied.

Table 1. Source String

Source	String
DOAJ	distributed denial of service or ddos; 2005-2017
IEEE Xplore	((distributed denial of service) OR ddos) and refined by Year: 2005-2017
Science Direct	pub-date ¿= 2005 and (distributed denial of service) and ddos
Springer	”distributed denial of service” or ”ddos”’within 2005 - 2017

Table 2. Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
Models, methods and techniques for detecting DDoS attacks	Detection submits proposals that do not include the experimental results
Proposed variables in the detection attacks	Present detection mechanisms in general botnets
Proposed components that make up the mechanism	Books, proceedings, posters, theses, workshops
Proposed tools in the detection mechanisms	Presented in its tracking attack flow
Directly answer the research questions	Submit contributions that aim to cloud computing environments, P2P networks, MANET, wireless local areas, data centers, high speed networks and DNS servers

2.2. Conducting the review

The search results obtained, according to the proposed strategy, were subjected to a selection process, according to the inclusion and exclusion criteria established. It was necessary to make a preliminary review of their content in order to determine their relevance to the present study and to determine whether these works apply to the detection of DDoS attacks. Most of the items were discarded because they corresponded to another subject under study, such as surveys, taxonomy and botnets. The process implemented and the results obtained at each stage are shown in Figure 1. Subsequently, we proceeded to analyze the articles in order to answer the research questions.

The results of the search performed showed a total of 1341 articles. Of these, 81 were selected, that met the inclusion and exclusion criteria established, as can be seen in Table 3.

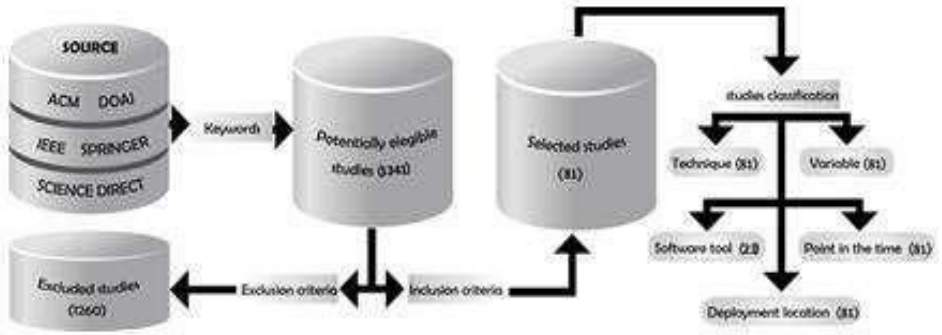

Figure 1. Process for exploring the literature

Table 3. Selected articles

Source	Potentially eligible studies	Selected studies (Journal article)
DOAJ	158	20
IEEE Xplore	80	21
Science Direct	843	30
Springer	260	10
Total	1341	81

2.3. Time trends of the publications

Figure 2 shows the temporal trend of the publications on the detection of DDoS attacks, selected from the methodology, by phase conducting the review sample. In it, you can see the increase in the number of publications over the past 13 years can be seen. The trend in the number of published papers reflects the importance that has been given to this subject of study by the scientific community.

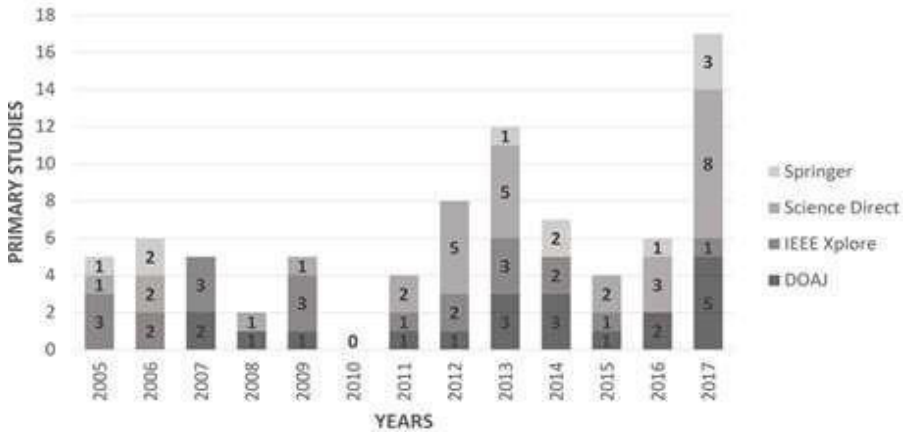

Figure 2. Time trend of publications on DDoS attack detection

2.4. Data Sources

The search results show that the largest number of articles were obtained from the Science Direct databases and IEEE Xplore. Moreover, reference sources that provided some information on the subject were Springer and DOAJ, as can be seen in Table 3.

2.5. Aspects

The following items on the detection of DDoS attacks were chosen: techniques, variables, tools, deployment location, point in time and detection accuracy. Table 4 shows these aspects together with their respective definitions:

Table 4. Definitions of aspects

Aspects	Definition
Technique	It refers to the set of procedures or resources used in a particular activity. In this paper we consider the techniques employed by detection mechanisms.
Variable	It is defined as the character that is measured in different individuals or objects. In this study it responds to the need to know the features used the mechanism for detecting DDoS attacks.
Software tools	These are computer programs that help the specialist in the design process and the development of software or documentation [14].
Deployment location	It refers to the location were the detection mechanism should be deployed i.e., at the source, in the network, at the destination or a hybrid of the above.
Point in time	It is defined as the moment when the detection mechanism should be activated i.e., before, during or after the attack.
Accuracy/Detection Rate	The overall value of all correctly classified instances i.e., both true positives and true negatives [15]

Table 5 shows the distribution of the 81 selected studies, in accordance with the aspects identified for detecting DDoS attacks as defined above. It can be seen that 100% of the selected papers presented at least one technique fo|r detecting attacks. Also, only 28% mention the tool used to implement the technique for detecting a DDoS attack.

Table 5. Aspects of DDoS attacks detection

Source	Technique	Variable	Tools	Deployment location	Point in time
DOAJ	[16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35]	[16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35]	[16] [17] [23] [26] [32] [35]	[16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35]	[16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35]
IEEE Xplore	[36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56]	[36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56]	[38] [46]	[36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56]	[36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56]
Science Direct	[57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86]	[57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86]	[59] [60] [61] [62] [63] [68] [70] [71] [72] [73] [74] [82] [86]	[57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86]	[57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86]
Springer	[87] [88] [89] [90] [91] [92] [93] [94] [95] [96]	[87] [88] [89] [90] [91] [92] [93] [94] [95] [96]	[92] [96]	[87] [88] [89] [90] [91] [92] [93] [94] [95] [96]	[87] [88] [89] [90] [91] [92] [93] [94] [95] [96]
Total	81	81	23	81	81

3. RESULTS AND DISCUSSION

The analysis of the information collected in Section 3 was performed on the basis of the research questions posed in Section 2. The results are presented in tables containing the description of the aspect to be analyzed, together with the names of the authors who used them.

3.1. Q1: What are the techniques used in the detection?

The techniques used in the detecting DDoS attacks are shown and described in Table 6. As can be appreciated in this table different techniques 48 have been proposed. The aspects of each technique are discussed in Table 6.

Table 6. Techniques used for detecting DDoS attacks

Id	Technique	Description
T1	Bagging	Representative of parallel ensemble learning methods. It employs Random Sampling in sampling data set.The algorithm focuses mainly on decreasing variance.
T2	Bat Algorithm	The bat algorithm uses the echo based location determining behavior of bats to solve both single objective and multi-objective optimization problems.
T3	Bloom filter	The Bloom filter is a kind of space-efficient hash data structure. We propose using a modified Bloom filter in order to construct a hash table that can record three-way TCP control packets at a limited storage cost.
T4	Change aggregation tree (CAT)	This CAT mechanism is designed for use at the router level for detecting abrupt changes in traffic flows.When a DDoS attack is launched, the routers observe changes in the space temporal distribution of traffic volumes.
T5	Cluster analysis	Cluster analysis is to group data so that objects in a given group are similar to each other and dissimilar from to those in other groups. By using cluster analysis, we can separate normal traffic and each phase of the DDoS forming clusters have dissimilarities among them attack into partitioned groups if the variables involved in
T6	Congestion Participation Rate (CPR)	Congestion Participation Rate (CPR) to identify LDDoS flows by measuring the intention of network flows to congest the network. To the best of our knowledge, it is the first metric that is able to recognize LDDoS flows by quantifying each flow*j*s intention to congest the network.
T7	Correlation analysis	The correlation is used to describe the similarity of different flows. However, in some cases, it may indicate zero correlation. Although the two flows are completely correlated there is a phase difference.
T8	Counter mechanism	Assigns a continuous value as opposed to a binary measure to each client session, and the scheduler utilizes these values to determine if and when to schedule a sessionjs requests.
T9	Cuckoo search	Technique stimulated by the parasite act of some Cuckoo birds. The species of type Cuckoo unable to complete its reproduction cycle without proper host.
T10	Cusum algorithm	A nonparametric cumulative sum (CUSUM) procedure commonly used for detection of wide range of possible shifts and is generally favored for its simplicity and low computational overhead.
T11	Entropy	Renyi's generalized entropies is a family of measures that characterize the distribution of a random variable. Shannon entropy has been used to conceptualize source address entropy and traffic cluster entropy.
T12	Firewall	Firewall function as above, giving the defender the option to set the value which is the threshold above which all the packets of a flow are dropped.
T13	Fuzzy logic	Fuzzy estimator on the mean packet between arrival times. It interprets the rules well but it suffers from the disadvantage of not being able to acquire the rules automatically.
T14	Genetic algorithms	A Genetic algorithm is a heuristic search that mimics the process of natural evolution. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using inheritance, mutation, selection, and crossover techniques inspired by natural evolution, such as
T15	Googles strategic position	The main idea of JUST-Google is to let ISPs edge routers allow traffic originating from sources that are approved by Google and destined to a victim within that ISP to pass while filtering all other traffic destined to the same victim. An HsMM algorithm that describes the stochastic process varying with time and monitors the App-DDoS attacks occurring during a flash crowd event.
T16	Hidden semi-Markov model (HsMM)	An HsMM algorithm that describes the stochastic process varying with time and monitors the App-DDoS attacks occurring during a flash crowd event.
T17	Hop-Count Filtering	The source IP address serves as the index in the table for retrieving the correct hop-count for this IP address. If the computed hop-count matches the stored hop-count, the packet has been authenticated.
T18	Information distance	A metric used to detect low-rate DDoS attacks by measuring the difference between legitimate traffic and attack traffic.
T19	Information divergence	Estimates the distances between the probability measurements independently of the parameters and detects the attacker and discards the adversarys packets for a fixed amount of time in an organized manner.
T20	Joint Deviation Rate (JDR)	Joint Deviation Rate (JDR), a new metric to describe the deviation rate of the network traffic states. JDR is a combination of the deviations of all the multiple features in Network Traffic State (NTS).
T21	K-nearest neighbors	The k-nearest neighbor algorithm is a method that predicts flow classes based on the k-closest training examples in feature space. A flow is classified by the majority vote of its neighbors and k is a positive integer, typically small.
T22	Kolmogorov Complexity	Kolmogorov Complexity states that the joint complexity measure of random strings is lower than the sum of the complexities of the individual strings when the strings exhibit some correlation.

Id	Technique	Description
T23	Mapping Service	A service provider registers the binding(s) from its domain name(s) to the IP addresses into the domain name system (DNS). When a customer wants to obtain a service from the service provider, his/her computer first queries domain name and then sends a request to the server that uses the returned IP address.the IP address corresponding to the service providerjs
T24	Mathematical model	Mathematical model for estimating the attack effect of this stealthy type of DDoS. By originally capturing the adjustment behavior of a victim in the TCP congestion window, our model can comprehensively evaluate the configured) and the affect of the attack on the network environment.combined impact of the attack pattern (i.e., how the attack is
T25	Multi-agent application	An agent is built as an aggregation of capabilities, and such capabilities are selected according to the primitive actions that a mechanism provides.
T26	Neural Network	A Neural network consists of processing elements called neurons. These neural networks are designed to learn a new pattern, new association, and new functional dependencies. The advantage of a neural network is a better generalization capability.
T27	Neyman Pearson cost minimization strategy	Neyman Pearson (NeP) theory where prior knowledge of the distribution of data is not known. The NeP hypothesis is useful in situations where different types of error have different consequences.
T28	Overlay network	Maintains virtual rings or shields of protection around registered customers. A ring is composed of a set of IPSs that are at the same distance (number of hops) from the customer.
T29	Packet filtering	Packet classification and filtering scheme to be implemented at the edge routers of the ISP network that contains the targeted system, and should be activated after a TCP-based reflector DDoS attack has been detected.
T30	Packet marking	A scheme that allows a DDoS victim to filter out attack packets on a per packet basis with a high accuracy after only a few attack packets have been received.
T31	Path identifiers	Used in negotiated between neighboring domains as interdomain routing objects.
T32	Pushback	Commands can contain some rate-limit requests, so that, when an upstream router receives the command, it will rate-limit the traffic to the victim and not cause congestion near the victim.
T33	Puzzle Solving	Captures complex temporal correlations across multiple time scales with very low computational complexity.
T34	Queuing model	Carries information about traffic characteristics and congestion properties.
T35	Random Forrest	Random Feature Selection is farther introduced in the training process for Random Forest.
T36	Rate-limit filters	The congested router starts with a local rate limit, and then progressively pushes the rate limit to some neighbor routers and further out, forming a dynamic rate-limit tree, which can be expensive to maintain.
T37	Ratio of Collective Flow (RCF)	Responsible for classifying a flow as legitimate, suspicious or attack flow based on the basis of packet information obtained from the monitoring module and the current load on an outgoing queue.
T38	Resilient Back Propagation (RBP)	The RBP algorithm was found to perform better. A single classifier commits errors on different training samples. So, by creating an ensemble of classifiers and combining their outputs, the total error can be reduced and the detection accuracy can be increased.
T39	Router throttling	Contributes to the fundamental understanding of router throttling as a mechanism against DDoS attacks. In particular, a control-theoretical model useful for understanding a system is behavior under a variety of parameters and operating conditions.
T40	Routing Information Protocol (RIP)	RIP (routing information protocol), a representative protocol of IGP (interior gateway protocol). RIP, which works by the exchange of tables among routers, operates inside AS (an autonomous system). RIP is used as the routing protocol on the inside of AS.
T41	Semantic traffic differentiation	Semantic traffic differentiation has two main advantages over per packet and per-user differentiation approaches: 1) It easily spots randomly generated attack traffic (with or without spoofing) since such traffic creates short-lived structures with no higher semantics. 2) It easily spots structures that are engaged in one-way communications, aggressively sending traffic to an unresponsive party.
T42	Signature based	Profiles which describe of characteristics of a known network security according to the security requirements of network objects on a network.
T43	Special Sequence Matrix	SSM is a dynamic spanning matrix. Used in bModel they are produced dynamically and cause the diameter of the matrix to grow dynamically as well.
T44	Spectral analysis	Spectra analysis can be applied to both training traffic and the incoming traffic streams to the tesbed. Leveraging spectral analysis, our hypothesis testing model make spectral template matching effective by detecting shrew DDoS attacks at traffic streaming level and by cutting off malicious flows at a refined flow level.
T45	Statistical Methods	The key idea is to prioritize a packet based on a score which estimates its legitimacy given the attribute values it carries.
T46	Support-vector data description (SVDD)	An anomaly-detection method that uses unlabeled data to find a model for unusual instances.
T47	TCP/IP and HTTP statistics	The following statistical values are computed for each incoming user: number of get requests, standard deviation of get, mean of flows per user, standard deviation of flows per user and standard deviation of posts, flows per minute per user, request per minute per user and so on.
T48	Wavelet Analysis	Captures a complex temporal correlation across multiple time scales with a very low computational complexity.

Table 6 shows the results of 81 studies that present DDoS attack detection techniques. While in Table 8, it can be seen that the Neural Network technique is used more frequently, having been applied in eight studies. Therefore, it is evident that this technique is the most commonly used due to its computational and logical capacity to identify anomalies between data flow entries. Entropy is used by 6 attack detection studies. This technique is used because it allows identifying certain characteristics of a data flow that would allow the detection of a DDoS attack. It can be concluded that the most commonly used techniques analyze the data flow for the detection of DDoS attacks and focus on the network layer.

3.2. Q2: What are the variables used in the detection?

In the studies analyzed a total of 28 variables for the detection of DDoS attacks were identified, as can be seen in Table 7. This table also provides a description of the variables that have been identified.

Table 7. Variables used by DDoS attack detection techniques

Id	Variable	Description
V1	Absolute bandwidth consumption	This feature represents the average bandwidth consumed by the requests found in absolute time interval defined. This feature also considered as significant since the estimation of bandwidth consumption is critical in load assessment.
V2	Absolute page access count	This feature represents the average number of requests in an absolute time interval defined. This feature also critical one among the considered features, since the page access count along with absolute session interval optimizes the detection of the load on target web server.
V3	Absolute page access time	This feature represents the average time spent on each page request in an absolute time interval defined. The motive to consider this feature is, load of requests with minimal access time of each page is suspicious.
V4	Absolute session count	This feature represents the average number of sessions found in an absolute time interval defined. This feature is considered since the load on any target web server estimated by the number of sessions in a given time interval.
V5	Absolute session interval	This feature represents the average time render each session in an absolute time interval defined. This feature is critical as the session time indicates the time spent by a source on the target web server with an intension of fair use or an attack.
V6	Absolute time interval	This denotes the absolute time taken by the set of sessions initiated at given threshold time frame. This feature considered as significant, as HTTP-flood is cumulative of multiple sessions and diversified packet flow. The features explored further for defined absolute time interval.
V7	ACK number	ACK number sent by the receiving terminal is the last Sequence Number when communication was successful.
V8	Click rates of web objects	Estimation of the click-through rate of available ads for a given search query. The more interactive it is, the higher the click-through rate is.
V9	Eminent source diversity ratio	This feature represents the average number of divergent sources those initiate the sessions in an absolute time interval defined. The request load from eminent sources is tolerable, hence this feature considered as significant.
V10	IP address	The only valid IP source address for packets originating from the PC is the one assigned by the ISP (whether statically or dynamically assigned).
V11	Network traffic	Remote logins and file transfers
V12	Number of connections	Behavioral characteristics of a connection in terms of number, type of various data items with respect to time. These features are used to determine the statistical properties, such as standard deviation and variance.
V13	Number of ICMP	Number of ICMP echo reply packets from the same source.
V14	Number of packets	Packets transmitted or received without errors.
V15	Number of requests	Requests for currently open windows and whether the number of requests for an open window of time is viable.
V16	Number of UDP	Number of UDP echo packets to a specified port
V17	Number of users	Set of real users accessing a server
V18	Packets	Packets carrying path information. The victim node can defend itself from DDoS attack by filtering the packets transmitting via/from an attacking node.
V19	Port	The I/O port determines which service ports are being used.
V20	Protocol	Internet Protocols (IP), there is now a standard for how general purpose computers, such as personal computers, workstations and servers can interchange data over the telephone system.
V21	Rate of packets	This feature is calculated on the packets sent from a particular sender.
V22	Ratio of incoming SMTP packets	A host that does not have any incoming connection is more likely to be a spammer than one that has incoming SMTP traffic
V23	Ratio of outgoing SMTP packets	Shows the outgoing SMTP traffic time series for a host known to have sent spam
V24	Session‘s requests	Session inter-arrival times between consecutive sessions
V25	TCP flows	Flows with a large amount of data to send, such as FTP transfers
V26	Traffic rate	Defined as the total number of bits received over a certain time interval.
V27	Type of packet	Fundamentally, all networks have essentially two kinds of packets. Data packets that belong to users and carry users or application traffic. Control packets belong to the network and are used to dynamically build and operate the network
V28	Variance of time	Variance of time difference between two consecutive packets

Table 8 shows the variables used by the detection techniques and the authors who used the mentioned techniques are summarized. It can be seen that the most commonly used variables are packets and IP addresses. The packages are used for detection because they contain data such as IP source, weight, speed, among others. While the IP address is used to identify the origin of the data flow, which allows to cut the traffic that is sent from them when it is identified as an attack. It is important to note that these variables are used in mechanisms that correspond to the detection of DDoS attacks in the network layer.

Table 8. Techniques, studies and variables used by the detection mechanisms of DDoS attacks

Technique	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28
T1											[34]														[34]			
T2		[81]			[81]	[81]											[81]	[81]										
T3																		[89]										
T4										[41]									[41]	[41]								
T5										[60] [32]				[60]					[60]								[60]	
T6											[68]																	
T7										[86]				[49]	[92]										[86]			
T8										[20] [24] [47]								[20]						[45]				
T9	[83] [94]	[94]	[83] [94]	[83] [94]	[83] [94]	[83] [94]			[84] [94]																			
T10						[79]						[79]			[79]						[79]							
T11					[77]					[47] [28] [77]	[19]							[26] [84]										
T12										[18]								[72] [90]										
T13						[69]						[70]	[70]			[70]						[70]	[70]					[70]
T14										[64] [29] [85]		[70]	[70]		[85]	[70]			[85]	[85]		[70]	[70]					[70]
T15										[22]																		
T16								[25]							[46]		[46]											
T17										[43]																		
T18										[54]	[30]							[31]										
T19										[65] [95]				[73]														
T20										[67]								[67]	[67]	[67]					[67]			
T21																		[74] [34]							[34]			
T22																		[88]										
T23										[51]				[51]														
T24																									[53]			
T25										[82]																		
T26										[29] [35] [78] [85]	[50] [52]	[70]	[70]	[44]		[70]		[78]	[85] [78]	[35] [85]	[35]	[70]	[70]	[85]				[70]
T27											[63]																	
T28															[48]													
T29							[58]			[57] [58]								[17] [27] [71]										
T30																		[16] [40] [42] [71]										
T31										[56]																		
T32																		[42]										
T33																								[55]				
T34										[62]								[61]										
T35																		[34]							[34]			
T36																		[36]										
T37																		[21]										
T38											[63]																	
T39																		[75]								[38]		
T40												[91]																
T41										[37]																		
T42										[23] [96]	[30]								[23] [96]	[23]								
T43															[92]													
T44										[59]									[59]	[59]								
T45										[33] [76]								[39] [76]	[76]	[76]					[76]			
T46												[93]						[93]		[93]								
T47						[69]								[69]	[69]													
T48											[87]																	

3.3. What are the tools used to implement the techniques?

The ten tools used for the implementation of detection techniques are shown in Table 9. The same evidence shows that the two most commonly used tools are Matlab and the Network simulator. This is due to the fact that these two tools present functionalities for the adequate implementation of the detection mechanisms [11].

Table 9. Tools used by detection techniques of DDoS attacks

Techniques	CRF++ toolkits	Globus Toolkit	LIBSVM toolkits	Matlab	Network simulator	SAS Enterpriser Miner	SSFNet simulator	Tstat	Weka	Preset resiliense simulator
[T5]				[32]		[60]			[32]	
[T6]					[68]					
[T7]				[86] [92]				[73]		
[T11]					[26]					
[T12]				[72]	[72]					
[T13]				[70]						
[T14]				[70]						
[T16]				[46]						
[T19]		[73]								
[T21]								[74]		
[T25]										[82]
[T26]				[35] [70]						
[T27]				[63]						
[T29]					[17] [71]					
[T30]					[71]		[16]			
[T33]				[61]		[62]				
[T37]				[63]						
[T38]					[38]					
[T41]	[23]		[23]						[23]	
[T42]				[92]	[96]					
[T43]					[59]					

3.4. Q4: Where are the detection techniques implemented?

DDoS attack detection techniques can be deployed in four locations: source, destination, network and hybrid. Source refers to the source of the attack, while destination is the target of the attack. Network is the place where the information traffic circulates and hybrid means that the detection is performed in multiple places and there is usually cooperation between the points of implementation. Table 10 shows the four sites of implementation together with the authors who use them.

Table 10. Deployment locations where detection mechanisms are implemented

Deployment location	Studies	Total
Source	[37] [50] [56]	3
Destination	[25] [28] [29] [33] [34] [45] [46] [58] [60] [61] [69] [74] [79] [81] [82] [83] [85] [90] [93] [94] [95]	21
Network	[16] [17] [19] [20] [22] [23] [24] [26] [27] [30] [31] [32] [35] [36] [38] [39] [40] [41] [43] [44] [47] [48] [49] [51] [52] [53] [54] [57] [59] [61] [63] [64] [66] [68] [70] [71] [72] [73] [75] [76] [77] [78] [84] [86] [87] [88] [91]	47
Hybrid	[18] [21] [42] [55] [65] [67] [80] [89] [92][96]	10
Total		81

Table 10 shows that Network is where most of the detection techniques have been implemented, that is, approximately 58% of the total amount. This is because the network is the place from which the characteristics of the data flow used by the detection mechanisms can be extracted. Therefore, Network is used by the mechanisms more frequently when implementing a detection technique. On the contrary, the Source is where the techniques are implemented on a smaller scale, because its implementation requires a high degree of cooperation between the data networks, which prevents the construction of a greater number of mechanisms that can predict an attack.

3.5. Q5: At what point in the time should the detection mechanism in an attack be activated?

The detection mechanism can act against a possible DDoS attack Before, During and After [7]. The point in time before refers to prevention of the attack before it happens, while during refers to the moment the attack is being made; and finally, after refers to when the attack occurs at the destination and so can be considered as mitigation. Table 11 shows the points in time in which the detection techniques can act together with the authors that employ them at each location.

Table 11. Point in time when the detection mechanisms are implemented

Point in time	Studies	Total
Before the attack	[21] [37] [42] [50] [56] [89] [96]	7
During the attack	[16] [17] [18] [19] [20] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [38] [39] [40] [41] [43] [44] [45] [46] [47] [48] [49] [51] [52] [53] [54] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [76] [77] [78] [79] [81] [82] [83] [84] [85] [86] [87] [88] [90] [91] [92] [93] [94] [95]	71
After the attack	[55] [75] [80]	3
Total		81

In Table 11 shows that During is the point in the time where most of the detection techniques have been implemented. This is because the detection in that place is executed in real time, when the attack flow has reached the target. This is because the mechanism analyzes the flow of data while entering the system, when an anomaly is detected this data flow is cut off. On the contrary, After is the moment in which detection techniques are less implemented because the mechanism would have to predict an attack before it affects the system. This process is difficult because continuous communication between the predecessor networks is required to achieve a prediction.

3.6. Q6: What is the precision with which the techniques detect a DDoS attack?

In this research work only studies that had either a detection or an accuracy rate greater than or equal to 98% were considered and where tests with datasets consisting of real flows and DDoS were performed. The detection rate is calculated by means of the following equation: TP detection is equal to $\frac{TP}{TP+FN}$, and accuracy corresponds to $\frac{TP+TN}{TP+TN+FP+FN}$ where, TP = number of true positives, TN = number of true negatives, FP = number of false positives and FN = amount of false negatives. The precision with which the techniques detect a DDoS attack are shown in Table 12.

Table 12. Detection mechanisms of DDoS Attack that showed the best ratios

Detection Rate (%)	Studies	Dataset
99.99	[34]	Knowledge Discovery and Data mining (KDD) Cup 1999
99.67	[86]	CAIDA, TUIDS and DARPA
99.4	[63]	CAIDA 2007, DARPA 2009, BONESI-generated
98.31	[29]	KDD Cup1999

Table 12 shows that the detection mechanism with the highest precision was achieved by [34]. The detection rate of this mechanism was 99.9%. For this, this mechanism uses a combination of three techniques (Random Forest, nearest K-neighbors and Bagging). In addition, the implementation of this mechanism is in the network, so that detection occurs during the attack, so its impact is mitigated when detected by the system. In [86] they proposed a mechanism that uses the correlation technique. The efficiency of this method reaches 99.67%. To do this, it uses the Matlab tool, as well as the implementation of the mechanism is performed on the network during the attack.

It is also observed that the highest efficiency percentages correspond to techniques implemented in the network layer [34], [63], [86]. These techniques employ variables used in the identification of the flow of data such as traffic and TCP flow, as shown in the first mechanism. Whereas in the second mechanism the variables IP address and TCP flow are used, that is, variables used also in the network layer. Therefore, this analysis can establish the need to have alternative mechanisms that evaluate not only the flow of data that circulates through the network, but also measure the user's interaction with the system. In addition, detection mechanisms could be developed that can use other techniques in combination with other variables to achieve greater detection efficiency. In this context, mechanisms could be proposed for detection in other layers where DDoS attacks also occur, such as the application layer. Since, in this layer is where the greatest number of attacks have occurred in recent years due to its easy execution and difficult detection [29].

4. CONCLUSIONS

The systematic review of the literature presented in this study has identified the main aspects involved with the detection of DDoS attacks, focusing on techniques, variables and tools, in addition to the place where it was implemented and the point of detection over time. An analysis of the results has provided answers to the six proposed research questions. In addition, forty eight techniques that are used in the

detection of DDoS attacks were identified. Also, a total of twenty eight variables were observed and it was evident that the most used tools are Matlab and Network simulator, due to the functionalities and advantages of information processing. The most used place for the implementation of a mechanism is the network, because the data flows are analyzed before they reach the server. The most used point in time for the deployment of a technique is during, because the detection is done in real time when the attack occurs. The most effective mechanism to achieve a high detection rate is that proposed by [34], which reached an accuracy of 99.9%, it uses the characteristics of the data flow that is extracted in the network during the attack.

REFERENCES

[1] Tripathi S, Gupta B, Almomani A, Mishra A, Veluru S. Hadoop based defense solution to handle distributed denial of service (ddos) attacks. *Journal of Information Security*, 2013, 4(03), 150.

[2] Waguih H. A data mining approach for the detection of denial of service attack. *IAES International Journal of Artificial Intelligence*, 2013, vol. 2, no 2, p. 99.

[3] Jain A, Singh A. K. Distributed denial of service (ddos) attacks-classification and implications. *Journal of Information and Operations Management*, 2012, vol. 3, no 1, p. 136.

[4] Ni T, Gu X, Wang H. Detecting DDoS Attacks Against DNS Servers Using Time Series Analysis. *Indonesian Journal of Electrical Engineering and Computer Science*, 2014, vol. 12, no 1, p. 753-761.

[5] Criscuolo P J. Distributed denial of service, tribe flood network 2000, and stacheldraht CIAC-2319, *Department of Energy Computer Incident Advisory Capability (CIAC)*. UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory, 2000.

[6] Choi J, Choi C, Ko B, Kim P. A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. Soft Computing, 2014, vol. 18, no 9, p. 1697-1703.

[7] Zargar S, Joshi J, Tipper D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE communications surveys & tutorials*, 2013, vol. 15, no 4, p. 2046-2069.

[8] Chen L C, Longstaff T A, Carley K M. Characterization of defense mechanisms against distributed denial of service attacks. *Computers & Security*, 2004, vol. 23, no 8, p. 665-678.

[9] Bhuyan M H, Kashyap H J, Bhattacharyya D K, Kalita J K. Detecting distributed denial of service attacks: methods, tools and future directions. *The Computer Journal*, 2013, 57(4), 537-556.

[10] Wei S, Mirkovic J. Building reputations for internet clients. *Electronic Notes in Theoretical Computer Science*, 2007, vol. 179, p. 17-30.

[11] Oikonomou G, Mirkovic J. Modeling Human Behavior for Defense against Flash-Crowd Attacks. *ICC*. 2009. p. 1-6.

[12] Devi S R, Yogesh P. A hybrid approach to counter application layer DDoS attacks. *International Journal on Cryptography and Information Security (IJCIS)*, 2012, vol. 2, no 2.

[13] Kitchenham B. Procedures for performing systematic reviews. Keele, UK, Keele University, 2004, vol. 33, no 2004, p. 1-26.

[14] Ul Haq M Z, and Suharjito S. Usability Analysis of Business Intelligence Tool Based Table Virtualization. *Indonesian Journal of Electrical Engineering and Computer Science*, 2018, vol. 9, no 2, p. 431-437.

[15] Liu L, Wan P, Wang Y, and Liu S. Clustering and hybrid genetic algorithm based intrusion detection strategy. *Indonesian Journal of Electrical Engineering and Computer Science*, 2014, vol. 12, no 1, p. 762-770.

[16] Beak C, Chaudhry J A, Lee K, Park S, Kim M. A novel packet marketing method in DDoS attack detection. *American Journal of Applied Sciences*, 2007, vol. 4, no 10, p. 741-745.

[17] Meenakshi S, Srivatsa S K. A distributed framework with less false positive ratio against distributed denial of service attack. *Information Technology Journal*, 2007, vol. 6, no 8, p. 1139-1145.

[18] Chen Y, Das S, Dhar P. Detecting and Preventing IP-spoofed Distributed DoS Attacks. *IJ Network Security*, 2008, vol. 7, no 1, p. 69-80.

[19] Yan R, Zheng Q. Using Renyi cross entropy to analyze traffic matrix and detect DDoS attacks. *Information Technology Journal*, 2009, vol. 8, no 8, p. 1180-1188.

[20] Liu H, Sun Y, Kim M S. A Scalable DDoS Detection Framework with Victim Pinpoint Capability. *JCM*, 2011, vol. 6, no 9, p. 660-670.

[21] Tiruchengode N. Dynamic approach to defend against distributed denial of service attacks using an adaptive spin lock rate control mechanism. *Journal of Computer Science*, 2012, vol. 8, no 5, p. 632-636.

[22] Al-Duwairi B, Al-Qudah Z, Govindarasu M. A novel scheme for mitigating botnet-based DDoS attacks. *Journal of Networks*, 2013, vol. 8, no 2, p. 297.

[23] Chen S W, Wu J X, Ye X L, Guo T. Distributed denial of service attacks detection method based on conditional random fields. *Journal of Networks*, 2013, vol. 8, no 4, p. 858.

[24] Udhayan J, Babu M R. Deteriorating distributed denial of service attack by recovering zombies using penalty scheme. *Journal of Computer Science*, 2013, vol. 9, no 11, p. 1618.

[25] Huang C, Wang J, Wu G, Chen J. Mining Web User Behaviors to Detect Application Layer DDoS Attacks. *JSW*, 2014, vol. 9, no 4, p. 985-990.

[26] Sachdeva M, Kumar K. A traffic cluster entropy based approach to distinguish DDoS attacks from flash event using DETER testbed. *ISRN Communications and Networking*, 2014, vol. 2014.

[27] Wang Y, Sun R. An IP-traceback-based packet filtering scheme for eliminating DDoS attacks. *Journal of Networks*, 2014, vol. 9, no 4, p. 874.
[28] Saleh M A, Abdul Manaf A. A novel protective framework for defeating HTTP-based denial of service and distributed denial of service attacks. *The Scientific World Journal*, 2015, vol. 2015.
[29] Johnson Singh K, Thongam K, De T. Entropy-Based Application Layer DDoS Attack Detection Using Artificial Neural Networks. *Entropy*, 2016, vol. 18, no 10, p. 350.
[30] Cepheli Ö, Büyükçorak S, Karabulut Kurt G. Hybrid intrusion detection system for ddos attacks. *Journal of Electrical and Computer Engineering*, 2016, vol. 2016.
[31] Zhou L, Liao M, Yuan C, Zhang H. Low-Rate DDoS Attack Detection Using Expectation of Packet Size. *Security and Communication Networks*, 2017, vol. 2017.
[32] Gu Y, Wang Y, Yang Z, Xiong F, Gao Y. Multiple-Features-Based Semisupervised Clustering DDoS Detection Method. *Mathematical Problems in Engineering*, 2017, vol. 2017.
[33] Mirvaziri H. A new method to reduce the effects of HTTP-Get Flood attack. *Future Computing and Informatics Journal*, 2017, vol. 2, no 2, p. 87-93.
[34] Jia B, Huang X, Liu R, Ma Y. A DDoS attack detection method based on hybrid heterogeneous multiclassifier ensemble learning. *Journal of Electrical and Computer Engineering*, 2017, vol. 2017.
[35] Peraković D, Periša M, Cvitić I, Husnjak S. Model for detection and classification of DDoS traffic based on artificial neural network. *Telfor Journal*, 2017, vol. 9, no 1, p. 26.
[36] Chen S, Song Q. Perimeter-based defense against high bandwidth DDoS attacks. *IEEE Transactions on Parallel & Distributed Systems*, 2005, no 6, p. 526-537.
[37] Mirkovic, J., & Reiher, P. D-WARD: a source-end defense against flooding denial-of-service attacks. *IEEE transactions on Dependable and Secure Computing*, 2005, vol. 2, no 3, p. 216-232.
[38] Yau D K, Lui J, Liang F, Yam Y. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Transactions on Networking (TON)*, 2005, vol. 13, no 1, p. 29-42.
[39] Kim Y, Lau W C, Chuah M C, Chao H J. PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE transactions on dependable and secure computing*, 2006, vol. 3, no 2, p. 141-155.
[40] Yaar A, Perrig A, Song D. StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense. *IEEE Journal on Selected Areas in Communications*, 2006, vol. 24, no 10, p. 1853-1863.
[41] Chen Y, Hwang K, Ku W S. Collaborative detection of DDoS attacks over multiple network domains. *IEEE Transactions on Parallel & Distributed Systems*, 2007, no 12, p. 1649-1662.
[42] Chen R, Park J M, Marchany R. A divide-and-conquer strategy for thwarting distributed denial-of-service attacks. *IEEE Transactions on Parallel and Distributed Systems*, 2007, vol. 18, no 5, p. 577-588.
[43] Wang H, Jin C, Shin K G. Defense against spoofed IP traffic using hop-count filtering. *IEEE/ACM Transactions on Networking (ToN)*, 2007, vol. 15, no 1, p. 40-53.
[44] Chonka A, Singh J, Zhou W. Chaos theory based detection against network mimicking DDoS attacks. *IEEE Communication Letters*, 2009, vol. 13, no 9, p. 717-719.
[45] Ranjan S, Swaminathan R, Uysal M, Nucci A, Knightly E. DDoS-shield: DDoS-resilient scheduling to counter application layer attacks. *IEEE/ACM Transactions on networking*, 2009, vol. 17, no 1, p. 26-39.
[46] Xie Y, Yu S Z. Monitoring the application-layer DDoS attacks for popular websites. *IEEE/ACM Transactions on Networking (TON)*, 2009, vol. 17, no 1, p. 15-25.
[47] Xiang Y, Li K, Zhou W. Low-rate DDoS attacks detection and traceback by using new information metrics. *IEEE transactions on information forensics and security*, 2011, vol. 6, no 2, p. 426-437.
[48] François J, Aib I, Boutaba R. FireCol: a collaborative protection network for the detection of flooding DDoS attacks. *IEEE/ACM Transactions on Networking (TON)*, 2012, vol. 20, no 6, p. 1828-1841.
[49] Yu S, Zhou W, Jia W, Guo S, Xiang Y, Tang F. Discriminating DDoS attacks from flash crowds using flow correlation coefficient. *IEEE Transactions on Parallel and Distributed Systems*, 2012, vol. 23, no 6, p. 1073-1080.
[50] Chen Y, Ma X, Wu X. DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory. *IEEE Communications Letters*, 2013, vol. 17, no 5, p. 1052-1054.
[51] Luo H, Lin Y, Zhang H, Zukerman M. Preventing DDoS attacks by identifier/locator separation. *IEEE network*, 2013, vol. 27, no 6, p. 60-65.
[52] Wu X, Chen Y. Validation of chaos hypothesis in NADA and improved DDoS detection algorithm. *IEEE Communications Letters*, 2013, vol. 17, no 12, p. 2396-2399.
[53] Luo J, Yang X, Wang J, Xu J, Sun J, Long K. On a Mathematical Model for Low-Rate Shrew DDoS. *IEEE Trans. Information Forensics and Security*, 2014, vol. 9, no 7, p. 1069-1083.
[54] Ma X, Chen Y. DDoS detection method based on chaos analysis of network traffic entropy. *IEEE Communications Letters*, 2014, vol. 18, no 1, p. 114-117.
[55] Wu Y, Zhao Z, Bao F, Deng R H. Software puzzle: A countermeasure to resource-inflated denial-of-service attacks. *IEEE Transactions on Information forensics and security*, 2015, vol. 10, no 1, p. 168-177.
[56] Luo H, Chen Z, Li J, Vasilakos A V. Preventing distributed denial-of-service flooding attacks with dynamic path identifiers. *IEEE Transactions on Information Forensics and Security*, 2017, vol. 12, no 8, p. 1801-1815.
[57] Lee F Y, Shieh S. Defending against spoofed DDoS attacks with path fingerprint. *Computers & Security*, 2005, vol. 24, no 7, p. 571-586.
[58] Al-Duwairi B, Manimaran G. Distributed packet pairing for reflector based DDoS attack mitigation. *Computer*

communications, 2006, vol. 29, no 12, p. 2269-2280.
[59] Chen Y, Hwang K. Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. *Journal of Parallel and Distributed Computing*, 2006, vol. 66, no 9, p. 1137-1151.
[60] Lee K, Kim J, Kwon K H, Han Y, Kim S. DDoS attack detection method using cluster analysis. *Expert systems with applications*, 2008, vol. 34, no 3, p. 1659-1665.
[61] Lu W Z, Gu W X, Yu S Z. One-way queuing delay measurement and its application on detecting DDoS attack. *Journal of Network and Computer Applications*, 2009, vol. 32, no 2, p. 367-376.
[62] Doron E, Wool A. Wda: A web farm distributed denial of service attack attenuator. *Computer Networks*, 2011, vol. 55, no 5, p. 1037-1051.
[63] Kumar P A R, Selvakumar S. Distributed denial of service attack detection using an ensemble of neural classifier. *Computer Communications*, 2011, vol. 34, no 11, p. 1328-1341.
[64] Lee S M, Kim D S, Lee J H, Park J S. Detection of DDoS attacks using optimized traffic matrix. *Computers & Mathematics with Applications*, 2012, vol. 63, no 2, p. 501-510.
[65] Rahmani H, Sahli N, Kamoun F. DDoS flooding attack detection scheme based on F-divergence. *Computer Communications*, 2012, vol. 35, no 11, p. 1380-1391.
[66] Shiaeles S N, Katos V, Karakos A S, Papadopoulos B K. Real time DDoS detection using fuzzy estimators. *Computers & security*, 2012, vol. 31, no 6, p. 782-790.
[67] Wang F, Wang H, Wang X, Su J. A new multistage approach to detect subtle DDoS attacks. *Mathematical and Computer Modelling*, 2012, vol. 55, no 1-2, p. 198-213.
[68] Zhang C, Cai Z, Chen W, Luo X, Yin J. Flow level detection and filtering of low-rate DDoS. *Computer Networks*, 2012, vol. 56, no 15, p. 3417-3431.
[69] Giralte L C, Conde C, De Diego I M, Cabello E. Detecting denial of service by modelling web-server behaviour. *Computers & Electrical Engineering*, 2013, vol. 39, no 7, p. 2252-2262.
[70] Kumar P A R, Selvakumar S. Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. *Computer Communications*, 2013, vol. 36, no 3, p. 303-319.
[71] Seo D, Lee H, Perrig A. APFS: adaptive probabilistic filter scheduling against distributed denial-of-service attacks. *Computers & Security*, 2013, vol. 39, p. 366-385.
[72] Spyridopoulos T, Karanikas G, Tryfonas T, Oikonomou G. A game theoretic defence framework against DoS/DDoS cyber attacks. *Computers & Security*, 2013, vol. 38, p. 39-50.
[73] Varalakshmi P, Selvi S T. Thwarting DDoS attacks in grid using information divergence. *Future Generation Computer Systems*, 2013, vol. 29, no 1, p. 429-441.
[74] Xiao P, Qu W, Qi H, Li Z. Detecting DDoS attacks against data center with correlation analysis. *Computer Communications*, 2015, vol. 67, p. 66-74.
[75] Malialis K, Kudenko D. Distributed response to network intrusions using multiagent reinforcement learning. *Engineering Applications of Artificial Intelligence*, 2015, vol. 41, p. 270-284.
[76] Kalkan K, Alagöz F. A distributed filtering mechanism against DDoS attacks: ScoreForCore. *Computer Networks*, 2016, vol. 108, p. 199-209.
[77] Sachdeva M, Kumar K, Singh G. A comprehensive approach to discriminate DDoS attacks from flash events. *Journal of Information Security and Applications*, 2016, vol. 26, p. 8-22.
[78] Saied A, Overill R E, Radzik T. Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing*, 2016, vol. 172, p. 385-393.
[79] Jazi H H, Gonzalez H, Stakhanova N, Ghorbani A A. Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. Computer Networks, 2017, vol. 121, p. 25-36.
[80] MIRVAZIRI H. A new method to reduce the effects of HTTP-Get Flood attack. *Future Computing and Informatics Journal*, 2017, vol. 2, no 2, p. 87-93.
[81] Sreeram I, Vuppala V P K. HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm. *Applied Computing and Informatics*, 2017.
[82] Nunes I, Schardong F, Schaeffer-Filho A. BDI2DoS: an application using collaborating BDI agents to combat DDoS attacks. *Journal of Network and Computer Applications*, 2017, vol. 84, p. 14-24.
[83] Prasad K M, Reddy A R M, Rao K V. BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web. *Journal of King Saud University-Computer and Information Sciences*, 2017.
[84] Behal S, Kumar K. Detection of DDoS attacks and flash events using novel information theory metrics. *Computer Networks*, 2017, vol. 116, p. 96-110.
[85] Singh K J, De T. MLP-GA based algorithm to detect application layer DDoS attack. *Journal of Information Security and Applications*, 2017, vol. 36, p. 145-153.
[86] Hoque N, Kashyap H, Bhattacharyya D K. Real-time DDoS attack detection using FPGA. *Computer Communications*, 2017, vol. 110, p. 48-58.
[87] Li L, Lee G. DDoS attack detection and wavelets. *Telecommunication Systems*, 2005, vol. 28, no 3-4, p. 435-451.
[88] Kulkarni A., Bush, S. Detecting distributed denial-of-service attacks using kolmogorov complexity metrics. *Journal of Network and Systems Management*, 2006, vol. 14, no 1, p. 69-80.
[89] Xiao B, Chen W, He Y. A novel approach to detecting DDoS attacks at an early stage. *The Journal of Supercomputing*, 2006, vol. 36, no 3, p. 235-248.
[90] Kang S H, Park K Y, Yoo S G, Kim J. DDoS avoidance strategy for service availability. *Cluster computing*, 2013, vol. 16, no 2, p. 241-248.
[91] Kang H S, Kim S R. sShield: small DDoS defense system using RIP-based traffic deflection in autonomous

system. *The Journal of Supercomputing*, 2014, vol. 67, no 3, p. 820-836.
[92] Zhou W, Jia W, Wen S, Xiang Y, Zhou W. Detection and defense of application-layer DDoS attacks in backbone web traffic. *Future Generation Computer Systems*, 2014, vol. 38, p. 36-46.
[93] Dick, U., & Scheffer, T. Learning to control a structured-prediction decoder for detection of HTTP-layer DDoS attackers. *Machine Learning*, 2016, vol. 104, no 2-3, p. 385-410.
[94] Prasad K M, Reddy A R M, Rao K V. BIFAD: Bio-inspired anomaly based HTTP-flood attack detection. *Wireless Personal Communications*, 2017, vol. 97, no 1, p. 281-308.
[95] Boro D, Bhattacharyya D K. DyProSD: a dynamic protocol specific defense for high-rate DDoS flooding attacks. *Microsystem Technologies*, 2017, vol. 23, no 3, p. 593-611.
[96] Merouane M. An approach for detecting and preventing DDoS attacks in campus. *Automatic Control and Computer Sciences*, 2017, vol. 51, no 1, p. 13-23.

ANEXO 2

New Features of User's Behavior to Distributed Denial of Service Attacks Detection in Application Layer

https://doi.org/10.3991/ijoe.v14i12.9439

Silvia Bravo (✉)
Technical University of Cotopaxi, Latacunga, Ecuador
ëáäî á~K=Äê~î çã] ì í ÅKÉÇì KÉÅ =

David Mauricio
National University of San Marcos, Lima, Peru

Abstract—Distributed Denial of Service (DDoS) attacks are a threat to the security of red. In recent years, these attacks have been directed especially towards the application layer. This phenomenon is mainly due to the large number of existing tools for the generation of this type of attack. The highest detection rate achieved by a method in the application capacity is 98.5%. Therefore, the problem of detecting DDoS attacks persists. In this work an alternative of detection based on the dynamism of the web user is proposed. To do this, evaluate the user's characteristics, mouse functions and right click. For the evaluation, a data set of 11055 requests was used, from which the characteristics were extracted and entered into a classification algorithm. To that end, it can be applied once in Java for the classification of real users and DDoS attacks. The results showed that the evaluated characteristics achieved an efficiency of 100%. Therefore, it is concluded that these characteristics show the dynamism of the user and can be used in a detection method of DDoS attacks.

Keywords—DDoS, user's behavior, application layer, attack detection

1 Introduction

The detection of DDoS attacks is one of the biggest problems facing the security architecture of the network. Therefore, it has become an important factor of study in the field of computer security. A DDoS attack occurs when an attacker coordinates their attacks using several machines, called zombies, towards a specific target or server. The aim of the attacker is to make massive requests to the victim machine to saturate it and that it stops serving the requests of real users.

To counteract this type of attack, several detection mechanisms have been proposed, both at the network level [1]-[49] and at the application level [50]-[58]. The highest detection rate obtained to date is 99.4%, and has been achieved by implementing a network-level method [1]. The dataset used in that work is KDD cup dataset, from which 300,000 connection records were extracted between DDoS attacks and real users. On the other hand, in the methods implemented at the application layer level, the best

detection rate obtained is 98.5% [50], of which the dataset used is not available, however for the tests, service requests were simulated and used Sslsqueeze and Slowloris for the generation of attacks.

The detection mechanisms, for the most part, focus their efforts on the network layer. However, currently the largest number of attacks have been directed to the application layer, because they are easy to execute because of the large amount of existing software [50], [58], and more difficult to detect because they are illegitimate requests that they camouflage themselves as requests from real users. So the present work focuses on the detection of attacks in the application layer.

All methods of detection of attacks in the application layer are based on characteristics, their efficiency depends on them. However, no detection method contemplates the user's interaction with the system, which is a feature that can differentiate between a human and a robot [55]. In this work we identify new features based on the interaction of the user with the system, specifically its interaction with the mouse (mouse movement and right click), and verify its influence on the detection of DDoS attacks.

This work is organized as follows. In section 2, a literature review of the characteristics for the detection of DDoS attacks at the application layer level is made. Section 3 presents the characteristics of user behavior for the detection of attacks, presents the methods used to capture the characteristics and proposes a classification algorithm to identify a real user and a robot, in section 4 the numerical experiments, in section 5 the results and discussions are shown and, finally, the conclusions are presented.

2 Literature review of features

The DDoS attacks in the application layer are characterized by the massive sending of requests, causing limitations in the access to the web services of legitimate users. Figure 1 shows, the transactionality of the system, we observe the requests made by the user or attacker to the web server. In the process of detecting this type of attacks, it is necessary to extract the characteristics of the requests sent to the server. For this, algorithms or procedures are used that filter information on characteristics such as distance measurements [59], [60] provided by the request flows [61]. Once the characteristics are obtained, algorithms or classification criteria are used to detect attacks. Machine learning algorithms are commonly used in the classification of real users and DDoS attacks [62]. There are also classification criteria based on Soft computing techniques and its hydrological approach [1]. Finally, when a DDoS attack is detected, these will be discarded from the set of requests, while the requests of the real users enter the web server to obtain a response.

Table 1 shows the characteristics of the data flow of each client, the characteristics of IP packets in a time interval and the behavior patterns of each user. They are extracted at intervals of time when a client connects to a domain [51]. These characteristics are of the statistical type and record the client's access to system resources and the frequency with which each client requests a resource in the domain.

The detection of DDoS attacks depends to a large extent on the characteristics that are used. The adequate selection of characteristics will allow to improve the detection

process in efficiency and processing time [1]. Therefore, in recent years, the efforts in the detection of DDoS attacks have focused on the search for features that contribute to the detection of attacks in the application layer. Table 1 shows 30 characteristics that are used in the detection of attacks.

The highest detection rate obtained to date is 98.5% and has been achieved using software generated in Python using the Intrusion Detection System (IDS) technique [50]. However, the resources available to attackers are evolving day by day. Therefore, despite the fact that attack detection mechanisms reach high rates, the problem persists.

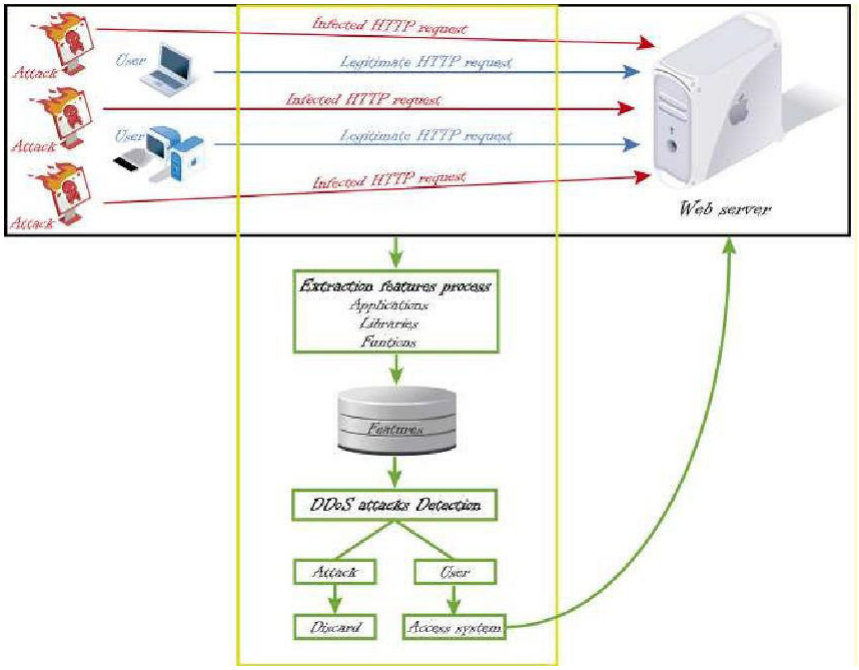

Fig. 1. Execution and detection of DDoS attack in the application layer

Table 1. Features of application layer

Feature	Description	Reference
Access pattern	Access pattern is constantly repeated, develop a frequent path detector which involves checking the requests of the complete flow.	[53]
Average length of query strings of client	Average of consultations made by clients.	[56]
Click number of web objects	The deviation from the entropy of the training data set fitting to the hidden semi-Markov model can be considered as the abnormality of the observed data set.	[55]
Client legitimacy	The legitimacy of a user sending an enormous number of requests is checked against the known client clusters.	[57]
Duration of the conversation	Conversations initiated by one client to the destination socket during some short time interval.	[50]

Entropy of request type (GET/POST/OTHER)	The fractions of request types per connection (GET, POST,or OTHER).	[56]
Entropy of the requests	Entropy to measure the amount of disorder in the flow of the packets or request in the form of an HTTP GET request at multiple time slots.	[58]
Flow similarity	Flow similarity is considered as a key parameter for discriminating between legitimate and illegitimate flows and a few works	[57]
Fraction of connections for domain that accepts any version of English	Connection (e.g., en-us) in Accept-Language.	[56]
Fraction of connections of client that request the most frequent resource path	A client accesses and also count how often each client requests the currently most common path on the domain.	[56]
Access pattern	Access pattern is constantly repeated, develop a frequent path detector which involves checking the requests of the complete flow.	[53]
Average length of query strings of client	Average of consultations made by clients.	[56]
HTTP GET request count	The operation of HTTP starts with a client by sending a request to the server in the form of a request method.	[58]
IP address	Source IP addresses, we are able to classify them into different traffic.	[54]
Maximal, minimal and average packet size	Average of these packet numbers and the mutual information of the fast Fourier Transform.	[50]
Maximal, minimal and average size of TCP window	Number of packets received at the current time horizon and at the previous one.	[50]
Maximal, minimal and average time to live (TTL)	Account time intervals between subsequent packets of the same flow.	[50]
Number of bytes sent in 1 second	Packets in bytes sent from the client to the server and from the server to the client.	[50]
Number of different resource paths of client	It includes the number of different resource paths that client has accessed.	[56]
Number of packets sent in 1 second	Packets sent from the client to the server and from the server to the client.	[50]
Number of request	Requests for the currently open windows and whether the number of requests for an open window.	[53]
Number of users	Set of real users accessing a server.	[53]
Percentage of encrypted packets with different properties	Since the traffic may be encrypted it is not always possible to define what web page these clients request.	[50]
Percentage of packets with different TCP flags	As it was mentioned in the previous section, in this study, we concentrate on the traffic transferred over TCP.	[50]
Session's requests	Requests for the currently open windows and whether the number of requests for an open window.	[52]
Sum of incoming payload of all clients of domain	If requests from attacking IP addresses were to be processed, inspected, and filtered based on the individual payload.	[56]
Sum of outgoing payload of all clients	If requests from attacking IP addresses were to be processed, inspected, and filtered based on the individual payload.	[56]

Sum of response times of all clients of domain	Properties of all clients that interact with the domain in the time interval	[56]
Sum of response times of client	Average durations until the first FIN packet is received and until the connection is closed, as well as the response time.	[56]
Users browsing process	We see average and total length of such browsing sequences.	[51]
Variance of the entropy	Variance of the entropy value, since the value of the variance provides the variations in the entropy value.	[58]
Web page requested	In the case of an application level DDoS attack, the attack packets are in the form of web page requests.	[57]

3 Feature of user behavior

3.1 Proposed features

The dynamism of the user is the user's interaction with the system and through it it is possible to know the behavior of a user and its difference with others [63]. The authentication of a user by means of his behavior has been a task studied from the point of view of information security [64]. Therefore, in order to avoid access by unauthorized users, several investigations [63]-[68] have focused their efforts on a process called biometric behavior. Within this process are: the use of keystrokes, mouse dynamics and the interaction with the graphical user interface (GUI) [64] for the identification of users.

Table 2 shows 24 characteristics that allow detecting the dynamism of the user and differentiating it from another. These characteristics are divided into two groups, these groups arise from the interaction of the mouse or keyboard and the user. In this paper, two characteristics are evaluated (mouse movement and right click), because in the data set used for the evaluation, these characteristics are present.

Mouse movement and right click allow to unequivocally identify a real user of a robot. In the case of mouse movement, a real user moves this peripheral to navigate through the web environment [69]. While right click is a special event that allows access to drop-down sub-menus, although it is not an event used regularly, it also identifies the dynamics of the user and the environment [68]. On the other hand, the robots are generated by specialized software to make the largest number of requests to a system [1], without the use of any peripheral.

It is worth mentioning that the characteristics presented in Table 2, despite being used in the biometric process to identify a user of another, these have not been proven in the differentiation of real users and robots.

Table 2. Features of the mouse and keyboard

ID	Mouse Features	Reference
M1	Single-click	[68]
M2	Double-click	
M3	Movement offset	
M4	Speed curve against time	
M5	Acceleration curve against time	

M6	Time	[69]
M7	Movement	
M8	Left or right button pressed or released	
M9	Coordinates of an event	
M10	Mouse position coordinates	[70]
M11	Mouse trajectory	
M12	Angle of the path in various directions	
M13	Curvature and its derivative	
M14	Mouse movement	
M15	Angular velocities	
M16	Tangential acceleration and jerk	
M17	Mouse movement coordinate	[71]
M18	Movement angle	
M19	Time to move	
M20	Time of mouse clicks	
	Keyboard Features	
K1	Number of key press events	[68]
K2	Average time between key press events	
K3	Average time per stroke	
K4	Number of times a given key has been pressed	

3.2 Features capture

Table 3 describes the characteristics of the mouse that can be captured and the techniques used for such purposes. These features can be captured using software developed in programming languages that incorporate libraries or special functions for this [68]-[71].

Table 3. Extraction Features of the mouse

ID	Extraction Method	Reference
M1	Windows application (written in C#)	[68]
M2		
M3		
M4		
M5		
M6	Java (kSquared.de library)	[69]
M7		
M8		
M9		
M10	NA	[70]
M11		
M12		
M13		
M14		

M15		
M16		
M17	Java applet and javascript	[71]
M18		
M19		
M20		

3.3 Classification algorithm

Figure 2 shows the classification algorithm that allows the identification of DDoS attacks by means of mouse features. The proposed characteristics allow to know if there is an attack or not, the process consists in verifying if the service request includes at least one of the proposed features, which is considered a human user otherwise it is considered a robot. The algorithm calculates the accuracy rate of DDoS attacks by verifying the number of attacks found by the algorithm between the numbers of actual attacks in the dataset.

```
Input: Dataset
begin
Query right click, mouse movement, request URL
Loop Dataset
if request URL is active
   if right click is active or mouse movement is active
      add user;
   else
      add attack;
Query abnormal URL
   accuracy is equal request - abnormal_URL;
end
output accuracy;
```

Fig. 2. Classification algorithm of real users and robots

4 Numerical experiments

4.1 Detection criteria

Figure 3 shows the architecture of the validation environment used for the construction of the classification algorithm of real users and DDoS attacks. In it, we consider the set of input data given by Lichman [12], and which is discussed in section 4.2. The use of the MySQL database manager was also observed for the extraction of the characteristics that were used in the validation, in order to create a new set of data with the selected characteristics. It enters the application created in Java for the classification process. It should be noted that the classification algorithm, the same one mentioned in section 3.3, allows the evaluation of the two interaction characteristics for the detection

of computer attacks, these being: mouse movement and right click. Finally, results reports are generated, in which the total number of DDoS attacks and actual users found is shown, as well as the total time spent executing the entire process.

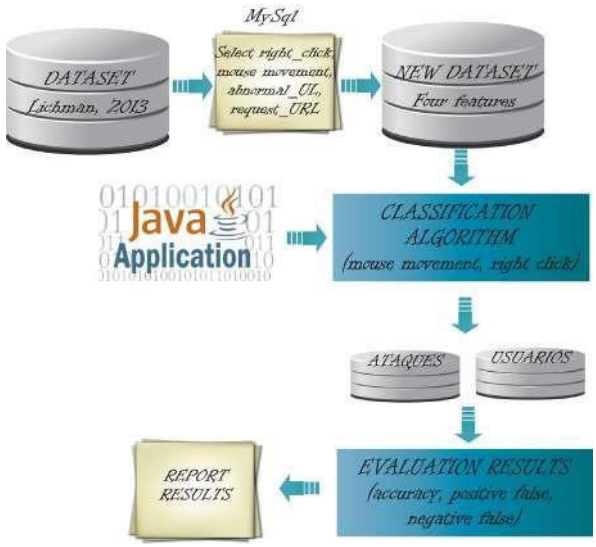

Fig. 3. Validation environment architecture

4.2 Dataset

The dataset used in this work for the validation process of the classification algorithm was created by Lichman [72]. It contains 11055, of which 9096 are real users and the rest are DDoS attacks. This data set was selected because it reports the characteristics of the mouse to be evaluated. In addition, this data set contains 31 attributes from which four were extracted to perform the validation (right click, mouse movement, abnormal URL and request URL). It should be noted that, through the URL request feature, it is known whether a request was made to the system or not. On the other hand, the abnormal URL allows identifying the requests that are computer attacks.

4.3 Feature extraction

Figure 4 shows the general algorithm that extracts the features proposed in this work. To do this, an active request is identified in the set with the data to then identify the proposed variables. The features are extracted by SQL queries to the database. After executing the consultations, all records are obtained where a service or resource has been requested for subsequent analysis and reporting of results.

```
Input: Request = active
begin
Open database
Query = Select mouse_movement,
right_click from dataset
Execute Query
end
```

Fig. 4. Algorithm used for the extraction of features

4.4 Results

The algorithm used to implement the classification criteria was created in Java version 1.8.0 using NetBeans IDE 8.2. The tests were developed on a machine whose processor is Intel (R) Core (TM) i7 CPU 2.60 GHz, 8 GB RAM, with Windows 10 operating system. Table 4 shows the attack detection rate obtained using the two characteristics of the mouse, this being 100%, both for the number of real users and for the number of DDoS attacks. This result shows that with the use of software designed for the detection of attacks and the use of the two characteristics of the user's dynamism, the highest precision rate is reached. It is worth mentioning that the time used by the application to perform the classification was 50 milliseconds. It should be mentioned that in this work it is difficult to identify false positives and negatives, because a dataset with exact data is used, where the interaction of the real user in the requests made is observed. Therefore, when a request is made, this is done through interaction with the mouse, otherwise it is a DDoS attack. However, it can be said that with the use of more features and means of data entry, there could be cases of false positives and negatives. These percentages show the importance of these characteristics for the detection of this type of computer attack.

Table 4. Detection efficiency of DDoS attacks

Users	Real data	Detection criteria	Compliance Rate (%)	Execution time (mls)
Real user	9096	9096	100	90
DDoS attacks	1959	1959	100	

4.5 Discussion

The results obtained in the tests carried out show that all DDoS attacks do not have the mouse and right click characteristics, so their detection is 100%. The evaluated characteristics (mouse movement and right click) show the dynamism of the user. Therefore, these characteristics allow to differentiate a real request from a computer attack. They use a low cost for the detection of an attack against other characteristics proposed in the literature, because the algorithm used consumes few resources because of the simplicity of the programmed code. These features also allow you to detect user behaviors that other features do not. For example, mouse operations that had not been

proposed in other works aimed at detecting DDOS attacks. It is worth mentioning that there are other characteristics of the dynamism of the user that can be considered for the identification of real users and robots (keyboard). However, with the use of more features and means of data entry, cases of false positives and negatives would appear. It should also be noted that with the advance in attack detection mechanisms, attackers find new alternatives to circumvent the mechanisms that are being proposed. Therefore, in the future attackers could falsify the variables that measure the characteristics of user behavior, simulating the input data and identifying a robot as a real user.

5 Conclusion

The review of the state of the art on the variables used in the detection of DDoS attacks at the application layer level shows that 30 variables have been used in the mechanisms published in the last 10 years. In this work we have introduced 24 new features based on the behavior of the web user. They are extracted from the transactionality of the user with the system in real time, therefore, they are computationally economic characteristics due to their easy obtaining. The numerical tests were performed using a dataset of 11055 requests between real users and attacks. The dataset used in the tests contains two of the 24 variables proposed in this paper for the detection of attacks in the application layer. The evaluation of the two variables (mouse movement and right click), using software designed in Java, managed to achieve 100% efficiency in the differentiation of real user and robot. Therefore, the right click and mouse movement variables are identified as characteristics of the user's dynamism. Therefore, these variables can be considered for their implementation in DDoS attack detection mechanisms.

6 Acknowledgment

The first author acknowledges the contributions, made by Professor Angel H. Moreno and the Technical University of Cotopaxi for the assigned doctoral scholarship.

7 References

[1] Kumar, P. A. R., & Selvakumar, S. (2011). Distributed denial of service attack detec-tion using an ensemble of neural classifier. Computer Communications, 34(11), 1328-1341. https://doi.org/10.1016/j.comcom.2011.01.012

[2] Beak, C., Chaudhry, J. A., Lee, K., Park, S., & Kim, M. (2007). A novel packet market-ing method in DDoS attack detection. American Journal of Applied Sciences, 4(10), 741-745. https://doi.org/10.3844/ajassp.2007.741.745

[3] Chen, Y., Das, S., Dhar, P., El-Saddik, A., & Nayak, A. (2008). Detecting and Prevent-ing IP-spoofed Distributed DoS Attacks. IJ Network Security, 7(1), 69-80.

[4] Yan, R., & Zheng, Q. (2009). Using Renyi cross entropy to analyze traffic matrix and detect DDoS attacks. Information Technology Journal, 8(8), 1180-1188. https://doi.org/10.3923/itj.2009.1180.1188

[5] Anurekha, R., Duraiswamy, K., Viswanathan, A., Arunachalam, V. P., Kumar, K. G., Rajivkannan, A. (2012). Dynamic approach to defend against distributed denial of service attacks using an adaptive spin lock rate control mechanism. Journal of Com-puter Science, 8(5), 632-636. https://doi.org/10.3844/jcssp.2012.632.636

[6] Chen, S. W., Wu, J. X., Ye, X. L., & Guo, T. (2013). Distributed denial of service at-tacks detection method based on conditional random fields. Journal of Networks, 8(4), 858. https://doi.org/10.4304/jnw.8.4.858-865

[7] Sachdeva, M., & Kumar, K. (2014). A traffic cluster entropy based approach to distin-guish DDoS attacks from flash event using DETER testbed. ISRN Communications and Networking, 2014. https://doi.org/10.1155/2014/259831

[8] Yau, D. K., Lui, J., Liang, F., & Yam, Y. (2005). Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. IEEE/ACM Trans-actions on Networking (TON), 13(1), 29-42. https://doi.org/10.1109/TNET.2004.842221

[9] Yaar, A., Perrig, A., & Song, D. (2005). FIT: Fast internet traceback. In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications So-cieties. Proceedings IEEE (Vol. 2, pp. 1395-1406). https://doi.org/10.1109/INFCOM.2005.1498364

[10] Xiang, Y., Li, K., & Zhou, W. (2011). Low-rate DDoS attacks detection and traceback by using new information metrics. IEEE transactions on information forensics and security, 6(2), 426-437. https://doi.org/10.1109/TIFS.2011.2107320

[11] Zhenwei, Y. (2011). Intrusion detection: a machine learning approach (Vol. 3). World Scientific.

[12] Ma, X., & Chen, Y. (2014). DDoS detection method based on chaos analysis of net-work traffic entropy. IEEE Communications Letters, 18(1), 114-117. https://doi.org/10.1109/LCOMM.2013.112613.132275

[13] Chen, Y., & Hwang, K. (2006). Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. Journal of Parallel and Distributed Computing, 66(9), 1137-1151. https://doi.org/10.1016/j.jpdc.2006.04.007

[14] Spyridopoulos, T., Karanikas, G., Tryfonas, T., & Oikonomou, G. (2013). A game theo-retic defence framework against DoS/DDoS cyber attacks. Computers & Security, 38, 39-50. https://doi.org/10.1016/j.cose.2013.03.014

[15] Seo, D., Lee, H., & Perrig, A. (2013). APFS: adaptive probabilistic filter scheduling against distributed denial-of-service attacks. Computers & Security, 39, 366-385. https://doi.org/10.1016/j.cose.2013.09.002

[16] Kumar, P. A. R., & Selvakumar, S. (2013). Detection of distributed denial of service at-tacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. Computer Communications, 36(3), 303-319. https://doi.org/10.1016/j.comcom.2012.09.010

[17] Kang, H. S., & Kim, S. R. (2014). sShield: small DDoS defense system using RIP-based traffic deflection in autonomous system. The Journal of Supercomputing, 67(3), 820-836. https://doi.org/10.1007/s11227-013-1031-7

[18] Xiao, P., Qu, W., Qi, H., & Li, Z. (2015). Detecting DDoS attacks against data center with correlation analysis. Computer Communications, 67, 66-74. https://doi.org/10.1016/j.comcom.2015.06.012

[19] Meenakshi, S., & Srivatsa, S. K. (2007). A distributed framework with less false posi-tive ratio against distributed denial of service attack. Information Technology Journal, 6(8), 1139-1145. https://doi.org/10.3923/itj.2007.1139.1145

[20] Liu, H., Sun, Y., & Kim, M. S. (2011). A Scalable DDoS Detection Framework with Victim Pinpoint Capability. JCM, 6(9), 660-670. https://doi.org/10.4304/jcm.6.9.660-670

[21] Udhayan, J., & Babu, M. R. (2013). Deteriorating distributed denial of service attack by recovering zombies using penalty scheme. Journal of Computer Science, 9(11), 1618. https://doi.org/10.3844/jcssp.2013.1618.1625
[22] Al-Duwairi, B., Al-Qudah, Z., & Govindarasu, M. (2013). A novel scheme for mitigat-ing botnet-based DDoS attacks. Journal of Networks, 8(2), 297. https://doi.org/10.4304/jnw.8.2.297-306
[23] Wang, Y., & Sun, R. (2014). An IP-traceback-based packet filtering scheme for elimi-nating DDoS attacks. Journal of Networks, 9(4), 874. https://doi.org/10.4304/jnw.9.4.874-881
[24] Chen, S., & Song, Q. (2005). Perimeter-based defense against high bandwidth DDoS at-tacks. IEEE Transactions on Parallel & Distributed Systems, (6), 526-537. https://doi.org/10.1109/TPDS.2005.74
[25] Kim, Y., Lau, W. C., Chuah, M. C., & Chao, H. J. (2006). PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. IEEE transactions on dependable and secure computing, 3(2), 141-155. https://doi.org/10.1109/TDSC.2006.25
[26] Chen, Y., Hwang, K., & Ku, W. S. (2007). Collaborative detection of DDoS attacks over multiple network domains. IEEE Transactions on Parallel & Distributed Sys-tems, (12), 1649-1662. https://doi.org/10.1109/TPDS.2007.1111
[27] Chen, R., Park, J. M., & Marchany, R. (2007). A divide-and-conquer strategy for thwarting distributed denial-of-service attacks. IEEE Transactions on Parallel and Distributed Sys-tems, 18(5), 577-588. https://doi.org/10.1109/TPDS.2007.1014
[28] Wang, H., Jin, C., & Shin, K. G. (2007). Defense against spoofed IP traffic using hop-count filtering. IEEE/ACM Transactions on Networking (ToN), 15(1), 40-53. https://doi.org/10.1109/TNET.2006.890133
[29] Chonka, A., Singh, J., & Zhou, W. (2009). Chaos theory based detection against net-work mimicking DDoS attacks. IEEE Communication Letters, 13(9), 717-719. https://doi.org/10.1109/LCOMM.2009.090615
[30] François, J., Aib, I., & Boutaba, R. (2012). FireCol: a collaborative protection network for the detection of flooding DDoS attacks. IEEE/ACM Transactions on Networking (TON), 20(6), 1828-1841. https://doi.org/10.1109/TNET.2012.2194508
[31] Chen, Y., Ma, X., & Wu, X. (2013). DDoS detection algorithm based on preprocessing net-work traffic predicted method and chaos theory. IEEE Communications Letters, 17(5), 1052-1054. https://doi.org/10.1109/LCOMM.2013.031913.130066
[32] Wu, X., & Chen, Y. (2013). Validation of chaos hypothesis in NADA and improved DDoS detection algorithm. IEEE Communications Letters, 17(12), 2396-2399. https://doi.org/10.1109/LCOMM.2013.102913.130932
[33] Luo, H., Lin, Y., Zhang, H., & Zukerman, M. (2013). Preventing DDoS attacks by iden-tifier/locator separation. IEEE network, 27(6), 60-65. https://doi.org/10.1109/MNET.2013.6678928
[34] Luo, J., Yang, X., Wang, J., Xu, J., Sun, J., & Long, K. (2014). On a Mathematical Mod-el for Low-Rate Shrew DDoS. IEEE Trans. Information Forensics and Security, 9(7), 1069-1083. https://doi.org/10.1109/TIFS.2014.2321034
[35] Mirkovic, J., & Reiher, P. (2005). D-WARD: a source-end defense against flooding denial-of-service attacks. IEEE transactions on Dependable and Secure Computing, 2(3), 216-232. https://doi.org/10.1109/TDSC.2005.35
[36] Lee, F. Y., & Shieh, S. (2005). Defending against spoofed DDoS attacks with path fin-ger-print. Computers & Security, 24(7), 571-586. https://doi.org/10.1016/j.cose.2005.03.005
[37] Al-Duwairi, B., & Manimaran, G. (2006). Distributed packet pairing for reflector based DDoS attack mitigation. Computer communications, 29(12), 2269-2280. https://doi.org/10.1016/j.comcom.2006.03.007

[38] Lee, K., Kim, J., Kwon, K. H., Han, Y., & Kim, S. (2008). DDoS attack detection meth-od using cluster analysis. Expert systems with applications, 34(3), 1659-1665. https://doi.org/10.1016/j.eswa.2007.01.040
[39] Lu, W. Z., Gu, W. X., & Yu, S. Z. (2009). One-way queuing delay measurement and its application on detecting DDoS attack. Journal of Network and Computer Applica-tions, 32(2), 367-376. https://doi.org/10.1016/j.jnca.2008.02.018
[40] Doron, E., & Wool, A. (2011). Wda: A web farm distributed denial of service attack atten-uator. Computer Networks, 55(5), 1037-1051. https://doi.org/10.1016/j.com-net.2010.05.001
[41] Zhang, C., Cai, Z., Chen, W., Luo, X., & Yin, J. (2012). Flow level detection and filter-ing of low-rate DDoS. Computer Networks, 56(15), 3417-3431. https://doi.org/10.1016/j.com-net.2012.07.003
[42] Wang, F., Wang, H., Wang, X., & Su, J. (2012). A new multistage approach to detect subtle DDoS attacks. Mathematical and Computer Modelling, 55(1-2), 198-213. https://doi.org/10.1016/j.mcm.2011.02.025
[43] Rahmani, H., Sahli, N., & Kamoun, F. (2012). DDoS flooding attack detection scheme based on F-divergence. Computer Communications, 35(11), 1380-1391. https://doi.org/10.1016/j.comcom.2012.04.002
[44] Lee, S. M., Kim, D. S., Lee, J. H., & Park, J. S. (2012). Detection of DDoS attacks using optimized traffic matrix. Computers & Mathematics with Applications, 63(2), 501-510. https://doi.org/10.1016/j.camwa.2011.08.020
[45] Varalakshmi, P., & Selvi, S. T. (2013). Thwarting DDoS attacks in grid using infor-mation divergence. Future Generation Computer Systems, 29(1), 429-441. https://doi.org/10.1016/j.future.2011.10.012
[46] Li, L., & Lee, G. (2005). DDoS attack detection and wavelets. Telecommunication Systems, 28(3-4), 435-451. https://doi.org/10.1007/s11235-004-5581-0
[47] Kulkarni, A., & Bush, S. (2006). Detecting distributed denial-of-service attacks using kol-mogorov complexity metrics. Journal of Network and Systems Management, 14(1), 69-80. https://doi.org/10.1007/s10922-005-9016-3
[48] Xiao, B., Chen, W., & He, Y. (2006). A novel approach to detecting DDoS attacks at an early stage. The Journal of Supercomputing, 36(3), 235-248. https://doi.org/10.1007/s11227-006-8295-0
[49] Kang, S. H., Park, K. Y., Yoo, S. G., & Kim, J. (2013). DDoS avoidance strategy for service availability. Cluster computing, 16(2), 241-248. https://doi.org/10.1007/s10586-011-0185-4
[50] Zolotukhin, M., Kokkonen, T., Hämäläinen, T., & Siltanen, J. (2016). On Application Layer DDoS Attack Detection in High-Speed Encrypted Networks.
[51] Dick, U., & Scheffer, T. (2016). Learning to control a structured-prediction decoder for de-tection of HTTP-layer DDoS attackers. Machine Learning, 104(2-3), 385-410. https://doi.org/10.1007/s10994-016-5581-9
[52] Xie, Y., & Yu, S. Z. (2009). Monitoring the application-layer DDoS attacks for popu-lar websites. IEEE/ACM Transactions on Networking (TON), 17(1), 15-25. https://doi.org/10.1109/TNET.2008.925628
[53] Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., & Knightly, E. (2009). DDoS-shield: DDoS-resilient scheduling to counter application layer attacks. IEEE/ACM Transactions on networking, 17(1), 26-39. https://doi.org/10.1109/TNET.2008.926503
[54] Giralte, L. C., Conde, C., De Diego, I. M., & Cabello, E. (2013). Detecting denial of ser-vice by modelling web-server behaviour. Computers & Electrical Engineering, 39(7), 2252-2262. https://doi.org/10.1016/j.compeleceng.2012.07.004

[55] Zhou, W., Jia, W., Wen, S., Xiang, Y., & Zhou, W. (2014). Detection and defense of application-layer DDoS attacks in backbone web traffic. Future Generation Computer Systems, 38, 36-46. https://doi.org/10.1016/j.future.2013.08.002

[56] Huang, C., Wang, J., Wu, G., & Chen, J. (2014). Mining Web User Behaviors to Detect Application Layer DDoS Attacks. JSW, 9(4), 985-990. https://doi.org/10.4304/jsw.9.4.985-990

[57] Saravanan, R., Shanmuganathan, S., & Palanichamy, Y. (2016). Behavior-based detec-tion of application layer distributed denial of service attacks during flash events. Turkish Journal of Electrical Engineering & Computer Sciences, 24(2), 510-523. https://doi.org/10.3906/elk-1308-188

[58] Johnson Singh, K., Thongam, K., & De, T. (2016). Entropy-Based Application Layer DDoS Attack Detection Using Artificial Neural Networks. Entropy, 18(10), 350. https://doi.org/10.3390/e18100350

[59] Gavrilis, D., & Dermatas, E. (2005). Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. Computer Networks, 48(2), 235-245. https://doi.org/10.1016/j.comnet.2004.08.014

[60] Nguyen, H. V., & Choi, Y. (2008). Proactive detection of DDoS attacks using k-NN classifier in an Anti-DDoS Framework. International Journal of Computer System Science and Engineering, 247-252.

[61] Wang, D., Chang, G., Feng, X., & Guo, R. (2008). Research on the detection of distrib-uted denial of service attacks based on the characteristics of IP flow. In IFIP Interna-tional Conference on Network and Parallel Computing (pp. 86-93). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88140-7_8

[62] Xiang, Y., & Zhou, W. (2005). Mark-aided distributed filtering by using neural net-work for DDoS defense. In GLOBECOM'05: IEEE Global Telecommunications Con-ference, 28 November-2 December 2005 St. Louis, Missouri, USA, discovery past and future (pp. 1701-1705). IEEE Globecom.

[63] Ghezzi, C., Pezzè, M., Sama, M., & Tamburrelli, G. (2014). Mining behavior models from user-intensive web applications. In Proceedings of the 36th International Con-ference on Software Engineering (pp. 277-287). ACM. https://doi.org/10.1145/2568225.2568234

[64] Stevanovic, D., & Vlajic, N. (2014). Application-layer DDoS in dynamic Web-domains: Building defenses against next-generation attack behavior. In Communica-tions and Network Security (CNS), 2014 IEEE Conference on (pp. 490-491). https://doi.org/10.1109/CNS.2014.6997519

[65] R. J. Urban. (2015). Detection of exit behavior of an Internet user. U.S. Patent Appli-cation No 14/829,409.

[66] Abramson, M., & Aha, D. W. (2013). User Authentication from Web Browsing Behav-ior. In FLAIRS conference (pp. 268-273).

[67] Kim, Y., & Kim, I. (2014). Involvers' Behavior-based Modeling in Cyber Targeted At-tack. Proceedings of SECURWARE.

[68] Shen, C., Cai, Z., Guan, X., Du, Y., & Maxion, R. A. (2013). User authentication through mouse dynamics. IEEE Transactions on Information Forensics and Security, 8(1), 16-30. https://doi.org/10.1109/TIFS.2012.2223677

[69] Salmeron-Majadas, S., Santos, O. C., & Boticario, J. G. (2014). An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective modeling in an educational context. Procedia Computer Science, 35, 691-700. https://doi.org/10.1016/j.procs.2014.08.151

[70] Graepel, T., Candela, J. Q., Borchert, T., & Herbrich, R. (2010). Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. Omnipress.
[71] Gamboa, H., & Fred, A. L. (2003). An Identity Authentication System Based On Hu-man Computer Interaction Behaviour. In PRIS (pp. 46-55).
[72] M. Lichman. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Ir-vine, CA: University of California, School of Infor-mation and Computer Science.

8 Authors

Silvia Bravo was born in Latacunga, Ecuador. She graduated from the Technical University of Cotopaxi in 2007, where she received the title of "Computer Science". She is currently pursuit a Ph.D. from National University of San Marcos within the Doctoral Program of "Computer and System". She is currently working as a professor and researcher at the Faculty of Engineering Science, in the Technical University of Cotopaxi. Her research activity is mainly focused on the software development and informatics security.

David Mauricio was born in Lima, Peru. He graduate from the National San Marcos University in 1987, wherehe received the title of "Computer Science". He obtained the title of "Master in Mathematics Applied" from the Federal University of Rio de Janeiro, Brazil, in 1991. In 1994, he obtained the title of "Doctor in Systems Engineering" from the Federal University of Rio de Janeiro. He is currently working as a professor at the Faculty of Systems Engineering, in the National Mayor de San Marcos University and scientific consultant in National Council for Science and Technology (CONCYTEC). His research activity is mainly focused on the Combinatorial optimization, Designs and analysis of algorithms, Heuristics search, Metaheuristics, Mathematical programming, Expert systems, Data mining, Artificial intelligence.

Article submitted 29 August 2018. Resubmitted 14 and 16 September 2018. Final acceptance14 October 2018. Final version published as submitted by the authors.

ANEXO 3

ISSN (Print) : 2319-8613
ISSN (Online) : 0975-4024

Agile method for detecting DDoS attacks in the application layer based on user's dynamism

Silvia Bravo [#1], David Mauricio [*2]
[#] Faculty of Engineering and Applied Sciences,
Technical University of Cotopaxi Latacunga, Ecuador
[1] silvia.bravom@utc.edu.ec
[*] Faculty in Systems Engineering and Computer Science,
National University of San Marcos Lima, Peru
[2] dmauricios@unmsm.edu.ec

Abstract— **DDoS attacks are one of the most damaging computer attacks of recent times. Attackers send large number of requests to saturate a victim machine and it stops providing its services to legitimate users. In general attacks are directed to the network layer and the application layer, the latter has been increasing due mainly to its easy execution and difficult detection. The present work proposes a low cost detection approach that consists of two steps: first, user characteristics are extracted in real time while browsing the web application; second, each extracted feature is used by an order sorter O(1) to differentiate a real user from a DDoS attack. A real user is identified by making requests using peripherals for navigation (user dynamism), while DDoS attacks are requests sent by robots and do not require the use of peripherals to make requests, therefore the characteristics of the user's dynamism are used for the detection of a DDoS attack. The results on the attack tests using the attack tools LOIC, OWASP and GoldenEye, show that the proposed method has a detection efficiency of 100%, and that the characteristics of the web user allow to differentiate between a real user and a robot.**

Keyword - Application layer, DDoS, user's dynamism, detection attacks, use of peripherals

I. INTRODUCTION

DDoS attacks have become one of the threats with the greatest impact on the security of computer systems. These attacks are aimed at consuming bandwidth or server resources, preventing legitimate users from accessing the services. These attacks can be in the network layer (protocols, hubs, switch) and in the application layer (system, CPU, resources), the latter has increased in recent years due to its easy execution and difficult detection, thus, the efforts in the mechanisms of detection are focusing to this type of attack. The attacks directed to the application layer are considered sophisticated because they mimic the requests of real users, so it is more difficult to detect them. The methods consider information of user requests, and some logic that allows to relate these with an attack or a user. The logic is given by techniques such as neural networks, genetic algorithms, support vector machine and statistical models that in general consume considerable resources. The excessive consumption of resources means that the detection process is slower and even more so with large amounts of information. The slowness of the process impacts the system causing saturation of the bandwidth and consumption of server resources. In addition, the mentioned techniques have a waiting time before detection, to know if it is an attack, which affects the productivity of the services. The most difficult task that detection methods have is to differentiate a request to identify it as a real user or attack. In [1] they introduces the characteristics of the user's dynamism, indicating that they come from the interaction between the user and the system. In [2] they specify that the characteristics of the user's dynamism allow differentiating a robot from a real user.

The attack detection mechanisms do not contemplate any of the characteristics of the user's dynamism. Therefore, in this work we propose a simple and low cost method based on the characteristics of the user's dynamism for the detection process of DDoS attacks in the application layer. To do this, keystrokes, mouse dynamics and interaction with the graphical user interface (GUI) [3] for the identification of real users are evaluated. The proposed method has been validated in a case study to evaluate its efficiency. For this, an algorithm has been implemented that detects the interaction of the user and the system. It was tested on a web system in real time. The system has a three-layer architecture to implement the user interface and the detection algorithm. The attacks were generated using the LOIC, OWASP and GoldeEye tools to provoke flood attacks. This work is organized as follows. In section 2, a review of the literature of DDoS attack detection methods at the application layer level is made. Section 3 proposes the agile method of detecting DDoS attacks by using the dynamism of the user. In section 4 the numerical experiments are carried out and the results are shown and, finally, the conclusions are presented in session 5.

II. RELATED WORK

The review of the literature regarding the detection methods of DDoS attacks in the application layer records nine proposed methods. Hidden semi-Markov Model is a method that analyses the statistics of the user's search process and access to web objects [4] [5]. However, it has been proven that robots are able to emulate search patterns and access statistics recorded in a session [6]. A mechanism that counts the requests made by a user in a session called Counter Mechanism was implemented to detect attacks [7]. However, robots can simulate statistics by mimicking requests from real users [8]. A Fuzzy Estimator implemented in an attack detection mechanism allows analysing the number of requests, number of users and access patterns in order to establish statistics to identify anomalies in the system [9]. Attackers have developed robots that are capable of generating requests by imitating the number of requests and users, as well as patterns of access to the system [5]. The correlation analysis has also been used in the detection of attacks, indicating the statistical probability of sending requests from the same group of IP addresses [10]. When the attackers have a group of computers under their control, they can make requests from different places avoiding correlation of the points where the request arises. [11] Support vector machine is used to analyse the statistics of the sessions of each client to later identify the anomalies [12]. For this, in this method the characteristics are used: strings of client, paths of client, all clients of domain, connections of client, response times, request type, payload of all clients. However, these characteristics correspond to statistics of user sessions that have to be processed by SVM, which implies a high computational cost and consumption of server resources. For the detection of attacks, prototype systems were also used, such as the mechanism called intrusion detection system (IDS) [13]. In this mechanism statistics of incoming requests were used as: duration of the conversation, number of packets, number of bytes, average packet size, size of TCP window, average time, percentage of packets, and percentage of encrypted packets. Despite being an innovative proposal, being built in Python, becoming an application aimed at detecting anomalies, resource costs turn out to be high. The Hellinger metric has also been used in the detection of computer attacks [14]. Two techniques have been used for attack detection, Neural Networks and Genetic algorithm. These techniques use the characteristics of incoming requests, analysing the entropy and variance of the captured characteristics. It should be noted that these mechanisms employ features that can be easily simulated by attackers (web page requested, request count) by employing robots that issue requests from low-speed users.

TABLE I. Detection Method and features for the detection of DDoS attacks in the application layer

Method	**Features**	**References**
Hidden semi-Markov Model (HsMM)	Users' browsing process Access to web objects	[4] [5]
Counter mechanism	Session's requests	[7]
Fuzzy estimator	Number of request Number of users Access pattern	[9]
Correlation analysis	IP address	[10]
Support vector machine (SVM)	Average length of query strings of client Number of different resource paths of client Sum of incoming payload of all clients of domain Fraction of connections of client that request the most frequent resource path Sum of response times of all clients of domain Sum of response times of client Fraction of connections for domain that accepts any version of English Entropy of request type (GET/POST/OTHER) Sum of outgoing payload of all clients	[12]
Intrusion Detection System Prototype	Duration of the conversation Number of packets sent in 1 second Number of bytes sent in 1 second Maximal, minimal and average packet size Maximal, minimal and average size of TCP window maximal, minimal and Average time to live (TTL) percentage of packets with different TCP flags Percentage of encrypted Packets with different	[13]

	properties	
Hellinger Distance Metric	Flow similarity Client legitimacy Web page requested	[14]
Neural Networks Genetic algorithm	HTTP GET request count Entropy of the requests Variance of the entropy	[15]

Table I shows the methods and features used by the DDoS attack detection mechanisms in the application layer. In total nine methods and thirty characteristics are observed. It is also observed that SVM uses the greatest number of features for the detection of attacks, which implies high computational costs. The detection mechanisms [13] [15] are the ones with the highest degree of detection, 98.5% and 98.32% respectively. This is mainly due to the fact that in the first case, a system is implemented for the exclusive detection of anomalies, it is implemented in Python. While in the second case, two techniques for data analysis are merged. However, in none of the two cases are the characteristics of user dynamism considered.

III. PROPOSED METHOD

In this work we present a low cost detection method that allows detecting DDoS attacks oriented to the application layer. For this, it uses characteristics of the dynamism of the user extracted in real time. These characteristics show the user's interaction with the system.

A. Architecture of the Detection Method

Figure 1 shows the architecture used for the implementation of the DDoS attack detection method. In the same it is observed the entrance of the requests coming from the Internet to the interface of the web application. The requests made generate a data bank where the established connections and the processes performed are recorded. The data bank generated in the application layer is analysed by an interaction detector. At the application level, the processes that the user generates are recorded (links, resources, forms, etc.). The detector records the activity between the user and the mouse and keyboard peripherals. The characteristics of Table I are extracted in real time by programming in PHP and Javascript. These characteristics are stored until the user executes the next request. Both the request and the characteristics of the user are sent to the detection algorithm of Figure 1 for evaluation. As indicated in the previous section, this algorithm is responsible for determining the existence of requests and interactions with the system, taking a decision between real user or computer attack.

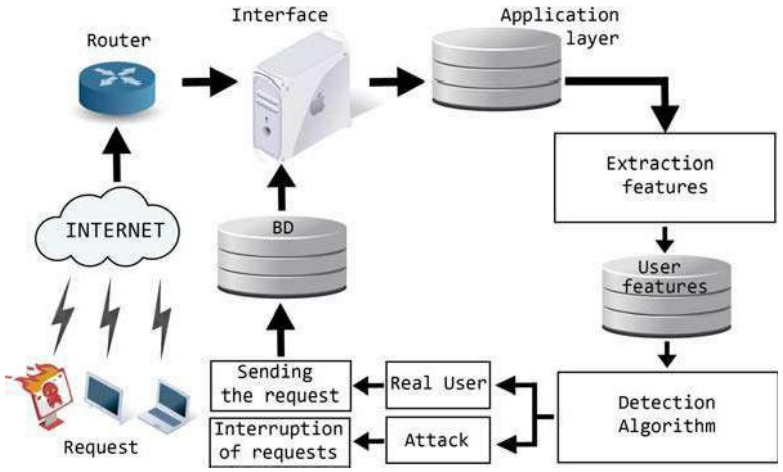

Fig. 1. Architecture of the detection method

B. User Dynamism

The present work considers the characteristics of the dynamism of the user in the computer system. The dynamism of the user arises when the user interacts with the system. In [16], mentions that the user's dynamics are the interests of the users and their preferences. The model of user requests and server responses provide limited knowledge about user behaviour. For better compression it is better to move to the client side. To do this, collect information such as mouse move, click, blur, or resize. In [17] they mentions that an alternative to predict the next web page to be opened by a user, comes from the dynamism of the movement of the mouse with the direction it takes in the graphical interface. In [18] they proposes a technique to identify users by grouping keystroke dynamics. In [19] they used the pulse dynamics, which uses the rhythm and the way in which an individual writes characters on the keyboard, it is used as behavioural biometrics. The keystroke rhythms of a

user, in terms of time, are measured to develop a unique biometric template of the user's typing pattern for future authentication. In [20] they evaluated the characteristics of the mouse to identify real users of DDOS attacks. Checking that the dynamism of the mouse provides unique characteristics to identify this type of attack.

C. Characteristics of the User's Dynamism

The characteristics of user behaviour are extracted from the processes between the peripherals used and the interaction with the system. In this work, the dynamism of the user is observed through the transactions that are made with the mouse and keyboard peripherals. Table II shows the user characteristics that are extracted and used in the proposed DDoS attack detection method. It is worth mentioning that these features are extracted using PHP and Javascript functions in real time.

TABLE II. Features of the user's dynamism

Id	Features	Description
f1	Mouse move	The mouse is moved to a location on the screen to perform an action.
f2	Mouse click	When a user presses and releases a mouse button and there are five types of click events that are recorded: left click, right click, and double left click.
f3	Mouse highlight	This action begins with a left mouse click/hold to begin the highlighting and ends with the mouse release.
f4	Mouse drag	When an object is dragged and dropped. This action begins with a left mouse click/hold and ends with the mouse release
f5	Mouse drop	
f6	Mouse scroll	The Mouse Wheel or Scroll is an event when the movement of the wheel or scroll has a net up or down effect. The resultant effect is based on the consecutive wheel or scroll movements.
f7	Mouse wheel	
f8	Key press	This happens when a user presses a key and slides the touch device (finger or stylus)

D. Architecture of the detection method

Figure 2 shows the algorithm used to capture the characteristics of the user's dynamism. The algorithm works every time the user performs an operation with the mouse or keyboard and its interaction with the graphical interface. When a user interacts with the mouse and keyboard it is registered by means of a Javascript function. The captured characteristics are stored in a register to be sent in the following to be checked by the detection algorithm. When a user requests a service, the user is forced to use a peripheral to make the request. The capture of the user's characteristics consists of taking the pulsations that the user is making with the peripherals. When a user interacts with a peripheral it is registered by a Javascript function in a data bank or registry.

```
1. begin catch
2. when interaction then
3. id = operation
4. if request then
5. submit id
6. end catch
```

Fig. 2. Algorithm for capture features (ACF).

E. Detection and mitigation algorithm

The main idea of the algorithm is to verify if the request made to the system presents any of the characteristics of the web user to differentiate a real user from a computer attack in real time.

1. Input request
2. begin verification
3. for i equal 1 to 8
4. if fi stores true then id stored true
5. end verification
6. begin send each request
7. when id stores true then execute query request
8. when id stores false then execute message
9. end send

Fig. 3. Algorithm for detection of DDoS attacks (ADDA).

In Figure 3 the proposed attack detection algorithm is presented, it uses the characteristics of the dynamism of the web user that are shown in Table I and verifies if they are active or not. For this purpose, 1) a request is made to the system, 2) the verification of the captured characteristics begins, 3) a repetitive loop is used that goes from one to the total of characteristics used in this work, 4) if the analysed characteristic they have been activated, the activations performed will be stored in another variable, 5) and the verification of the characteristics of the user's dynamism is completed. 6) When a request is made, the characteristics of the user's dynamism must be verified. 7) When the variable that stores the verification is active, the request is made. 8) When the variable that stores the verification is inactive, a message is sent that must be answered by the user otherwise the request will not be given. This last step is in mitigating the algorithm that requests from attackers are sent to the server.

IV. Numerical Experiments

A. *Experimental Design*

To validate the proposed algorithm, the web services of a hotel in the city of Detroit in the United States were considered. It receives 3100 passengers annually. The hotel has a restaurant service and the information of it is in a dedicated server, it uses Linux CentOS, 4-core processor, 8 GB memory, 1TB disk space. Figure 4 shows the built-in validation environment that allows the incorporation of three levels for the detection of DDoS attacks. The first level involves the user interface, where the user interacts with the system making requests for links, videos, graphics, etc. In the second level are located the functions that load of information to all the applications of the system. In this level are the functions that perform the call to the ACF algorithm. Finally, on the third level is the ADDA algorithm. The response of the algorithm has two outputs, execute the request made by the user or send a verification message.

Fig. 4. Validation environment for detection of DDoS attacks.

The server used in this work has been subjected to a series of simulated attacks to verify the efficiency of the proposed method. The results obtained have been extracted using the same attack tools for later analysis.

B. Simulation of Attacks

To generate DDoS attacks LOIC software (Low Orbit Ion Canon) [21], OWASP DOS HTTP POST [22] and GoldenEye HTTP [23] were used. It is worth mentioning that these tools were selected because they are the most used for the generation of this type of attacks, due to their simplicity and effectiveness [23]. To do this, several attacks were made with each tool towards the hotel server (victim), in order to evaluate the attack rate needed to overload the server. In each attack, the overload values were obtained, which would then be evaluated using the proposed detection algorithm. Table III shows the tools that were used to simulate the attack on the web system, the amount of solitudes generated and the time it took the system to overload.

TABLE III. Evaluation Results without detection method

Tool attack	Number of request	System overload time (min)
LOIC	4800	2.15
OWASP DOS HTTP POST	4000	1.30
GoldenEye HTTP Denial Of Service Tool	5300	3.20

The results of Table III show that the computer attacks generated by the LOIC, OWASP and GoldenEye software use about two minutes to overload the system, causing inaccessibility to resources and services for real users. It is also observed that the number of requests used to overcharge the system varies between 4000 and 5300.

C. Results

Table IV shows the results obtained using the proposed detection method. It shows 100% of attacks generated by the tools have been detected effectively. The time used in the detection was on average 60 milliseconds and the same amounts of simulation requests were used to generate the attack. It is worth mentioning that there are no dataset related to DDoS attacks for tests. In addition, the works with the highest detection rate in the application layer [13] and [14] do not show the tools that were used to evaluate the proposed methods.

TABLE IV. Evaluation results with the detection method

Tool attack	Number of request	Detection time (mil)	Detection rate %
LOIC	4800	60	100
OWASP DOS HTTP POST	4000	58	100
GoldenEye HTTP Denial Of Service Tool	5300	63	100

Table IV shows that the detection mechanism developed through the use of web user dynamism features is effective with a 100% detection rate for the three attack generation tools. In addition, the time spent is around 60 milliseconds. These results show the effectiveness of the detection method through user interaction with the system through the peripherals used. It should be mentioned that with the improvement of the detection mechanisms, the attackers also improve their attack strategies, so the possibility that the input values of the user characteristics evaluated in this work can be supplanted is not ruled out.

V. CONCLUSION

This paper presents an agile and effective detection mechanism based on the characteristics of the web user's dynamism for the detection of DDoS attacks in the application layer. This mechanism employs eight new characteristics of user behavior that have not been used in any other similar work. The method of detecting DDoS attacks using the characteristics of user behavior has a 100% effectiveness in detection. This result shows the influence of the characteristics that identify a user when interacting with the system. The tests in a real-time platform and the application of the attack tools LOIC, OWASP and GoldenEye allow to evaluate the algorithm under a simulated attack environment. These simulations allowed to verify that the algorithm reaches an optimal result when processing large quantities of requests.

REFERENCES

[1] I. Brosso, A. La Neve, G. Bressan, and W. V. Ruggiero, “A continuous authentication system based on user behavior analysis”. Availability, Reliability, and Security, 2010. ARES'10 International Conference on IEEE, 2010.
[2] G. Oikonomou, and J. Mirkovic. "Modeling human behavior for defense against flash-crowd attacks." Communications, 2009. ICC'09. IEEE International Conference on IEEE, 2009.
[3] M. Abramson, & D. W. Aha, User Authentication from Web Browsing Behavior. FLAIRS conference, 2013.
[4] Y. Xie and S. Z. Yu, Monitoring the application-layer DDoS attacks for popular websites. IEEE/ACM Transactions on Networking (TON), vol. 17, no 1, p. 15-25, 2009.
[5] C. Huang, J. Wang, G. Wu, and J. Chen, Mining Web User Behaviors to Detect Application Layer DDoS Attacks. JSW, 9(4), 985-990, 2014.
[6] F. Yu, Y. Xie, and Q. Ke, Sbotminer: large scale search bot detection. Proceedings of the third ACM international conference on Web search and data mining, pp. 421-430, 2010.

[7] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci and E. Knightly, DDoS-shield: DDoS-resilient scheduling to counter application layer attacks. IEEE/ACM Transactions on Networking (TON), vol. 17, no 1, p. 26-39, 2009.
[8] C. Ye, and K. Zheng, Detection of application layer distributed denial of service. Computer science and network technology (ICCSNT) 2011 International Conference, Vol. 1, pp. 310-314, 2011.
[9] L. C. Giralte, C. Conde, I. M. De Diego, E. Cabello, Detecting denial of service by modelling web-server behaviour. Computers \& Electrical Engineering, vol. 39, no 7, p. 2252-2262, 2009.
[10] 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, 38, 36-46, 2014.
[11] N. Hoque, H. Kashyap, and D. K. Bhattacharyya, Real-time DDoS attack detection using FPGA. Computer Communications, 110, 48-58, 2017.
[12] U. Dick, T. Scheffer, Learning to control a structured-prediction decoder for detection of HTTP-layer DDoS attackers. Machine Learning, 104(2-3), 385-410, 2016.
[13] M. Zolotukhin, T. Kokkonen, T. Hämäläinen and J. Siltanen, On Application-Layer DDoS Attack Detection in High-Speed Encrypted Networks. International Journal of Digital Content Technology and its Applications, 2016.
[14] R. Saravanan, S. Shanmuganathan and Y. Palanichamy, Behavior-based detection of application layer distributed denial of service attacks during flash events. Turkish Journal of Electrical Engineering \& Computer Sciences, 24(2), 510-523, 2016.
[15] K. Johnson Singh, K. Thongam and T. De, Entropy-Based Application Layer DDoS Attack Detection Using Artificial Neural Networks. Entropy, 18(10), 350, 2016.
[16] L. A. Leiva, Mining the browsing context: Discovering interaction profiles via behavioral clustering. User Modeling, Adaptation and Personalization (UMAP), 19, 2011.
[17] A. Kundu, Dynamic web prediction using asynchronous mouse activity. Computational Social Networks Springer, London, pp. 257-280, 2012.
[18] S. Sznur, Advances in Keystroke Dynamics Techniques to Group Users Sessions. International Journal of Information Security Science, 4(2), 26-38, 2015.
[19] G. Kulkarni, R. Chandorkar and N. Chavan, A Security By Biometric Authentication. International Journal of Computer Science and Engineering Research and Development (IJCSERD), 2(1), 7-14, 2012.
[20] S. Bravo, D. Mauricio and Á. H. Moreno, Mouse Features for DDoS Attacks Detection in the Application Layer. Proceedings of the 9th International Conference on Information Management and Engineering, pp. 177-181, 2017.
[21] L. Y. Zhang, Q. I. A. N. Ming and Y. B. Chi, DDoS Attack Detection Using Sliding Window Method. DEStech Transactions on Computer Science and Engineering, 2017.
[22] M. Y. Arafat, M. M. Alam and M. F. Alam, A Practical Approach and Mitigation Techniques on Application Layer DDoS Attack in Web Server. International Journal of Computer Applications, 131(1), 2015.
[23] J. Jazi, H. Gonzalez, N. Stakhanova and A. A. Ghorbani, Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. Computer Networks, 121, 25-36, 2017.

AUTHOR PROFILE

Silvia Bravo was born in Latacunga, Ecuador. She graduated from the Technical University of Cotopaxi in 2007, where she received the title of "Computer Science". She is currently pursuit a Ph.D. from National University of San Marcos within the Doctoral Program of "Computer and System". She is currently working as a professor and researcher at the Faculty of Engineering Science, in the Technical University of Cotopaxi. Her research activity is mainly focused on the software development and informatics security.

David Mauricio was born in Lima, Peru. He graduate from the National San Marcos University in 1987, where he received the title of "Computer Science". He obtained the title of "Master in Mathematics Applied" from the Federal University of Rio de Janeiro, Brazil, in 1991. In 1994, he obtained the title of "Doctor in Systems Engineering" from the Federal University of Rio de Janeiro. He is currently working as a professor at the Faculty of Systems Engineering, in the National Mayor de San Marcos University and scientific consultant in National Council for Science and Technology (CONCYTEC). His research activity is mainly focused on the combinatorial optimization, designs and analysis of algorithms, heuristics search, metaheuristics, mathematical programming, expert systems, data mining, and artificial intelligence.

ANEXO 4

Mouse Features for DDoS Attacks Detection in the Application Layer

Silvia Bravo
Universidad Técnica de Cotopaxi
Ave. Simón Rodríguez
Barrio El Ejido, Sector San Felipe,
Latacunga, Ecuador
(+593) 03 2252205 Ext. 139
silvia.bravom@utc.edu.ec

David Mauricio
Universidad Nacional de San Marcos
Calle Germán Amézaga, N° 375
Lima 1, Lima, Perú
(+0051) 619-7000
dmauricios@unmsm.edu.pe

Ángel H. Moreno
Universidad Técnica de Cotopaxi
Ave. Simón Rodríguez
Barrio El Ejido, Sector San Felipe,
Latacunga, Ecuador
(+593) 3 2252205 Ext. 139
angel.hernandez@utc.edu.ec

ABSTRACT
DDoS attacks are a threat to the security of the network. In recent years these attacks have been directed especially against the application layer. This phenomenon is mainly due to the large number of existing tools available for the easy generation of this type of attack. The methods used in the application layer reach detection rates of between 98.5 and 98.32%. Therefore, the problem of detecting DDoS attacks persists. In this work we propose a detection alternative based on the dynamism of the web user. To do this, two mouse characteristics are evaluated: movement and right click. A dataset of 11055 applications was also used, from which the two characteristics were extracted and entered into a classification algorithm. To this end, a Java application was developed for the classification of real users and those behind DDoS attacks. The results show that the proposed characteristics achieve an efficiency of 100%. It is concluded that these characteristics reveal the dynamism of the user and can be used as a method of detection of DDoS attacks.

CCS Concepts
• **Security and privacy Systems security; Denial-of-service attacks**

Keywords
DDoS; denial of service; mouse features; user dynamism

1. INTRODUCTION
The detection of Distributed Denial of Service (DDoS) attacks is one of the biggest problems facing the security architecture of the network. Therefore, it has become an important factor of study in the field of computer security. A DDoS attack occurs when an attacker coordinates his attacks using several machines, called zombies, against a specific objective or server. The aim of the attacker is to direct massive requests at the victim machine to saturate it so that it stops serving the requests of real users.

ICIME 2017, October 9-11, 2017, Barcelona, Spain

ACM ISBN 978-1-4503-5337-3/17/10$15.00
DOI: https://doi.org/10.1145/3149572.3149609

In order to counteract this type of attack several detection mechanisms have been proposed, both at the network level [1-49] and at the application level [50-56]. The highest detection rate obtained to date, 99.4%, has been achieved by implementing a network-level method [1]. The dataset used for this work is the KDD cup dataset, from which 300,000 connection records were extracted between DDoS attacks and real users. On the other hand, with the methods implemented at the application layer level the obtained detection rate is 98.5 [50], from of which there is no information from the dataset used. Nevertheless a virtual network was used to simulate the generation of real users and tools for attack generation (Sslsqueeze and Slowloris). This proposal also measures the following characteristics: source IP address, source port, destination IP address and destination port. However, the efficiency of detection depends not only on the characteristics and but also on the test dataset used. Until now most attacks have been directed at the application layer, because they are easy to execute with the large amount of existing software [50, 56]; and they are more difficult to detect because they are illegitimate requests that are camouflaged as requests from real users. Detection mechanisms, for the most part, focus their efforts on the network layer. Therefore, the interaction between the user and the system is not considered. However, the characteristics of user dynamism are significant for differentiating between a human user and a robot [53]. Accordingly, the main objective of the study is to evaluate two of the characteristics of the user behavior, mouse movement and right click, for its possible implementation in the detection mechanisms of DDoS attacks.

2. RELATED WORK
In the field of computer security, the detection of attacks denominated DDoS, is a rapidly expanding field of investigation because of its high impact on computer systems and its easy implementation [1]. In the mechanisms of detection of this type of attack, at the application layer level, characteristics that measure the behavior of the user have been studied [50-55]. However, these methods do not take into account the dynamism of the web user. For example, Huang et al. [53] calculate the rate of links established in a search and if it experiences search inconsistencies, it becomes clear that an attack is occurring. Xie *et al*. [51] propose the analysis of the web search process of a user by determining attributes, such as the number of users and the number of requests. For this purpose, they evaluate, by means of a route detector, whether the same access pattern is being constantly. Ranjan *et al*. [52] measure the number of requests direct at the system in each session, taking as attributes the time between the request or session and the workload profile. Measuring these attributes makes it possible to determine whether there is work overload in a session, which is indicative of

an attack. Table 1 shows the 30 application layer-level features that are used in the detection of DDoS attacks. It should be noted that in the last two years 25 new features have been incorporated in attack detection methods at the application layer level, which shows the interest of the scientific community in finding new features that might improve detection rates. The highest detection rate obtained to date is between 98.5 and 98.32 and has been achieved using the Intrusion Detection System (IDS) and Neural Networks in conjunction with a Genetic Algorithm, using the allowing characteristics: duration of the conversation; maximal, minimal and average packet size; maximal, minimal and average size of the TCP window; maximal, minimal and average time to live (TTL); the number of bytes sent in 1 second; the number of packets sent in 1 second; the percentage of encrypted packets with different properties; the percentage of packets with different TCP flags [50] and the entropy of the requests, the HTTP GET request count, the variance of the entropy [56], respectively.

3. FEATURES FOR DDOS ATTACK DETECTION

The dynamism of the user involves a process by which the behavioral tendencies that exist between users are measured [57]. Authentication of a user from his behavior is a task that has been studied from the point of view of informatics security [58]. In order to avoid access by unauthorized users, several investigators [57-62] have focused their efforts on a process called biometric behavior. This process involves: the use of keystrokes, mouse dynamics and interaction with the graphical user interface (GUI) [60] for the identification of users. In the present work, two mouse interaction characteristics are proposed for the detection of DDoS attacks. These features are: Mouse movement and Right click.

3.1 Mouse movement

This feature is evidenced when a real user deploys the mouse through the application interface [57]. This dynamism between the user and the system indicates that a human user is making the server real requests. In DDoS attacks the user makes the server a request directly without scrolling the mouse. A DDoS attack does not generate any kind of dynamics with the system [58]. This inactivity allows a real user to be clearly differentiated from a DDoS attack.

3.2 Right Click

A user can perform operations on the client machine by executing a click event [59]. This action constitutes a unique event that allows to a real user to be unequivocally identified [60], because a DDoS attack performs requests without employing this type of special event [61].

Table 1 describes the characteristics of the mouse that can be captured and the techniques that are used for such purposes. These features can be captured using software developed in programming languages that incorporate libraries or special functions for it. It should be mentioned that some of these techniques have also been used to captures features of other peripherals, such as the keyboard [62-65].

Table 1. Methods for capturing mouse features

Id	Mouse features	Library / Programming language	Ref.
M1	Single-click	Windows application (written in C#)	[62]
M2	Double-click		
M3	Movement offset		
M4	Speed curve against time		
M5	Acceleration curve against time		
M6	Time	Java (kSquared.de library)	[63]
M7	Movement		
M8	Left or right button pressed or released		
M9	Coordinates of an event		
M10	Mouse position coordinates	NA	[64]
M11	Mouse trajectory		
M12	Angle of the path in various directions		
M13	Curvature and its derivative		
M14	Mouse movement		
M15	Angular velocities		
M16	Tangential acceleration and jerk		
M17	Mouse movement coordinate	Java applet and javascript	[65]
M18	Movement angle		
M19	Time to move		
M20	Time of mouse clicks		

3.3 Detection Criteria

Figure 1 shows the classification algorithm that makes it possible to identify DDoS attacks by means of the characteristics of the mouse. The proposed algorithm verifies whether the characteristics studied in this work (right click and mouse movement) are active or not. When a request is made to the system and both features are inactive, the algorithm considers it as a DDoS attack. When this happens the algorithm stores the request in a variable called attack. On the other hand, when there is a request and at least one of the two features is active, it becomes apparent that it is a real user. When this happens the software saves it in a variable called user. Finally, the algorithm calculates the accuracy rate. The software is loaded with a dataset of 11055 requests between real users and DDoS attacks. By analyzing the dataset using the MySql database manager, it is possible to know from the beginning how many real users and DDoS attacks it contains. The algorithm calculates the accuracy rate of DDoS attacks by dividing the number of attacks encountered by the algorithm between the numbers of actual attacks in the dataset.

```
Input: D: Dataset
Output: Accuracy detection
begin
    while D.next
        if D.click=-1 and D.mouse_movement=-1 then
            attack++;
        else
            user ++;
        end if
    end while
    Query D.n_attack;
    accuracy=(attack*100)/n_attack;
end
```

Figure 1. Classification algorithm for the differentiation of DDoS attacks and users.

4. EXPERIMENTS

4.1 Dataset

The dataset selected for the validation process of the classification algorithm contains 11055 requests, of which 9096 are real users and the rest are DDoS attacks. This dataset was selected because it contains the two characteristics of the mouse to be evaluated. In addition, this dataset contains 31 attributes from which four were extracted to perform the validation (right click, mouse movement, abnormal URL and request URL). It should be noted that before using the request URL feature, it must be known whether a request was made to the system or not. On the other hand, abnormal URL identifies requests that are attacks [66].

4.2 Results

The algorithm used to implement the classification criteria was created in Java version 1.8.0 using NetBeans IDE 8.2. The tests were developed on a machine whose processor is Intel (R) Core i7 CPU 2.60 GHz, 8 GB RAM, with a Windows 10 operating system. Table 2 shows the classification efficiency obtained using the two mouse features, in this case 100%, both for the number of real users and for the number of DDoS attacks. This result shows that by using software designed for the detection of attacks and the two characteristics of user dynamism, the highest rate of accuracy is achieved. It should also be mentioned that the time taken by the application algorithm to perform the classification was 50 milliseconds. These percentages underline the importance of these characteristics for the detection of this type of computer attack.

Table 2. DDoS Attack Detection Efficiency

Type of user	Dataset	Algorithm	Efficiency (%)
Real users	9096	9096	100
DDoS attacks	1959	1959	100

4.3 Discussion

The results obtained in the tests performed show an efficiency level of 100%. The characteristics evaluated (mouse movement and right click) evidence the dynamism of the user. Therefore, it has been verified that the characteristics of the dynamism of the user makes it possible to differentiate a real request from a computer attack. It should be mentioned that there are other characteristics of user dynamism that can be considered for the identification of real users and robots (e.g. keyboard). However, with the advance in the mechanisms of detection of attacks, the attackers are constantly finding new alternatives to elude the mechanisms that are being proposed.

5. CONCLUSIONS

The state-of-the-art review of the variables used in the detection of DDoS attacks at the application layer level reveals that 30 variables have been used in published mechanisms in the last 10 years. The dataset used in this work contains two of the 30 variables included in the literature for the detection of DDoS attacks in the application layer. The evaluation of the two variables (mouse movement and right click), using a software designed in Java, managed to achieve 100% efficiency in differentiating between a real user and a robot. To do this, we used the right click and mouse movement variables, which are identified as characteristics of the user's dynamism. Therefore, these variables should be considered for implementation in the detection mechanisms of DDoS attacks.

6. REFERENCES

[1] Kumar, P. A. R., & Selvakumar, S. 2011. Distributed denial of service attack detection using an ensemble of neural classifier. *Computer Communications. 34,* 11, 1328-1341. DOI= https://doi.org/10.1016/j.comcom.2011.01.012

[2] Beak, C., Chaudhry, J. A., Lee, K., Park, S., & Kim, M. 2007. A novel packet marketing method in DDoS attack detection. *American Journal of Applied Sciences*. *4*, 10, 741-745. DOI=10.3844/ajassp.2007.741.745

[3] Chen, Y., Das, S., Dhar, P., El-Saddik, A., & Nayak, A. 2008. Detecting and Preventing IP-spoofed Distributed DoS Attacks. *IJ Network Security*. *7*, 1, 69-80.

[4] Yan, R., & Zheng, Q. 2009. Using Renyi cross entropy to analyze traffic matrix and detect DDoS attacks. *Information Technology Journal*. *8*, 8, 1180-1188. DOI=10.3923/itj.2009.1180.1188

[5] Anurekha, R., Duraiswamy, K., Viswanathan, A., Arunachalam, V. P., Kumar, K. G., and Rajivkannan, A. 2012. Dynamic approach to defend against distributed denial of service attacks using an adaptive spin lock rate control mechanism. *Journal of Computer Science.* 8, 5, 632.

[6] Chen, S. W., Wu, J. X., Ye, X. L., & Guo, T. 2013. Distributed Denial of Service Attacks Detection Method Based on Conditional Random Fields. *JNW. 8,* 4, 858-865. DOI=10.4304/jnw.8.4.858-865

[7] Sachdeva, M., & Kumar, K. 2014. A traffic cluster entropy based approach to distinguish DDoS Attacks from flash event using DETER testbed. *ISRN Communications and Networking*, *2014*. DOI= http://dx.doi.org/10.1155/2014/259831

[8] Yau, D. K., Lui, J., Liang, F., & Yam, Y. 2005. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Transactions on Networking (TON). 13,* 1, 29-42. DOI=10.1109/IWQoS.2002.1006572

[9] Yaar, A., Perrig, A., & Song, D. 2005. FIT: Fast internet traceback. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* (Vol. 2, pp. 1395-1406). IEEE. DOI=10.1109/INFCOM.2005.1498364

[10] Xiang, Y., Li, K., & Zhou, W. 2011. Low-rate DDoS attacks detection and traceback by using new information metrics. *IEEE Transactions on Information Forensics and Security. 6,* 2, 426-437. DOI=10.1109/TIFS.2011.2107320

[11] Yu, Z., and Tsai, J. J. 2011. Intrusion detection: a machine learning approach (Vol. 3). World Scientific.

[12] Ma, X., & Chen, Y. 2014. DDoS detection method based on chaos analysis of network traffic entropy. *IEEE Communications Letters. 18,* 1, 114-117. DOI=10.1109/LCOMM.2013.112613.132275

[13] Chen, Y., & Hwang, K. 2006. Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. *Journal of Parallel and Distributed Computing. 66,* 9, 1137-1151. DOI= https://doi.org/10.1016/j.jpdc.2006.04.007

[14] Spyridopoulos, T., Karanikas, G., Tryfonas, T., and Oikonomou, G. 2013. A game theoretic defence framework against DoS/DDoS cyber-attacks. Computers & Security. 38, 39-50. DOI= https://doi.org/10.1016/j.cose.2013.03.014

[15] Seo, D., Lee, H., & Perrig, A. 2013. APFS: adaptive probabilistic filter scheduling against distributed denial-of-service attacks. *Computers & Security. 39*, 366-385. DOI= https://doi.org/10.1016/j.cose.2013.09.002

[16] Kumar, P. A. R., & Selvakumar, S. 2013. Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. *Computer Communications. 36,* 3, 303-319. DOI= https://doi.org/10.1016/j.comcom.2012.09.010

[17] Kang, H. S., & Kim, S. R. 2014. sShield: small DDoS defense system using RIP-based traffic deflection in autonomous system. *The Journal of Supercomputing*. *67*, 3, 820-836. DOI https://doi.org/10.1007/s11227-013-1031-7

[18] Xiao, P., Qu, W., Qi, H., & Li, Z. 2015. Detecting DDoS attacks against data center with correlation analysis. *Computer Communications.* 67, 66-74. DOI= https://doi.org/10.1016/j.comcom.2015.06.012

[19] Meenakshi, S. and Srivatsa, S. K. 2007. A distributed framework with less false positive ratio against distributed denial of service attack. *Information Technology Journal.* 6, 8, 1139-1145. DOI=10.3923/itj.2007.1139.1145

[20] Liu, H., Sun, Y., and Kim, M. S. 2011. A scalable DDoS detection framework with victim pinpoint capability. *Journal of Communications.* 6, 9, 660-670. DOI= 10.4304/jcm.6.9.660-670

[21] Udhayan, J. And Babu, M. R. 2013. Deteriorating distributed denial of service attack by recovering zombies using penalty scheme. *Journal of Computer Science.* 9, 11, 1618. DOI=10.3844/jcssp.2013.1618.1625

[22] Al-Duwairi, B., Al-Qudahy, Z., and Govindarasu, M. 2013. A novel scheme for mitigating botnet-based DDoS attacks. *Journal of Networks.* 8, 2, 297-306. DOI=10.4304/jnw.8.2.297-306

[23] Wang, Y. and Sun, R. 2014. An IP-Traceback-based Packet Filtering Scheme for Eliminating DDoS Attacks. *Journal of Networks.* 9, 4, 874-881. DOI=10.4304/jnw.9.4.874-881

[24] Chen, S., & Song, Q. 2005. Perimeter-based defense against high bandwidth DDoS attacks. *IEEE Transactions on Parallel and Distributed Systems. 16,* 6, 526-537. DOI=10.1109/TPDS.2005.74

[25] Kim, Y., Lau, W. C., Chuah, M. C., & Chao, H. J. 2006. PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE transactions on dependable and secure computing. 3,* 2, 141-155. DOI=10.1109/TDSC.2006.25

[26] Chen, Y., Hwang, K., & Ku, W. S. 2007. Collaborative detection of DDoS attacks over multiple network domains. *IEEE Transactions on Parallel and Distributed Systems*. 18, 12, 1649-1662. DOI=10.1109/TPDS.2007.1111

[27] Chen, R., Park, J. M., & Marchany, R. 2007. A divide-and-conquer strategy for thwarting distributed denial-of-service attacks. *IEEE Transactions on Parallel and Distributed Systems*. 18, 5, 577-588. DOI=10.1109/TPDS.2007.1014

[28] Wang, H., Jin, C., and Shin, K. G. 2007. Defense against spoofed IP traffic using hop-count filtering. *IEEE/ACM Transactions on Networking (ToN).* 15, 1, 40-53. DOI= 10.1109/TNET.2006.890133

[29] Chonka, A., Singh, J., and Zhou, W. 2009. Chaos theory based detection against network mimicking DDoS attacks. *IEEE Communications Letters*. 13, 9, 717-719. DOI=10.1109/LCOMM.2009.090615

[30] François, J., Aib, I., and Boutaba, R. 2012. FireCol: a collaborative protection network for the detection of flooding DDoS attacks. *IEEE/ACM Transactions on Networking (TON).* 20, 6, 1828-1841. DOI= 10.1109/TNET.2012.2194508

[31] Chen, Y., Ma, X., and Wu, X. 2013. DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory. *IEEE Communications Letters.* 17, 5, 1052-1054. DOI=10.1109/LCOMM.2013.031913.130066

[32] Wu, X. and Chen, Y. 2013. Validation of Chaos Hypothesis in NADA and Improved DDoS Detection Algorithm. *IEEE Communications Letters*. 17, 12, 2396-2399. DOI=10.1109/LCOMM.2013.102913.130932

[33] Luo, H., Lin, Y., Zhang, H., and Zukerman, M. 2013. Preventing DDoS attacks by identifier/locator separation. *IEEE Network*. 27, 6, 60-65. DOI=10.1109/MNET.2013.6678928

[34] Luo, J., Yang, X., Wang, J., Xu, J., Sun, J., and Long, K. 2014. On a Mathematical Model for Low-Rate Shrew DDoS. *IEEE Transactions on Information Forensics and Security.* 9, 7, 1069-1083. DOI=10.1109/TIFS.2014.2321034

[35] Mirkovic, J. and Reiher, P. 2005. D-WARD: a source-end defense against flooding denial-of-service attacks. *IEEE Transactions on Dependable and Secure Computing.* 2, 3, 216-232. DOI=10.1109/TDSC.2005.35

[36] Lee, F. Y. and Shieh, S. 2005. Defending against spoofed DDoS attacks with path fingerprint. *Computers and Security.* 24, 7, 571-586. DOI= https://doi.org/10.1016/j.cose.2005.03.005

[37] Al-Duwairi, B. And Manimaran, G. 2006. Distributed packet pairing for reflector based DDoS attack mitigation. *Computer communications.* 29, 12, 2269-2280. DOI= https://doi.org/10.1016/j.comcom.2006.03.007

[38] Lee, K., Kim, J., Kwon, K. H., Han, Y., and Kim, S. 2008. DDoS attack detection method using cluster analysis. *Expert Systems with Applications.* 34, 3, 1659-1665. DOI= https://doi.org/10.1016/j.eswa.2007.01.040

[39] Lu, W. Z., Gu, W. X., and Yu, S. Z. 2009. One-way queuing delay measurement and its application on detecting DDoS attack. *Journal of Network and Computer Applications.* 32, 2, 367-376. DOI= https://doi.org/10.1016/j.jnca.2008.02.018

[40] Doron, E., Wool, A. 2011. WDA: Web farm distributed denial of service attack attenuator. *Computer Networks.* 55, 5, 1037-1051. DOI=https://doi.org/10.1016/j.comnet.2010.05.001

[41] Zhang, C., Cai, Z., Chen, W., Luo, X., and Yin, J. 2012. Flow level detection and filtering of low-rate DDoS. *Computer Networks.* 56, 15, 3417-3431. DOI= https://doi.org/10.1016/j.comnet.2012.07.003

[42] Wang, F., Wang, H., Wang, X., and Su, J. 2012. A new multistage approach to detect subtle DDoS attacks. *Mathematical and Computer Modelling*. 55, 1, 198-213. DOI= https://doi.org/10.1016/j.mcm.2011.02.025

[43] Rahmani, H., Sahli, N., and Kamoun, F. 2012. DDoS flooding attack detection scheme based on F-divergence. *Computer Communications*. 35, 11, 1380-1391. DOI= https://doi.org/10.1016/j.comcom.2012.04.002

[44] Lee, S. M., Kim, D. S., Lee, J. H., and Park, J. S. 2012. Detection of DDoS attacks using optimized traffic matrix. *Computers and Mathematics with Applications*. 63, 2, 501-510. DOI= https://doi.org/10.1016/j.camwa.2011.08.020

[45] Varalakshmi, P. and Selvi, S. T. 2013. Thwarting DDoS attacks in grid using information divergence. *Future Generation Computer Systems.* 29, 1, 429-441. DOI= https://doi.org/10.1016/j.future.2011.10.012

[46] Li, L. and Lee, G. 2005. DDoS attack detection and wavelets. *Telecommunication Systems.* 28, 3-4, 435-451. DOI=https://doi.org/10.1007/s11235-004-5581-0

[47] Kulkarni, A. and Bush, S. 2006. Detecting distributed denial-of-service attacks using kolmogorov complexity metrics. *Journal of Network and Systems Management*. 14, 1, 69-80. DOI=https://doi.org/10.1007/s10922-005-9016-3

[48] Xiao, B., Chen, W., and He, Y. 2006. A novel approach to detecting DDoS Attacks at an Early Stage. *The Journal of Supercomputing.* 36, 3, 235-248. DOI=https://doi.org/10.1007/s11227-006-8295-0

[49] Kang, S. H., Park, K. Y., Yoo, S. G., and Kim, J. 2013. DDoS avoidance strategy for service availability. *Cluster computing*, 16, 2, 241-248. DOI=https://doi.org/10.1007/s10586-011-0185-4

[50] Zolotukhin, M., Kokkonen, T., Hämäläinen, T., and Siltanen, J. 2016. On Application-Layer DDoS Attack Detection in High-Speed Encrypted Networks. *International Journal of Digital Content Technology and its Applications.* 10.

[51] Xie, Y., & Yu, S. Z. 2009. Monitoring the application-layer DDoS attacks for popular websites. *IEEE/ACM Transactions on Networking (TON).* 17, 1, 15-25. DOI= 10.1109/TNET.2008.925628

[52] Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., & Knightly, E. 2009. DDoS-shield: DDoS-resilient scheduling to counter application layer attacks. *IEEE/ACM Transactions on Networking (TON).* 17, 1, 26-39. DOI= 10.1109/TNET.2008.926503

[53] Huang, C., Wang, J., Wu, G., & Chen, J. 2014. Mining Web User Behaviors to Detect Application Layer DDoS Attacks. *JSW*. 9, 4, 985-990. DOI=10.4304/jsw.9.4.985-990

[54] Dick, U. and Scheffer, T. 2016. Learning to control a structured-prediction decoder for detection of HTTP-layer DDoS attackers. *Machine Learning*. 104, 2-3, 385-410. DOI=https://doi.org/10.1007/s10994-016-5581-9

[55] Saravanan, R., Shanmuganathan, S., and Palanichamy, Y. 2016. Behavior-based detection of application layer distributed denial of service attacks during flash events. *Turkish Journal of Electrical Engineering & Computer Sciences*. 24, 2, 510-523. DOI=10.3906/elk-1308-188

[56] Johnson Singh, K., Thongam, K., and De, T. 2016. Entropy-Based Application Layer DDoS Attack Detection Using Artificial Neural Networks. *Entropy.* 18, 10, 350. DOI=10.3390/e18100350

[57] Ghezzi, C., Pezzè, M., Sama, M., and Tamburrelli, G. 2014. Mining behavior models from user-intensive web applications. *In Proceedings of the 36th International Conference on Software Engineering*. ACM; 2014: p. 277-287.

[58] Stevanovic, D. and Vlajic, N. 2014. Application-layer DDoS in dynamic Web-domains: Building defenses against next-generation attack behavior. *IEEE Conference Communications and Network Security (CNS).* 2014; 2014: p. 490-491. DOI=10.1109/CNS.2014.6997519

[59] Urban, R. J. 2015. Detection of exit behavior of an Internet user. U.S. Patent Application No 14/829,409.

[60] Abramson, M. and Aha, D. W. 2013. User authentication from web browsing behavior. Naval Research Lab Washington DC.

[61] Kim, Y., Kim, I. 2014. Involvers' Behavior-based Modeling in Cyber Targeted Attack. Proceedings of Securware.

[62] Shen, C., Cai, Z., Guan, X., Du, Y., and Maxion, R. A. 2013. User authentication through mouse dynamics. *IEEE Transactions on Information Forensics and Security.* 8, 1, 16-30. DOI=10.1109/TIFS.2012.2223677

[63] Salmeron-Majadas, S., Santos, O. C., and Boticario, J. G. 2014. An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective modeling in an educational context. *Procedia Computer Science*. 35, 691-700. DOI= https://doi.org/10.1016/j.procs.2014.08.151

[64] Graepel, T., Candela, J. Q., Borchert, T., and Herbrich, R. 2010. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 13-20).

[65] Gamboa, H. and Fred, A. 2003. An identity authentication system based on human computer interaction behaviour. In *Proc. of the 3rd Intl. Workshop on Pattern Recognition in Information Systems*, Angers, France, pp. 46–55

[66] Lichman, M. 2013. UCI Machine Learning Repository [http://archive.icc.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

ANEXO 5

2018 International Conference on Information and Computer Technologies

DDoS Attack Detection Mechanism in the Application Layer Using User Features

Silvia Bravo
Faculty of Engineering and Applied Sciences
Technical University of Cotopaxi
Latacunga, Ecuador
e-mail: silvia.bravom@utc.edu.ec

David Mauricio
Faculty in Systems Engineering and Computer Science
National University of San Marcos
Lima, Peru
e-mail: dmauricios@unmsm.edu.pe

***Abstract*—DDoS attacks are one of the most damaging computer aggressions of recent times. Attackers send large number of requests to saturate a victim machine and it stops providing its services to legitimate users. In general attacks are directed to the network layer and the application layer, the latter has been increasing due mainly to its easy execution and difficult detection. The present work proposes a low cost detection approach that uses the characteristics of the Web User for the detection of attacks. To do this, the features are extracted in real time using functions designed in PHP and JavaScript. They are evaluated by an order 1 classifier to differentiate a real user from a DDoS attack. A real user is identified by making requests interacting with the computer system, while DDoS attacks are requests sent by robots to overload the system with indiscriminate requests. The tests were executed on a computer system using requests from real users and attacks using the LOIC, OWASP and GoldenEye tools. The results show that the proposed method has a detection efficiency of 100%, and that the characteristics of the web user allow to differentiate between a real user and a robot.**

Keywords-DDoS; denial of service; dynamism user; features user

I. INTRODUCTION

DDoS attacks have become one of the threats to the security of computer systems. These attacks are aimed at consuming bandwidth or server resources, preventing legitimate users from accessing them. In recent years there has been an increase in the number of these attacks, especially to the application layer, due to its easy execution and difficult detection. Therefore, the efforts in the detection mechanisms of DDoS attacks have focused on the mitigation of this type of attack.

This type of attack is considered sophisticated because it mimics the requests of real users, so it is more difficult to detect. Therefore, the detection mechanisms at the application layer level analyze the characteristics of the user [1]-[10]. In the mechanism implemented by Zolotukhin *et al*. [1] Intrusion Detection System techniques are used using the characteristics, duration of the conversation number of packets sent in 1 second, number of bytes sent in 1 second maximum, minimum and average packet size, maximum, minimum and average size of TCP window, maximal, minimal and average time to live (TTL), percentage of packets with different TCP flags: URG, ACK, PSH, RST, SYN and FIN, percentage of encrypted packets with different properties: handshake, alert, etc.

This mechanism reaches a detection rate of 98.5%. While the mechanism proposed by Johnson *et al.* [9] uses the techniques Neural Netwoks and Genetic Algorithm in conjunction with the entropy characteristics of the requests, HTTP GET request count and variance of the entropy. This method reaches an attack detection percentage of 98.32%. It should be noted that these mechanisms determine statistics of user requests that circulate through the network and do not analyze the user's interaction while the user is browsing the system. This work proposes characteristics of the dynamism of the user for the detection process of DDoS attacks. Therefore, keystrokes, mouse dynamics and interaction with the graphical user interface (GUI) [11] for the identification of real users are evaluated. For this purpose, an algorithm that detects the interaction of the user and the system has been implemented. It was tested on a web system in real time. This web system was developed under a three-layer architecture to implement the user interface and the detection algorithm. To evaluate the proposed method the LOIC, OWASP and GoldenEye tools were used to generate flood attacks. The results show that the features and the algorithm used allow to detect DDoS attacks in a 100%.

II. RELATED WORKS

The detection of DDoS attacks is an issue that has been growing due to its great impact on computer systems, in addition to its easy implementation [1]. In the detection mechanisms of this type of attack, a level of application capacity, characteristics of user behavior are employed. However, these methods do not take into account the dynamism of the web user with the computer system. Xie *et al.* [2] proposes in its mechanism a search analysis of a user's websites to determine their characteristics, such as the number of users and the number of requests. To do this, it uses a route detector that measures the access pattern. Ranjan *et al.* [3] use in its detection method the number of requests in each session, taking time in the session or workload profile, in order to overload work in a session. Huang *et al.* [8] uses a mechanism that calculates the rate of links in a search, if there are abrupt search changes, it is clear that an attack is occurring.

The detection mechanisms in the application layer have obtained to date a detection rate between 98.5% [1] and 98.32% [9] and has been achieved using the intrusion

978-1-5386-5384-5/18/$31.00 ©2018 IEEE

detection system (IDS) and neural networks together with the algorithm genetic, using the characteristics, duration of the conversation, maximum, minimum and average packet size, maximum minimum and average size of the TCP window, maximum and maximum minimum lifetime (TTL), number of bytes sent in 1 second, number of packets sent in 1 second, percentage of encrypted packets with different properties, percentage of packets with different TCP flags [1] and the entropy of the requests, the number of HTTP GET requests, the variance of entropy [10], respectively.

III. Method of Detection of DDoS Attacks in the Application Layer

A. Architecture of the Detection Method

Figure 1 shows the architecture used for the implementation of the DDoS attack detection method. It shows the entry of requests from the Internet to the interface of the web application. The requests made generate a data bank where the established connections and the processes performed are registered. The data bank generated in the application layer is analyzed by an interaction detector. At the application level, the processes that the user generates are recorded (links, resources, forms, use of peripherals, etc). The detector records the procedures performed by the mouse and keyboard peripherals. The features are extracted in real time by Javascript programming, these characteristics are stored until the user executes the next request. The requests and the characteristics of the user are sent to the classification algorithm for evaluation. This algorithm is responsible for determining the existence of requests and interactions with the system, taking a decision between real user or computer attack.

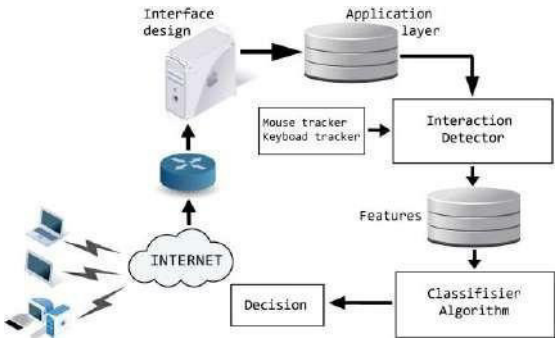

Figure 1. Architecture of the detection method.

B. Characteristics of the User's Dynamism

The characteristics used by the detection mechanism proposed in this work are taken from the interaction between the user and the system through the peripherals and the graphical user interface when the user executes a request. In this work, the characteristics of the user's dynamism must be recorded in real time before processing the next request. Table I shows the characteristics of the user that are employed by the proposed detection mechanism. It is worth mentioning that these features are extracted using PHP and JavaScript functions.

TABLE I. Web User Features

Features	*Description*
Mouse Move	The mouse is moved to a location on the screen to perform an action.
Mouse Click	The Mouse Click feature occurs when a user presses and releases a mouse button and there are five types of click events that are recorded: left click, right click, and double left click.
Mouse highlight	This action begins with a left mouse click/hold to begin the highlighting and ends with the mouse release.
Mouse drag and drop	When an object is dragged and dropped. This action begins with a left mouse click/hold and ends with the mouse release.
Mouse Wheel and Scroll	The Mouse Wheel or Scroll is an event when the movement of the wheel or scroll has a net up or down effect. The resultant effect is based on the consecutive wheel or scroll movements.
Key press and release event	This happens when a user presses a key and slides the touch device (finger or stylus)

C. Classification Algorithm

The use of web user characteristics and the classification algorithm allow the identification of a real user and a computer attack in real time. Figure 2 shows the classification algorithm proposed in this work, it uses the classification criteria based on the characteristics of the web user. The proposed algorithm verifies if each of the characteristics shown in Table 1 are active or not. When a request is made to the system and the extracted characteristics are inactive, it is considered as an attack. When this happens the algorithm generates a captcha verification process, which will be deactivated when the user resolves the test generated by the captcha. Once the captcha test has been resolved, the request made is processed. Otherwise, the request will not be executed until the captcha is resolved. On the other hand, when there is a request and at least one of the extracted characteristics is active, it will be understood that there is interaction between the user and the system, therefore, it is a real user. When this happens, the algorithm allows the requests made by the user to be processed.

```
Input user_request
begin capture
When document mousemove then id stores true
When document onClick then id stores true
When document ondblClick then id stores true
When document onmousedown then id stores true
When document onmouseup then id stores true
When document onscroll then id stores true
When document onwheel then id stores true
When document onkeypress then id stores true
end capture

begin verify each request
When id stores true then execute query request
When id stores false then execute captcha verification
end verify
```

Figure 2. Algorithm for detection of DDoS attacks.

IV. NUMERICAL EXPERIMENTS

A. Validation Environment

The validation environment is a web application and a dedicated server, Linux Linux CentOS, 4-core processor, 4 GB memory, 1TB disk space. The computer application works in real time and is loaded with information about your business. Figure 3 shows the design of the validation environment. It is composed of three levels. The first level involves the user interface, here the functions of the system are used as links, videos, graphics, etc. In the second level, located are the functions that load the information to all the applications of the system. It should be noted that, at this level, there are the functions that perform the call to the mouse user's characteristics detector (mouse, keyboard and graphical user interface). Hence, you can register the user characteristics immediately detected in an activity in the graphical interface.

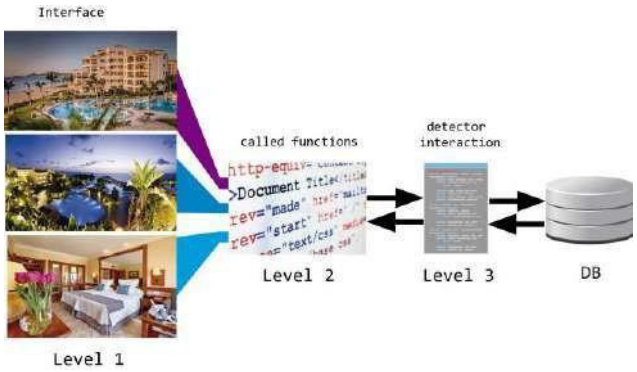

Figure 3. Validation environment.

Finally, in the third level is the validation algorithm proposed in section III, C. The characteristics of the user behavior captured in the second level are analyzed and classified in this level. When a request is sent, it is verified that at least one of the characteristics of the user is active and is considered a real user, otherwise, if you have a computer attack. In this way, the user is authenticated every time he makes requests to the system.

B. Extraction of Characteristics of the User's Dynamism

PHP and JavaScript functions are used to extract user characteristics. These functions are loaded into the system when the user interface is built. They register the activities that the user executes in the system, for example: clicks on resources, keystrokes, and general manipulation of the interface. The characteristics are stored in variables in a second plane within the architecture. For this, each one of the eight characteristics proposed in this work has a specific function that allows its capture. It is worth mentioning that the records of the features are made in real time and stored for user authentication in their next application.

C. Attack Detection

Once the user's characteristics have been extracted, they are validated as a real user or attack. For this, the real users are defined as requests made to the system that maintained interaction with it, this is verified by registering the characteristics of the User. To this end, these characteristics of user behavior are analyzed in the following application. When a request is sent, and at least one of the characteristics is true, it is considered a real user, and if not, if none of the characteristics is active, it is considered a DDoS attack and a captcha validation system is activated. The characteristic extractor is responsible for assigning the value of active or non-active to the variables of the characteristics. These values are sent to the classification function. It verifies the activity registered by the user, allowing the application to continue to be sent additional requests, in case of identifying a real user and otherwise a captcha test is issued that must be resolved for user validation.

V. RESULTS AND DISCUSSION

In this work, the accuracy of the proposed detection method was evaluated. To generate DDoS attacks, LOIC (Low Orbit Ion Canon), OWASP DOS HTTP POST and GoldenEye HTTP software were used. For this, several attacks were made with each tool towards the victim server, in order to evaluate the attack rate necessary to overload the server. In each attack, the overload values were obtained, which would then be evaluated using the proposed detection algorithm. Table II shows the tools that were used to simulate the attack on the web system, the amount of solitudes generated and the time it took the system to overload.

TABLE II. RESULTS OF EVALUATION WITHOUT DETECTION METHOD

Tool	*Number of requests*	*System drop time (min)*
LOIC (Low Orbit Ion Cannon)	4800	2.15
OWASP DOS HTTP POST	4000	1.30
GoldenEye HTTP Denial Of Service Tool	5300	3.20

The results of Table II show that the computer attacks generated by the software LOIC, OWASP and GoldenEye use about two minutes to overload the system, causing inaccessibility to resources and services for real users. It is also observed that the number of requests used to overcharge the system varies between 4000 and 5300. It should be mentioned that these tools were selected because they are the most used for the generation of this type of attacks, due to their simplicity and effectiveness [12].

TABLE III. RESULTS OF EVALUATION WITH DETECTION METHOD

Tool	*Number of request*	*Detection time (mls)*	*Detection rate (%)*
LOIC (Low Orbit Ion Canon)	4800	60	100
OWASP DOS HTTP POST	4000	58	100
GoldenEye HTTP Denial Of Service Tool	5300	63	100

Table III shows the results obtained using the proposed detection method. It shows 100% of attacks generated by the tools have been detected effectively. The time used in the detection was on average 60 milliseconds and between 4000 and 5300 requests were used to generate the attack. It is worth mentioning that there are no dataset related to DDoS attacks for tests. In addition, the works with the highest detection rate in the application layer [1] and [8] do not show the tools that were used to evaluate the proposed methods. Table 3 shows that the detection mechanism developed through the use of web user characteristics is effective with a detection rate of 100% for the three attack generation tools. In addition, the time spent is around 60 milliseconds. These results show the effectiveness of the detection method by using the characteristics of the user's dynamism through the interaction with the system. It should be mentioned that with the improvement of the detection mechanisms, the attackers also improve their attack strategies, so the possibility that the input values of the user characteristics evaluated in this work can be supplanted is not ruled out.

VI. CONCLUSION

The method of detecting DDoS attacks using the characteristics of user behavior has a detection rate of 100%. This result demonstrates the importance of registering the characteristics of a user when he interacts with the system. The registration of the dynamism of the user in real time is used by the classification algorithm to execute the next request, identifying a real user of an attack. The tests in a computer system in real time, using real users and the LOIC OWASP and GoldenEye attack tool, allow for evaluation, the characteristics and the algorithm under a simulated attack environment. The executed tests were able to verify the efficiency of the algorithm reaching the most optimal result in large numbers of requests.

ACKNOWLEDGMENT

The authors thank the National Council of Science, Technology and Technological Innovation (CONCYTEC) - Peru and Technical University of Cotopaxi for the partial funding of this work and Professor Angel H. Moreno for their contributions to this work.

REFERENCES

[1] M. Zolotukhin, T. Kokkonen, T. Hämäläinen, J. Siltanen, On Application-Layer DDoS Attack Detection in High-Speed Encrypted Networks. International Journal of Digital Content Technology and its Applications, 2016, 10.

[2] Y. Xie, S. Yu, Monitoring the application-layer DDoS attacks for popular websites. IEEE/ACM Transactions on Networking (TON): vol. 17, no 1, 2009, p. 15-25.

[3] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, E. Knightly, DDoS-shield: DDoS-resilient scheduling to counter application layer attacks. IEEE/ACM Transactions on Networking (TON): vol. 17, no 1, 2009, p. 26-39.

[4] L. Giralte, C. Conde, I. De Diego, E. Cabello, Detecting denial of service by modelling web-server behaviour. Computers \& Electrical Engineering: vol. 39, no 7, 2013, p. 2252-2262.

[5] 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, 38, 2014, 36-46.

[6] C. Huang, J. Wang, G. Wu, J. Chen, Mining Web User Behaviors to Detect Application Layer DDoS Attacks. *JSW*, *9*(4), 2014, 985-990.

[7] U. Dick, T. Scheffer, Learning to control a structured-prediction decoder for detection of HTTP-layer DDoS attackers. Machine Learning, 104(2-3), 2016, 385-410.

[8] R. Saravanan, S. Shanmuganathan, Y. Palanichamy, Behavior-based detection of application layer distributed denial of service attacks during flash events. Turkish Journal of Electrical Engineering \& Computer Sciences, 24(2), 2016, 510-523.

[9] K. Johnson Singh, K. Thongam, T. De, Entropy-Based Application Layer DDoS Attack Detection Using Artificial Neural Networks. Entropy, 18(10), 2016, 350.

[10] D. Stevanovic, N. Vlajic, Application-layer DDoS in dynamic Web-domains: Building defenses against next-generation attack behavior. En Communications and Network Security (CNS), 2014 IEEE Conference on. IEEE; 2014: p. 490-491.

[11] M. Abramson, D. Aha, User authentication from web browsing behavior. Naval Research Lab Washington DC; 2013.

[12] I. Riadi, A. Muhammad, Neural network-based ddos detection regarding hidden layer variation. Journal of Theoretical & Applied Information Technology, 2017, vol. 95, no 15.

ANEXO 6

CÓDIGO DE PRIMERA CAPA

Extracto de la programación de la interfaz de página y activación de características del usuario para la validación. La documentación del código se encuentra en inglés, según requerimientos de la empresa.

```
function twentysixteen_post_thumbnail_sizes_attr( $attr, $attachment, $size ) {
	if ( 'post-thumbnail' === $size ) {
		is_active_sidebar( 'sidebar-1' ) && $attr['sizes'] = '(max-width: 709px) 85vw, (max-width: 909px) 67vw, (max-width: 984px) 60vw, (max-width: 1362px) 62vw, 840px';
		! is_active_sidebar( 'sidebar-1' ) && $attr['sizes'] = '(max-width: 709px) 85vw, (max-width: 909px) 67vw, (max-width: 1362px) 88vw, 1200px';
	}
	return $attr;
}
add_filter( 'wp_get_attachment_image_attributes', 'twentysixteen_post_thumbnail_sizes_attr', 10 , 3 );

/**
 * Modifies tag cloud widget arguments to have all tags in the widget same font size.
 *
*
 * @param array $args Arguments for tag cloud widget.
 * @return array A new modified arguments.
 */
function twentysixteen_widget_tag_cloud_args( $args ) {
	$args['largest'] = 1;
	$args['smallest'] = 1;
	$args['unit'] = 'em';
```

```
	return $args;
}
add_filter( 'widget_tag_cloud_args', 'twentysixteen_widget_tag_cloud_args' );

// Called for activation of features
add_action('wp_enqueue_scripts','dcms_insertar_js');
function dcms_insertar_js(){
	wp_register_script('miscript',get_template_directory_uri().'/js/script.js',array(jquery),'
1',true);
	wp_enqueue_script('miscript');
}
```

CÓDIGO DE SEGUNDA CAPA

Extracto del script en donde se encuentran las funciones necesarias para enviar las solicitudes y están puedan regresar información. Nótese que la función de activación de funciones se encuentra en este script, con el fin de dar a conocer las activaciones de características realizadas.

```
<?php
/**
 * The main template file
 *
 * This is the most generic template file in a WordPress theme
 * and one of the two required files for a theme (the other being style.css).
 * It is used to display a page when nothing more specific matches a query.
 * E.g., it puts together the home page when no home.php file exists.
 *
 * @link http://codex.wordpress.org/Template_Hierarchy
 *
```

```
 * @package WordPress
 * @subpackage Twenty_Sixteen
 * @since Twenty Sixteen 1.0
 */

get_header(); ?>

	<div id="primary" class="content-area">
		<main id="main" class="site-main" role="main">
		<?php if ( have_posts() ) : ?>

			<?php if ( is_home() && ! is_front_page() ) : ?>
				<header>
					<h1 class="page-title screen-reader-text"><?php single_post_title(); ?></h1>
				</header>
			<?php endif; ?>

			<?php
			// Start the loop.
			while ( have_posts() ) : the_post();

				/*
				 * Include the Post-Format-specific template for the content.
				 * If you want to override this in a child theme, then include a file
				 * called content-___.php (where____is the Post Format name) and that will be used instead.
				 */
// Called for activation of features
				get_template_part( 'template-parts/content', get_post_format(), dcms_insertar_js() );
```

```
                        // End the loop.
                        endwhile;

                        // Previous/next page navigation.
                        the_posts_pagination( array(
                                'prev_text'          => __( 'Previous page', 'twentysixteen' ),
                                'next_text'          => __( 'Next page', 'twentysixteen' ),
                                'before_page_number' => '<span class="meta-nav screen-reader-text">' . __( 'Page', 'twentysixteen' ) . ' </span>',
                        ) );
                // If no content, include the "No posts found" template.
                else :
                        get_template_part( 'template-parts/content', 'none' );
                endif;
                ?>
                </main><!-- .site-main -->
        </div><!-- .content-area -->
<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

CÓDIGO DE TERCERA CAPA

Extracto del script en donde se encuentran la verificación de las características, comprobando cuales de ellas se encuentran activadas.

```
var operation;
// Feature activation
document.getElementById('primary')
document.getElementById('primary'). dcms_insertar_js()=function(){
```

```
        operation=true;
        funcion();
}
// Verification of activation, in case of being false, pause in sending data
if(document.getElementById('primary').operation=false){
<?php
                comment_form( array(
                        'title_reply_before'  =>  '<h2  id="reply-title"  class="comment-reply-
pause">',
                        'title_reply_after'  => '</h2>',
                ) );
        ?>
}
```