UAEM | Universidad Autónoma del Estado de México

CENTRO UNIVERSITARIO XICO VALLE DE CHALCO

C.U. Valle de Chalco

DESARROLLO DE UN PROCESO DE SEGURIDAD PARA LA PREVENCIÓN DE INTRUSIONES EN UNA RED PRIVADA

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRA EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A

ANA LAURA ALCÁNTARA RAMÍREZ

TUTORA ACADÉMICA
DRA. MARÍA DE LOURDES LÓPEZ GARCÍA

TUTOR ADJUNTO
DR. JUVENAL RUEDA PAZ
TUTORA ADJUNTA
DRA. CRISTINA JUÁREZ LANDÍN

VALLE DE CHALCO SOLIDARIDAD, MÉXICO OCTUBRE 2017.

Valle de Chalco Solidaridad, Edo de Méx. a lunes, 16 de octubre de 2017

DR. EN C. JUVENAL RUEDA PAZ
COORDINADOR DE LA MAESTRÍA CIENCIAS DE LA COMPUTACIÓN
DEL CENTRO UNIVERSITARIO *UAEM* VALLE DE CHALCO.

P R E S E N T E.

Por este medio le comunico a usted que la comisión revisora designada para realizar la tesis denominada: **"Desarrollo de un proceso de seguridad para la prevención de intrusiones en una red privada"**, como parte de los requisitos para obtener el grado académico de Maestría en **Ciencias de la Computación** presenta **Ana Laura Alcántara Ramírez**, con número de cuenta **0423563** para sustentar el acto de evaluación de grado, ha dictaminado que dicho trabajo reúne las características de contenido para proceder a la impresión del mismo

A T E N T A M E N T E

Tutor adjunto	Tutora Académica	Tutor Adjunto
Dra. Cristina Juárez Landín	**Dra. María de Lourdes López García**	**Dr. Juvenal Rueda Paz**

Valle de Chalco Solidaridad, Estado de México lunes, 16 de octubre de 2017

ANA LAURA ALCÁNTARA RAMÍREZ
CANDIDATO A GRADO DE MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN
CENTRO UNIVERSITARIO UAEM VALLE DE CHALCO

Presente

De acuerdo con el Reglamento de Estudios Avanzados de la Universidad Autónoma del Estado de México y habiendo cumplido con todas las indicaciones que la Comisión Revisora realizó con respecto a su trabajo **Tesis** titulado **"DESARROLLO DE UN PROCESO DE SEGURIDAD PARA LA PREVENCIÓN DE INTRUSIONES EN UNA RED PRIVADA"** la Coordinación de la Maestría en **Ciencias de la Computación** del Centro Universitario UAEM Valle de Chalco concede la autorización para que proceda a la impresión de la misma.

Sin más por el momento, le reitero la seguridad de mi especial consideración y estima.

CENTRO UNIVERSITARIO UAEM
VALLE DE CHALCO
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

DR. EN C. JUVENAL RUEDA PAZ
COORDINADOR DE LA MAESTRÍA CIENCIAS DE LA COMPUTACIÓN
CENTRO UNIVERSITARIO UAEM
VALLE DE CHALCO

CARTA DE CESIÓN DE DERECHOS DE AUTOR

El que suscribe **Ana Laura Alcántara Ramírez** Autor del trabajo escrito de evaluación profesional en la opción de Trabajo terminal de Grado con el título **"Desarrollo de un proceso de seguridad para la prevención de intrusiones en una red privada"**, por medio de la presente con fundamento en lo dispuesto en los artículos 5, 18, 24, 25, 27, 30, 32 y 148 de la Ley Federal de Derechos de Autor, así como los artículos 35 y 36 fracción II de la Ley de la Universidad Autónoma del Estado de México; manifiesto mi autoría y originalidad de la obra mencionada que se presentó en el **Centro Universitario UAEM Valle de Chalco** para ser evaluada con el fin de obtener el Grado de **Maestría En Ciencias de la Computación.** Así mismo expreso mi conformidad de ceder los derechos de reproducción, difusión y circulación de esta obra, en forma NO EXCLUSIVA, a la Universidad Autónoma del Estado de México; se podrá realizar a nivel nacional e internacional, de manera parcial o total a través de cualquier medio de información que sea susceptible para ello, en una o varias ocasiones, así como en cualquier soporte documental, todo ello siempre y cuando sus fines sean académicos, humanísticos, tecnológicos, históricos, artísticos, sociales, científicos u otra manifestación de la cultura.

Entendiendo que dicha cesión no genera obligación alguna para la Universidad Autónoma del Estado de México y que podrá o no ejercer los derechos cedidos.

Por lo que el autor da su consentimiento para la publicación de su trabajo escrito de evaluación profesional.

Se firma la presente en la ciudad de Valle de Chalco, a los 16 días del mes de octubre de 20 17 .

Ana Laura Alcántara Ramírez
Nombre y firma de conformidad

Dedicatoria

A Castañita...

Agradecimientos

Agradezco al Consejo Nacional de Ciencias y Tecnología (CONACYT) por el apoyo económico otorgado durante mis estudios de maestría, al Departamento de Computación del Centro de Investigación de Estudios Avanzados del IPN (CINVESTAV-IPN) por el apoyo otorgado en mi proceso de titulación y al Centro Universitario UAEM Valle de Chalco por las becas de escolaridad otorgadas y por hacer de mí, una alumna orgullosa de su Institución.

A la Doctora María de Lourdes López García, mi asesora, por confiar en mí, aceptando desarrollar este proyecto, brindándome el apoyo necesario, haciendo todo y más de lo que hubiese podido esperar, gracias por compartirme de su experiencia, conocimiento, dirigiéndome con seguridad y firmeza para concluir este trabajo de Tesis.

De igual manera, a los Doctores Juvenal Rueda Paz y Cristina Juárez Landín, ambos tutores adjuntos, quienes con sus aportaciones enriquecieron y mejoraron este trabajo.

Al Maestro Francisco Raúl Salvador Ginez por su valiosa cooperación, toda la ayuda que me brindó, así como sus consejos, me sirvieron en muchas ocasiones que me sentía estancada. Al Doctor Santiago Domínguez Domínguez por sus aportaciones determinantes para esta investigación.

A la plantilla de profesores pertenecientes a la Maestría en Ciencias de la Computación por los conocimientos impartidos.

Con todo mi amor a mis hijos Santiago y Samantha, así como, a mis tres ángeles Monserrat, Sebastián y José Eduardo, los cinco son lo que más amo y mi motor de vida. Muy especialmente a mi esposo Eduardo, por ser mi cómplice, por todo su apoyo y comprensión. A mi madre por su apoyo y confianza, sin ella no hubiese logrado esto. A mi padre que sigue siendo mi ejemplo que seguir.

Finalmente, a Dios, quien es la fuente de mi fe y ha estado conmigo siempre, levantándome cuando he caído, dándome la esperanza y fuerza para seguir intentándolo.

Resumen

El control de acceso no autorizado en redes informáticas es un problema que inicia desde el surgimiento de los sistemas de información computarizados, donde la seguridad y la privacidad de la información son factores importantes.

Una solución conveniente para resolver este problema es el uso de un Sistema de Detección de Intrusos (IDS, por sus siglas en inglés). La eficiencia de un IDS está determinada por la certeza en la detección, misma que depende de una correcta clasificación, que tendrá lugar si se cuenta con un vector que contenga las características adecuadas del objeto o entidad a clasificar.

En esta tesis, se propone un proceso para la generación de un vector característico, a partir de información real proveniente de la red que permita realizar una correcta interpretación sobre el comportamiento de los procesos habituales para los cuales la red fue creada, para así, discernir entre lo autorizado y no permitido en la red. Para comprobar la eficacia de la detección se utilizan 5 clasificadores incluido en ellos una red neuronal y árbol de decisión. Así, la certeza de una evaluación precisa de la red, permitirá protegerla de usuarios maliciosos que intenten invadirla sin ser detectados.

Abstract

Unauthorized access detection in a computer networks is a problem that starts from the beginning of computerized information systems, where the security and privacy of the information are important factors. A good solution to solve this problem is the use of an Intrusion Detection System (IDS). The efficiency of an IDS is measured by the precision in the detection which depends on the an accurate classification, that can be possible, using a vector with the appropriate characteristics of the object or entity to be classified.

In this thesis, the process for the generation of a characteristic vector based on real information from the network is proposed. The vector allows classifiers to do a correct interpretation of the behavior of the common processes for which the network was created, in order to discern between what is authorized or non authorized on the network. To verify the effectiveness of the detection, five classifiers are used, including a neural network and decision tree. Thus, the certainty of an accurate evaluation of the network, will protect it from malicious users who try to invade it undetected.

Índice general

Índice de figuras

Índice de tablas

1. Introducción

La importancia de la comunicación radica en la necesidad de compartir información entre entidades. Las redes computacionales son un medio de comunicación que permite compartir información a grandes distancias de manera rápida, fácil y en diferentes formatos. El canal de transmisión de los datos puede ser público o privado, sin embargo, en ambos casos debe proveerse seguridad ante información importante o secreta.

La mayoría de los usuarios puede acceder a las redes de computadoras, que les permite tener una comunicación desde su ubicación hacia cualquier punto donde la red mantenga conexión. Utilizar este canal, sobre todo si es público, implica tener conocimiento sobre cómo usarlo, pero no necesariamente sobre cómo funciona. Ésta diferencia, hace que las entidades maliciosas se aprovechen de los usuarios ingenuos para vulnerar el canal de comunicación usado y lograr ataques como una intrusión no deseada en la red.

Para proteger los datos de quienes usan estos canales, se implementan protocolos de seguridad, así como, la aplicación de métodos y herramientas especializadas para ciertas tareas. Los Sistemas de Detección de Intrusiones (IDS, por sus siglas en inglés) son útiles en la búsqueda de la seguridad, brindando un medio de detección ante una intrusión no autorizada en una red. En esta herramienta, se incluyen métodos para el tratamiento y análisis de los datos que se transmiten, tales como la minería de datos, los algoritmos genéticos y la inteligencia artificial, entre otros. De tal manera que sea posible clasificar el tipo de tráfico y decidir cuándo se presenta o no, un ataque de intrusión.

1.1. Planteamiento del problema

La comunicación es un elemento indispensable para el hombre, las redes proveen un canal de comunicación eficiente, donde se comparte información cada vez más sensible o de carácter más personal, por ello, es necesario encontrar herramientas que brinden protección contra riesgos como robo o suplantación de la identidad de los usuarios, la información privada puede ser violada, los cortafuegos pueden ser corrompidos, las medidas de seguridad violadas o las redes saboteadas.

La necesidad de intercambiar información es igual de vital como el protegerla. Para la protección de estos datos, se deben establecer mecanismos que validen la información que se puede recibir y transmitir. Analizando las soluciones existentes, se encuentra que es viable la creación de un proceso de seguridad, el cual, tendrá como base el grado de certeza obtenido durante la fase de clasificación del flujo de datos de la red, logrando una correcta distinción de un flujo representativo de ataque y de un flujo normal.

Por lo anterior, es importante desarrollar un sistema de clasificación eficiente, enfocado a la protección de la red ante ataques como la denegación de servicios (DoS), que es uno de los ataques más frecuentes de la red y del cual aún no se logra una solución total. Para ello se emplea el proceso propuesto que es guia para la elaboración de las fases durante la creación del sistema.

1.2. Hipótesis

La investigación prueba la correcta clasificación de la base de conocimiento obtenida mediante el uso de algoritmos clasificadores de datos, frente a los resultados obtenidos por otros sistemas clasificadores que empleen algoritmos similares, propuestos como solución para una intrusión no autorizada, mediante una correcta distinción de flujo anómalo contra el flujo normal de una red.

Si se conocen y entienden a fondo los elementos que participan en un ataque de denegación de servicio hacia un servidor web se pueden detectar dichos elementos dentro del flujo de datos de una red para determinar cuándo ocurre un ataque de este tipo, para así, brindar protección a la red.

1.3. Objetivos

General

Determinar un proceso para la clasificación del tráfico de una red en estado normal o bajo ataque, a través de la generación de un vector caracterísito basado en el comportamiento de la red.

Específicos

1. Poner en marcha una red de computadoras que contenga un servidor Web.
2. Identificar ataques sobre una red, principalmente, los ataques de intrusión o denegación de servicios.
3. Reproducir ataques de red de tipo denegación de servicio en el escenario propuesto.
4. Capturar el tráfico de red en el escenario propuesto bajo ataque y en estado normal.
5. Analizar el tráfico capturado e identificar patrones, para la generación de un perfil o de un vector característico.
6. Determinar cuáles algoritmos de clasificación, pertenecientes a técnicas de inteligencia artificial, son aplicables y recomendados para el perfil generado.
7. Realizar las pruebas apropiadas para la validación del funcionamiento del vector característico obtenido.

8. Comparar los resultados obtenidos del proceso de clasificación con lo reportado en el estado del arte.

1.4. Metodología

Se han determinado los siguientes tipos de investigación, descritos a continuación, por la eficiencia que muestran durante su aplicación.

1. Investigación documental: para obtener conocimiento sobre la tarea a realizar, comprender los conceptos principales y encontrar otros estudios e investigaciones referentes al tema de estudio y se basa en consultas de tipo bibliográfico tales como libros, trabajos de tesis, artículos científicos, memorias, revistas, bases de datos especializados, páginas web, videos y otros recursos especializados.

2. Investigación experimental: se emplea una investigación de esta índole durante la realización de pruebas y análisis de los datos obtenidos de la red, para determinar las características y elementos presentes en un flujo normal de red y el emitido al momento de reproducir un ataque de DoS.

3. Metodología en capas: su uso se ve reflejado en la realización de fases, ordenadas jérarquicamente y que obedecen a una necesidad de ejecución, basada en la lógica evolutiva del proyecto. Se determinan las capas necesarias y dentro de estas las actividades que las componen.

1.5. Organización del documento

El trabajo se compone de seis capítulos. El primero de ellos contiene las siguientes partes: introducción, planteamiento del problema, hipótesis, objetivos, metodología, así como la presente descripción del documento; estas conforman la explicación, justificación y razones en las que esta cimentado el desarrollo de este trabajo de tesis.

El capítulo 2 denominado preliminares, comprende diversos temas como el panorama referente a la seguridad actual, los tipos de ataques que se generan con la intrusión de una red, las técnicas que comúnmente se emplean ante una intrusión a la red, la definición y características de un Sistema de Detección de Intrusiones y algunos algoritmos aplicables a la generación de un vector característico, así como, el estado del arte.

En el capítulo 3 se describe el proceso de clasificación propuesta, dando la explicación de las actividades que componen cada fase, como lo son, el diseño de la red, la captura de datos en los dos estados posibles de la red (normal y ataque), la selección de atributos, la generación del vector característico, entre otros.

El capítulo 4 habla de los escenarios que se desarrollaron durante este trabajo de investigación, dividiéndose en Escenario1 y Escenario2, en cada uno de estos apartados se desglosan las actividades desarrolladas y apegadas a las fases del proceso propuesto. Todos los datos contenidos fundamentan y comprueban las bases de la propuesta realizada.

Posteriormente el capítulo 5 contiene una discusión sobre los resultados obtenidos, haciendo una comparativa con algunos trabajos contenidos en el estado del arte. La información presentada en este capítulo tiene como finalidad establecer una visión cualitativa de los resultados obtenidos.

Finalmente, el capítulo 6 muestra las conclusiones a las que se llegaron, tras el desarrollo de este proyecto, enfatizando en los objetivos alcanzados y los puntos claves que pudieron encontrarse durante la tesis. Se proporciona además una visión del trabajo futuro que genero este trabajo.

2. Preliminares

El desconocimiento del funcionamiento de un sistema, así como de su uso y la violación de reglas establecidas para el mantenimiento de la seguridad en una red, son las principales causas de generar vulnerabilidad a una red y de que factores externos o internos puedan generar una incidencia en la red, ocasionando pérdida de información, pérdida de servicios o robo de la información que se maneja.

El presente capítulo tiene la finalidad de adentrar al lector en los elementos relacionados en este trabajo, para así, entender los conceptos y la relación que mantienen entre sí. En un principio se presenta un panorama de la seguridad actual para comprender la importancia de este trabajo de investigación. Posteriormente se incluyen algunos puntos que desarrollan los usuarios de la red, los cuales repercuten directamente en el decremento de la seguridad, al realizar malas prácticas que pueden generar incidentes y así mermar la seguridad que existe.

Le sigue la descripción de los tipos y características que poseen algunos de los ataques más comunes hacia una red, para comprender el modo en que estos operan. Le complementa la descripción de algunos indicios que delatan una red vulnerada, es decir, una red que presenta los resultados de una intrusión en la red.

El Análisis de soluciones descritas en la literatura, se presenta posteriormente, junto con una síntesis de las técnicas empleadas ante una intrusión en la red. Para complementar se describen los sistemas de identificación de intrusiones como una buena solución ante los problemas de seguridad que pueden presentarse en una red, además de, describir la arquitectura de estos sistemas. Finalmente, se introducen los conceptos de algoritmos de clasificación, describiendo las características de los algo-

ritmos empleados en este trabajo de investigación.

2.1. Panorama referente a la seguridad actual

El hombre por naturaleza es un ser sociable, la comunicación, por tanto, será siempre algo imprescindible para él. Actualmente, se han aprovechado las nuevas tecnologías para lograr un proceso de comunicación más ágil y productivo. Las ventajas, privilegios y comodidades a distancia son factores que ayudan a utilizarlas cada vez con mayor frecuencia. Las redes informáticas proporcionan este tipo de comunicación, valiéndose del empleo de infraestructuras ya sean cableadas o inalámbricas. Sin embargo, los sistemas desarrollados aún no logran garantizar un funcionamiento libre de errores o vulnerabilidades, comprometiendo los datos sensibles a un acceso no autorizado durante su transmisión o al servidor donde se encuentran almacenados.

Los principios de protección de los datos manejados a través de una red se presentaron en los ámbitos militares, el uso de sistemas, técnicas, dispositivos y todo lo relacionado con seguridad en una red fue implementado como un lujo en las empresas, posteriormente, como una necesidad general. Las empresas tenían la responsabilidad de no dejar sus datos desprotegidos. Actualmente, las empresas entienden que invertir en seguridad es beneficioso. Para un usuario la seguridad y protección de sus datos también es muy importante, utilizando aplicaciones y dispositivos que garantizan la seguridad de su información.

Por tanto, la seguridad en la red es un tema que cada vez adquiere más interés debido al avance tecnológico en las comunicaciones. Como es esperado, la información transmitida en una red está expuesta a múltiples ataques. Hasta el momento, se han propuesto diversas técnicas para evitar ataques al acceso no autorizado. Por desgracia, no existe una técnica que contemple todos los ataques conocidos. Por lo anterior, los administradores de red tratan de cubrir los huecos de seguridad empleando un mecanismo que combine varias técnicas y contrarreste diversos ataques.

Por otro lado, el envío y recepción de información a través de una red es un proceso

que está protegido por los protocolos de red empleados, los dispositivos utilizados y por la eficiencia de herramientas, sistemas, entre otros elementos que en conjunto operen para brindar seguridad a la red por donde viaja la información.

Existen protocolos de red que son los encargados de garantizar el envío y recepción de la información como son UDP y TCP. En tanto, protocolos como IPSec, L2TP y SSH son los responsables de proveer un canal seguro para la comunicación. Finalmente, existen otros protocolos de cifrado de datos que buscan la confidencialidad de la información transmitida, como los contenidos en IPSec y que son DES, 3DES y AES, entre otros (Dordoigne, 2015).

Otros elementos que buscan proporcionar seguridad en la transmisión como lo son los firewalls con su filtrado de información, las Listas de Control de Acceso (ACL) con el envío restrictivo de información, cifrado, las Redes Virtuales Privadas (VPN) que proporcionan un canal seguro, la implementación de protocolos seguros y las VLAN (Aguirre, 2013).

El uso de todos los elementos existentes que dan seguridad es posible gracias a la compatibilidad que tengan entre sí para su implementación conjunta, lograrlo representa un reto. Todos los protocolos de red existentes y pertenecientes a las diferentes capas o niveles de los estándares OSI y TCP, los lenguajes de programación en los que se basan las aplicaciones que se emplean en seguridad y los dispositivos o hardware que se emplean en una red para el proceso de comunicación, deben obedecer y estar diseñados en base a los estándares ya validados a fin de garantizar la compatibilidad de uso con otros elementos de red (Dordoigne, 2015).

De igual manera, las versiones de protocolo de red IPV4 e IPV6 presentan problemas de seguridad. El protocolo IPV6 posee ventajas ante IPV4 gracias a la encriptación y autenticación que IPSec y al manejo de un rango de direcciones bastante grande, es decir es eficiente contra cierto tipo de ataques, pero aún es vulnerable a otros que también afectan a IP4, además la compatibilidad que presenta IPV6 con algunas técnicas de seguridad existentes no es posible como la implementación del mecanismo seguro de vecinos (SEND) gracias a las llaves públicas (PKI) que IPV6 maneja (Tamayo,

2016).

Así entonces, el tema de seguridad es un tema importante debido al alza de vulnerabilidades en la red, así como, el aumento del número de riesgos en la seguridad, lo que lleva a una demanda de elementos que garanticen un mayor nivel en la seguridad y actualizaciones con las nuevas tecnologías emergentes, para seguir garantizando la compatibilidad entre los sistemas.

Los tipos de amenazas a la seguridad de un sistema o red computacional, según el estándar ISO 7498-2 se pueden dividir en cuatro categorías generales y que son (Bertolín, 2008):

1. *Interrupción,* afecta la disponibilidad al eliminar un factor o elemento perteneciente al proceso de comunicación.

2. *Intercepción,* amenaza a la confiabilidad al obtener el acceso a un factor o elemento perteneciente al proceso de comunicación.

3. *Modificación,* amenaza a la integridad debido no sólo al acceso sino a la posibilidad de cambio de los datos

4. *Fabricación,* es también una afectación a la integridad debido a la implantación de información falsa al sistema.

2.2. Incidentes y vulnerabilidades

Los niveles de incidentes de seguridad en Internet se han incrementado como se puede constatar en la página del Equipo de Respuesta ante Emergencias Informáticas (CERT) (http://cert.org/). Estos valores se relacionan con la explotación de vulnerabilidades que se presenta como consecuencia de un desconocimiento del uso correcto de estándares, dispositivos y normas generadas para salvaguardar la integridad de una red.

Prevenir la incidencia de este tipo de amenazas está relacionado, principalmente, con el aprendizaje de los usuarios sobre el correcto funcionamiento de los dispositivos;

la acotación por parte de los administradores de la red a los derechos y responsabilidades que posea cada área perteneciente a la organización o empresa que opere la red; el uso de buenas prácticas; y la generación de políticas de seguridad adecuadas para la red. Respecto a este último punto, se debe considerar que para la generación de una política y su implantación es necesario considerar las siguientes fases recomendadas (Bertolín, 2008):

1. Análisis y valoración de riesgos
2. Construcción de la política
3. Implantación de la política
4. Mantenimiento de la política
5. Implicación de todo el componente humano

El componente humano es uno de los factores más importantes para el mantenimiento de seguridad, ya que, aunque existan herramientas que logren garantizar un nivel grande de seguridad, pueden verse vulneradas ante el uso normal de la red por usuarios que no respetan las reglas de seguridad que están estipuladas, ya sea por ingenuidad o por exceso de confianza. Una de las principales creencias dañinas para no cumplir las reglas de seguridad, es pensar que la información contenida en la computadora no es de importancia para el atacante (Tamayo, 2016).

En este sentido, el término *ingeniería social* se refiere al uso de técnicas basadas en la interacción con la víctima a fin de obtener información confidencial como lo son contraseñas que permitan tener acceso a un punto de la red, para así, poder suplantar a la víctima, obtener información sobre otros componentes de la red, e inclusive desde cualquier punto retransmitir un virus informático (Salamanca, 2017).

Esta comprobado que dejar a la vista datos sensibles, como contraseñas, teléfonos y claves en papeles, libretas, pizarrones, entre otros, a los que se pueda tener acceso, proporciona al atacante un medio de obtención de información, por ejemplo al tirar sin

romper minuciosamente los comprobantes que contengan datos relacionados con la banca o con la escritura de claves, contraseñas o cualquier tipo de dato sensible que no se destruya en su totalidad puede ser blanco de entidades maliciosas (Tamayo, 2016; Salamanca, 2017).

Otras ideas erróneas que tienen los usuarios es que el no abrir archivos desconocidos o contar con un antivirus, usar firewall o sistemas operativos no comerciales como Windows o como Unix o Mac los libra de ser víctimas de un ataque.

Los usuarios y de los administradores de la red son el factor humano que tiene la responsabilidad de uso y mantenimiento del equipo o los equipos asignados, así como de la información que se maneja (Bertolín, 2008).

Para proteger la información es necesario comprender el tipo de empresa u organización que genera dicha información, los medios por los cuales se transmiten los datos, los encargados de transmitir la información, los elementos utilizados para el proceso de comunicación, así como, las deficiencias y vulnerabilidades que tienen inicialmente estos sistemas. Un modo correcto de estudiar todos estos puntos es generar un análisis de riesgos.

Un análisis de riesgos proporciona una lista de los elementos que intervienen, ordenándola jerárquicamente, para así, definir niveles de afectación que se tendrían en caso de que alguno de estos elementos se vea vulnerado (Dordoigne, 2015; Bertolín, 2008).

2.3. Tipos de ataques

La prevención de ataques es primordial para la seguridad de una red. Los puntos mencionados en la sección pasada dejan ver que no se debe subestimar al enemigo y es necesario protegerse hasta de lo más improbable. Si bien es cierto que es imposible una protección total, se debe analizar continuamente el comportamiento de la red, emplear actualizaciones, incluir técnicas y herramientas de protección a la red.

Lo anterior, se basa en que la generación de puntos de afectación crece exponen-

cialmente a medida que se incrementa el número de usuarios de una red; va de la mano con la necesidad de incluir nuevo software que contenga puntos vulnerables, es decir, a medida que la red crece y se hace más compleja, el grado de vulnerabilidad se incrementa.

La clasificación de ataques se puede realizar en función de la afectación que generan o bien al modo de operar para su realización, bajo esta comparación existen dos tipos de ataques: pasivos y activos. Esta clasificación está basada en el grado de acción, profundidad en la intrusión generada y modificación en la estructura u operación de la red que se realiza. A continuación, se describen algunas características de los ataques pertenecientes a estos grupos.

- Ataques pasivos

  Identificar este tipo de ataque puede ser muy complicado debido a que no hay un grado notorio de afectación en el sistema, pero puede darse, ya que se requiere efectuar un ataque pasivo para poder generar un ataque del tipo activo. Los administradores deben prevenir este tipo de ataque protegiendo la información mediante el uso de contraseñas y generación de perfiles, también debe cuidarse el acceso a los lugares físicos donde se encuentren los equipos sensibles, entre otra clase de medidas.

  Los ataques pasivos buscan obtener u observar información importante que no necesariamente implica su modificación o alteración. Ejemplos de estos ataques son las monitorizaciones y capturas de tráfico. La Tabla 2.1 presenta una lista simplificada de los ataques pasivos indicando su objetivo y las acciones que realiza para alcanzarlo.

- Ataques activos

  Los ataques activos ejecutan una acción maliciosa que afecta directamente a la organización. Como, por ejemplo, se realizan modificaciones o falsificaciones en el flujo de datos, se pretende ser alguien que no es, es decir suplantar a fin de obtener información o colapsar algún servicio (Pellejero, Andreu, y Lesta,

Tabla 2.1: Listado de ataques pasivos. [Fuente propia]

Nombre	Objetivo	Acciones
Espionaje	Recopilar información sobre la topología de la red	Emplear hardware o software especial
Monitorización de la red	Captura de información sensible (MAC, IP, contraseñas, etc)	Configurar una tarjeta WLAN en modo promiscuo o monitor
Descubrimiento de contraseñas	Descifrar contraseñas	Emplear métodos de fuerza bruta o diccionario

2006). Este tipo de ataques se puede subdividir en enmascaramiento, repetición, modificación de mensajes y denegación de servicios (Cobo, 2011).

La diversificación de estos ataques es muy grande, además se generan nuevas formas de lograr un mismo efecto; los atacantes investigan como vulnerar las nuevas tecnologías aprovechándose de las ya conocidas para generar todo tipo de ataques, combinando muchas veces varios tipos de ataques para generar uno de mayor afectación.

Existen puntos ya conocidos como vulnerables, los cuales siguen siendo explotados en tanto no se logre corregirlos totalmente. Los diseñadores de sistemas operativos, creadores de protocolos de red, creadores de normas y estándares, programadores de software, diseñadores de red, todos tienen que lograr mantener la compatibilidad con las tecnologías emergentes y sanear los puntos débiles de la seguridad.

Es muy importante entender que, invertir en la protección de la red es primordial, es responsabilidad de quien otorga el servicio garantizar la seguridad de la información y de los usuarios utilizar los sistemas correctamente. La Tabla 2.2 corresponde a algunos tipos de ataques activos donde se muestra el nombre, objetivo al que busca llegar dicho ataque y las acciones que debe realizar para completarlo (Miranda, 2014; Aguilera, 2011).

Tabla 2.2: Listado de ataques activos.[Fuente propia]

Nombre	Objetivo	Acciones
Puntos de acceso no autorizado	Establecer una conexión directa a la red	Conectarse a la red como un usuario normal
Suplantación	Obtener claves de acceso para romper filtros basados en MAC	Suplantar la identidad del usuario empleando credenciales o identificadores estáticos obtenidos previamente
Hombre de en medio	Interponerse entre dos puntos de red que hayan establecido comunicación	Emplear dos interfaces para simular un punto de acceso o un usuario válido a fin de obtener usuario y contraseña del emisor y receptor que establecieron la comunicación
Secuestro de sesión	Robo de sesión de uno o de los dos elementos que han establecido una comunicación	Tomar una conexión existente entre dos dispositivos, generar tráfico que parezca venir de una de las partes, para obtener respuesta y robar la sesión
Denegación de servicio	Inutilizar a la red para que no se pueda acceder a ella	Colapsar total o parcialmente un servidor, saturándolo con múltiples peticiones de servicio desde una PC o varias que hayan sido establecidas previamente como zombies
Duplicación	Obtener nombres de usuario, claves y contraseñas enfocadas a actividades bancarias	Se duplica una página web del sitio oficial del banco para que se ingresen las claves al intentar acceder a la página
Spam	Saturación de la red	Envío de correos masivos que no hayan sido solicitados por el usuario
Código malicioso	Introducir hardware, software o firmware infectado	Envío de correos que contengan incrustadas ligas que al darle clic ejecuten virus o bien generen cadenas que saturen al servidor de correos
Hoax	Obtener direcciones de correo de usuarios válidos	Envío de correos que generen cadenas, algunos de ellos con virus incrustados

2.4. Indicios de una intrusión en la red

Es importante responder a los incidentes que comprometan la seguridad de un host o de una red, para ello, es imprescindible conocer los principales indicadores de un incidente de seguridad. Una lista que representa los puntos elementales que delatan una intrusión en la red son los propuestos en (Gonzalez, 2010), la cual se muestra a continuación:

- Uso excesivo de los recursos de un sistema (memoria, procesador, pila)
- Modificación de la integridad de la información almacenada en un sistema
- Accesos a los sistemas en horarios diferentes a los habituales
- Intentos excesivos para descifrar las claves de acceso hacia un sistema
- Reconocimiento de puertos de forma remota hacia un sistema (barrido de puertos)
- Aparición de pantallas emergentes que no pertenecen al sistema

La exploración de puertos de Internet es el indicador más común de un ataque, se pueden utilizar varias herramientas para supervisar los archivos de registro de suceso como *Microsoft Operations Manager* o herramientas gratuitas como *Event Comb* y *Dumpel*; si se ha logrado penetrar las defensas, es posible que se borren las pistas que puedan delatar tal ataque, esto se logra borrando o modificando los registros de sucesos, borrar estos registros genera un suceso 517. La presencia de dicho suceso o la ausencia total de registros es por tanto una señal de un ataque efectuado satisfactoriamente (Smith y Komar, 2003).

No tener acceso a recursos de red, es otra señal, de que la seguridad se ha comprometido (conocido como denegación de servicio). Otro tipo de señal es la actividad reflejada en el procesador, que puede delatar una intrusión efectuada; el poder ejecutar procesos ocultos del administrador de tareas es otro tipo de señal, que se ve reflejada

en los porcentajes de utilización del procesador; los servicios que se deberían ejecutar y se han pausado o detenido; los servicios nuevos o los que ya no están; son muestra de que el atacante ha modificado el sistema para satisfacer sus fines.

Siguiendo con la descripción de señales que delatan una intrusión se tiene la presencia de archivos incluidos o carpetas faltantes, que arrojen un descenso en el espacio de sistema; cambios de fecha/hora en el sistema o en los archivos del algoritmo hash (algoritmo empleado para convertir mensajes o datos en un valor numérico) (Dulaney, 2012); la presencia de nuevos controladores es otra forma de borrar pistas para un intruso. El cambio en los permisos de usuario, pertenencia a un grupo u otra directiva de seguridad administrativa también son muestras habituales de un ataque efectuado (Smith y Komar, 2003).

Es importante notar que la seguridad no es necesaria sólo en un host a nivel local, también lo es en una red empresarial. La protección de los sistemas de información requiere de un profundo estudio de los sistemas, las redes, los programas y el hardware para prevenir amenazas futuras o ya existentes. Evaluando los riesgos, conociendo la infraestructura, conectividad y protección de redes, determinando amenazas y vulnerabilidades, conociendo los mecanismos de protección física y lógica, controlando los accesos, seguridad y vulnerabilidad en las redes, la recuperación ante ataques o respuesta a estos mismos, y determinar medidas para que los usuarios sepan protegerse.

2.5. Técnicas de seguridad ante una intrusión en la red

Una técnica (del griego, arte, técnica u oficio) según la Real Academia de la lengua española es el conjunto de procedimientos y recursos del que se sirven una ciencia o un arte, habilidad para ejecutar cualquier cosa, o para conseguir algo. Para las áreas tecnológicas, una técnica es un procedimiento, conjunto de reglas, normas o protocolos que tiene como objetivo obtener un resultado determinado y efectivo.

La técnica requiere tanto destrezas manuales como intelectuales, frecuentemente el uso de herramientas y de conocimientos varios. Los encargados de la seguridad en

una red deben poseer los conocimientos especializados en la materia, así como estar al tanto de las herramientas que puedan emplearse, mismas que van actualizándose día a día.

Las técnicas empleadas para la protección de la red deben proporcionar los 5 principios básicos que son (Aguilera, 2011):

1. *Integridad*, asegura que los datos del sistema no han sido alterados ni cancelados y que el contenido recibido de los mensajes es el correcto;

2. *Confidencialidad*, proporciona protección contra la muestra de los datos de forma deliberada o accidental;

3. *Disponibilidad* de la información cuando se requiera autenticación o identificación. El sistema debe ser capaz de verificar la autenticidad del usuario que intenta acceder a la red;

4. *No repudio o irrenunciabilidad* que proporciona a un sistema evidencias irrefutables de la autoría de un hecho. Consiste en no negar la transmisión de un mensaje emitido o la recepción de un mensaje, esto basado en pruebas de envío y recepción que certifican la identidad de emisor y receptor;

5. *Control de acceso* que se refiere al control de los recursos a los que pueden acceder los usuarios autorizados.

2.5.1. Técnicas de protección a la red comúnmente empleadas

En la búsqueda de técnicas que logren garantizar la seguridad de los datos durante su transmisión y la seguridad en los equipos que resguardan dicha información, según (Castro, Díaz, y Sancristóbal, 2014), los encargados de resguardar esta seguridad emplean uno o varios de los siguientes sistemas:

- *Defensa de seguridad de sistemas operativos,* enfocados a servidores y dispositivos móviles. Son reglas, introducidas en las medidas de seguridad y otras, que

requieren ser configuradas, o que vienen por default en los sistemas operativos. Un ejemplo de esto se vio en la versión de UAC (User Account Control) que implementó Microsoft a partir de Windows Vista, este sistema de seguridad ejecutaba todos los programas en modo restringido lo que lograba que para todas las aplicaciones que requerían permisos administrativos, el usuario permitiera o denegara los permisos de ejecución de dichas aplicaciones.

- *Sistemas de identificación o autenticación seguros:* se hace referencia a contraseñas, sistemas biométricos, certificados digitales o tarjetas de identificación. Un sistema tradicional de identificación personal realiza una autenticación de una entidad relacionada con la persona a través de algo que la persona tiene como una llave, una credencial, etc.; algo que la persona sabe cómo una clave, un pin, una contraseña, entre otros; algo que la persona es, como un rasgo personal fisiológico, una huella, la forma de su cara, la palma de la mano, etc.; finalmente algo que la persona genera como un patrón de comportamiento, tono de voz, firma escrita, etc. (Simon, 2003).

- *Un sistema cortafuegos:* es una combinación de hardware y software, que tiene como meta que todo el flujo saliente y entrante tiene que pasar por él; únicamente el flujo permitido y especificado previamente penetra en la red. El cortafuego está en medio de dos redes que manejan el mismo método de cifrado y descifrado. Se clasifican de acuerdo con su configuración y puede ser, por establecimiento de conexión, por filtrado de paquetes o la combinación de los dos anteriores (Esparza, 2013). Restringe las conexiones entrantes para el acceso de servidores que se sitúan en una red aislada y denominada zona desmilitarizada (DMZ). El acceso también puede ser de una red privada a Internet a través de proxies intermediarios que realizan peticiones mediante conversiones (NAT) o adaptación de puertos (Socks) (Rosado, 2014).

- *Los sistemas criptográficos:* consisten en técnicas, principalmente de cifrado de datos que, si bien es cierto es posible romper, se requiere de métodos más sofis-

ticados y especializados de ataque. Otorgan la posibilidad de emplear algoritmos que cifren los mensajes que son transmitidos, en el caso de ser captados por un tercero, la información contenida en estos mensajes no pueda ser obtenida en un formato claro y tampoco es posible descifrarla (Sanchez, 2012). Emplean algoritmos de llave privada, llave pública o la combinación de ambos. La seguridad está basada en la existencia y lo complicado que resulta resolver las operaciones matemáticas inversas, de las operaciones en las que se encuentren cimentados los algoritmos empleados y el factor aleatorio que pueda agregarse a estos algoritmos (Ochoa, 2013).

- *Sistemas antivirus:* basados en software que permiten contrarrestar los efectos de los virus informáticos. El concepto de virus agrupa todos los tipos de malware, que incluye virus, gusanos, troyanos, rootkits, spyware, adware, crimeware y software malicioso e indeseado. El malware provoca daños en el equipo o equipos una vez que se ha esparcido, que pueden ser saturación del equipo, borrado de archivos, obtención de claves, o el inicio de un ataque más elaborado como el *phising* que busca la obtención de datos bancarios (M., Becerra, y Guevara-Juárez, 2010).

- *El sistema de análisis de vulnerabilidades:* se refiere a la búsqueda de protección para los activos de una red (hardware, software, datos, entre otros). En general, las vulnerabilidades se pueden agrupar en función de su diseño, implementación, uso y vulnerabilidad del día cero. El diseño se refiere a la deficiencia por los protocolos empleados o políticas de seguridad ineficientes; la implementación abarca errores de programación, existencia de puertas traseras en los sistemas informáticos por descuido de los creadores de las herramientas que se usan en la red; uso, es el empleo inapropiado de las herramientas adoptadas por mala configuración o desconocimiento; y vulnerabilidad del día cero se refiere a las deficiencias conocidas en los sistemas que no se pueden cubrir (Mifsud, 2012).

- *Los estándares para sistemas de gestión de seguridad:* se refiere al empleo de

normas, reglas y protocolos enfocados a las redes que pretenden establecer los lineamientos de correcta estructuración y uso de herramientas y técnicas empleadas. Ejemplo de estándar es IEC 27001 y como ejemplos de protocolos están TCP, ISO, RADIUS, TACACS, SSH, SSL (Castro y cols., 2014). Los estándares de calidad reciben aportaciones de todo tipo para su redacción pueden ver la seguridad en todos sus puntos de vista, establecer una metodología para su implementación y uso (Bertolín, 2008).

2.5.2. Herramientas de seguridad empleadas contra intrusiones de red

Un administrador de seguridad debe identificar los puntos vulnerables en la red y determinar las mejores herramientas que contrarresten dichas vulnerabilidades. Para elegir correctamente las herramientas que se requieren, debe conocer el funcionamiento de cada una de ellas e implementarla acatando las reglas de uso. Dependiendo de los elementos de una red, se derivarán las herramientas que puedan emplearse para garantizar su seguridad (Dordoigne, 2015).

Las herramientas tradicionales, que los administradores de red usan son: Redes Virtuales Privadas (VPN), Sistemas de Detección de Intrusiones (IDS), Snort y Sistema de Prevención de Intrusiones (IPN), descritas brevemente a continuación.

- *Una red privada virtual:* establece una conexión de red privada que se utiliza en una red pública. Este tipo de red puede conectar una red de área local (LAN) a través de Internet a otras redes públicas. Una red VPN requiere software y hardware en los servidores y terminales. Generalmente, emplean un protocolo de túnel, por ejemplo, L2TP (Layer 2 Tunneling Protocol, Protocolo de túnel de capa 2), IPSec (Internet Protocol Security, Protocolo de túnel punto a punto) o PPTP (Point-to-point Tunneling Protocol, Protocolo de túnel punto a punto) (Dulaney, 2012). A través de una red VPN los datos se transmiten cifrados, el destinatario y el emisor pueden descifrar estos datos. Para garantizar la seguridad en una

VPN, se deben cumplir principios básicos: autenticación, no repudio, integridad y confiabilidad (Aguilera, 2011).

- *Los sistemas de detección de intrusiones:* es un mecanismo que escucha el tráfico de red, a fin de detectar actividades anormales y reducir el riesgo de intrusión. Es común que los atacantes intenten comprometer los sistemas de detección de intrusos, saturando el tráfico o enviando información falsa de lo que sucede en la red. Los IDS utilizan cuatro enfoques basados en: comportamiento, firmas, anomalías y heurísticos (Dulaney, 2012). Puede estar instalado en computadoras individuales basado en host (HIDS) parecido a un cortafuego. También puede estar basado en la red (NIDS) (Bradley y Carvey, 2008). Se retomará esta herramienta para describirla con mayor profundidad más adelante.

- *Snort:* es un sistema de detección de intrusos a nivel de red. Su objetivo es monitorizar el tráfico, empleando un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder a través de patrones o firmas previas. En general, sus funciones son similares a un analizador de tráfico (*sniffer*). La primera versión de Snort surgió en los años 80 y fue propuesta por Marty Roesch. Su arquitectura se compone de un módulo de captura de tráfico, decodificador, preprocesadores, motor de detección, archivo de reglas, plugins de detección y plugins de salida. Además, está disponible bajo licencia GLP, por tanto, es gratuito y funciona para plataformas Windows y GNU/Linux (López, 2009). Es muy efectivo ya que verifica los paquetes de red y puede interceptar un rango grande de ataques conocidos o actividad perjudicial (Bradley y Carvey, 2008).

- *Sistema de Prevención de Intrusiones (IPS):* previene e identifica anomalías de actividad, se sitúa dentro del tráfico de la red a fin de evitar las intrusiones (Tejada, 2015). Un IPS puede modificar las reglas del cortafuego para bloquear todo el tráfico de un puerto como medida de control e implementa las mismas técnicas de un IDS, pero además de detectar también responderá a un ataque e

intentará detener la intrusión. Monitoriza el tráfico de las capas de red y transporte, analizando los contenidos y la carga de los paquetes en búsqueda de ataques sofisticados que puedan agregar datos maliciosos a las capas enlace de datos y aplicación (Stewart, 2007). Las plataformas IPS en equipos de red marca Cisco, se basan en una mezcla de tecnologías de detección basadas en firmas, en perfil y en análisis de protocolo (Ariganello y Sevilla, 2014).

- *Los HoneyPots (Tarros de Miel):* técnica que es una copia parcial o total de la red a proteger, los elementos copiados no se implementan con protección a fin de dejarlos vulnerables premeditadamente, para que sirvan de señuelos. Este engaño permite observar las metodologías y técnicas que el atacante emplea en la intrusión, a fin de recabar información para protección o contraataque (Gonzalez, 2010). La red emulada debe estar aislada de la red en producción; configurada y administrada por expertos para protegerla como a la real (León-Jaramillo, 2011). Una variante de este tipo de técnica es el HoneyNets que se compone de granjas de servidores o emulan redes corporativas muy grandes lo que implica la creación de una red muy costosa que sirva de señuelo y hasta sea complicado romper su seguridad (Gonzalez, 2010).

2.6. Otras soluciones

Las redes han evolucionado, todos los elementos que actualmente pueden conformar una red hacen que esta se vuelva más compleja. Aunado a esto, las herramientas empleadas para la seguridad de la red pueden ampliar las áreas de ataque y crear nuevas vulnerabilidades al no emplearse de la manera correcta.

Entre las múltiples herramientas de seguridad existentes, se encuentran las que son desarrolladas para el análisis de datos como: NSM (Network Security Monitoring), SIEM (Security Information and Event Management), SEM (Security Event Management), SIM (Security Information Management), PNA (Passive Network Audit), entre otras (Santillán Arenas, 2015).

2.6.1. Monitorización de la seguridad de la red (NSM)

Es un modelo de análisis de tráfico de red que permite la creación de un framework que incluye técnicas de monitoreo, detección y retención de datos que evidencian una intrusión. Las técnicas empleadas son IDS, analizadores de flujo de datos (sniffers), entre otros.

El proceso de atención a incidentes de red cuenta con cuatro fases que son Plan, protección, detección y respuesta. NSM está vinculado a la fase de detección, específicamente con dos procesos: contención pronta del incidente, el cual se compone de la información que se tiene sobre la intrusión detectada; y en emergencia, donde la metodología del incidente fue detectada y están las evidencias del ataque. Proporciona un modelo de referencia de intrusiones el cual tiene cuatro tipos de datos que se enlistan a continuación (Santillán Arenas, 2015):

- Datos de contenido completo: captura bit-a-bit.
- Datos de sesión: distribución de protocolos y acumulación de tráfico.
- Datos estadísticos: registro de conversaciones entre dispositivos.
- Datos de alerta: información extraída de IDS

En este modelo se busca indicar no sólo qué herramientas pueden emplearse, sino dónde y cuándo, detalles de implementación, áreas de monitoreo, zonas vulnerables, entre otros (Bejtlich, 2013).

2.6.2. Información de seguridad y administración de eventos (SIEM)

El modelo SIEM implementa la minería de datos para lograr la extracción de modelos o patrones descriptivos de una gran cantidad de datos, mediante su interpretación como producto de un análisis estadístico, estos patrones permiten una auditoria de los datos. Además de la minería de datos, se emplean otras técnicas y herramientas

como IPS, IDS, firewalls, routers, bitácoras de sistemas, etcétera. Entre las características principales que los SIEM proporcionan están los siguientes (Santillán Arenas, 2015):

- Acumulación de datos en el motor de análisis centralizado.
- Correlación: interpretar y establecer las relaciones.
- Alertas.
- Cumplimiento: revisión del cumplimiento sobre lo ya establecido.
- Retención: almacenamiento de datos históricos.
- Análisis forense: reconstrucción de los *hechos* con ayuda de una línea de tiempo.
- Inteligencia: toma de decisiones en base a un análisis de la seguridad.

Este modelo busca una correlación y reducción de los datos, es decir, un filtrado que primero aislé el evento genere sesiones y reglas (Bejtlich, 2013). Los SIEM combinan características de los SIM con el análisis en tiempo real y los SEM con el almacenamiento a largo plazo de registros de eventos (Santillán-Arenas, 2014; Bejtlich, 2005).

2.6.3. Auditoria pasiva de tráfico en la red (PNA)

Este modelo realiza un análisis de bitácoras de sistema y la correlación de datos. Su principal elemento de análisis es el tráfico de red, partiendo de ésta para la generación de reportes. Las herramientas que emplea son como su nombre lo indica pasivas, es decir, que su análisis y obtención no genera ninguna alteración o intervención en la operación habitual de la red que se analiza, de manera interna.

La acumulación de datos involucra su procesamiento y decodificación, para la creación de las bitácoras, esta aproximación a los datos implica una interpretación, lo que será la base para la creación de una firma para identificar y describir una actividad o

sistema. Este análisis permite identificar protocolos, versiones de software, dominios, alertas de IDS, flujos, por mencionar algunos.

PNA también se conoce como Identificación Pasiva de Red (Passive Network Discovery) (Arkin, 2012) que permite al administrador de red responder algunas cuestiones acerca de qué y cómo está compuesta la red y cómo opera, empleando técnicas de análisis forense de eventos, identificación de vulnerabilidades y perfiles. La desventaja de este modelo es la limitación en su fuente de información que puede provocar una imprecisión de los datos, lo que lo hace poco confiable para definir el estado de la red (Santillán-Arenas, 2014).

2.7. Sistemas de Detección de Instrusos (IDS)

Son herramientas que escuchan el tráfico de la red y son capaces de detectar actividades inusuales, para así, reducir el riesgo de una intrusión no permitida. Pueden evaluar la información en tiempo real, o que esté contenida en una base de datos. Según el área que protegen estos pueden clasificarse en (Horng y cols., 2011; Bradley y Carvey, 2008):

- HIDS es un sistema de detección de intrusiones basado en host, que tiene como objetivo identificar ataques con base en la observación de los encabezados de los paquetes, para detectar a una entidad que intenta violar o modificar la seguridad del host. Recogen y analizan los datos que originan en un equipo que recibe un servicio, como un servidor web. Además de detectar actividad desautorizada, los sistemas IDS de host son también eficaces en la detección de modificaciones de archivos. Un HIDS permite identificar el registro de eventos de seguridad de los sistemas operativos.

- NIDS es un sistema de identificación de intrusiones de red y su detección se basa en el análisis de los paquetes de red y de los protocolos que se emplean para la transmisión de los datos, ambos pueden ser en tiempo real o no. Analiza

los paquetes de datos que viajan sobre la red real, estos paquetes se examinan y se comparan con datos empíricos para verificar su naturaleza: malévolo o benigno. Los NIDS tienden a ser distribuidos. En vez de analizar la información que originan en un host, se analizan técnicas red, basadas en las aplicaciones de identificación de datos TCP/IP o de otros paquetes del protocolo que viajan a lo largo de la red.

La Tabla 2.3 muestra los enfoques que puede tener un IDS, de acuerdo al modo de detección que maneja: comportamiento, firmas, anomalías o heurístico (Liao, Lin, Lin, y Tung, 2013; Khan, Awad, y Thuraisingham, 2007; Depren, Topallar, Anarim, y Ciliz, 2005; Portnoy, Eskin, y Stolfo, 2001).

2.7.1. Arquitectura del IDS

La estructura de este sistema cuenta con los elementos: base de conocimiento, base de hechos, motor de inferencia e interfaz; cuenta además con módulos de justificación y la colaboración del elemento humano que funge como la parte experta del sistema.

La base de conocimiento contiene los datos que son recolectados para su análisis, dichos datos son obtenidos mediante el empleo de una aplicación, que funge como sniffer, por ejemplo, Wireshark, que captura muestras del flujo de datos, que se establece durante la comunicación y transmisión de datos de la red.

La base de hechos es el conjunto de reglas y parámetros establecidos, para el IDS son los datos filtrados, que serán evaluados y que están contenidos, en una base de datos, que será ingresada y sometida al sistema que contiene el motor de inferencia. Normalmente los datos a evaluar son el tamaño de la carga, la presencia de las banderas SYN y ACK en estado encendido, así como, el valor del tiempo, por mencionar algunos.

El motor de inferencia tendrá como labor el análisis de los datos, capturados y filtrados para su evaluación por la base de hechos.

Tabla 2.3: Clasificación de IDS [Fuente propia]

Enfoque	Descripción
Comportamiento	*Funcionalidad*: busca variaciones de costumbres, como un tráfico elevado. *Ventajas*: método simple y efectivo para detectar ataques conocidos. Detalla el análisis contextual. *Desventajas*: inefectivo para ataques no conocidos, o variantes de los conocidos. Difícil mantener las firmas y patrones actualizados. Requiere mucho tiempo para aprender.
Firmas o MD-IDS	*Funcionalidad*: clasifica ataques con base en firmas y auditorias. *Ventajas*: efectivo para detectar vulnerabilidades nuevas, es menos dependiente del sistema operativo y puede detectar el abuso de privilegios. *Desventajas*: los perfiles cambian constantemente y no es efectivo en su reconstrucción.
Anomalías o AD-IDS	*Funcionalidad*: busca elementos fuera de lo común, se centra en patrones de tráfico. *Ventajas*: está basado en comportamiento de protocolos de red, detecta secuencias anormales de comandos. *Desventajas*: no distingue ataques que simulen un comportamiento usual en los protocolos y puede ser incompatible con algunos navegadores.
Heurístico	*Funcionalidad*: emplea algoritmos para analizar el tráfico que pasa por red. *Ventajas*: puede predecir eventos y ser autodidacta, distingue secuencias de comando. *Desventajas*: consume muchos recursos y es de funcionamiento complejo.

Dentro del proceso que desarrolla el motor de inferencia, se encuentra la evaluación de los módulos de justificación que son algoritmos empleados durante el proceso de clasificación. Los módulos de justificación, que evaluarán según las reglas establecidas, la presencia de los datos, que determinan si ha sido efectuado un ataque o intento de intrusión en la red. Los módulos de justificación estarán ligados al establecimiento de una conclusión ante los datos evaluados. Es en esta fase entran en función las firmas determinadas en el sistema, producto del análisis y filtrados de los datos. Se determinan los patrones que definen si existe o no una intrusión en la red.

2.8. Herramientas de clasificación

El proceso de clasificación es uno de los más útiles y comunes en el tratamiento de datos, ya que permite analizar el comportamiento de una o más variables dentro de un conjunto de información. Dicho conjunto es formado por datos agrupados y dependientes del atributo al que pertenecen, los datos son sometidos al sistema clasificador para así, determinar a qué clase pertenece. Los clasificadores requieren una fase de entrenamiento o construcción de la base de conocimientos (Heady, Luger, Maccabe, y Servilla, 1990; Zhang, Li, Manikopoulos, Jorgenson, y Ucles, 2001). A continuación, se describen 5 clasificadores, que según el estudio del estado del arte son los más comúnmente utilizados para clasificar los ataques de intrusión en una red:

1. Red neuronal: compuesta de varias neuronas (unidad mínima de procesamiento de la información, representa un dato de entrada) que están divididas en varias capas. Las neuronas de una capa se conectan con las neuronas de la capa siguiente y les pasan información. La arquitectura consiste en una capa de entrada que recibe la información del exterior; capas intermedias (ocultas) que realizan el trabajo de la red y una capa de resultados que muestra los resultados de la última capa intermedia (Zhang y cols., 2001; Snapp y cols., 1991).

2. Algoritmo J48: derivada del algoritmo C4.5. Es un árbol de decisión C4.5 para la clasificación que crea un árbol binario (Patil y Sherekar, 2013). Se basa en la

utilización del criterio ratio de ganancia (gain ratio) para evitar que las variables con mayor número de presencia salgan beneficiadas en la selección. Además, el algoritmo incorpora una poda del árbol una vez que éste ha sido inducido (André, Gulnara, Muñoz, y Montalvo, 2010).

3. Random Forest: emplea una selección aleatoria de atributos y genera un conjunto de árboles predictores que serán evaluados posteriormente (Tolosi y Lengauer, 2011). Cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de harpillero que construye una amplia colección de árboles no correlacionados y promediados posteriormente (Deng, Runger, y Tuv, 2011).

4. Naive Bayes: asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, variable, tabulador, parámetro o atributo. Se evalúan de modo independiente sin establecer relaciones o coincidencias. Se puede entrenar en un ambiente de aprendizaje supervisado. Puede ser entrenado con pocos datos, obteniendo las medias y las varianzas de las variables necesarias para la clasificación. Debido a que las variables independientes se asumen, solo es necesario determinar las varianzas de las variables de cada clase y no toda la matriz de covarianza (K., Eibe, Pfahringer, y Holmes, 2004; Demichelis, Magni, Piergiorgi, Rubin, y Bellazzi, 2006).

5. Decision Table: Llamada DTM (Decision Table Majority), se compone de un conjunto de características que se incluye en la Tabla atributos y por instancias etiquetadas (reglas). En su procesamiento cada dato de entrada se asigna a la clase con la que ha tenido mayor número de correspondencias. De esta forma, a partir de un dato no etiquetado el clasificador busca correspondencias de este dato de entrada con el total de reglas para todos los atributos. Si no se encuentra ninguna correspondencia, la Tabla DTM asigna el dato a la clase mayoritaria

(Berdun, Armentano, y Amandi, 2016; Kohavi, 1995).

Los algoritmos toman un conjunto de datos, encuentran la relación, congruencia o resultado representativo de los datos analizados, obteniendo la interpretación de un fenómeno. En el IDS, los algoritmos dan explicación a la relación causa-consecuencia durante la transmisión de datos en un estado normal y de ataque.

2.9. Estado del arte

La información generada de procesos de red es cuantiosa, y tiende a crecer en cuanto la arquitectura de la red y los servicios que proporcionan se incrementan. Garantizar la seguridad de esta información obliga a buscar mejores herramientas. La base de un correcto funcionamiento de estas herramientas y su efectividad depende de lo certero que sea el juicio generado, es decir, la capacidad de distinguir correctamente el flujo que circula para así discernir entre lo permitido y lo no permitido.

Dentro de la literatura que contiene los avances en este tipo de trabajos podemos encontrar que siguen distintas vertientes, algunos apuestan por la variación de clases para evaluar correctamente, otros autores mencionan la necesidad de acotar las variables a evaluar y ser más mesurados en la cantidad de clases con las que se trabaja. Otro aspecto que se considera es el enfoque de aprendizaje supervisado, no supervisado, semi-supervisado, entre otros; que garanticen una mejor evaluación de las relaciones entre los datos.

Siguiendo con estos puntos, acciones tales como considerar un pre-procesado en los datos, proponer algoritmos que combinen algoritmos ya existentes, trabajar con bases de conocimientos previamente generadas o proponer el manejo de datos reales, se enfocan a conseguir la muestra apropiada y el evaluador o clasificador preciso que brinde confianza en su predicción.

En (Rivero, Ribeiro, y Kadir, 2016) analizan el conjunto de datos denominado KDD99 que cuenta con 41 atributos distintos, de los cuales, se seleccionaron 23 para su clasificación. El entrenamiento se realizó con el 10 % de los 51 millones de instan-

cias contenidas en la base de datos, aplicándoles tres variantes de preprocesamiento, para después hacer una comparación basada en el uso de algoritmos representativos del aprendizaje automático. Entre estos algoritmos se encuentran, una Red Neuronal Perceptron Multicapa (MLP), SMO que es una variante empleada en WEKA del algoritmo de Máquinas de Soporte Vectorial (SVM), el algoritmo J48, Naive Bayes y el algoritmo basado en instancias K con valores 3, 5 y 7.

Los resultados presentados arrojaron porcentaje del 98.14 % para Naive Bayes y un 99.02 % para J48, siendo éste el más preciso. Para la variante de pre-procesado 2 se tiene a J48 con un 97.43 % ante lo obtenido con SMO con un 99.23 %. Finalmente, los resultados con la variante 3 presenta al algoritmo J48 con 95.85 % y MLP con 98.4 %.

El (Ashfaq, Wang, Huang, Abbas, y He, 2017)se propone un sistema de identificador de intrusiones que use un clasificador basado en aprendizaje semi-supervisado. Los algoritmos empleados para el tratamiento de los datos son J48, Naive Bayes, NB tree, Random Forest, Random tree, Red Neuronal y SVM. Se implementan dos variantes que consisten en el uso de la base de conocimiento KDDCUP99 con los 41 atributos que la integran y una variación de esta base compuesta por 21 atributos. Los resultados de certeza en la clasificación son SVM con 69.52 % que presenta los valores más bajos con un 42.29 % para la segunda variante de la base de conocimiento que cuenta con 21 atributos.

En (Zhu, Liang, Chen, y Ming, 2017) se encuentran tres variantes propuestas con distinto enfoque para evaluar los datos, previo al proceso del clasificador. Los datos son obtenidos de las bases KDD99 y Gure KDD. Tiene 6 posibles clases que representan 5 ataques a la red y una clase que describe un comportamiento normal. Dentro de los ataques que se incluyen en el evaluador, se encuentra el de Denegación de Servicio, además, de un algoritmo para clasificar las clases.

Los resultados porcentuales de certeza se presentan en una Tabla que los divide en los tres enfoques de clasificación trabajados. Para el método de clasificación 1 aplicada a 8 algoritmos se obtuvieron los siguientes resultados, 80.67 % para Random

Forest y 99.21 % para una variante de su algoritmo propuesto. El segundo método de clasificación se aplicó a tres algoritmos donde su propuesta obtuvo 82.10 % frente al 96.5 % de Naive Bayes. Por último, la tercera clasificación, aplicada a 4 algoritmos, presentaron los valores 98.38 % para Decision Tree Based y 99.27 % para el algoritmo de su propuesta, con lo cual, garantizan una clasificación precisa.

3. Proceso de clasificación propuesto

Para la correcta interpretación de un fenómeno o hecho, los datos que se obtienen de la observación y posteriormente de las pruebas, deben apegarse a métodos de investigación que sigan un modelo de investigación, similar a emplear el método científico. Es recomendable trazar un plan de seguimiento que contemple las fases a desarrollar para la solución de un problema, saber qué hacer y cuándo hacerlo implica conocer los elementos y el orden de su participación. Los modelos ayudan a conocer el orden de las fases que componen un procedimiento. En esta investigación, el desarrollo de un modelo mejora la comprensión de los pasos a realizar, las fases que se derivan a lo largo de este trabajo se describen con más detalle en las secciones de este capítulo. Este modelo puede ser aplicado en casos similares, ya que proporciona un panorama sobre lo que tiene que efectuarse durante un proceso de distinción de tráfico anómalo que refleje una posible intrusión realizada o de un tráfico normal de la red, que es el principio funcional de un IDS. El proceso presentado puede ser referencia para otras investigaciones, ya que muestra de modo sintetizado y claro las etapas a realizar para el desarrollo de un sistema clasificador del flujo de datos obtenidos en una red, con ello se logra presentar un panorama general que evite perderse en los procesos y elementos que deben usarse, describiendo las fases sin profundizar en la elección de los posibles elementos utilizados ya que estos pueden variar. Los proyectos basados en una metodología bien cimentada y estructurada facilitan la ejecución de las tareas al proporcionar un orden y seguimiento de las etapas, permite ubicar los elementos, realizar una retroalimentación y saber si la etapa concluyó satisfactoriamente. Este proceso comprende ocho estados: diseño de la red, captura de datos, análisis de la

lectura, elección de los atributos, generación del vector, tratamiento de los datos, clasificación y generación de resultados. La Fig. 3.1 presenta el diagrama sobre las fases o pasos que componen el proceso propuesto. Los pasos se describen a detalle en la siguiente sección.

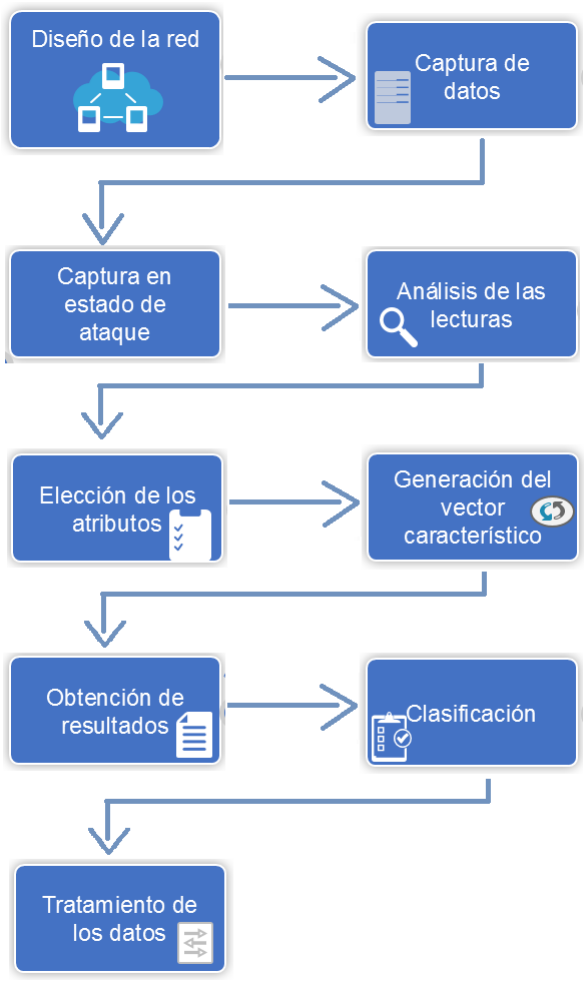

Figura 3.1: Diagrama del proceso de clasificación propuesto [Fuente propia].

3.1. Diseño de red

El diseño de la red es el primer paso que realizar, parte de las necesidades y elementos que debe cubrir una red con determinados servicios. Un diseño básico de red interconectada cuenta con cierto número de host, un switch que los comunique entre

sí, dividiendo la red en distintos segmentos y un ruteador que establezca conexión en los segmentos que forman la red. En cuanto a la seguridad, es recomendable que el diseñador de la red y encargado de la configuración de la misma, establezcan desde un inicio medidas básicas de protección como lo son, empleo de cortafuegos, uso de Vlans que permitan un mejor control de la administración de la red, deshabilitar puertos que no se usen, entre otras medidas. Complementariamente, la elección de la topología de red debe estar definida, antes de comenzar a configurar o añadir cualquier elemento a una red, Existen varios tipos de topología y se mencionan brevemente a continuación, de igual forma, se describen otros aspectos fundamentales para la construcción del escenario de red como es la configuración de los equipos, la configuración del servidor y del atacante.

- Topología de la red: existen varios tipos de topología como la de anillo que consta de un canal que los interconecta formando un anillo, de bus donde todos los nodos se conectan al mismo canal, de anillo doble que consta de dos canales conectados en anillos para proporcionar redundancia, estrella donde todos los elementos se encuentran interconectados a un punto central de la red o de malla donde todos los equipos están entrelazados entre sí. Para los fines de este trabajo la topología más eficiente es la de estrella.

- Configuración de los equipos: este punto comprende la interconexión entre los equipos. El primer paso que realizar es la configuración del router hacia los switches. La configuración entre estos contiene la habilitación de los puertos que se conectan de los switches al router, la configuración de los segmentos e IP que tendrán los equipos, lo mismo para los equipos host donde es necesario configurar una IP del equipo, habilitar la IP de default Gateway y la habilitación del nodo.

- Configuración de los protocolos de red: esta parte se enfoca en los servicios que posee la red. Es el conocimiento de los que maneja la red para brindar tareas o servicios y de los que son necesario proteger. Lo que incluye el conocimiento de

técnicas de protección y uso de estos protocolos.

- Configuración del servidor: este elemento se describe por separado para distinguirlo de los equipos de red y de los host debido a que generalmente, es elemento que proteger; la configuración de este elemento es referente al servidor web que tiene y las páginas web que tienen salida gracias al servidor configurado.

- Configuración del atacante: este elemento se incluye cuando sea un ambiente controlado, para generar pruebas y conocimiento, sobre el funcionamiento de un sistema de detección de intrusiones, es decir en un escenario con fines didácticos y no en un ambiente real, pues en redes reales la configuración de un equipo atacante no pertenece a los administradores de la red.

La configuración específica de los equipos, así como, la descripción detallada de como efectuar los ataques de tipo DoS son omitidos en este trabajo debido a que no es de interés académico mostrar los pasos para atacar una red.

3.2. Captura de datos en estado normal de la red

Esta etapa del proyecto se realiza una vez que se prueba la conexión de los equipos y los servicios configurados en la misma. El estado definido como normal, se refiere al tráfico que contiene peticiones y procesos válidos y permitidos en la red. Es el conjunto de procesos autorizados para los que fue diseñada la red. Toda tarea ejecutada en la red, que no tenga permisos de ser ejecutada no debe contemplarse al tomar las capturas de este tipo de tráfico, ya que podría afectar los resultados del análisis provocando posibles omisiones del comportamiento real de la red durante una transmisión si anomalías. Es necesario contar con un ejemplo de flujo ideal, para saber qué es lo que se tiene que transmitir para dejar a todo lo fuera de este tipo de flujo como un flujo que muestre una anomalía o intento de intrusión en la red. Para realizar una captura de los datos que se transmiten en la red se determina la herramienta para tomar las lecturas, se puede recurrir a algunas aplicaciones que funcionen como escucha, entre ellas

se encuentran Wireshark y Tcpdump. Estas aplicaciones pueden ser propietarias o de dominio público, con ellas es posible determinar el tipo de información que circula por la red y el impacto que puede llegar a tener sobre la misma red y la capa de donde se toma la lectura. Existen aplicaciones que logran capturar lecturas de capa 2 y 3(enlace de datos y de red). Es importante mencionar que, el análisis del tráfico de red se basa habitualmente en la utilización de una escucha con interfaz ethernet conectadas al bus o a un punto del switch. Dicha escucha, con su interfaz ethernet (puerto) funcionando en modo promiscuo, captura el tráfico a analizar. Los puntos por considerar, durante la captura del tráfico en redes de área local, son la cantidad de información promedio que se transfiere, a través del canal de comunicación y la velocidad de transferencia ya que son indicadores importantes para evaluar la eficiencia en la red. El desempeño de la red se caracterizó utilizando los siguientes parámetros:

1. Cantidad de tráfico: cantidad de información promedio que se transfiere a través del canal de comunicación.

2. Tasa de transferencia: velocidad de transmisión que pasa por una línea de telecomunicación.

3. Porcentaje de utilización: relación entre de tráfico medido al tráfico máximo que el puerto puede administrar.

Los paquetes capturados pertenecen a las capas 2 y 3 del modelo TCP en su mayoría, ya que son el tipo de tráfico que capturan usualmente las aplicaciones. Se pueden utilizar comandos del sistema operativo que ofrecen información sobre el equipo y los puertos que abre para la comunicación con otros dispositivos.

3.3. Captura de datos en estado de ataque

De la misma forma que en la captura de flujo normal de la red, durante esta etapa, se emplea una aplicación de escucha o monitoreo de la red, empleando un puerto, que

sirve de escucha, en modo promiscuo. Se establece un origen y destino del proceso de comunicación. Por ejemplo, el equipo de origen es el equipo atacante y el equipo destino es el servidor web, o equipo a proteger. Para descartar información innecesaria, se recomienda tomar las lecturas al momento de generar el ataque. Se puede recurrir a la ejecución de pruebas iniciales que permitan familiarizarse con las actividades a realizar para ejecutar el ataque, a fin de agilizar el proceso de ejecución y con ello disminuir el tamaño de las muestras. Existen muchos detalles al momento de configurar y reproducir las pruebas para la toma de lecturas por lo que se recomienda hacer uso de la herramienta de bitácora a fin de llevar orden de esos detalles que pueden dar razón a los detalles encontrados en el tráfico analizado. No es recomendable hacer la toma de todas las lecturas en un mismo día o misma hora, hacerlo podría generar perder detalles que pueden ser más claros con una toma registrada en circunstancias diferentes. Dependiendo del tipo de ataque que se esté estudiando y del cual se generan las reproducciones para analizarlo, se pueden emplear aplicaciones o sistemas operativos que ayuden a generar las acciones necesarias o bien que puedan brindar un análisis complementario del ataque. Kali es un tipo de sistema operativo que brinda una gama de herramientas para poder reproducir y apoyar en el análisis de distintos tipos de ataques.

3.4. Análisis de las lecturas

Una vez obtenidas las lecturas de la red es necesario analizarlas, en busca de características que cualifiquen el comportamiento de la red y a su vez, que todas esas características puedan cuantificarse para poder optimizar el análisis. La búsqueda de esta información puede ser de tres tipos:

1. *Búsqueda completa:* logra encontrar los atributos mediante el análisis exhaustivo de todos los elementos, lo que implica realizar una búsqueda exhaustiva.

2. *Búsqueda secuencial:* genera subconjuntos de atributos que buscan ser seleccionados como mejores atributos, se parte de un vector vacío al que van agre-

gándole las características evaluadas como relevantes o bien puede partir de un vector lleno de todas las características que van siendo evaluadas a fin de determinar si son relevantes o no a fin de permanecer o ser eliminadas.

3. *Búsqueda aleatoria:* crea los subconjuntos de atributos seleccionados aleatoriamente, estos atributos permanecen o se evalúan en base a una evaluación efectuada también de modo aleatorio

Sea cual sea el método de análisis de la información que se realice, esta será sometida a una evaluación posterior una vez que se hayan determinado los atributos finales. Por tanto, los puntos relevantes encontrados en esta etapa, mismos que se desarrollan en las siguientes etapas son: Determinar el tipo de selección de características (filtro de atributos y formación del vector característico), tipo de Discretización (tratamiento de los datos), tipo de algoritmo (algoritmos aplicados en el proceso de clasificación)

3.5. Elección de los atributos

La ejecución de esta fase es determinante para el proyecto, de esta depende que el sistema tenga la facultad de clasificar correctamente. Poder mejorar el rendimiento del sistema entre otros beneficios. Su ejecución da la pauta para continuar el proceso y es una de las etapas en las que generalmente se invierte más tiempo buscando métodos y herramientas que ayudaran a completarla. Como su nombre lo indica es la fase en la que se lleva a cabo la elección de atributos que representan las cabeceras de la base de conocimiento. Se recomienda la selección de atributos que generen un subconjunto óptimo de elementos con características determinantes, en una base de datos que logre disminuir la dimensión de la muestra, eliminar el ruido (representado por datos irrelevantes que no afectan ni modifican la salida) y mejorar el desempeño del algoritmo de aprendizaje, es decir, deben evitarse atributos redundantes o irrelevantes. La selección de las características relevantes permite no consumir recursos

innecesarios. Se basa en una fase de búsqueda y en una fase de evaluación de los datos seleccionados. El proceso de selección de atributos concluye al encontrar un conjunto de atributos eficiente con el estándar de comparación impuesto inicialmente. Para la elección de los atributos es necesario un análisis independientemente ya que las lecturas tomadas arrojan comportamientos distintos en cada escenario. Por ello, se recomienda la creación de un vector característico a fin de comparar los datos correspondientes de flujo normal y de ataque. Durante esta fase, además de la elección de los atributos del vector característico, se emplea la discretización, que consiste en convertir en valores numéricos la información que se maneja, esto se describe más a detalle en la sección de tratamiento de datos. El tipo de información es sometida a un proceso de aprendizaje supervisado al conocer las clases que componen al vector, en el estudio de ataques a una red las clases posibles serian el flujo normal y el flujo de ataque que se analice, aun si son varios ataques por analizar se recomienda dividirlos por escenarios donde puedan tomarse lecturas del flujo permitido y de un flujo de ataque.

3.6. Generación del vector característico

Generalmente el proceso de Selección de atributos finaliza cuando es alcanzado el umbral que se ha establecido, determinándolo como el óptimo, tras una búsqueda completa, encontrando el subconjunto de atributos óptimo. Esta fase comprende el llenado de la base de conocimiento con los datos de las capturas tomadas extrayendo únicamente los datos correspondientes a los atributos determinados. El producto de esta fase es la base de conocimiento que contiene los datos relacionados con los atributos y las posibles clases a las que puedan pertenecer.

3.7. Tratamiento de los datos

Para el análisis de los datos es vital el proceso de tratamiento de datos que puede optimizar las características de la base de conocimiento o conjunto de todos los datos muestrales. Empleando métodos que permitan compactar la muestra a fin de no desperdiciar recursos en el análisis de datos irrelevantes. La posible transformación de los datos permite la aplicación de procedimientos que proporcionen un proceso y un resultado más certero. Dentro de los métodos de tratamiento de datos la normalización puede encontrar su partición, es un proceso que puede realizarse en distintas fases del desarrollo del Sistema de Detección de Intrusos. Los procesos de normalización hacen que los datos difieran sobre todo en aplicaciones lineales, polinomiales y gaussianas. La normalización de datos está relacionada con la mejora de rendimiento y el mejoramiento de convergencia cuando se entrenan los clasificadores. Otro método de tratamiento de datos es la discretización de los datos, que es el proceso de convertir un valor continuo o nominal en un valor numérico. Los tipos de valores que pueden ser tratados en proceso de discretización son los expuestos a continuación *cuantitativas* que pueden ser discretas (tienen un número finito o contable de valores, en general números enteros o binarios) y continuas (atributo continuo con valores finitos, representado por números reales o de punto flotante) *cualitativas o categóricas* estas pueden ser nominales(sin orden, nombran al elemento que refieren) y ordinales(tienen un orden que puede establecer orden en los valores)

La discretización permite el análisis de los tipos de datos mencionados anteriormente y su posible transformación de valores continuos a discretos. La discretización permite la construcción de modelos de clasificación más compactos y sencillos, además de mejorar la precisión del clasificador y optimizar la fase de aprendizaje en el proceso de clasificación. La base de conocimiento se construye a partir de la transformación de las lecturas obtenidas en Weka a un formato de texto plano sometido al procedimiento de selección de atributos para la construcción del vector característico que es el conjunto de cabeceras(atributos) y datos; es en esta fase, donde se realiza

el método de discretización al transformar datos cualitativos a discretos, estos datos, son contenido en un archivo Excel para su posterior proceso de normalización, para lograr la obtención de valores continuos comprendidos entre el cero y el uno.

3.8. Clasificación

La correcta interpretación de datos contenidos en extensas bases de datos se basa en el análisis óptimo de las características presentes en los datos. La minería de datos es un área que se especializa en encontrar las características y relaciones contenidas en la información analizada. Dentro de la minería de datos se estudian tres elementos: el modelo, el criterio de preferencia y el algoritmo de búsqueda. *El modelo* posee dos factores importantes, la función que determina la acción a realizar sobre los datos y la interpretación del conocimiento. *El criterio de preferencia* es la base que determina los atributos del vector característico. *El algoritmo de búsqueda* es el método procedimental que se ejecuta para la obtención de los atributos. Las funciones de los modelos usados con más frecuencia en la minería de datos son:

- **Clasificación:** método que asigna una etiqueta o clase a una muestra. Para asignar esta etiqueta, el proceso se basa en categorías predefinidas en la fase de aprendizaje. Los algoritmos de clasificación pueden basarse en operaciones de tipo discriminación lineal, árbol de decisión o reglas y estimación densidad.
- **Regresión:** clasifica un caso o una variable predictora y evalúa una busca crear una condición predictiva sobre las futuras muestras.
- **Clustering:** asigna las clases en base a la agrupación de datos evaluándolas por similaridad, densidad de probabilidad o distancia.
- **Sumarización:** proporciona una muestra evaluada en combinación de operaciones combinadas como la media, desviación estándar entre otras.

- **Modelado de Sistemas:** describe las relaciones y dependencias entre las variables basadas en conexiones cuantitativas o cualitativas obtenidas mediante la aplicación de reglas asociativas.
- **Análisis de conexiones:** se basa en la búsqueda de las relaciones y dependencias de los atributos y no de los datos.
- **Análisis de secuencias:** modela patrones basados en variaciones de tiempo, busca la relación entre variables y el tiempo, dando énfasis a las modificaciones de las variables a través del tiempo.

En el diseño de Sistemas de Detección de Intrusiones las funciones más utilizadas son clasificación, clustering, modelado de secuencia o análisis de secuencias. En esta investigación la función elegida para ser empleada es la clasificación.

3.9. Obtención de resultados

Durante esta fase se obtienen los resultados, se originan tras el proceso de clasificado, los datos arrojados por cada uno de los algoritmos de filtrado brindan ponderadores de eficiencia en el proceso de clasificación mediante una matriz con datos sobre el tamaño de la muestra, la cantidad de datos, la eficiencia lograda, el tiempo de ejecución de cada algoritmo de clasificación por ser un método que asigna etiquetas, basándose en las categorías predefinidas durante la fase de aprendizaje, además de que en su ejecución se recomienda el uso de algoritmos de discriminación lineal o árboles de decisión o reglas, los cuales, son los recomendados para el tratamiento de datos que no muestran relaciones entre sí o que no son dependientes. Al analizar las salidas de los procesos de clasificación se encuentra que el existen algoritmos que obtienen mejores resultados, de ahí la importancia sobre la buena elección de los algoritmos aplicables en la fase de clasificación, que brindan durante los resultados un panorama sobre su uso.

4. Escenarios propuestos

La teoría llevada a la práctica permite la comprobación de la hipótesis, así en un proceso de investigación una fase importante y determinante, es la fase experimental donde a través de la ejecución de pruebas o laboratorios se puede comprobar lo que inicialmente se planteó, en esta investigación permitirá obtener los datos muestrales necesarios para entender el fenómeno estudiado y desarrollar la fase de observación del escenario para posteriormente analizarlo.

El trabajo realizado se divide en dos casos de estudio para los cuales se describen las condiciones físicas en que se desarrollaron, las características que presentan y que dan como resultado un conjunto de datos obtenidos, que posteriormente son analizados para identificar los atributos que representen el flujo de datos que se presenta en las consultas realizadas y en las réplicas de ataques realizadas a la red. Estos atributos conforman el vector característico que contiene los datos que serán sometidos al proceso de clasificación. Todas las acciones ejecutadas siguiendo el orden determinado en el proceso propuesto.

4.1. Escenario 1

Los IDS de red protegen a un conjunto de computadoras, de usuarios no autorizados, incluyendo posiblemente al personal interno. Durante la fase de aprendizaje, el IDS construye un modelo predictivo (es decir, un clasificador) capaz de distinguir entre conexiones normales y las conexiones anormales, llamadas intrusiones o ataques. De toda la información obtenida por el analizador de tráfico, se aplicaron filtros para omitir

información irrelevante para el detector de intrusiones.

En el escenario propuesto se proporcionó un conjunto estándar de datos a ser auditados, obtenidos del proceso de filtrado ejecutado con la herramienta Wireshark durante peticiones normales a la red y posteriormente en escenario de ataque.

El ataque aplicado a la red definida en el escenario uno, es denegación de servicio (DoS), el cual busca la interrupción del flujo de datos y reduce la disponibilidad que otorga un servicio activo. El modo en que opera consiste en enviar paquetes con formato permitido en grandes cantidades para lograr la saturación del servidor, de tal manera que ya no le sea posible atender las solicitudes. Para lograr la saturación del servidor web se transmitieron paquetes del protocolo ICMP con carga elevada. Es importante mencionar que, el interés en este artículo es detallar una solución al ataque de DoS por inundación y no describir cómo se efectúa.

4.1.1. Desarrollo del proceso propuesto en el escenario 1

En esta sección se describen las actividades desarrolladas en cada fase que forma el proceso propuesto por esta tesis para llevar acabo el desarrollo del vector característico y así ejecutar un proceso de clasificación que muestre los resultados obtenidos por el flujo de datos de la red, sometidos a el proceso de clasificación a fin de concluir si se trata de un flujo de datos de ataque o de un flujo de datos normal de la red.

1. **Diseño de la red**. La arquitectura del escenario de la red es en esencia la misma para ambos casos de estudio. Consiste en una red constituida por dos segmentos de red que están interconectadas por medio de un ruteador y posteriormente, por un dispositivo switch que se encarga de administrar y redistribuir el tráfico de red hacia los hosts que se interconectan al switch. Entre ellos se encuentra un servidor web, desarrollado en php y tiene como sistema operativo a Ubuntu V14 de Linux. Las computadoras se conectan al equipo switch a fin de lograr la comunicación entre estos.

   El sistema operativo en los hosts es Windows, mientras que el sistema operativo

de red (IOS) del router y switch son C1841-ADVIPSERVICESK9-M VERSIÓN 12.4 y C295-I6Q4L2-M, versión 12.1 respectivamente. En tanto el equipo que será adversario cuenta con un sistema operativo Kali de Linux, este sistema operativo contiene características importantes que permiten emplearlo para ejecutar los ataques de DoS que se analizan en esta investigación.

Durante esta fase se configuran los servicios de con los que contará el escenario para el escenario uno y dos se requiere la configuración del servicio DHCP para generar una consulta web. La configuración de los protocolos de enrutamiento en este caso el protocolo empleado es EIGRP tomando en cuenta que la red pudiera conectarse a redes externas mediante un proveedor de servicio o la configuración del servicio DHCP para que la asignación de IP a los equipos sea de forma automática.

La Fig. 4.1 muestra el escenario definido para el análisis de tráfico. Como puede observarse, consiste en dos redes conectadas a través de un ruteador. La red 1 se compone de un servidor web, el cual será atacado, y varios hosts, mientras que la red 2 contiene por lo menos una computadora que fungirá como adversario.

El escenario se construyó físicamente en las instalaciones del Centro Universitario Valle de Chalco, Edificio D, Laboratorio de redes. Las Tablas 4.1 y 4.2 muestran una descripción de los elementos que conforman el escenario de red empleado, en software y hardware, respectivamente.

Para determinar el correcto funcionamiento del escenario, una de las pruebas básicas es comprobar conexión entre los equipos mediante la prueba de ping, otra forma es ingresar el comando *tracert* para comprobar la ruta que recorre el tráfico emitido de un origen al destino. Se habilitaron únicamente los puertos utilizados, en el switch se configura un puerto en modo promiscuo.

2. **Captura de datos en estado normal de la red.** La información resultante de los laboratorios desarrollados para el estudio del caso uno, se compone de capturas

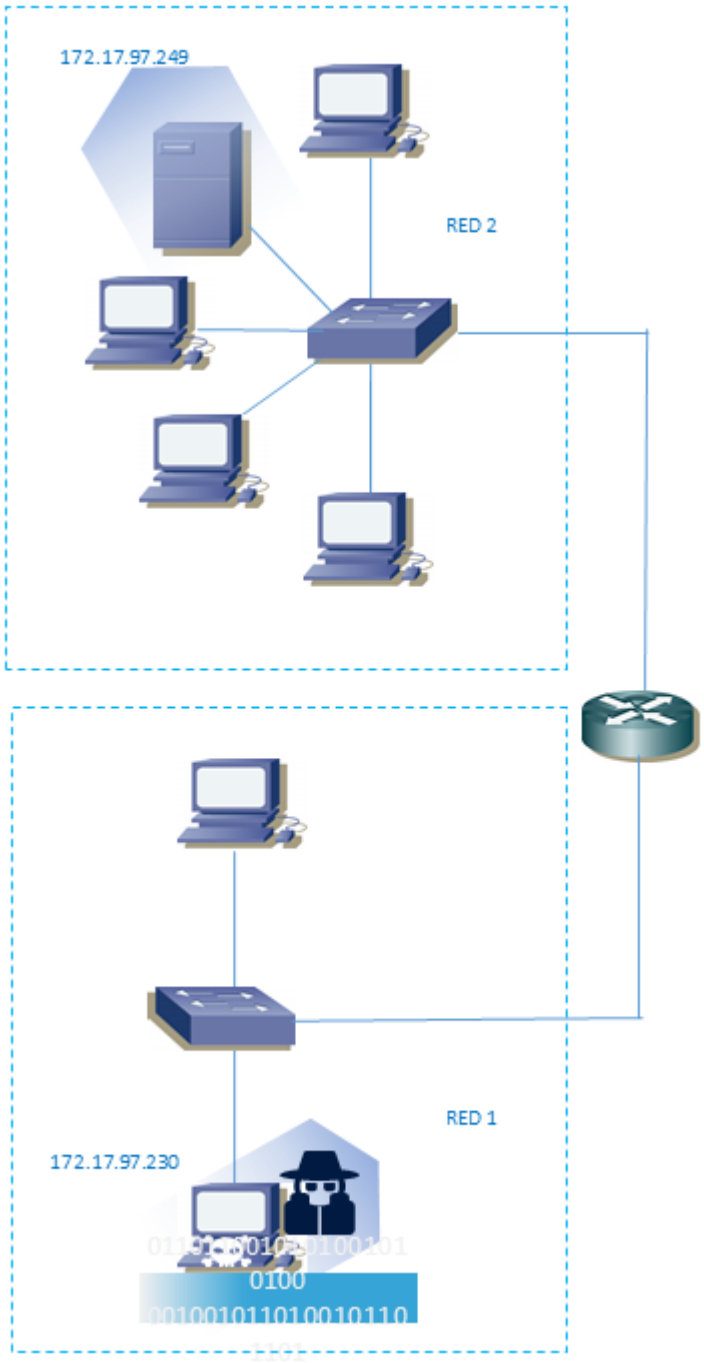

Figura 4.1: Escenario de red1 [Fuente propia].

Tabla 4.1: Herramientas de software utilizadas [Fuente propia].

Nombre	Descripción	Uso
Linux Ubuntu	SO V14, 60GB DD, 8GB RAM, AMD64	Contiene al servidor
Linux Kali	SO V10, 60GB DD, 8GB RAM, AMDD64	Para efectuar ataques
Windows	V7, 1T DD, 12 RAM, AMD64	Contiene al virtualizador
Windows	V10, 1T DD, 12 RAM, AMD64	Contiene al virtualizador
VMWare	V12, 64b	Virtualizador
Wireshark	V2.2.1, 64b	Analizador de tráfico
Xap	V3.2.2, 64b	Servidor de servicios (Apache, MYSQL)

Tabla 4.2: Herramientas de hardware utilizadas [Fuente propia].

Nombre	Descripción	Uso
PC	HP, AMD64, 1T, 12RAM	Servidor
PC	HP, AMD64, 1T, 12RAM	Atacante
PC	HP, AMD32, 1T, 4RAM	Host de red
Router	2901 CISCO 2P GE, 2HWIC	Gateway
Switch	2600 CISCO 48P	Puente
Modem	Linksys 6P	Servidor de Internet

que pertenecen a peticiones de tipo TCP/IP, del tipo DNS al solicitar la página e ICMP al enviar un ping, obtenida a través de Wireshark.

Existen varias aplicaciones para capturar el tráfico de una red. En este trabajo, es utilizada la aplicación Wireshark 2.2.7, conocido inicialmente como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica. Añade una interfaz gráfica y muchas opciones de organización y filtrado de información, estableciendo la configuración del puerto escucha en modo promiscuo. También incluye una versión basada en texto llamada *tshark*. Permite examinar datos de una red en tiempo real o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete.

Wireshark incluye un lenguaje completo para filtrar lo que se quiera ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP. Soporta más de 110 protocolos que pueden manejarse por medio de los filtros. Es software libre, y se ejecuta sobre la mayoría de sistemas operativos Unix y compatibles, incluyendo Linux, Solaris, FreeBSD, NetBSD, OpenBSD, Android, y Mac OS X, así como en Microsoft Windows.

El tráfico analizado consta de dos tipos, uno con una transmisión normal y otro bajo ataque por inundación de paquetes. La escucha fue considerada en un lapso de 5 segundos y para 45 lecturas de cada caso. El tamaño de las muestras obtenidas va de 1KB a los 16KB. Se obtienen datos de volcado TCP sin procesar para una red de área local (LAN) que simula una típica LAN, de igual forma, los datos con múltiples ataques.

Tomando en cuenta que el tráfico en redes de área local se mide como la cantidad de información promedio que se transfiere, a través del canal de comunicación. La velocidad es un indicador importante para evaluar la eficiencia en la red. El desempeño de la red se caracterizó utilizando los siguientes parámetros:

a) Cantidad de tráfico: cantidad de información promedio que se transfiere a través del canal de comunicación.

b) Tasa de transferencia: velocidad de transmisión que pasa por una línea de telecomunicación.

c) Porcentaje de utilización: relación entre de tráfico medido al tráfico máximo que el puerto puede administrar.

Los datos se obtienen en tiempos bien definidos, para homogenizar se estableció un rango de 5 segundos aproximadamente, donde los datos fluyen hacia y desde una dirección IP de origen a una dirección IP de destino bajo algún protocolo, como lo son TCP, ICMP, entre otros. Con la finalidad de proporcionar una perspectiva general de los datos capturados, la trama del tipo TCP/IP y el tamaño que esta tiene puede observarse en la Fig. 4.2.

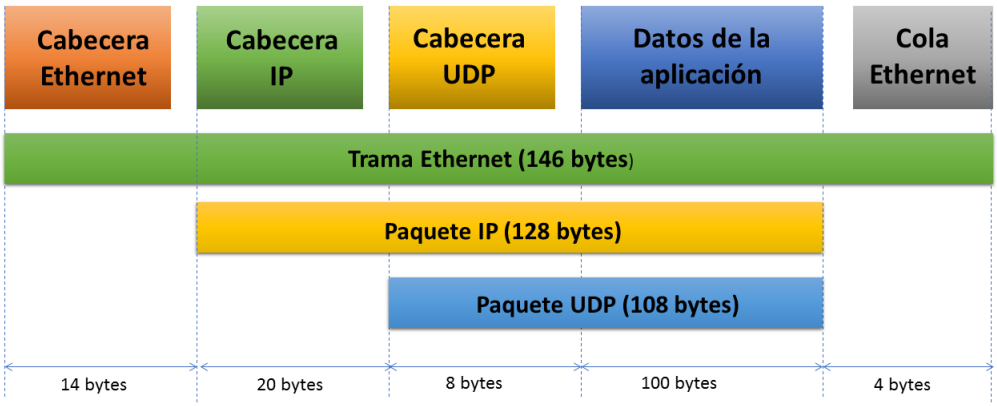

Figura 4.2: Trama TCP/IP [Fuente Propia]

Los datos capturados con la aplicación Wireshark poseen un formato que debe ser tratado para emplearlos correctamente en las siguientes fases del proceso. La Fig. 4.3 muestra la exportación del formato manejado por Wireshark a texto plano para su tratamiento y análisis.

3. **Captura de datos en estado de ataque.** La captura de datos requiere ser obtenida al momento exacto de la ejecución del ataque y detenerse al finalizar el

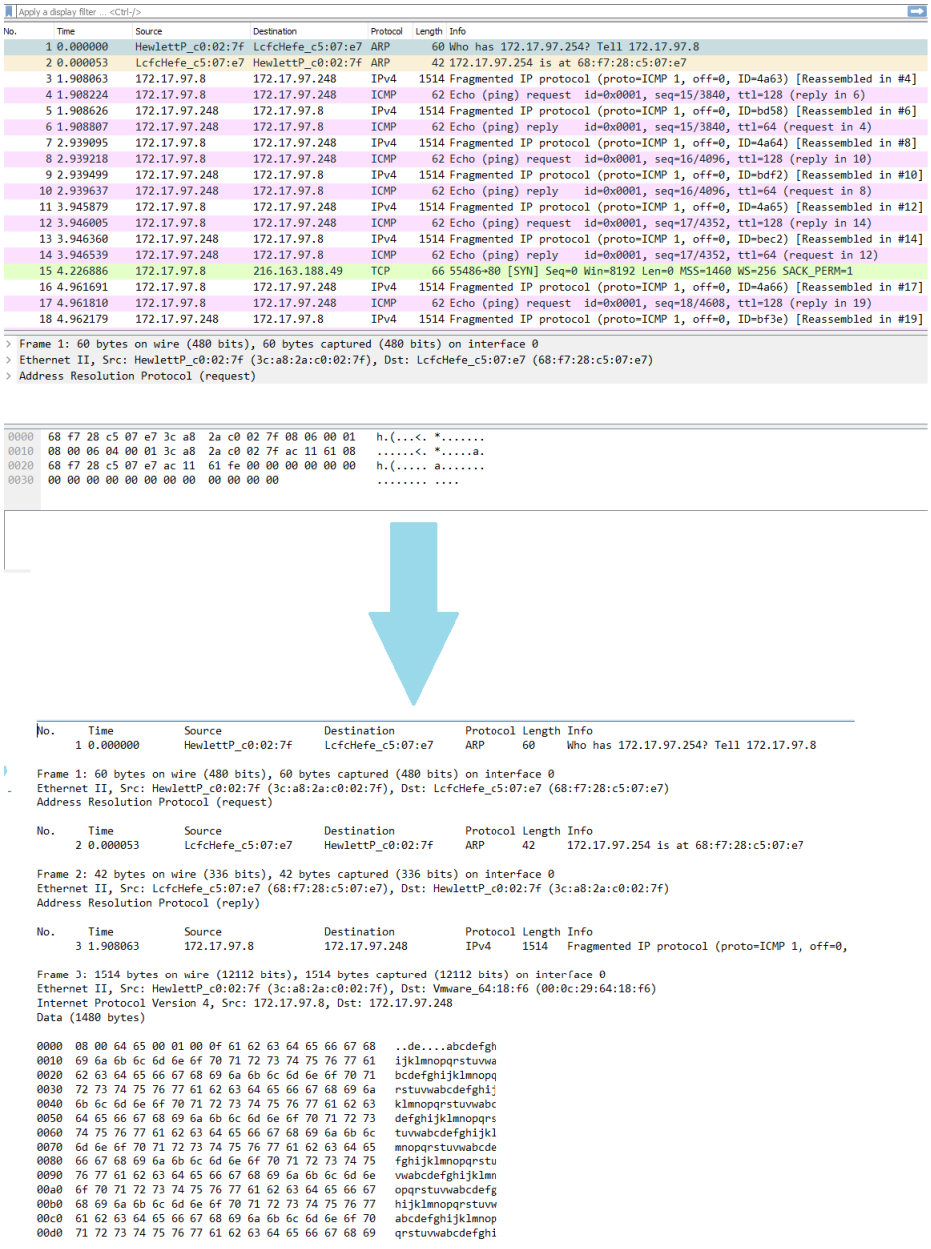

Figura 4.3: Datos capturados y exportados a texto plano [Fuente propia].

efecto de ataque en la red. La información obtenida durante esta fase puede ser muy grande dependiendo del tiempo que se requiera para efectuar el ataque y el tiempo en que tarde en recuperarse la red una vez que el ataque concluye. El tamaño de las muestras va de 3 KB a 138KB. Para minorizar los tiempos es necesario tener dominio de la ejecución del ataque, además es recomendable que de las capturas obtenidas se obtengan muestras aleatorias de distintos momentos del ataque y de las variaciones que este pudiera presentar al realizarlo.

En el escenario uno, al realizar el ataque por inundación de ping, una de las variantes que podía presentarse es el tamaño de los paquetes empleados, la velocidad de los paquetes a transmitir entre otros. Es enriquecedor para la base de información obtenida contar con datos muestrales que contengan estas variaciones. Todos los laboratorios efectuados se registraron adicionalmente con la herramienta de bitácora

El tamaño de los archivos seleccionados fue definido por el rango de duración del ataque, acotado a 5 segundos. Estos datos tomados por Wireshark son obtenidos del servidor y no del equipo adversario, desde el puerto que funciona en modo promiscuo o de escucha.

Para efectuar el envío de una gran carga en un ping desde un solo equipo, Kali permite generar ráfagas de ping que ponen al servidor web fuera de servicio, con paquetes con carga de 10 000 Megas. Sin embargo, para este escenario las peticiones fueron realizadas desde un equipo perteneciente a la red, pero de otro segmento de red, con un sistema operativo Windows 7, este host, envía al servidor web, pines con carga con diferentes tamaños a fin de corroborar el tamaño necesario para dejar fuera al servidor web. Las muestras contemplan tiempos de 2 a 3 minutos que es el promedio de tiempo que dura la generación del ataque y la recuperación de los servicios en la red. De igual forma, una vez construida la base con estos datos, se le brinda un cambio de formato para ser empleados en las siguientes fases.

4. **Análisis de la lectura.** Una vez que se tienen las lecturas del tráfico de red a estudiar, se efectúa un análisis para lograr identificar las características de una petición normal hacia el servidor y de una petición maliciosa hacia el servidor. El análisis se ve beneficiado con algunas cualidades de la aplicación Wireshark, como el empleo de filtros a fin de que se muestren datos de modo sintetizados referentes a un punto a analizar, filtrar basándose en una dirección IP que envía tráfico a una dirección destino, síntesis en base a un protocolo utilizado, entre otros.

   El análisis de estos datos determina los atributos que formaran el vector característico y esta cimentada en la elección de características que logren diferenciar los dos tipos de tráfico posible, el normal y el anómalo. Cada registro de conexión consta de aproximadamente 7.5 kb hasta 2.5 Megas bytes. Con esto se obtienen dos clases posibles c0 y c1 con los que se entrena la red, aplicando los distintos algoritmos empleados en este trabajo, los registros de prueba obtienen la clase durante la fase de prueba. Con el proceso de análisis de datos empleado, se obtienen datos contenidos en un archivo de Excel que es el principio de la construcción de la base de conocimiento.

   Para hacer posible la correcta interpretación de los datos, es necesario dar tratamiento al formato en que los datos son obtenidos, Wireshark convierte los datos a un texto plano con el que es posible trabajar con un analizador léxico para buscar relaciones y presencia de determinadas características definidas como clave para dar explicación al comportamiento que presenta la red. Estas acciones dan origen a la siguiente fase del proceso.

5. **Elección de los atributos.** Esta es una fase determinante en el proceso, el éxito o fracaso de la correcta clasificación del IDS dependen de una buena elección de atributos. Una de las ideas que se retoma de los trabajos revisados en el estado del arte consiste en la disminución de los atributos a fin de mejorar la

fase de entrenamiento y mejorar el rendimiento del algoritmo de clasificación. En este trabajo los datos fueron sometidos al análisis del flujo de datos de la red y el análisis con ayuda del analizador léxico.

Una vez obtenidos los resultados estadísticos del análisis efectuado con el analizador léxico, se procede a vaciar dichos datos en la base que formará la base de conocimiento con que cuente el IDS. Esta base de datos está elaborada en Excel. Los patrones de estudio que conforman la lista de atributos a la que se logró llegar y que son característicos de los fenómenos estudiados forman las cabeceras de los vectores característicos que contienen la información seleccionada, misma que representa los atributos que constituyen el vector característico, mismo que debe hallarse en las muestras de flujo tomadas. Comprende 18 atributos de los cuales los primeros ocho atributos corresponden a datos físicos de la red y los 10 restantes de los datos obtenidos del proceso de filtrado y del análisis de los datos obtenidos.

La tabla4.3 enlista los atributos empleados, dando su descripción y el tipo de valor posiblemente asignado: continuo para valores constantes y discreto para valores aleatorios. Los datos que corresponden a la red como tal se enfocan a la dirección ip de destino y origen, además de incluir el tiempo que se toma de la muestra, los datos restantes se desprenden de las características encontradas entre los datos.

La lista de los atributos definidas se convierte a las variables a0 a a17 y es el modo en que se identificaran en el vector característico con formato inicial de Excel para su posterior normalización y exportación de formato nuevamente a fin de poseer la extensión *.cvs* que maneja Weka. Llegar a la obtención del vector característico fue una de las fases que requirió más tiempo invertido, por el análisis y determinación del proceso de tratamiento para los datos a fin de determinar las relaciones y características entre los datos.

6. **Generación del vector característico.** De la fase anterior se tendrán definidos

Tabla 4.3: Lista de atributos, escenario 1 [Fuente propia].

Atributo	Descripción	Tipo
a0	Ip origen	continuo
a1	Ip origen	continuo
a2	Ip origen	continuo
a3	Ip origen	continuo
a4	Ip destino	continuo
a5	Ip destino	continuo
a6	Ip destino	continuo
a7	Ip destino	continuo
a8	Tiempo de transmisión	continuo
a9	Patrones totales	discreto
a10	Número de patrones distintos	discreto
a11	Densidad léxica	discreto
a12	Total de sentencias	discreto
a13	Longitud promedio comando	discreto
a14	Longitud máxima de comando	continuo
a15	Longitud mínima de comando	continuo
a16	Legibilidad1	discreto
a17	Legibilidad2	discreto

los atributos, estos formarán las cabeceras del vector característico, posteriormente, se procede al filtrado y obtención de los campos que fueron definidos como incluyentes durante la transmisión de los datos. El caso uno se compone de dos vertientes, la primera incluye al flujo obtenido en lo que hemos determinado como flujo normal de la red, por otro lado, en la segunda vertiente se tendrá un flujo perteneciente al momento de efectuar el ataque.

Los dos vectores característicos pertenecen a una clase cada uno $c0$ y $c1$, donde, $c0$ representa un flujo normal y $c1$ un flujo clasificado como sospechoso o de ataque. Los dos vectores se unen en un solo vector resultante, que comprende 18 atributos. Los atributos son representados con las variables *a0* a *a17*.

El producto final de esta fase es el archivo Excel que contiene todos los datos de la selección y que si pertenecen a los atributos definidos. Debemos recordar que en este escenario el ataque es denegación de servicio del servidor web producida por las ráfagas de ping enviadas de un equipo atacante que logra conectarse a la red.

7. **Tratamiento de los datos.** Esta fase se hace presente en otras fases del proceso, pero concluye antes del proceso de clasificación. El primer tratamiento de datos es necesario tras la captura del flujo de datos de la red, cuando Wireshark realiza el primer cambio de formato para los datos obtenidos, en archivos de texto plano, posteriormente, se presenta un segundo tratamiento de datos con los datos obtenidos de los filtrados que se concentraron en un archivo Excel, el cual fue posteriormente normalizado con apoyo de las herramientas de programa, empleando del menú formulas la formula denominada Normalizar que tiene como base la operación que implica el empleo de los datos, la utilización de la media obtenida y de la desviación estándar calculada para así obtener un vector característico.

   Es importante que durante el tratamiento de datos se evite la perdida de funcionalidad del archivo o de datos. Para esta investigación se emplearon los métodos

de normalización de los datos y discretización al transformar los datos contenidos en la base de datos a valores finitos además de lograr trasformar datos cualitativos como la presencia de un protocolo o IP en datos cuantitativos. La base de datos posee datos cuantitativos y cualitativos, tratados para su conversión en datos discretos. Además, se realiza un cambio de datos discretos a continuas al transformarlos en números pertenecientes al rango de 0 a 1.

La importancia de esta fase ésta basada en la mejora de la base de conocimiento y es fundamental su realización a fin de que la realización de la siguiente fase que es la clasificación se vea mejorada y se pueda obtener un proceso más eficiente.

8. **Clasificación.** Esta fase consiste en el manejo de los datos mediante la aplicación de una acción, esta acción es la clasificación, que es un tipo de función perteneciente a la minería de datos, consiste en evaluar a qué clase de las existentes para el proceso, pertenece un conjunto de datos. Para llevarla acabó, se vale de los algoritmos de clasificación que han sido determinados. En esta investigación fueron determinados cinco algoritmos empleados en la minería de datos, para el entendimiento e interpretación de grandes cantidades de datos.

   La aplicación conocida como Weka fue empleada en esta fase, es una herramienta desarrollada en la universidad de Waikato, enfocada en el aprendizaje automático y la minería de datos, escrito en Java y distribuido en licencia GNU-GPL. Tiene portabilidad y su interfaz gráfica hace muy amigable su uso. Soporta realizar tareas como preprocesamiento de datos, clustering, clasificación, regresión, entre otras.

   Los ponderadores obtenidos al ejecutar los algoritmos clasificadores pueden ser analizados en base a su valor. El rasgo más importante para definir la eficiencia de los clasificadores está determinado por el porcentaje de clasificaciones correctas, clasificaciones incorrectas, tiempo de ejecución entre otros que hacen entender con cual se obtuvieron los mejores puntajes.

Los resultados obtenidos por los algoritmos clasificadores en Weka durante el proceso de clasificación arrojan varias variables y ponderadores dentro de los cuales resaltan, los referentes al grado de certeza que presenta el algoritmo.

El primer algoritmo que se muestra en la Fig. 4.4 es Naive Bayes, este algoritmo evalúa cada instancia por separado, la presencia o ausencia de una característica no se relaciona con la presencia o ausencia de otra característica. Logrando obtener ponderadores estadísticos como resultado.

```
Naive Bayes Classifier
Time taken to build model: 0.03 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          88       97.7778 %
Incorrectly Classified Instances         2        2.2222 %
Kappa statistic                          0.9555
Mean absolute error                      0.0222
Root mean squared error                  0.1491
Relative absolute error                  4.4438 %
Root relative squared error             29.7813 %
Total Number of Instances               90
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall  F-Measure
0.977     0.022     0.977       0.977   0.977
0.978     0.023     0.978       0.978   0.978
MCC       ROC Area  PRC Area  Class
0.956     0.979     0.934     c0
0.956     0.968     0.968     c1
Weighted Avg.
0.978     0.022     0.978       0.978   0.978
0.956     0.974     0.951
=== Confusion Matrix ===
  a  b   <-- classified as
 43  1 |  a = c0
  1 45 |  b = c1
```

Figura 4.4: Resultado del algoritmo Naive Bayes

El segundo algoritmo presentado en la Fig. 4.5 para el caso uno es el correspondiente a Red Neuronal del tipo Backpropagation de 3 capas. Este algoritmo muestra un orden de selección filtrando en cada capa las neuronas que resultan selectas de la capa anterior a fin de proporcionar un resultado filtrado en la última capa.

```
Backpropagation-red neuronal
Time taken to build model: 0.63 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          88       97.7778 %
Incorrectly Classified Instances         2        2.2222 %
Kappa statistic                          0.9555
Mean absolute error                      0.0235
Root mean squared error                  0.1346
Relative absolute error                  4.6944 %
Root relative squared error             26.8865 %
Total Number of Instances               90
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall  F-Measure
0.977     0.022     0.977       0.977   0.977
0.978     0.023     0.978       0.978   0.978
MCC      ROC Area  PRC Area  Class
0.956    0.996     0.995     c0
 0.956    0.996     0.996     c1
Weighted Avg.
0.978     0.022     0.978       0.978   0.978
0.956    0.996     0.996
=== Confusion Matrix ===
  a  b   <-- classified as
 43  1 |  a = c0
  1 45 |  b = c1
```

Figura 4.5: Resultado del algoritmo Red Neuronal

El tercer algoritmo clasificador del que se presentan resultados en la Fig. 4.6 es Decision Table, que presenta una tabla de correspondencias o relaciones, busca acomodar los datos en una clase existente, en el caso de que el registro no tenga una clase se asignará a la clase mayoritaria.

```
Desición table
Time taken to build model: 0.17 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          89       98.8889 %
Incorrectly Classified Instances         1        1.1111 %
Kappa statistic                          0.9778
Mean absolute error                      0.0444
Root mean squared error                  0.1089
Relative absolute error                  8.8637 %
Root relative squared error             21.7479 %
Total Number of Instances               90
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall   F-Measure
1.000     0.022     0.978       1.000    0.989
0.978     0.000     1.000       0.978    0.989
MCC       ROC Area  PRC Area  Class
0.978     0.981     0.965     c0
0.978     0.981     0.990     c1
Weighted Avg.
0.989     0.011     0.989       0.989    0.989
 0.978     0.981     0.978
=== Confusion Matrix ===
  a  b   <-- classified as
 44  0 |  a = c0
  1 45 |  b = c1
```

Figura 4.6: Resultado del algoritmo Decision Table

Random Forest es el cuarto algoritmo clasificador, mostrado en la Fig. 4.7 . Aplica una selección aleatoria de atributos con los cuales construye arboles independientes que posteriormente serán evaluados y promediados. Este algoritmo es uno de los tres algoritmos que presentan mejores resultados.

Finalmente, el quinto algoritmo es J48, se muestra en la Fig. 4.8, este algoritmo perteneciente a Weka, es un algoritmo derivado de C45 que busca la creación de árboles binarios a los que adicionalmente aplica una operación de poda. Establece un criterio de evaluación que tiene la finalidad de evitar que las clases ma-

```
Random Forest
Time taken to build model: 0.2 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          88       97.7778 %
Incorrectly Classified Instances         2        2.2222 %
Kappa statistic                          0.9555
Mean absolute error                      0.0607
Root mean squared error                  0.1349
Relative absolute error                 12.1236 %
Root relative squared error             26.9434 %
Total Number of Instances               90
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall   F-Measure
0.977     0.022     0.977       0.977    0.977
0.978     0.023     0.978       0.978    0.978
 MCC      ROC Area  PRC Area  Class
0.956     0.997     0.997     c0
0.956     0.997     0.997     c1
Weighted Avg.
0.978     0.022     0.978       0.978    0.978
0.956     0.997     0.997
=== Confusion Matrix ===
  a  b   <-- classified as
 43  1 |  a = c0
  1 45 |  b = c1
```

Figura 4.7: Resultado del algoritmo Random Forest

yoritarias se vean beneficiadas en el proceso de clasificación. Es uno de los tres algoritmos con mejor desempeño y adicionalmente proporciona el mejor tiempo de ejecución registrada.

```
J48 pruned tree
Time taken to build model: 0.06 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          89       98.8889 %
Incorrectly Classified Instances         1        1.1111 %
Kappa statistic                          0.9778
Mean absolute error                      0.0221
Root mean squared error                  0.1067
Relative absolute error                  4.4138 %
Root relative squared error             21.3139 %
Total Number of Instances               90
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall  F-Measure
1.000     0.022     0.978       1.000   0.989
0.978     0.000     1.000       0.978   0.989
MCC      ROC Area  PRC Area  Class
0.978    0.979     0.954     c0
0.978    0.979     0.989     c1
Weighted Avg.
0.989     0.011     0.989       0.989   0.989
0.978    0.979     0.972
=== Confusion Matrix ===
  a  b   <-- classified as
 44  0 |  a = c0
  1 45 |  b = c1
```

Figura 4.8: Resultados del algoritmo J48

Es importante mencionar que estos resultados no determinan que algoritmo clasificador sea el mejor en general, pero si cual presento ventajas para el caso de estudio que se presenta, tomando en cuenta la naturaleza de los datos obtenidos, la condición en la que se desarrolla, así como el tamaño de la muestra, la particularidad de los atributos propuestos y los métodos aplicados para obtener el filtrado de los datos que componen la muestra de información para la fase de aprendizaje.

9. **Generación de resultados.** La interpretación de las matrices de confusión está

Tabla 4.4: Resultados obtenidos por los clasificadores en Caso 1 [Fuente propia].

Clasificador	Correctos	Incorrecto	T Ejecución (seg)
Red Neuronal	98.8889 %	1.1111 %	0.21
J8	98.8889 %	1.1111 %	0.02
Random Forest	98.8889 %	1.1111 %	0.07
Naive Bayes	97.7778 %	2.2222 %	0.001
Decision Table	95.5556 %	4.4444 %	0.04

basada en la diagonal que se observa, de ésta se muestran los datos que pertenecen a cada clase (para este estudio clase a y b) separándolos en los que fueron clasificados correctamente y cuales incorrectamente. Para la Red Neuronal se tiene que de los 44 registros pertenecientes a la clase a todos fueron clasificados correctamente ya que ninguno se catalogó como clase b, en tanto, para los 46 registros que pertenecían a la clase b 45 fueron clasificados correctamente, es decir, como pertenecientes a la clase b y 1 catalogado como clase a.

La relación de los datos resultantes que fueron clasificados correctamente o incorrectamente, así como el tiempo de ejecución que tubo cada uno de ellos se ve reflejado en la Tabla 4.4 que se presenta a continuación

Los algoritmos Red Neuronal, J48 y Random Forest obtuvieron los mismos niveles de datos correctos en la clasificación y numero de datos incorrectos, aunque J48 es el que presenta un mejor resultado en Tiempo de ejecución. Naive Bayes y Decision Table se muestran como los algoritmos más débiles.

4.1.2. Análisis de los resultados obtenidos en el escenario 1

Una manera de comprobar que los resultados obtenidos estaban correctamente clasificados y que esto obedecía a la unión de atributos establecida inicialmente en la creación del vector característico, es la inyección de datos aleatorios, que no correspondían al patrón o naturaleza que mostraban los datos reales. Es así como para cada uno de los vectores creados se realizó la prueba de inyección de datos aleatorios. Para

el escenario uno, se aplicó para el vector (compuesto de datos normales y de datos de ataque) la inclusión de datos aleatorios.

El tamaño de la muestra aleatoria fue de 32 datos para cada atributo, recordando que son 18 atributos, el modo de obtener estos datos fue empleando la formula Aleatorio de Excel, estableciendo un rango de 0 a 1 con la finalidad de que coincidieran con el rango manejado de los datos ya normalizados en el vector original del escenario uno. Las clases definidas para los datos aleatorios en la fase de aprendizaje fue la mitad de la muestra para la clase $c0$ y los restantes para la clase $c1$.

La Fig.4.9 muestra las matrices de cofusión del proceso de clasificación realizado en Weka en el caso uno durante la prueba de inserción de aleatorios, en donde puede observarse que los datos son clasificados decrementando el nivel de eficiencia en la correcta clasificación, esto se debe a que los algoritmos presentan errores al intentar clasificar datos que no obedecen a la naturaleza que rige a las muestras de datos reales.

Random Forest

```
=== Confusion Matrix ===

  a  b   <-- classified as
 49 10 |  a = c0
 12 51 |  b = c1
```

J48

```
=== Confusion Matrix ===

  a  b   <-- classified as
 47 12 |  a = c0
 11 52 |  b = c1
```

Red Neuronal

```
=== Confusion Matrix ===

  a  b   <-- classified as
 48 11 |  a = c0
 26 37 |  b = c1
```

Decision Table

```
=== Confusion Matrix ===

  a  b   <-- classified as
 47 12 |  a = c0
 13 50 |  b = c1
```

Naive Bayes

```
== Confusion Matrix ===

  a  b   <-- classified as
 31 28 |  a = c0
 42 21 |  b = c1
```

Figura 4.9: Matriz de confusión en prueba de aleatorios para Caso 1 [Fuente propia].

Los niveles de certeza durante la clasificación obtenida por los algoritmos de clasificación se ven decrementados, pero mantienen su posición, comparándolos con los resultados de la Tabla 4.5 mostrados en el escenario uno, aunque se observa la mejora de Decision Table quien mejora en una posición dejando a Naive Bayes en quinto lugar en la tabla quien presenta un nivel por debajo de la media muestral total, es decir

Tabla 4.5: Resultados obtenidos por los clasificadores en prueba de aleatorios para Caso 1 [Fuente propia].

Clasificador	Correctos	Incorrecto	T Ejecución (seg)
Random Forest	81.9672 %	18.0328 %	0.16
J48	81.1475 %	18.8525 %	0.06
Decision Table	79.5082 %	20.4918 %	0.14
Red Neuronal	69.6721 %	30.3279 %	0.64
Naive Bayes	42.623 %	57.377 %	0.02

que no logra clasificar ni la mitad de los datos que pertenecen al vector característico que se maneja. Los niveles de los tiempos de ejecución y la interpretación porcentual de los resultados obtenidos se pueden ver reflejados.

Lo que puede observarse en la tabla es que Random Forest es el algoritmo que presenta mejores resultados, lo que es notorio es el despunte de Decisión Table ganándole la posición en la tabla al algoritmo de Red Neuronal en tanto Naive Bayes se perfila como el algoritmo más débil en la prueba de aleatorios para el escenario número 2. El comportamiento de los algoritmos ante la inserción de valores aleatorios que no siguen el patrón del vector original es diferente a lo que se esperaría ya que algunos mejoran su capacidad de clasificación ante el escenario planteado, en tanto otros pierden certeza en la clasificación. Lo anterior no representa que un algoritmo sea mejor que otro o que las clasificaciones realizadas sean del todo correctas ya que el valor real de los datos que si obedecen al comportamiento real de la red es el 75 % el otro 25 % refleja los datos aleatorios.

4.2. Escenario 2

El caso de estudio dos está enfocado al estudio de una variante del ataque Denegación de Servicio que se genera por la explotación de la vulnerabilidad que aún presentan algunas páginas web que manejan o emplean el protocolo HTTP. Para caracterizar el escenario se emplean las dos variaciones del tráfico: estado normal de la red y estado de ataque.

En el caso de estado normal de la red se generan peticiones de un equipo al servidor web. Para las capturas en estado de ataque se genera desde un host la petición maliciosa al servidor web que habrá de dejarlo fuera. La captura de información de la red emplea Wireshark y se hace uso de otras aplicaciones también empleadas en el escenario uno. Las acciones ejecutadas para este escenario se apegan al orden del proceso propuesto, mismo que fue empleado, en el caso uno.

4.2.1. Desarrollo del proceso propuesto en el escenario 2

Del mismo modo, en esta sección, se describen las actividades desarrolladas en cada fase que forma el proceso propuesto por esta tesis para llevar acabo el desarrollo del vector característico y así ejecutar un proceso de clasificación que muestre los resultados obtenidos por el flujo de datos de la red, sometidos a el proceso de clasificación a fin de concluir si se trata de un flujo de datos de ataque o de un flujo de datos normal de la red para el escenario planteado.

1. **Diseño de la red.** La arquitectura de la red contempla algunas variaciones, como cambio de sistema operativo que maneja el equipo adversario y segmento a la que pertenece la red. La Fig. 4.10 muestra el escenario definido para el análisis de tráfico. Como puede observarse, consiste en dos redes conectadas a través de un ruteador, mientras la red 1 se compone de un servidor web, el cual será atacado, y varios hosts, la red 2 contiene por lo menos una computadora que fungirá como adversario.

   Las capturas se realizan del puerto configurado como promiscuo en el switch. La configuración del equipo que genera el ataque hacia el servidor web está basada en la instalación de Kali como sistema operativo, para lograrlo, se instala una máquina virtual en un host a fin de generar un sistema operativo virtual con Kali instalado. Se opto por esta opción por la versatilidad y ventajas que ofrecen, entre ellas la portabilidad que permiten.

   La aplicación que se emplea para la generación de máquinas virtuales es VM-

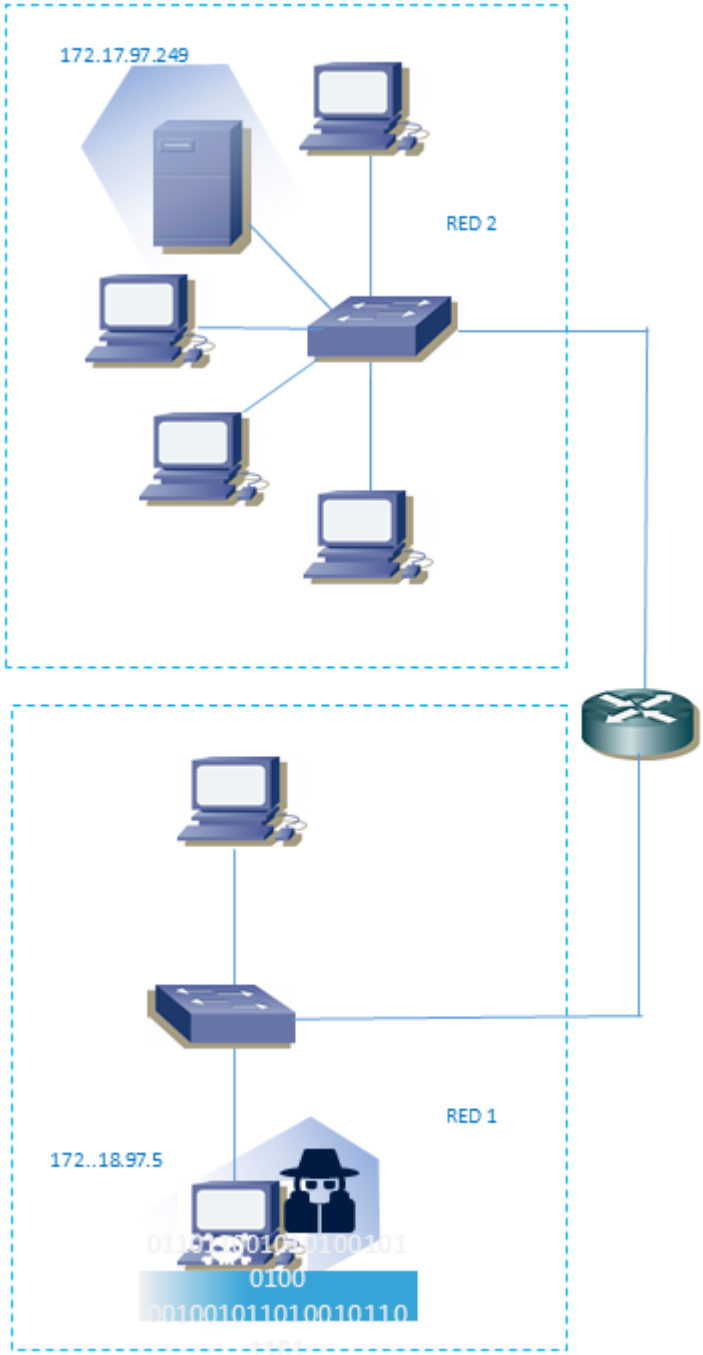

Figura 4.10: Escenario de red2 [Fuente propia].

ware que es una herramienta de virtualización, tiene versiones de Windows y Linux. No es necesario particionar el disco ni reiniciar el ordenador, una vez instalada. Una gran ventaja de esta aplicación es su capacidad para soportar varias máquinas virtuales que pueden ejecutarse al mismo tiempo. Al conectar dos monitores al equipo que contenga las virtualizaciones se pueden visualizar ambas máquinas virtuales.

2. **Captura de datos en estado normal.** Para obtener esta información se generan peticiones de un host con sistema operativo Ubuntu hacia el servidor Web en Apache, configurado en un equipo con Ubuntu como sistema operativo, la IP de origen es la del equipo y como IP destino se queda la del servidor. Se tomaron 45 lecturas para este tipo de flujo. La escucha fue sintetizada en un lapso de 5 segundos aproximadamente lo que deja con archivos que tienen un tamaño que va de los 3KB a los 23 KB.

3. **Captura de datos en estado de ataque.** El equipo atacante para este escenario está basado en un sistema operativo Kali configurado en una máquina virtual. Kali es un sistema operativo de Linux que tiene como sucesor a BackTrack es un sistema operativo basado en auditoria de seguridad en sistemas y herramientas de evaluación de la penetración que a su vez una mejora de SLAX(WHAX)y Auditor.

   La eficiencia de Kali Linux se basa en su capacidad de trabajo en ambientes virtuales, portabilidad, versatilidad en su instalación (comparada con otros sistemas Linux), Interfaz gráfica gracias a que pertenece al tipo de distribución GNOME para el consumo de menos recursos, usa recursos mínimos para su instalación, además de la posibilidad de modificar el kernel, contiene en su menú, herramientas para ejecutar, como el escaneo de puertos, escaneo de vulnerabilidades, descifrado de claves inalámbricas, permite configurar un Snort para que actúe como un IDS, además de que la comunidad lo actualiza constantemente.

   Kali puede ser instalado en una máquina virtual empleando VMware Workstation

que es una herramienta de virtualización, que tiene versiones para Windows y Linux. No es necesario particionar el disco ni reiniciar el ordenador, una vez instalada. Una gran ventaja de esta aplicación es su capacidad para soportar varias máquinas virtuales que pueden ejecutarse al mismo tiempo. Además, al conectar dos monitores al equipo que contenga las virtualizaciones se pueden visualizar ambas máquinas virtuales.

Para obtener estas muestras de información, es necesario preparar el equipo que generaría el ataque, además de ello, este escenario genera el ataque mediante un archivo configurado en lenguaje Pert el cual recibe el nombre de slowloris.pl ejecutado desde Kali, este genera un conjunto de peticiones de inicio de sesión al servidor web. Se genera un cliente http, dicho cliente intenta abrir tantas conexiones como pueda al servidor web e intenta mantenerlas abiertas tanto tiempo como sea posible. Periódicamente para evitar que el servidor web cierre la conexión va añadiendo headers a la petición HTTP sin llegar a finalizarla nunca.

Esto provoca que en determinados servidores web se vayan quedando las conexiones abiertas hasta llegar al máximo, bloqueando las peticiones legítimas. Entre los servidores web afectados se encuentra tanto Apache 1.x como Apache 2.x. Para finalizar la petición debería dejar una línea en blanco. Pero, en el caso de slowloris, deja el comando incompleto y al cabo de un tiempo vuelve introduce un header sin sentido para mantener la petición activa.

Se puede reducir el impacto del ataque modificando la variable Timeout, que por defecto está muy alto, en el programa se trabajó con un valor de 600 pero se realizaron pruebas variando este valor para corroborar el efecto causado, valorando su aumento o disminución. Este parámetro afecta por separado a los siguientes tiempos para la recepción de la petición:

a) Tiempo que tarda en recibir una petición GET

b) Tiempo entre la recepción de paquetes TCP en las peticiones POST y PUT

c) Tiempo entre ACKs en las respuestas de la petición

Uno de los puntos observados sobre el funcionamiento del programa Slowloris, es que si dejamos el valor demasiado bajo para la variable Timeout pueden finalizar peticiones legítimas y que en tanto se siga ejecutando el código, se sigue afectando la red, por eso es necesario parar las peticiones de ejecución desde el menú de tareas ejecutables en Kali.

Se generaron 45 ejecuciones del ataque para tomar una muestra en cada uno de ellos, de los archivos obtenidos se seleccionaron datos pertenecientes a un lapso aproximado de 5 segundos en las que se observaran distintos momentos del ataque. Los archivos resultantes van de los 12KB a los 29KB.

4. **Análisis de los datos.** Para trabajar con los datos obtenidos y generar un análisis de ellos, se recurre al primer tratamiento de los datos mediante Wireshark, pasando las lecturas obtenidas a un formato de texto plano en donde se pueden observar características presentes. El análisis es la base para la realización de la siguiente fase.

   Durante la observación de los archivos se encuentra que los datos de origen y destino son factores determinantes para detectar un posible ataque hacia la red, el tamaño de la carga es otro aspecto relevante, así como observar que se generan muchas peticiones en un periodo corto, desde un mismo equipo y aunque estas peticiones tienen un formato valido no concluyen el proceso de comunicación como se hace en las muestras que se obtuvieron del escenario con un flujo normal de la red.

5. **Elección de los atributos.** Una vez determinados los elementos presentes en una petición de comunicación al servidor web en estado normal, así como, los presentes cuando se genera un ataque al servidor web, se estructuran las cabeceras de los vectores característicos, que serán unificados en un vector resultante contemplando ambos casos, generando con ello las dos posibles clases, *c0* para representar el vector resultante de un estado normal de la red y *c1* formado con los datos del vector en un estado de ataque a la red.

Tabla 4.6: Lista de atributos, escenario 2 [Fuente propia].

Atributo	Descripción	Tipo
a0	Patrones totales	discreto
a1	Número de patrones distintos	discreto
a2	Densidad léxica	discreto
a3	Capacidad de lectura	discreto
a4	Total de comandos SYN	discreto
a5	Total de comandos ACK	discreto
a7	Total de sentencias	discreto
a7	Logitud promedio mínima de comando	discreto
a8	Longitud promedio comando	discreto
a9	Longitud máxima de comando	continuo
a10	Longitud mínima de comando	continuo
a11	Legibilidad2	discreto
a12	Ip origen	continuo
a13	Ip origen	continuo
a14	Ip origen	continuo
a15	Ip origen	continuo
a16	Ip destino	continuo
a17	Ip destino	continuo
a18	Ip destino	continuo
a19	Ip destino	continuo
a20	Tiempo de transmisión	continuo

Se recurre nuevamente a la opción de disminuir el número de atributos para buscar una ejecución optima en la clasificación, evitando emplear datos redundantes o que no aporten verdadera información de la red. Los vectores se componen de 21 cabeceras o tabuladores. Los primeros 11 correspondientes a atributos obtenidos tras el análisis de las relaciones entre los datos muestrales y los 10 restantes a datos de la red. Los atributos son representados con las variables *a0* a *a20*. La Tabla 4.6 enlista los atributos usados, con la descripción y el tipo de valor posiblemente asignado: continuo para valores constantes y discreto para valores aleatorios.

6. **Generación del vector.** Una vez concluido el proceso de análisis de los datos y de la selección de los atributos se establecen las cabeceras del vector característico para el caso dos que contiene las muestras de datos en peticiones normales a la red y las muestras de datos al reproducir el ataque hacia el servidor web. Las clases obtenidas son $c0$ y $c1$, donde, $c0$ representa un flujo normal y $c1$ un flujo clasificado como sospechoso o de ataque. Esta fase se realiza con el llenado, de los vectores de las dos vertientes: flujo normal de la red y en estado de ataque. El vector resultante entonces es la unión de los dos anteriores. Los archivos con los que se cuenta son archivos en texto plano que serán manejados y tratados mediante el analizador léxico para lograr obtener los datos pertenecientes a los 21 atributos estipulados como necesarios en el vector característico perteneciente a este escenario.

   El analizador léxico hace la labor de búsqueda en estos archivos y genera resultados cuantitativos sobre los atributos que se piden. Estos resultados obtenidos del analizador léxico, son vertidos en un archivo Excel con lo que se forma la base de datos, también conocida como base de conocimiento, a la cual, se les realiza un tratamiento a los datos a fin de lograr normalizarlos para su uso en las siguientes fases del proceso.

7. **Tratamiento de datos.**El procedimiento para el tratamiento de los datos fue el mismo que el realizado para el escenario uno, es decir, se emplean la fórmula de normalización de Excel. Con el mismo fin, lograr agilizar el proceso de ejecución del algoritmo clasificador.

   Previo a esta normalización, se presenta un primer cambio en el formato de datos para su manejo, cuando cambia el formato inicial de las capturas obtenidas en Wireshark a formato de texto plano. Además de disminuir el tamaño de la muestra, el tratamiento de datos permite agilizar el proceso

8. **Clasificación.** Esta fase contempla que los datos contenidos en la base de conocimiento ya normalizados son sometidos a la ejecución de los algoritmos cla-

sificadores. Para el proceso de clasificación se emplea Weka. Los algoritmos de clasificación son cinco, los mismos que en el caso uno: Red neuronal, Random Forest, Naive Bayes, Decision table y J48. La eficiencia de los clasificadores está determinada por el porcentaje de clasificaciones correctas, clasificaciones incorrectas, tiempo de ejecución entre otros que hacen entender con cual se obtuvieron los mejores puntajes.

Los resultados que presentan los algoritmos clasificadores en Weka durante el proceso de clasificación arrojan ponderadores como el grado de certeza en la clasificación. El primer algoritmo del que se muestran resultados es Naive Bayes en la Fig. 4.11.

```
Naive Bayes Classifier
Time taken to build model: 0.03 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          89        97.8022 %
Incorrectly Classified Instances         2         2.1978 %
Kappa statistic                          0.956
Mean absolute error                      0.0239
Root mean squared error                  0.1489
Relative absolute error                  4.7723 %
Root relative squared error             29.7607 %
Total Number of Instances               91
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall  F-Measure
0.977     0.021     0.977       0.977   0.977
0.979     0.023     0.979       0.979   0.979
MCC      ROC Area  PRC Area  Class
0.956    0.988     0.975     c0
0.956    0.983     0.990     c1
Weighted Avg.
0.978     0.022     0.978       0.978   0.978
0.956    0.985     0.982
=== Confusion Matrix ===
  a  b   <-- classified as
 43  1 |  a = c0
  1 46 |  b = c1
```

Figura 4.11: Resultados del algoritmo Naive Bayes

El segundo algoritmo mostrado en la Fig.4.12 clasificador empleado y del que a continuación se muestran los resultados es Decision Table. Los algoritmos

empleados para este escenario son los mismos empleados en el escenario uno a fin de establecer posibles comparaciones entre los resultados obtenidos.

```
Decision Table:
Time taken to build model: 0.17 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          90         98.9011 %
Incorrectly Classified Instances         1          1.0989 %
Kappa statistic                          0.978
Mean absolute error                      0.0449
Root mean squared error                  0.1086
Relative absolute error                  8.9836 %
Root relative squared error             21.7029 %
Total Number of Instances               91
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall  F-Measure
1.000     0.021     0.978       1.000   0.989
0.979     0.000     1.000       0.979   0.989
MCC      ROC Area  PRC Area  Class
0.978    0.980     0.954     c0
0.978    0.980     0.990     c1
Weighted Avg.
0.989     0.010     0.989       0.989   0.989
0.978    0.980     0.972
=== Confusion Matrix ===
  a  b   <-- classified as
 44  0 |  a = c0
  1 46 |  b = c1
```

Figura 4.12: Resultados del algoritmo Decision Table

El tercer clasificador emplea el algoritmo J48 que el algoritmo que para el caso uno logra empatar en resultados con Random Forest y Red Neuronal; para este escenario, este algoritmo se sigue perfilando como uno de los que obtiene mejores resultados y se muestra a continuación, en la Fig.4.13.

El cuarto algoritmo utilizado es Random Forest, este algoritmo al igual que J48 obedecen a principios de clasificación que favorece aprendizaje supervisado, se muestra en la Fig. 4.14.

Por último el la Fig.4.15 el clasificador Red Neuronal del Tipo Backpropagation de 3 capas es mostrado a continuación.

```
J48 pruned tree
Time taken to build model: 0.03 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          90        98.9011 %
Incorrectly Classified Instances         1         1.0989 %
Kappa statistic                          0.978
Mean absolute error                      0.0218
Root mean squared error                  0.1061
Relative absolute error                  4.3688 %
Root relative squared error             21.2077 %
Total Number of Instances               91
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall  F-Measure
1.000     0.021     0.978       1.000   0.989
0.979     0.000     1.000       0.979   0.989
MCC      ROC Area  PRC Area  Class
0.978    0.980     0.954     c0
0.978    0.980     0.990     c1
Weighted Avg.
0.989     0.010     0.989       0.989   0.989
0.978    0.980     0.972
=== Confusion Matrix ===
  a  b   <-- classified as
 44  0 |  a = c0
  1 46 |  b = c1
```

Figura 4.13: Resultados del algoritmo J48

```
RandomForest
Time taken to build model: 0.17 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          89       97.8022 %
Incorrectly Classified Instances         2        2.1978 %
Kappa statistic                          0.956
Mean absolute error                      0.0367
Root mean squared error                  0.1332
Relative absolute error                  7.3337 %
Root relative squared error             26.6182 %
Total Number of Instances               91
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall   F-Measure
0.977     0.021     0.977       0.977    0.977
0.979     0.023     0.979       0.979    0.979
MCC       ROC Area  PRC Area    Class
0.956     0.983     0.957       c0
0.956     0.983     0.990       c1
Weighted Avg.
0.978     0.022     0.978       0.978    0.978
0.956     0.983     0.974
=== Confusion Matrix ===
  a  b   <-- classified as
 43  1 |  a = c0
  1 46 |  b = c1
```

Figura 4.14: Resultados del algoritmo Random Forest

```
Red Neuronal
Time taken to build model: 1.35 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          89       97.8022 %
Incorrectly Classified Instances         2        2.1978 %
Kappa statistic                          0.956
Mean absolute error                      0.0248
Root mean squared error                  0.1235
Relative absolute error                  4.9669 %
Root relative squared error             24.6863 %
Total Number of Instances               91
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall  F-Measure
0.977     0.021     0.977       0.977   0.977
0.979     0.023     0.979       0.979   0.979
MCC      ROC Area  PRC Area  Class
0.956    0.981     0.956     c0
0.956    0.981     0.990     c1
Weighted Avg.
0.978     0.022     0.978       0.978   0.978
0.956    0.981     0.974
=== Confusion Matrix ===
  a  b   <-- classified as
 43  1 |  a = c0
  1 46 |  b = c1
```

Figura 4.15: Resultados Red Neuronal

La matriz de confusión de los cinco algoritmos empleados se presenta en la Fig.4.16 donde se pueden comparar fácilmente.

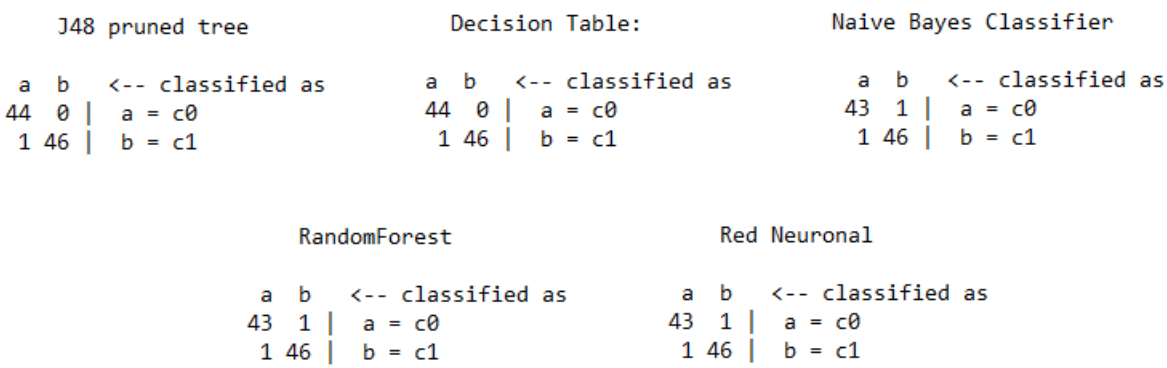

Figura 4.16: Matriz de confusión del caso 2 ataque DoS por HTML [Fuente propia].

9. **Generación de resultados.** La interpretación de las matrices de confusión está basada en la diagonal que se observa, de ésta se muestran los datos que pertenecen a cada clase (para este estudio clase a y b) separándolos en los que fueron clasificados correctamente y cuales incorrectamente. Para la Red Neuronal se tiene que de los 44 registros pertenecientes a la clase a todos fueron clasificados correctamente ya que ninguno se catalogó como clase b, en tanto, para los 46 registros que pertenecían a la clase b 45 fueron clasificados correctamente, es decir, como pertenecientes a la clase b y 1 catalogado como clase a.

   La relación de los datos resultantes que fueron clasificados correctamente o incorrectamente, así como el tiempo de ejecución que tubo cada uno de ellos se ve reflejado en la Tabla 4.7.

   El algoritmo Decision Table que presenta en el caso uno los resultados que lo posicionan como el más débil ante los demás clasificadores, logra en el escenario dos estar por encima de Random Forest y Red Neuronal. Lo mismo logra Naive Bayes quien el escenario uno quedo en el cuarto lugar. En este escenario quien presenta menor grado porcentual de certeza en la clasificación es el algo-

Tabla 4.7: Resultados obtenidos por los clasificadores en Caso 2 [Fuente propia].

Clasificador	Correctos	Incorrecto	T Ejecución (seg)
J48	98.9011 %	1.0989 %	0.03
Decision Table	98.9011 %	1.0989 %	0.17
Naive Bayes	97.8022 %	2.1978 %	0.03
Random Forest	97.8022 %	2.1978 %	0.17
Red Neuronal	97.8022 %	2.1978 %	1.35

ritmo Red Neuronal. J48 en tanto, sigue manteniéndose como el algoritmo que muestra mejores resultados durante el proceso de clasificación. Los tiempos de ejecución para todos los algoritmos se ven ligeramente incrementados en este escenario.

4.2.2. Análisis de los resultados obtenidos en el escenario 2

Para el escenario 2 se realizó un procedimiento que incluye la inserción de valores aleatorios para comprobación de la clasificación realizada. Esta inserción ayuda a verificar el comportamiento del algoritmo durante la clasificación. La prueba consistió en aumentar el número muestral del vector original, aumentando con valores aleatorios a los que se les asigno una clase.

El tamaño original del vector era de 92 datos por cada atributo, recordando que el vector característico del escenario dos se compone de 21 atributos. La magnitud total para el vector con aleatorios es de 123 datos para cada uno de los 21 atributos, es decir, se inyectan 31 elementos aleatorios para cada cabecera. Las clases asignadas para los datos aleatorios quedan en 15 calificados arbitrariamente como clase $c0$ y 16 clasificados arbitrariamente como clase $c1$

La Fig.4.17 presenta las matrices de confusión que se obtuvieron del proceso de clasificación empleando los cinco algoritmos. La clasificación es realizada en Weka como en los casos anteriores.

Como puede observarse en las matrices de confusión el algoritmo que presento mayor dificultad para lograr clasificar las clases correctamente es Naive Bayes. Todos

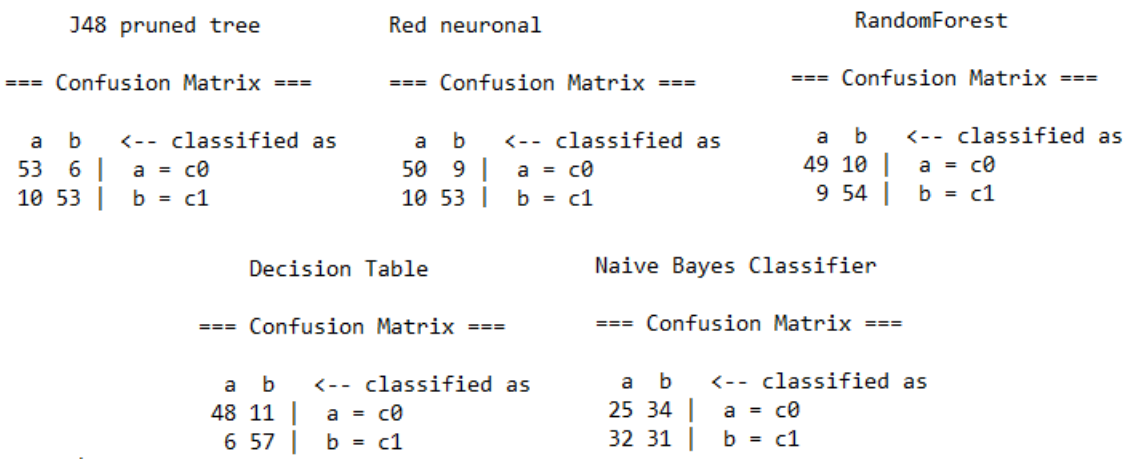

Figura 4.17: Matriz de confusión prueba de aleatorios en Caso 2 [Fuente propia].

Tabla 4.8: Resultados obtenidos por los clasificadores en prueba de aleatorios para Caso 2 [Fuente propia].

Clasificador	Correctos	Incorrecto	T Ejecución (seg)
J48	86.8852 %	13.1148 %	0.08
Decision Table	86.0656 %	13.9344 %	0.18
Random Forest	84.4262 %	15.5738 %	0.21
Red Neuronal	84.4262 %	15.5738 %	0.79
Naive Bayes	45.9016 %	54.0989 %	0.03

los algoritmos se ven modificados en los niveles de certeza logrados, se esperaba que se lograra una correcta clasificación de los datos del vector original, pero al insertar los aleatorios las relaciones encontradas entre los atributos se ven afectadas. El algoritmo con mayor número de datos clasificados es J48, seguido de Red neuronal, Random Forest, Decisión Table y por último Naive Bayes. La tabla muestra los resultados porcentuales de los niveles de certeza y de error en la clasificación obtenidos por los clasificadores.

Los resultados porcentuales obtenidos por los algoritmos se ven reflejados en la Tabla 4.8. De estos resultados se observa que los algoritmos en general presentan mayor dificultad para clasificar los datos al insertar en la muestra datos aleatorios, rompiendo los patrones de orden o análisis en los que se basan los algoritmos.

De los datos que se observan en la tabla muestran que el algoritmo Decision Table

cae del segundo lugar al quinto lugar en la prueba de inserción de aleatorios. Debemos recordar que Decision Table evalúa de manera independiente los datos y la presencia o ausencia de un elemento no interfiere en su método para asignar la clase a la que pertenece un conjunto de datos. Sin embargo, no logra asignar el 75 % de la muestra que sería el resultado ideal ya que el otro 25 % corresponde a datos aleatorios a los que no se les debería encontrar relación alguna con las clases existentes. Es decir que los algoritmos que presentan valores mucho mayores al 75 % de los datos asignados a una clase no necesariamente realizan una correcta clasificación.

Decision Table y J48 son los algoritmos que para este escenario logran mantenerse en primer y segundo lugar respectivamente, seguidos de Random Forest y Red Neuronal que recuperan una posición en la tabla en la prueba de aleatorios. En cuanto al tiempo de ejecución Naive Bayes mantiene el mismo porcentaje, en trato que los otros algoritmos varían presentando en algunos casos mejoras o decrementos en estos valores.

5. Análisis de resultados

Los resultados que los clasificadores reportan al usar el vector característico propuesto presentan porcentajes elevados de precisión. Para hacer un análisis de su efectividad, es necesario realizar una comparación con respecto a los trabajos publicados en el estado del arte vistos en la sección 2.9.

Es importante no perder de vista, que la información de comparación tiene variación para cada trabajo publicado, considerando el enfoque de aprendizaje empleado, los ataques de intrusión a los que está dirigido el detector, la base de datos de conocimiento usada, los atributos y las clases empleadas o si es supervisado o no. Para una mejor comprensión, se muestra una lista de las coincidencias y otra de las diferencias, para tomar los elementos más importantes en cada comparación.

- Diferencias:

    1. El origen de los datos empleados en la generación de los vectores característicos propuestos en el escenario1 y escenario2, proviene de información real de la red, mientras que los trabajos reportados en el estado del arte utilizan, principalmente, la base de datos KDD99. El número de variables
    2. Los trabajos relacionados usan más atributos y más clases posibles, en contraste con la propuesta realizada, que tiene sólo en el escenario1 con 18 atributos y 21 atributos para el vector del escenario2, con dos clases posibles para ambos escenarios.
    3. Los ataques de intrusión son variados, este trabajo se enfoca en el ataque de intrusión por denegación de servicio.

Tabla 5.1: Tabla de comparación en el porcentaje de precisión. [Fuente propia].

	Red Neuronal	Alg. J48	Naive Bayes	Random Forest
(Rivero y cols., 2016)	98.52 %	99.02 %	98.14 %	NA
(Ashfaq y cols., 2017)	77.41 %	81.05 %	76.56 %	80.67 %
(Zhu y cols., 2017)	NA	NA	76.56 %	80.67 %
Vector Escenario1	98.88 %	98.88 %	97.77 %	98.88 %
Vector Escenario2	97.80 %	98.90 %	97.80 %	97.80 %

- Similitudes de comparación

    1. La precisión con la que el clasificador reporta resultados.
    2. Los algoritmos de clasificación J48, Redes neuronales y Naive Bayes y Random Forest.
    3. La variación más precisa reportada en cada trabajo relacionado.

La Tabla 5.1 presenta la relación entre los algoritmos empleados y el porcentaje de precisión que muestran en la clasificación. Como se puede apreciar, la propuesta en este trabajo, reporta un elevado porcentaje de precisión en todos los clasificadores, muy cercano a lo propuesto en (Rivero y cols., 2016) y en contraste con (Ashfaq y cols., 2017), que reporta un 81.05 % con el algoritmo J48, sin embargo, con el algoritmo Naive Bayes cae hasta el 76.56 %.

Finalmente, en la Tabla 5.2 se presenta el tipo de información utilizada, en donde, claramente se identifica que el vector característico propuesto usa información obtenida de un escenario de ataque real a la red y en estado normal, mientras que los restantes, se apoyan de información generalizada, almacenada en la base de datos DKK99 y sus variaciones.

La comparación realizada entre los resultados obtenidos en este trabajo, frente a los mostrados por trabajos estudiados la sección 2.9, sirve para corroborar que los resultados obtenidos son óptimos, lo que también prueba el funcionamiento del proceso propuesto para la creación de un sistema de detección de intrusiones.

Esta investigación se enfoca en la detección de intrusiones generada por los ataques de tipo denegación de servicio, en las dos variantes tratadas. Otro factor impor-

Tabla 5.2: Tabla de comparación usando tipos de muestra y enfoque [Fuente propia].

	Muestra	Enfoque
Rivero (Rivero y cols., 2016)	KDD99	Supervisado
Ashfaq (Ashfaq y cols., 2017)	KDDCUP99	Semi-supervisado
Zhu (Zhu y cols., 2017)	KDD99 y Gure KDD	Supervisado
Vector Escenario1	Lectura directa de la red	Supervisado
Vector Escenario2	Lectura directa de la red	Supervisado
Vector Aleatorio1	Lectura directa de la red	Semi-supervisado
Vector Aleatorio2	Lectura directa de la red	Semi-supervisado

tante encontrado es la utilización de los algoritmos de clasificación pertenecientes a las herramientas de minería de datos que son empleados con más frecuencia. Los algoritmos de tipo árbol son los que mostraron mejores ponderadores, indicando que, en escenarios similares, este tipo de algoritmos son eficientes.

6. Conclusiones

La aplicación de técnicas de minería de datos que emplean algoritmos pertenecientes a la inteligencia artificial ofrece una gran ventaja para entender el comportamiento de un fenómeno, en este caso reconocer cuando se produce un ataque de intrusión en una red, el cual, logra valerse del envío de paquetes permitidos por los protocolos de red, pero en un formato mayor, con lo cual, se consigue una saturación del servidor y dejarlo fuera al servicio.

En esta investigación se presenta un modelo que se compone de ocho pasos (diseño de la red, captura de datos, análisis de la lectura, elección de los atributos, generación del vector, tratamiento de los datos, clasificación, generación de resultados). Para llevar a cabo un proceso de clasificación que tiene como finalidad la detección de intrusos en una red, particularmente, del ataque de denegación de servicio.

El ataque en el primer caso es provocado por inundación de paquetes como resultado del envío de ráfagas de ping, esto descrito en el caso 1 y un segundo escenario que provoca también un ataque de DoS producido por la explotación de una debilidad presente en el protocolo HTTP. La propuesta consta de la generación de un vector característico para cada uno de los escenarios descritos a partir de información obtenida de una red. Las fases desarrolladas permiten diferenciar entre los registros provenientes de un ataque y los registros provenientes de un flujo normal para ambos casos de esta investigación.

El vector característico en el escenario uno está compuesto por 18 atributos que corresponden a información propia de la red como las IP de origen y destino, el tiempo de captura y toda la información que participa en la transmisión, de tal manera, que

permita una clasificación correcta. En el escenario dos el vector característico obtenido se compone de 21 atributos donde los primeros 11 corresponden a información obtenida de la red y los 10 restantes a datos propios de la red.

Para comprobar que la información contenida en los vectores característicos permite distinguir claramente un ataque o no, se probaron en 5 clasificadores, tales como, una red neuronal, el algoritmo J48 y el Naive Bayes, entre otros. Los resultados reportados por los clasificadores indican que los vectores característicos propuestos permiten una categorización precisa y eficiente.

En comparación con los trabajos relacionados, la propuesta planteada en este trabajo, consigue obtener un porcentaje elevado de precisión con diferentes tipos de clasificadores, usando información real proveniente de la red, en un enfoque supervisado, al estar constituido con únicamente 18 atributos para su caracterización para el caso uno y 21 atributos para el caso dos.

El algoritmo J48 en general presento los mejores resultados ambos escenarios y aún en los escenarios de prueba donde se trabajó con un vector alterado con datos aleatorios. Algunos algoritmos presentaron variaciones en los escenarios como Decisión Table que presento una mayor certeza en la clasificación en el caso2 y mantuvo este comportamiento en el escenario de prueba con aleatorios. Naive Bayes logro mejorar su posición en el escenario dos, sin embargo, para la prueba de aleatorios sus niveles de precisión en la clasificación se vieron debilitados. Random Forest y Red Neuronal lograron los mismos resultados que J48 en el escenario uno, pero perdieron esta característica para el escenario dos y lo mismo en el escenario de prueba.

Los escenarios de prueba tuvieron como finalidad la comprobación de eficiencia de los vectores propuestos, mismo que quedo comprobado con los resultados obtenidos, al mostrar un decremento considerable en la capacidad de asignar una clase a los datos que componen los vectores modificados con valores aleatorios.

Dentro de las fases que comprenden la generación de un Sistema de Detección de Intrusiones la generación de la base de conocimiento y la determinación del motor de inferencia, son las fases indispensables para lograr un correcto funcionamiento. La

certeza en el proceso de clasificación depende de estos dos factores, es decir, de una base con datos que mantengan relaciones entre sí que respondan a las características que se presentan con un escenario estudiado. En este caso, los datos que sean representativos de un ataque de denegación de servicio y de un flujo normal en una red, para la formación del vector característico que cumpla con los atributos y datos que reflejen bien el comportamiento de la red en las circunstancias mencionadas.

De igual importancia es la selección del algoritmo de clasificación que logre interpretar los datos correctamente a fin de lograr una buena identificación del flujo de datos que se maneja en una red. Por ello, la detección de los elementos que participan y las fases necesarias para llevarlo a cabo apoyadas en el seguimiento de un modelo apropiado para su desarrollo logran generar un enfoque y ahorro de tiempo para llevar acabo dichas fases.

6.1. Trabajo a futuro

Hasta este punto, los clasificadores sólo identifican los ataques de denegación de servicio. Una extensión de la detección de intrusos es agregar otros ataques similares, que permitan realizar la clasificación correcta.

Una vez realizados los ataques e identificados en los clasificadores, es importante, desarrollar el proceso a nivel de red. De tal forma que, el tráfico sea analizado en tiempo real y sean detectadas las intrusiones antes de convertirse en ataques.

Referencias

Aguilera, P. (2011). *Redes seguras (seguridad informática)*. Editex.

Aguirre, L. (2013). *Rediseño de la red MPLS con soporte de IPv6 empleando las mejores prácticas de seguridad para el sistema autónomo de Telconet SA de la ciudad de Quito* (B.S. thesis). QUITO/EPN/2013.

André, M., Gulnara, M., Muñoz, V., y Montalvo, M. (2010). Minería de datos aplicada a la formación de equipos de proyectos de software. *AHCIET: revista de telecomunicaciones*, *7*(121).

Ariganello, E., y Sevilla, E. (2014). *Redes cisco: guía de estudio para la certificación CCNA Routing y Switching*. Ra-Ma.

Arkin, O. (2012). Demystifying the myth of passive network discovery and monitoring systems.

Ashfaq, R., Wang, X., Huang, J., Abbas, H., y He, Y. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, *378*, 484–497.

Bejtlich, R. (2005). *Extrusion detection: security monitoring for internal intrusions*. Addison-Wesley Professional.

Bejtlich, R. (2013). *The practice of network security monitoring: understanding incident detection and response*. No Starch Press.

Berdun, F., Armentano, M., y Amandi, A. (2016). Inferencia de roles de equipo a partir de conductas colaborativas detectadas en interacciones textuales. En *Simposio argentino de inteligencia artificial (asai 2016)-jaiio 45 (tres de febrero, 2016)*. (pp. 78–85).

Bertolín, J. y. o. (2008). *Seguridad de la información: Redes, informática y sistemas de información*. Editorial Paraninfo.

Bradley, T., y Carvey, H. (2008). *Manual imprescindible de protección del pc y seguridad en internet*. Anaya Interactiva.

Castro, A., Díaz, I., G.and Alzórriz, y Sancristóbal, E. (2014). *Procesos y herramientas para la seguridad de redes*. Editorial UNED.

Cobo, A. (2011). *Estudio científico de las redes de ordenadores*. Visión Libros.

Demichelis, F., Magni, P., Piergiorgi, P., Rubin, M., y Bellazzi, R. (2006). A hierarchical naive bayes model for handling sample heterogeneity in classification problems: an application to tissue microarrays. *BMC bioinformatics*, *7*(1), 514.

Deng, H., Runger, G., y Tuv, E. (2011). Bias of importance measures for multi-valued attributes and solutions. *Artificial neural networks and machine Learning–ICANN 2011*, *6792*(6792), 293–300.

Depren, O., Topallar, M., Anarim, E., y Ciliz, M. (2005). An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert systems with Applications*, *29*(4), 713–722. doi: 10.1016/j.eswa.2005.05.002

Dordoigne, J. (2015). *Redes informáticas-nociones fundamentales* (5.ª ed.). Ediciones ENI.

Dulaney, E. (2012). *Seguridad informática, COMPTIA SECURITY+*. Editorial Anaya Multimedia-Anaya Interactiva.

Esparza, J. (2013). *Implementación de un firewall sobre plataforma linux en la empresa de contabilidad armas & asociados* (B.S. thesis). Universidad de Quito.

Gonzalez, V. (2010). *Dectector de intrusos basado en sistema experto* (Tesis Doctoral no publicada).

Heady, R., Luger, G., Maccabe, A., y Servilla, M. (1990). *The architecture of a network level intrusion detection system*. University of New Mexico. Department of Computer Science. College of Engineering.

Horng, S., Su, M., Chen, Y., Kao, T., Chen, R., Lai, J., y Perkasa, C. (2011). A novel intrusion detection system based on hierarchical clustering and support vector

machines. *Expert systems with Applications*, *38*(1), 306–313. doi: 10.1016/j .eswa.2010.06.066

K., A., Eibe, F., Pfahringer, B., y Holmes, G. (2004). Multinomial naive bayes for text categorization revisited. En *Australian conference on artificial intelligence* (Vol. 3339, pp. 488–499).

Khan, L., Awad, M., y Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. *The International Journal on Very Large Data Bases*, *16*(4), 507–521. doi: 10.1007/s00778-006-0002-5

Kohavi, R. (1995). The power of decision tables. *Machine learning: ECML-95*, *912*(912), 174–189.

León-Jaramillo, M. (2011). *Wireless hacking, protegido con IDS, firewalls y honeypots* (B.S. thesis). Quito: Universidad Israel.

Liao, H., Lin, C., Lin, Y., y Tung, K. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), 16–24. doi: 10 .1016/jnca.2012.09.004

López, J. (2009). *Optimización de sistemas de detección de intrusos en red utilizando técnicas computacionales avanzadas* (Vol. 266). Universidad Almería.

M., V., Becerra, I., y Guevara-Juárez, O. (2010). Virus informáticos, todo un caso, pero no perdido. *CienciaUAT*, *4*(4), 56–61.

Mifsud, E. (2012). Introducción a la seguridad informática. *Observatorio Tecnológico*, *9*(18), 1–8.

Miranda, C. (2014). *Redes telemáticas*. Ediciones Paraninfo, SA.

Ochoa, J. (2013). *Función picadillo determinista al grupo g2 y su aplicación en autenticación para dispositivos móviles* (B.S. thesis). Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional.

Patil, T., y Sherekar, S. (2013). Performance analysis of naive bayes and j48 classification algorithm for data classification. *International Journal of Computer Science and Applications*, *6*(2), 256–261.

Pellejero, I., Andreu, F., y Lesta, A. (2006). *Fundamentos y aplicaciones de seguridad*

en redes WLAN: de la teoría a la práctica. Marcombo.

Portnoy, L., Eskin, E., y Stolfo, S. (2001). Intrusion detection with unlabeled data using clustering. En *In proceedings of ACM CSS Workshop on data mining applied to security (dmsa-2001.*

Rivero, J., Ribeiro, B., y Kadir, H. (2016). Comparación de algoritmos para detección de intrusos en entornos estacionarios y de flujo de datos. *Revista Universidad y Sociedad*, *8*(4), 32–42.

Rosado, C. (2014). *Virtualización de una red lan con servidores de código abierto para evaluar los niveles de seguridad* (B.S. thesis). Universidad Católica de Santiago de Guayaquil.

Salamanca, O. (2017). Sistema de gestión de seguridad para redes de área local para empresas desarrolladoras de software. *Enlace*, *13*(3), 114–130.

Sanchez, A. (2012). *Implementación de la criptografía basada en atributos en un dispositivo móvil* (B.S. thesis). Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional.

Santillán-Arenas, J. (2014). Frameworks para monitoreo, forense y auditoría de tráfico de red-i.

Santillán Arenas, J. (2015). Frameworks para monitoreo, forense y auditoría de tráfico de red-ii (poc). *Seguridad Cultura de prevención TI*, *1*(24), 8–20.

Simon, D. (2003). *Reconocimiento automático mediante patrones biométricos de huella dactilar* (Tesis Doctoral no publicada). Telecomunicacion.

Smith, B., y Komar, B. (2003). *Seguridad en microsoft windows: kit de recursos*. Editorial McGarw-Hill.

Snapp, S., Brentano, J., Dias, G., Goan, T., Heberlein, L., Ho, C.-L., ... others (1991). Dids (distributed intrusion detection system)-motivation, architecture, and an early prototype. En *Proceedings of the 14th national computer security conference* (Vol. 1, pp. 167–176).

Stewart, B. (2007). *CCNP BSCI Official Exam Certification Guide (Exam Certification Guide)*. Cisco Press.

Tamayo, J. (2016). Usuarios y hackers, un riesgo en la transmisión de datos financieros en Colombia. *ID Revista de investigaciones*, *2*(2), 90–99.

Tejada, E. (2015). *Gestión de servicios en el sistema informático. ifct0509*. IC Editorial.

Tolosi, L., y Lengauer, T. (2011). Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, *27*(14), 1986–1994. doi: 10.1093/bioinformatics/btr300

Zhang, Z., Li, J., Manikopoulos, C., Jorgenson, J., y Ucles, J. (2001). Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. En *Proc. ieee workshop on information assurance and security* (pp. 85–90).

Zhu, Y., Liang, J., Chen, J., y Ming, Z. (2017). An improved nsga-iii algorithm for feature selection used in intrusion detection. *Knowledge-Based Systems*, *116*, 74–85.