Diseño e Implementación de un Sistema de Detección de Intrusiones para redes Wifi usando herramientas de Big Data y Machine Learning

Naula López, Edgar Rodrigo

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

Ing. Romero Gallardo, Carlos Gabriel

8 de junio del 2021

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Curiginal

Document Information

Analyzed document	Tesis_Escrito_Final_indice.docx (D110456433)
Submitted	7/14/2021 5:01:00 PM
Submitted by	
Submitter email	cgromero@espe.edu.ec
Similarity	1%
Analysis address	cgromero.espe@analysis.urkund.com

Sources included in the report

SA	**TESIS_ESTEBAN_OTAÑEZ (1).docx** Document TESIS_ESTEBAN_OTAÑEZ (1).docx (D96300799)	2
SA	**20-21_2_ASIX_M14B_Sergio_Guillen_Llaneza.pdf** Document 20-21_2_ASIX_M14B_Sergio_Guillen_Llaneza.pdf (D107940780)	1
W	URL: http://repositorio.puce.edu.ec/handle/22000/13137 Fetched: 7/14/2021 5:01:00 PM	1

Firmado electrónicamente por:
CARLOS GABRIEL ROMERO GALLARDO

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación, "**Diseño e Implementación de un Sistema de Detección de Intrusiones para redes Wifi usando herramientas de Big Data y Machine Learning**" fue realizado por el señor **Naula López Edgar Rodrigo,** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 8 de junio del 2021

Firma:

Firmado electrónicamente por:
CARLOS GABRIEL
ROMERO GALLARDO

.....

Ing. Romero Gallardo, Carlos Gabriel

C. C: 1712198066

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

RESPONSABILIDAD DE AUTORÍA

Yo, **Naula López Edgar Rodrigo,** con cédula de ciudadanía n°1721132999, declaro que el contenido, ideas y criterios del trabajo de titulación: **Diseño e Implementación de un Sistema de Detección de Intrusiones para redes Wifi usando herramientas de Big Data y Machine Learning,** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 8 de junio del 2021

.....

NAULA LÓPEZ EDGAR RODRIGO

C.C.: 1721132999

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Naula López Edgar Rodrigo**, con cédula de ciudadanía n°1721132999, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño e Implementación de un Sistema de Detección de Intrusiones para redes Wifi usando herramientas de Big Data y Machine Learning**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son mi responsabilidad.

Sangolquí, 8 de junio del 2021

NAULA LÓPEZ EDGAR RODRIGO

C.C.: 1721132999

Dedicatoria

A todos quienes permitieron que esto haya sido posible. En especial al Padre, todopoderoso y la Madre Dolorosa por ayudarme y acompañarme siempre. A mis padres por ser mi ejemplo y apoyo. A mis hermanas por aliento brindado. A mis tías Lidia y Noemi por ser pieza fundamental para la realización de este trabajo.

Edgar Rodrigo

Agradecimiento

En primer lugar, a Dios y mi Madre Dolorosa por acompañarme en los momentos más duros de mi vida y darme fuerza para salir adelante.

A mis padres por el apoyo, cariño y amor incondicional brindado durante mis 23 años de vida.

A mis hermanas alegrar mis días.

A mis tías, Lidia y Noemi por apoyarme en uno de los momentos más difíciles que he atravesado.

A mi tutor, Ing. Carlos por apoyarme en la realización de esta tesis y por todos los conocimientos brindados.

Edgar Rodrigo

Índice de Tablas

Índice de Figuras

Resumen

Las redes inalámbricas Wifi hoy en día son las redes que más abundan debido a los beneficios que estas presentan, como son el de movilidad. Sin embargo, a pesar de que el estándar que define su funcionamiento, el estándar 802.11, se desarrolló hace muchos años, este todavía cuenta con falencias y vulnerabilidades que no han sido corregidas hasta el día de hoy. Hoy en día existen diferentes ataques a redes inalámbricas Wifi que comprometen uno de los aspectos claves que toda red debe poseer: disponibilidad. Ataques de denegación de servicio comprometen y entorpecen la comunicación entre los dispositivos que pertenecen a la red inalámbrica Wifi. Además de estos ataques de denegación de servicio, existen otros tipos de ataques que intentan apropiarse de los datos sensibles de los mismos, usando técnicas y métodos que engañan a los usuarios y hacen que estos se conecten a redes falsas creadas por atacantes. El presente proyecto tiene como objetivo diseñar e implementar un sistema de detección de intrusiones para detectar ataques a redes inalámbricas Wifi. Para la implementación del sistema se utilizó herramientas de Big Data como son: Apache Spark, Kafka y Elasticsearch. El sistema usa el modelo de machine learning, Random Forest, para clasificar todo el tráfico de la red y diferenciar tramas normales de las tramas maliciosas creadas por un atacante. Los resultados se analizan y visualizan en un Dashboard creado en Kibana.

Palabras Clave:

- **BIG DATA**
- **MACHINE LEARNING**
- **SISTEMA DE DETECCIÓN DE INTRUSIONES**

Abstract

Wi-Fi wireless networks are nowadays the most abundant networks due to the benefits they offer, such as mobility. However, even though the standard that defines their operation, the 802.11 standard, was developed many years ago, it still has flaws and vulnerabilities that have not been corrected to date. Today there are several attacks on wireless Wi-Fi networks that compromise one of the key aspects that every network must have: availability. Denial-of-service attacks compromise and hinder communication between devices belonging to the wireless Wi-Fi network. In addition to these denial-of-service attacks, there are other types of attacks that attempt to appropriate sensitive data from them, using techniques and methods that deceive users and make them connect to fake networks created by attackers. The present project aims to design and implement an intrusion detection system to detect attacks on wireless Wi-Fi networks. For the implementation of the system Big Data tools were used such as: Apache Spark, Kafka and Elasticsearch. The system uses the machine learning model, Random Forest, to classify all network traffic and differentiate normal frames from malicious frames created by an attacker. The results are analyzed and visualized in a Dashboard created in Kibana.

Keywords:

- **BIG DATA**
- **MACHINE LEARNING**
- **INTRUSION DETECTION SYSTEM**

Capítulo I

Introducción

Antecedentes

Hoy en día la conexión inalámbrica ha superado enormemente a la conexión a través de cables, en gran medida por todas las ventajas que el medio inalámbrico ofrece, especialmente el de la movilidad y portabilidad (Osterhage, 2018). Es por ello, que la familia de redes inalámbricas con el estándar IEEE 802.11 están presentes en casi todos los lugares en los cuales se requiere una conexión a internet, desde empresas, pequeñas oficinas y hogares familiares. Otro de los aspectos importantes que afianza aún más la tecnología inalámbrica es el internet de las cosas (IoT, por sus siglas en inglés), la cual tiene como objetivo conectar diversos dispositivos a la red.

Otro campo importante que ha tomado mucha relevancia en estos años es el campo de Ciberseguridad. No solo la conectividad es importante, sino también cuidar la información que se comparte en la red es otro de los aspectos, sino el más importante, que se deben tener en cuenta. Es deseable para cualquier red que solo los usuarios autorizados tengan acceso a la misma, lo cual, evidentemente es algo muy complicado de lograr en las redes inalámbricas por su característica inherente que la distingue. Es por ello, que estas son fácilmente vulnerables a diversos ataques, tanto pasivos como activos, pues sencillamente se podría llegar a escuchar todo el tráfico de una red, o, realizar ataques de denegación de servicio.

A pesar de todas los avances y esfuerzos de todos los protocolos que año tras años son actualizados por diversas instituciones (*IEEE* por ejemplo) que se encargan de desarrollar mejoras a cada estándar, siempre existen vulnerabilidades que los hackers puede explotar, pues, en primera instancia descubrieron un sin número de vulnerabilidades en el

protocolo de encriptación WEP (Wired Equivalent Privacy , por su siglas en inglés) que comprometían seriamente la seguridad de las redes inalámbricas que utilizaban este protocolo (Fleck & Potter, 2002), y aún existen vulnerabilidades en los protocolos más actuales como WPA/WPA2.

Justificación e Importancia

Las redes inalámbricas, en especial las definidas por el estándar 802.11, son las que predominan en casi todos los ámbitos: grandes empresas, universidades, hogares y espacios públicos. Son estas redes las que utilizan los usuarios para acceder a información sensible, como por ejemplo, cuentas bancarias o información privada perteneciente a una empresa, por lo cual, se vuelve fundamental e indispensable proteger estas redes para proteger los datos que usan el medio inalámbrico, el cual es inseguro y al cual puede acceder cualquier persona. Este último aspecto mencionado es la vulnerabilidad más explotada por hackers, ya que estos pueden fácilmente "escuchar" todo el tráfico de la red sin que el administrado o alguien se puede percatar de esta acción , es decir, es indetectable. Además, actualmente existen herramientas de fácil acceso para atacar las redes inalámbricas; Kali Linux, es una conocida herramienta utilizada para pruebas de penetración, la cual cuenta con un sin número de herramientas (Aircrack-ng, por ejemplo) que permiten descubrir puntos débiles de las redes inalámbricas, e incluso vulnerar los protocolos de encriptación. Estas herramientas, como se mencionó, son de fácil acceso y podrían ser utilizadas incluso por la persona con pocos o casi ningún conocimiento sobre redes inalámbricas, lo cual supone un riesgo, que para una empresa, por ejemplo, supone un punto a tener muy en cuenta, pues manejan información sensible que podría comprometer a la empresa misma (Ramachandran & Buchanan, 2015).

Alcance del Proyecto

El proyecto tiene como finalidad desarrollar e implementar un sistema de detección de intrusos en tiempo real para redes inalámbricas que utilizan el estándar 802.11, más comúnmente conocidas, como redes Wifi. Para la implementación del sistema de detección de intrusos se va a utilizar herramientas utilizadas en Big Data y un modelo de Machine Learning. Por la parte de Big Data, tenemos soluciones como Apache Spark, que permite procesar grandes cantidades de datos y además cuenta con un módulo de machine learning que permite implementar modelos de aprendizaje supervisado; Apache Kafka, que es un sistema de mensajería que se utilizará para enviar todo el tráfico en tiempo real a Spark para su debido procesamiento; Elasticsearch, que desde un punto de vista muy general, es un sistema de almacenamiento y búsqueda que servirá para almacenar todo el tráfico de la red inalámbrica procesado y que se podrá analizar y visualizar en Kibana, que es una herramienta tipo Dashboard que complementa a Elasticsearch. El sistema de detección tiene como objetivo detectar ataques a redes inalámbricas Wifi en tiempo real y alertar al usuario sobre este tipo de ataque.

Objetivos

Objetivo general

Desarrollar e implementar un sistema de detección de intrusos para monitorear redes inalámbricas en tiempo real que permita ataques a la red usando herramientas de Big Data y Machine Learning.

Objetivos Específicos

Investigar y analizar los principales ataques que se realizan en redes inalámbricas con el estándar 802.11.

Configurar e instalar Apache Kafka en un dispositivo para que actúe como sensor de la red y capture todo el tráfico de la red inalámbrica Wifi.

Generar ataques de denegación de servicio e Impersonificación para analizar el rendimiento del sistema de detección de intrusiones.

Entrenar el modelo de Machine Learning, Random Forest, con la base de datos AWID

Organización

El presente proyecto está divido de la siguiente manera: en el primer capítulo se presenta una introducción del proyecto, y a su vez se presenta la justificación e importancia que motivan el desarrollo de este. También se presenta el alcance del proyecto y sus objetivos, tanto general como específicos.

En el segundo capítulo se detalla el marco teórico del proyecto, el cual presenta todos los aspectos necesarios que se necesitan comprender para desarrollar el proyecto y entenderlo. Se describen las herramientas de Big Data utilizadas y el rol que cumplen en el sistema de detección de intrusiones, además de presentar los ataques y vulnerabilidades que existen actualmente en las redes inalámbricas Wifi.

En el tercer capítulo se describe el sistema de detección de intrusiones y su implementación. Se detalla su funcionamiento y todos los parámetros considerados para desarrollar el mismo, además de explicar de cómo está conformado y su funcionamiento.

En el cuarto capítulo se presenta el análisis del sistema de detección puesto a prueba en tiempo real bajo ataques a la red Wifi. Se describe que ataques fueron realizados y se examina el funcionamiento del sistema.

Por último, en el quinto capítulo se presenta las conclusiones del proyecto y recomendaciones. A su vez se presentan trabajos a futuro que deberían ser considerados para futuros desarrollos o mejoras del sistema de detección de intrusos.

Capítulo II

Estado del Arte

Estándar 802.11

El estándar 802.11 fue desarrollado por la *IEEE* (Institute of Electrical and Electronics Engineers, por sus siglas en inglés) y oficialmente lanzado al mercado en el año de 1997. Este estándar, es el conjunto de una serie de protocolos que forjan la base para cualquier red inalámbrica Wifi. Esta serie de protocolos describe los primeros dos niveles de la arquitectura del modelo OSI: la capa física y la capa de enlace (Gast, 2002) .

Arquitectura y Topología del estándar 802.11

Una red con el estándar 802.11 puede estar formada por diversas estaciones que se comunican entre sí y dan origen a lo que se conoce como WLAN (Wireless Local Area Network, por sus siglas en inglés). Los protocolos del estándar 802.11 permiten que estas estaciones puedan comunicarse entre sí, es decir, establece reglas para la conectividad entre los dispositivos y acceso al medio (Rackley, 2007). Los elementos más básicos que conforman una red inalámbrica con el estándar 802.11 son: estaciones (STA, abreviatura utilizada para la palabra en inglés Station) y los puntos de acceso (Access Point, por sus siglas en inglés). A la unión de estaciones dentro de un área de cobertura, se le denomina BSA (Basic Service Area, por sus siglas en inglés), en la cual, estos dispositivos pueden comunicarse entre sí y se tiene definido una función de coordinación.

Otro aspecto fundamental que describe el estándar 802.11, y es esencial para comprender la arquitectura y topología del estándar 802.11, es la función de coordinación. Esta última hace referencia básicamente a la lógica aplicada a cada dispositivo que determina cuando

una estación puede acceder al medio , pues, el medio inalámbrico es compartido por todas las estaciones y para evitar errores en la transmisión, debido a colisiones de tráfico, es necesario establecer precisamente una lógica de coordinación entre todas las estaciones que pertenecen a la WLAN. El estándar define dos funciones de coordinación: DFC (Distributed Coordination Function, por sus siglas en inglés) , el cual establece que cada dispositivo primero debe sensar el medio antes de transmitir y hacerlo solo en el caso de que el medio se encuentre desocupado, y PCF (Point Coordination Function) en el cual un dispositivo, generalmente el AP controla el acceso al medio y el envió de datos a cada dispositivo (Hiertz & Denteneer, 2010).

El estándar presenta dos topologías principales: IBSS y BSS. En la topología Independient Basic Service Set (IBSS) las estaciones pertenecientes a la misma se comunican entre si sin utilizar un punto que controle la comunicación, es decir, la comunicación entre las estaciones pertenecientes a la red no utiliza ningún punto de acceso para comunicarse con otros dispositivos.

Figura 1

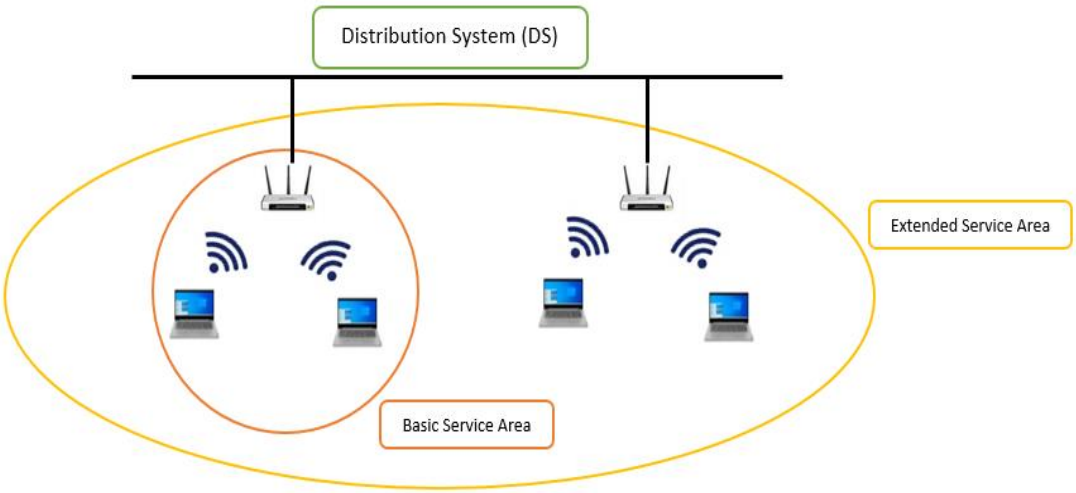
Comunicación inalámbrica entre dos dispositivos

En la topología BSS (Basic Service Set, por sus siglas en inglés) o modo infraestructura, a diferencia de IBSS, existe un nodo en la red mediante el cual todas las otras estaciones

pertenecientes a la red pueden conectarse y comunicarse entre sí. Cabe recalcar que se puede dar el caso en que se puedan conectar los puntos de acceso que permiten la conectividad en una BSS, es decir, se puede conectar un AP para poder extender la cobertura de la red (Póser & Kozlovszky, 2019). En este caso se le denomina ESS (Extented Service Set, por sus siglas en inglés).

Figura 2

Topología BSS de una red inalámbrica



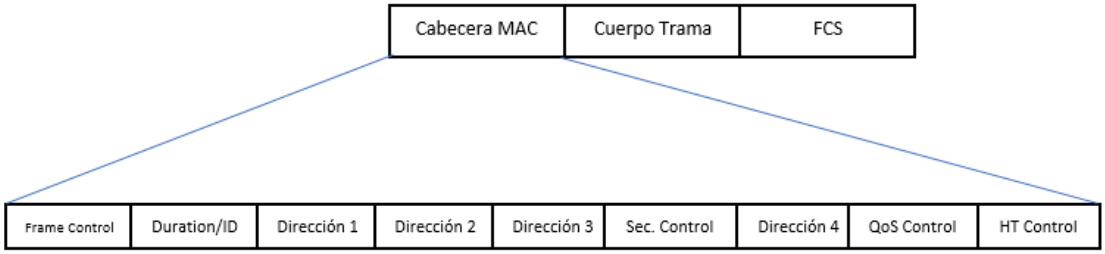
Estructura de una trama 802.11

El estándar 802.11 se encuentra las capas física y MAC del modelo OSI. El objetivo del estándar 802.11 es establecer protocolos que permitan la transmisión de datos. La forma en que viajan los datos a través del medio es a través de lo que se conoce como trama MAC. Esta trama MAC está formada por campos previamente establecidos: en primer lugar se encuentra la cabecera MAC, seguidamente está el cuerpo de la trama, la misma que tiene una longitud que varía, pues esta contiene información que describe el tipo de trama (Gast, 2002), y finalmente

se tiene al código de redundancia cíclica CRC, que también se lo conoce como FCS (Frame Check Sequence, por sus siglas en inglés).

Figura 3

Estructura de una trama 802.11



A continuación se describirá brevemente en detalle cada campo que compone la trama MAC.

Frame Control

El primer campo de la cabecera de la trama MAC es el *Frame Control*, el cual tiene una longitud de dos bytes y contiene la siguiente información:

- Protocol Version: Este es un campo de 2 bits. El valor por defecto de este campo es 0, por lo cual todos los valores restantes están reservados para futuras versiones del protocolo.

Figura 4

Cabecera de la trama 802.11

- Type: Tiene 2 bits de longitud y este campo indica el tipo de trama; puede ser de gestión (valor binario 00), control (valor binario 01), datos (valor binario 10) y reservado (valor binario 11) (Gast, 2002).

- Subtype: Tiene una longitud de 4 bits y juntamente con el campo Type , identifican el tipo de trama, pues, cada tipo de trama ya sea control, gestión o de datos, tienen diferentes subtipos que tiene un propósito especifico, por ejemplo, ya sea para autenticarse a la red o para enviar avisar a los demás dispositivos que se va a transmitir información (Gast, 2002).

- To DS y From Ds: Campos de una longitud de un solo bit cada uno, y sirven para poder reconocer la dirección de la comunicación (Gast, 2002). A continuación se detallan las posibles combinaciones:

Tabla 1

Combinaciones para el campo To/From DS

Combinaciones To/From DS	**Significado**
To DS = 0 From DS = 0	Trama enviada de una estación a otra en un IBSS. En las tramas de control y gestión estos bits tienen un valor de 0
To DS = 1 From DS = 0	Trama que tiene como destino DS
To DS = 0 From DS = 1	Trama que es enviada por el DS
To DS = 1 From DS = 1	Trama es enviada de un AP a otro AP en un DS inalámbrico

- More Fragments: Campo de un solo bit. Sirve de flag para poder reconocer que la trama que se está enviado es de un solo fragmento, y posteriormente se enviaran el resto de los fragmentos con los cuales se reconstruirá toda la trama. Exclusivamente se aplica solo a tramas del tipo de gestión o de datos (Gast, 2002). Si este flag tiene un valor de 0 es porque no ha sido fragmentado.

- Retry: Campo de longitud de 1 bit. Este campo sirve para eliminar tramas duplicadas, y tiene un valor de 1 en todas las tramas de gestión o datos cuando estas son una réplica de una trama que se envió anteriormente (Gast, 2002).

- Power Management: Campo de un solo bit. Este bit se utiliza para establecer el estado de energía en el que se encontrara la estación que realiza una transmisión después de finalizar la misma (Gast, 2002). Cuando este campo tiene un valor de 1, indica que pasará a un estado de modo de ahorro de energía, por el contrario, si el valor es de 0, indica que después de la transmisión continuará en estado activo.

- More Data: Campo de un solo bit. Si tiene un valor de 1, es un flag que indica que existe más información que la estación emisora desea enviar a la estación receptora (Gast, 2002).

- Protected Frame: Campo de 1 bit. Flag que indica que el campo Frame Body ha sido encriptado con un algoritmo de cifrado (Gast, 2002). Esto solo aplica para tramas del tipo de datos o gestión, pero de esta última solamente al tipo de Autenticación.

- Order: Campo de un solo bit. Tiene un valor de 1 para indicar a la estación que recibe la información que la procese según el orden en el cual lleguen las tramas (Gast, 2002).

Duration/ID

Campo que tiene una longitud de 2 bytes. Este campo puede ser utilizado para almacenar información con respecto a la duración o el ID de la trama, lo cual varía según el tipo de esta. Para las tramas del tipo Power Save, contienen el identificador de la estación. Para las otras tramas, contiene el valor reservado para la transmisión, el cual servirá para el cálculo de NAV (*Network* Allocation Vector, por sus siglas en inglés).

Campos Address 1-4

Estos campos contienen las direcciones MAC, las cuales tienen una longitud de 48 bits, de la estación que genera la trama (Source Address), la estación a la cual está dirigida la trama (Destination Address), la estación que transmitirá la trama (Transmitter Address) y la que recibirá la trama (*Receiver Address*) (Gast, 2002).

Tabla 2

Tabla de Direcciones para el DS

To DS	**From DS**	**Dirección 1**	**Dirección 2**	**Dirección 3**	**Dirección 4**
0	0	Destino	Origen	BSSID	N/A
0	1	Destino	BSSID	Origen	N/A
1	0	BSSID	Origen	Destino	N/A
1	1	Receptor	Transmisor	Destino	Origen

Control de Secuencia

Este campo, de dos bytes, que está conformado por dos indicadores: número de fragmento y el número de secuencia. El primero indica el número de fragmento que le corresponde a esta trama, dicho de otra forma, tiene un valor de 0 para el primer fragmento, y se irá incrementado sucesivamente y conserva el mismo valor para todas las retransmisiones del mismo segmento. El segundo indicador identifica el número de secuencia de un MSDU. Este campo es útil cuando se desea enviar una trama muy larga, por lo cual es necesario fragmentarla y se las envía independientemente. Con este campo se puede retransmitir los fragmentos que se enviaron con errores (Gast, 2002).

Frame Body

Este campo tiene una longitud variable. El rango en el cual puede variar es de 0 bytes hasta 2312 bytes. Este campo contiene información que dependerá al tipo y subtipo de trama, ya sea de gestión, control o si es una trama de datos (Gaitán, 2017) .

FCS

Tiene una longitud de 32 bytes. Este campo contiene el algoritmo CRC-32 para realizar el checksum, lo cual permitirá detectar errores en los bits de la trama enviada (Gaitán, 2017).

Tipos de Tramas

Tramas de Gestión

Este tipo de tramas permiten establecer la comunicación entre cualquier dispositivo y el AP. Además de iniciar la comunicación, es importante también mantenerla activa para que el dispositivo conectado puede transmitir información en cualquier momento; esta función también se lo realiza a través de las tramas de gestión (Gaitán, 2017).

Para lograr iniciar la comunicación, el AP envía tramas Beacon en toda el área de cobertura para poder advertir de su presencia. En este tipo de trama el AP envía información que será usado por cualquier dispositivo para iniciar la comunicación , por ejemplo: nombre del AP, tasa de transmisión y tipo de encriptación soportada por el AP. Después de establecer la comunicación y cuando el dispositivo desea dar por terminada la comunicación, se envían tramas de gestión del tipo de desautenticación para permitir abandonar la red. Existen otros tipos de tipos de tramas de gestión que cumplen diversas funciones siempre en dirección a establecer la comunicación (Gast, 2002). A continuación se mencionan:

Tabla 3

Tipos de tramas de Gestión

Tipo	**Subtipo**	**Nombre**
00	0000	Association Request
00	0001	Association Response
00	0010	Reassociation Request
00	0011	Reassociation Response
00	0100	Probe Request
00	0101	Probe Response
00	1000	Beacon
00	1011	Authentication

Tramas de Control

Este tipo de tramas controlan el acceso al medio y controlan el flujo de tramas entre el AP y cualquier dispositivo conectado o viceversa. Este tipo de tramas permiten evitar colisiones en el medio y de esta manera poder controlar todas las comunicaciones para que cada dispositivo pueda acceder al medio de manera ordenada (Gaitán, 2017). Los tipos de tramas de control son:

Tabla 4

Tipos de tramas de Control

Tipo	**Subtipo**	**Nombre**
01	1011	Request to Send
01	1100	Clear to Send
01	1101	ACK

Tramas de Datos

Las tramas de datos son usadas para transmitir la información que se produce en capas superiores, desde la capa 3 hasta la capa 7 según el modelo OSI. Las tramas de control transmiten los datos a través de lo que se conoce como MSDU (Mac Service Data Unit, por sus siglas en inglés). Todas las tramas de datos son encriptados, para elevar la seguridad de la comunicación, dependiendo del tipo de encriptación soportado por el AP. Si bien es cierto que estas tramas solo se utilizan para transmitir información, también estas dependen o varían, pues

estas pueden traer información adicional o tener habilitado QoS (Quality of Service, por sus siglas en inglés), además de que existen tramas de datos nulas las cuales no contiene información y más bien se transmiten exclusivamente de un STA (dispositivo conectado a la red) hacia un AP para comunicar un cambio en su estado de inactividad (Vallejo, 2016), generalmente generado cuando este se pone en modo ahorro de energía.

Seguridad

El gran riesgo de las redes inalámbricas se debe precisamente a que el medio es accesible para cualquiera que se encuentre en el área de cobertura de la red, incluso, podría cubrir mucho un área mucho más grande de lo planificado. A diferencia de una red cableada, una red inalámbrica no se le puede brindar seguridad físicamente, como sucede en el caso de una red cableada en la cual cualquier usuario que quiera conectarse a la red primero debe conectarse mediante un cable, en una red inalámbrica cualquiera que se encuentre en el rango de cobertura del AP, por ejemplo, puede escuchar todo el tráfico de la red. Generalmente una red, ya sea cableada o inalámbrica, para que se la considere segura, debe cumplir con tres condiciones: disponibilidad, integridad y confidencialidad. Disponibilidad hace referencia a que el usuario debe poder acceder a la red cuando él lo requiera, en cualquier tiempo (Osterhage, 2018). El factor que más afecta la disponibilidad de la red es la interferencia y otro tipo de ataques como denegación de servicio que se describirá en breve. Integridad se refiere a que los datos deben ser enviados al destino sin que estos sufran ningún cambio. La confidencialidad juega un rol importante en cualquier red, ya que es importante que solo los usuarios autorizados pueden acceder a la información (Osterhage, 2018). Como se ha mencionado, en una red inalámbrica cualquiera que se encuentre dentro del rango de cobertura puede escuchar el tráfico, por lo cual una medida para asegurar los datos es la encriptación.

A pesar de que algunos de los ataques se centran en las vulnerabilidades presentes en las capas física y MAC, el fin último de cualquiera que quiera comprometer la red es acceder o afectar los datos de la capa de aplicación (Fleck & Potter, 2002). A continuación se describen algunos de los ataques que se podrían realizar sobre las redes inalámbricas 802.11 en las capas física y MAC (Elhigazi, y otros, 2020):

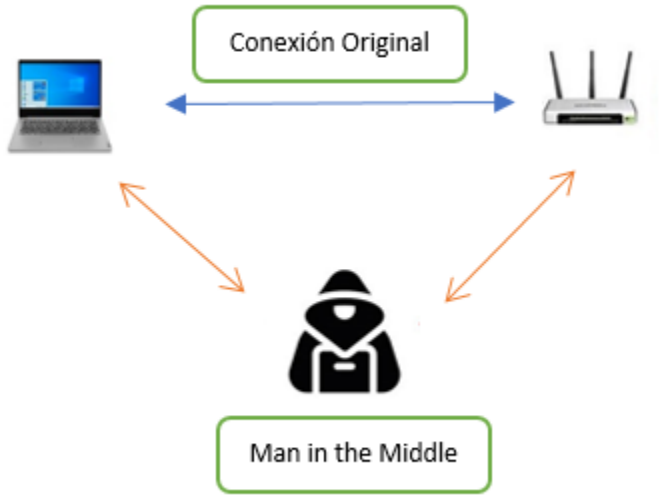
- Denegación de Servicio (DoS): un atacante puede inundar la red con tráfico excesivo por lo cual la disponibilidad de la red inalámbrica se ve afectada. La mayoría de los ataques de denegación de servicio utilizan el mecanismo de enviar masivamente tramas de gestión previamente falsificadas (Dasari & Univeristy, 2017). Esto se debe a que las tramas de gestión se transmiten sin utilizar ningún de seguridad para encriptarlas. El ataque más conocido y potente es el ataque de desautenticación, ya que es fácil de llevar a cabo y es muy eficiente (Thing & Cluster, 2017). Como se mencionó los paquetes de gestión no tienen protección, por lo cual puede ser fácilmente falsificados y enviados por toda la red, en especial aquellos paquetes de gestión de desautenticación. Para llevar a cabo este ataque lo único que se necesita es identificar la dirección MAC, tanto del cliente como del AP, lo cual se puede llevar cabo escuchando todo el tráfico de la red. Un atacante puede enviar paquetes de desautenticación, hacia al AP en nombre de un cliente, o viceversa, lo cual provocara que el AP ya no lo considere un cliente de la red y de esta manera pierde conectividad. También se pueden enviar paquetes de disociación, en lugar de paquetes de desautenticación, provocando los mismos resultados y usando la misma lógica que se usa para enviar paquetes de desautenticación. Cabe recalcar que estos dos tipos de ataques se puede realizar sobre un cliente o sobre todo la red. En este último escenario, un ataque puede enviar en nombre del AP, paquetes de desautenticación o disociación a todos los

clientes de la red, comprometiendo la disponibilidad y accesibilidad de todos los clientes (Ramachandran & Buchanan, 2015).

- Man-in-the-Middle: Este tipo de ataques tienen como objetivo hacer que el usuario se conecte a un falso AP en lugar del real o a una red administrada por un atacante malicioso. En el primero caso tenemos ataques como Evil Twin, en el cual una atacante puede crear una red con el mismo ESSID de una red real e intentar lograr que el usuario se conecte a su red maliciosa (Allahdadi & Morla, 2016). Generalmente las redes que suelen ser víctimas de este ataque son redes abiertas (sin ningún tipo de seguridad) y los usuarios que caen en estas redes falsas son usuarios que se encuentran muy cerca del atacante, pues el dispositivo del usuario se conectará a la red con más señal, de esta manera el dispositivo del usuario preferirá conectarse a la red falsa si es que esta tiene mayor señal que de la original. Otro tipo de ataque similares a este son los conocidos Honeypots que son AP abiertos que buscan atraer a usuarios ingenios para realizar varios ataques sobre los datos que estos envían al AP malicioso. Otra forma de este tipo de ataque son los Rogue Access Points, que son dispositivos que se instalan en la red sin autorización mediante la cual el atacante busca acceder a la red y atraer a clientes ingenuamente pues por lo general este tipo de red se suele dejar sin seguridad de autenticación (Ramachandran & Buchanan, 2015).

Figura 5

Ataque Man in the Middle



- Dictionary Attack: Este ataque es de fuerza bruta y generalmente se usa para descifrar claves débiles en redes con WPA/WPA2. Básicamente el atacante debe esperar capturar los paquetes correspondientes al handshake, que se utiliza cuando un usuario desea autenticarse a la red . Si el atacante no logra capturar el handshake, el primer paso que debe seguir es desautenticar a un usuario para que vuelva a solicitar al AP unirse a la red, proceso en el cual, el atacante escucha el tráfico y puede capturar el handshake. Una vez capturado el handshake, el atacante utiliza un diccionario para verificar la clave de la red, para lo cual utiliza el handshake para comparar cual es la clave que coincide (Ramachandran & Buchanan, 2015).

- Ataque sobre WEP: Aunque actualmente casi ninguna red inalámbrica usa el protocolo de encriptación WEP para brindar seguridad a la misma, se mencionará los ataques más comunes sobre este protocolo que permiten recuperar la clave WEP, aprovechando la

vulnerabilidad del algoritmo de encriptación que se utilizó, el RC4. Entre los ataques más comunes están: FMS, KoreK, PTW, inyección de paquetes ARP, ChopChop, ataques de fragmentación, Caffe Latte y HIre Attack. Si bien es cierto que estos ataques son efectivos en WEP, no tiene relevancia estudiarlos pues ahora prácticamente todas las redes inalámbricas han dejado obsoleto el uso de WEP (Ramachandran & Buchanan, 2015).

Big Data y Apache Spark

En estos últimos años la tecnología ha avanzado rápidamente, y junto con ella la cantidad de datos que ahora se maneja. Básicamente la tecnología está inmersa en cada aspecto del diario vivir de cualquier persona: social, personal y profesional (Salloum, Dautov, & Chen, 2016). Esto implica que en cada actividad se genera una gran cantidad de datos: empresas que buscan recopilar todo tipo de datos que su tipo de actividad puede generar, en redes sociales se ve una gran cantidad de información con la cual todos los usuarios interactúan e inclusive cuando personas hacen deporte con un reloj inteligente siempre hay algún tipo de datos que se genera y que por ende debe ser guardado, analizado o procesado de alguna manera conveniente según el tipo de este. Es por eso, que actualmente el término Big Data, cobra cada vez más importancia. Big data permite recolectar información de diferentes fuentes, ya sean sensores desplegados en diferentes lugares, aplicaciones móviles, dispositivos conectados entre sí, etc. Solo recolectarlos no es suficiente, pues para sacar provecho de dichos datos, es necesario organízalos y procesarlos para posteriormente descubrir hechos, patrones o información valiosa que los datos recolectados pueden ofrecer. Precisamente esto es lo que Big Data permite, analizar estos datos para poder sacar el mejor provecho de ellos (Chambers & Zaharia, 2018). Si bien es cierto, el almacenamiento de los datos no es un problema hoy en día, pues existe recursos al alcance de la mano que permiten lidiar con este problema. Sin embargo, en el contexto de Big Data, se deben tomar en cuenta otros aspectos: velocidad, que implica la

rapidez con la cual los datos son recolectados, almacenados y procesados; variedad de los datos, pues no todos los datos son del mismo, y el sistema debe estar en la capacidad de almacenar cualquier tipo de datos: correos electrónicos, imágenes, videos, etc.; otro aspecto fundamental de los datos es su visualización, un punto muy importante a tomar en cuenta ya que es este aspecto el que permite entender los datos.

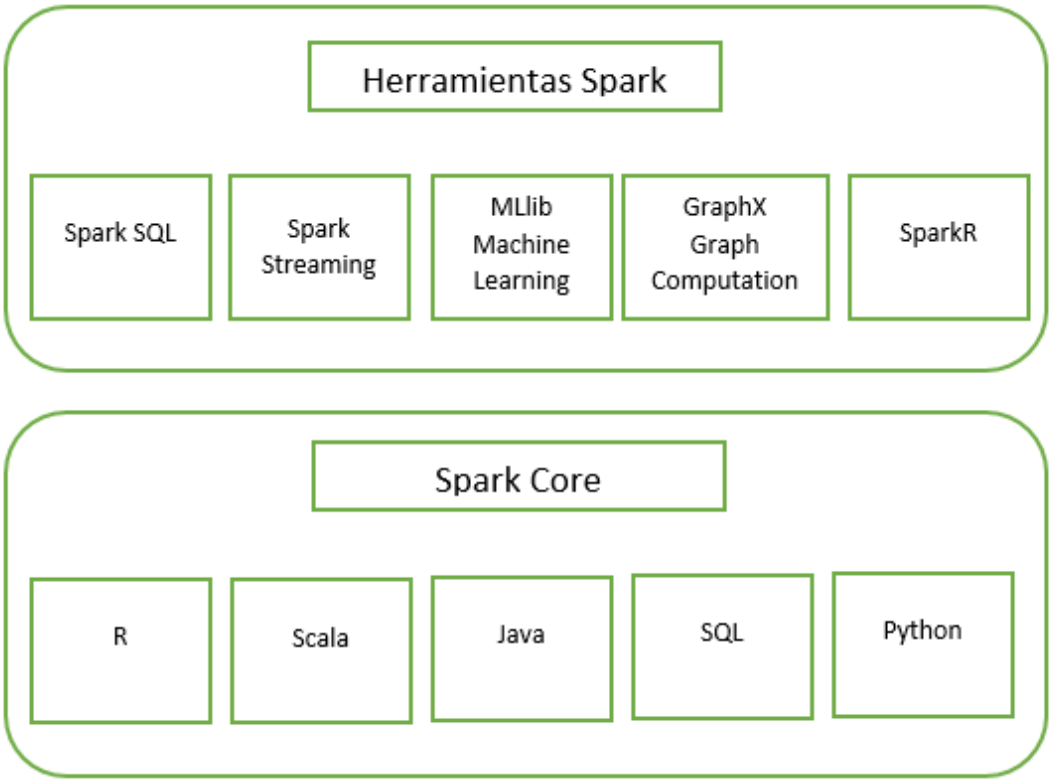
Apache Spark

Una de las herramientas de Big Data que predominan actualmente en el mercado es Apache Spark. Esta herramienta es una poderosa plataforma de procesamiento que permite manejar grandes cantidades de datos, incluso en tiempo real. No solo puede manejar grandes cantidades de datos, sino que también puede realizar el procesamiento de manera rápida y eficiente, pues utiliza el concepto de in-memory (Chambers & Zaharia, 2018), mediante el cual Spark solo accede al disco para cargar los mismos o guardar los resultados finales, es decir, todos los resultados del procesamiento de los datos se los almacena en la memoria, no directamente accediendo al disco donde se guardan los datos normalmente. Otras de las características de Spark se mencionan a continuación.

- Spark soporta una gran cantidad de lenguajes para su programación: Python, Java, Scala y R. Esto permite a los programadores a utilizar el lenguaje de su preferencia y además permite ser muy accesible a cualquier programador (Chambers & Zaharia, 2018).

Figura 6

Ecosistema de Apache Spark



- Incluye herramientas avanzadas para el análisis de los datos. Tiene librerías incluidas, como por ejemplo: machine learning, Spark streaming para analizar datos en tiempo real, GraphX para análisis gráficos y Spark SQL que soporta procesamiento de datos estructurados (Chambers & Zaharia, 2018).

- Puede correr en casi cualquier plataforma, como por ejemplo: Apache Hadoop YARN, Mesos, EC2 y Kubernetes (Chambers & Zaharia, 2018).

Los usos de Spark son muy variados. Aquí algunos ejemplos en los cuales Spark juega un papel muy importante en la industria:

- Sector de la Salud: mediante la plataforma de Spark se puede realizar un análisis de los historiales médicos de los pacientes para identificar posibles enfermedades que estos puedan sufrir en el futuro en base a su expediente médico. Un ejemplo es la aplicación MyFitnessPal que utiliza el sistema de procesamiento de Apache Spark (Chambers & Zaharia, 2018).

- Sector Financiero: Mediante el análisis de los datos de empresas o bancos, Spark permite ayudar a estas empresas sobre la toma de decisiones en aspectos esenciales como son: segmentación del mercado, riesgos de crédito bancario, etc. Los bancos además usan Big Data para predecir cuándo se produce un fraude bancario, incluso en tiempo real (Chambers & Zaharia, 2018).

- Sector E-commerce: Spark es muy usado en esta industria. Sus usos varían: recolectar y organizar todas las transacciones que se realizan en la plataforma, así como también mediante la información de cada usuario utilizar modelos de recomendaciones para las compras de los clientes. Alibaba y eBay son empresas que usan Apache Spark (Chambers & Zaharia, 2018).

- Sector del Entretenimiento: Spark permite a empresas como Netflix, Yahoo o Pinterest, reconocer patrones en la información de cada usuario en tiempo real, para poder realizar publicidad personalizada para cada usuario (Chambers & Zaharia, 2018).

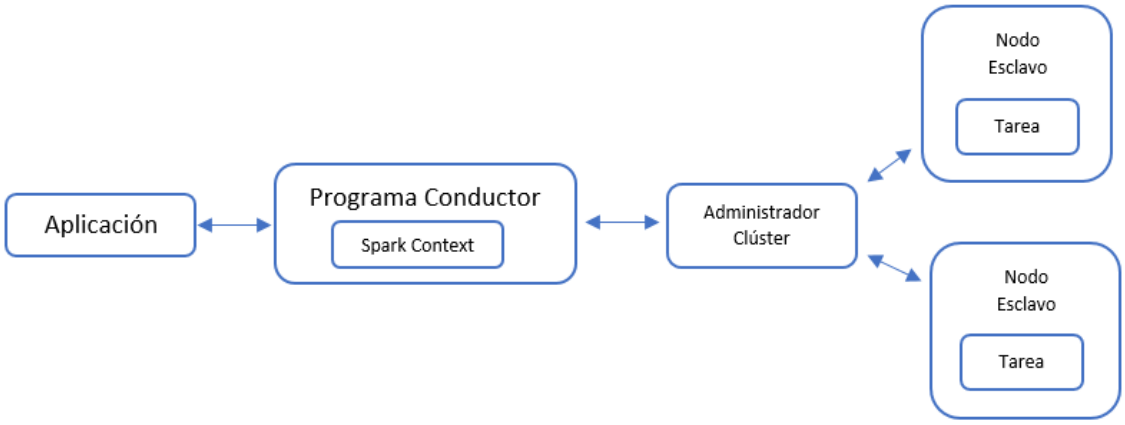
Arquitectura de Apache Spark

El elemento primordial en la arquitectura de Apache Spark es el programa conductor (Driver Program, por sus siglas en inglés). Este es el responsable por ejecutar la aplicación en sí y elaborar todas las tareas necesarias para llevar a cabo la misma. El nodo maestro es el que administra los recursos del clúster y está en contacto permanente con el programa conductor

que se ejecuta en un nodo separado al nodo maestro. Aquí en el nodo donde se ejecuta el programa conductor se crea el contexto de Spark, el cual sirve como un punto o puerta de comunicación (Gateway, por sus siglas en inglés) para todas las funciones de Apache Spark. Este contexto de Spark, permite elaborar todas las tareas que se destinaran a todos los nodos esclavos que pertenecen al clúster según la disponibilidad de recursos (Chambers & Zaharia, 2018). El nodo maestro siempre tiene en cuenta los recursos de todo el clúster, y mantiene un contacto permanente con el programa conductor para informar la disponibilidad de los nodos esclavos. Los nodos esclavos son los encargados de realizar las tareas que el nodo maestro les asigna y después de realizar dicha tarea regresa los resultados al nodo donde se ejecuta el programa conductor.

Figura 7

Arquitectura de Apache Spark



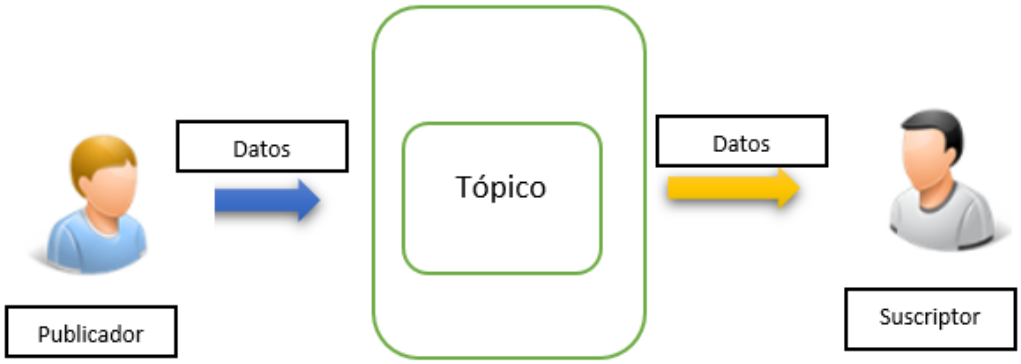
Apache Kafka

Apache Kafka es una plataforma de transmisión de datos de software libre. Esta herramienta permite recolectar datos en tiempo real de diversos lugares y reenviándolos a

todos los lugares donde sean necesarios (Garg, 2015). El modelo de funcionamiento en el que se basa Apache Kafka es el de publicación/suscripción.

Figura 8

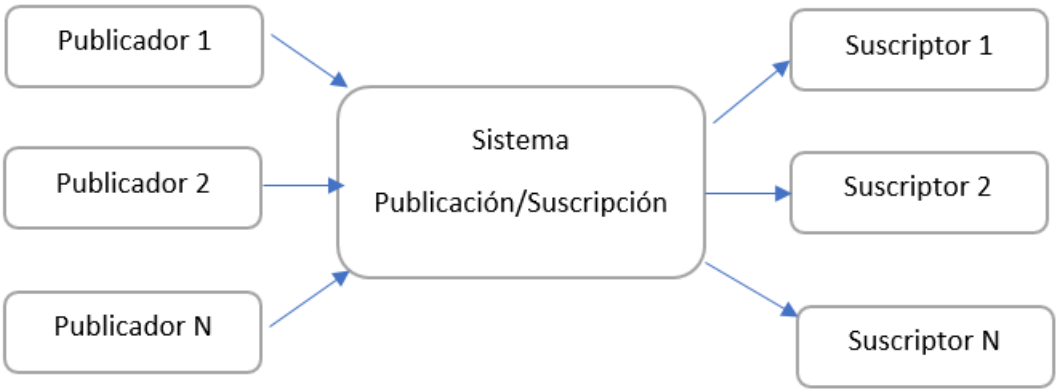
Sistema de mensajería Publicación/Suscripción



Este modelo funciona de la siguiente manera: cualquier dispositivo o aplicación (productor) envía los datos a un dispositivo (bróker) que los almacena ordenadamente un lugar específico (tópico). Posteriormente un suscritor o consumidor que desea recibir dichos datos deberá suscribirse al tópico que es de su interés y recuperar de ese tópico toda la información que ha sido publicada por el productor (Garg, 2015). Cabe recalcar que puede haber muchos productores que envían datos a un mismo o varios tópicos en el bróker y a su vez muchos consumidores suscritos al mismo. Una de las mayores ventajas de Kafka es que un sistema a prueba de fallos, pues dentro del sistema o clúster de Apache Kafka se pueden crear replicas para evitar pérdidas de datos.

Figura 9

Sistema de Publicación/Suscripción con varios usuarios



Algunas de las características de Apache Kafka se mencionan a continuación:

- Apache Kafka es un sistema de mensajería publicación/suscripción que puede resolver miles de megabytes de lecturas y escrituras en tiempo real de cientos de clientes (Garg, 2015).
- Apache Kafka es fácilmente escalable, pues está diseñado para expandirse fácilmente sin la necesidad de cesar sus operaciones (Garg, 2015).
- Su sistema soporta un alto flujo de datos (throughput), de los cuales se pueden crear réplicas, lo cual lo hace un sistema tolerante a fallos (Garg, 2015).
- Los datos almacenados permanecen el tiempo suficiente para que los consumidores puedan acceder a ellos.
- Apache Kafka tiene baja latencia en el envío de mensajes. Es un sistema muy rápido que soporta hasta 2 millones de escrituras por segundo.

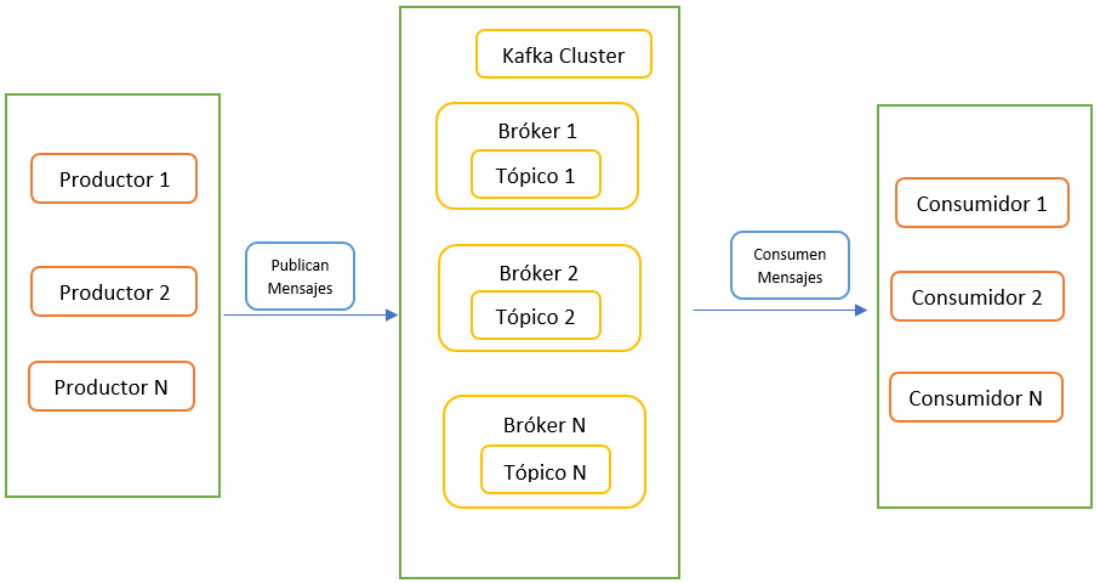
Componentes de Apache Kafka

Los elementos que conforman todo el sistema de Apache Kafka son los siguientes:

- Tópico: Los datos de los productores o publicadores son almacenados conjuntamente de acuerdo con su categoría o tipo. A este conjunto característico de un tipo de datos se le denominan tópico. Se lo puede considerar un contenedor de mensajes (Narkhede, Shapira, & Palino, 2017).
- Partición: Cada tópico esta divido en particiones, según el usuario lo requiera. Cada partición contiene los mensajes en una secuencia ordenada (Narkhede, Shapira, & Palino, 2017).
- Réplicas: Básicamente es el respaldo de los datos de un tópico.
- Bróker: Es el sistema responsable de mantener los datos publicados. Cada bróker puede tener uno o más particiones por tópico (Narkhede, Shapira, & Palino, 2017).
- Kafka Clúster: Es la unión de dos o más brókeres. Es recomendable conformar un clúster con el número suficiente número de brókeres para poder tener unos disponibles para los datos en sí y otros para utilizarlos como réplicas.
- Productores/Publicadores: Son los que publican los mensajes a uno o más tópicos. Envían los datos hacia los brókeres (Narkhede, Shapira, & Palino, 2017).
- Consumidores: Leen los datos de los brókeres. Estos se pueden suscribir a uno o más tópicos y consumir los datos publicados en los mismos (Narkhede, Shapira, & Palino, 2017).

Figura 10

Clúster de Kafka



Elasticsearch y Kibana

Elasticsearch y Kibana son parte de lo que se conoce como ELK Stack. Este último es un conjunto de tres productos de código abierto (Elasticsearch, Logstash y Kibana) que proporciona un sistema de almacenamiento y registro centralizado para identificar problemas en servidores o aplicaciones. Específicamente, Elasticsearch es un sistema de búsqueda de código abierto construido en base a Apache Lucene (Gormley & Tong, 2015) . Elasticsearch fue escrito en Java y aprovecha todo el potencial de Lucene para el sistema de búsqueda. Además Elasticsearch permite almacenar documentos en tiempo real para posteriormente poder realizar análisis en tiempo real mediante Kibana. Elasticsearch se caracteriza por ser de fácil implementación, de un alto grado de confiabilidad y almacenar todos los datos y los análisis sobre estos de manera centralizada, además de que está diseñada para trabajar sobre un gran volumen de datos

(Gormley & Tong, 2015). El principal uso que se le da a esta herramienta es el de motor de búsqueda, aunque también se puede aplicarla para los siguientes casos: búsqueda de aplicaciones, búsqueda de sitios web, análisis de logs, monitoreo de rendimiento de aplicaciones, analítica de seguridad o de negocios. A continuación se menciona algunas de las características de Elasticsearch:

- Se puede almacenar cualquier tipo de datos
- Tiene interfaz web conocida como REST API con el formato de salida JSON
- Búsquedas en tiempo real
- Soporta geolocalización
- Almacena los documentos sin esquema, basado en JSON y REST
- Puede realizar filtrado en los datos para obtener información
- Puede realizar un escalamiento vertical y horizontal
- Es un sistema de búsqueda muy rápido
- Es tolerante a fallas

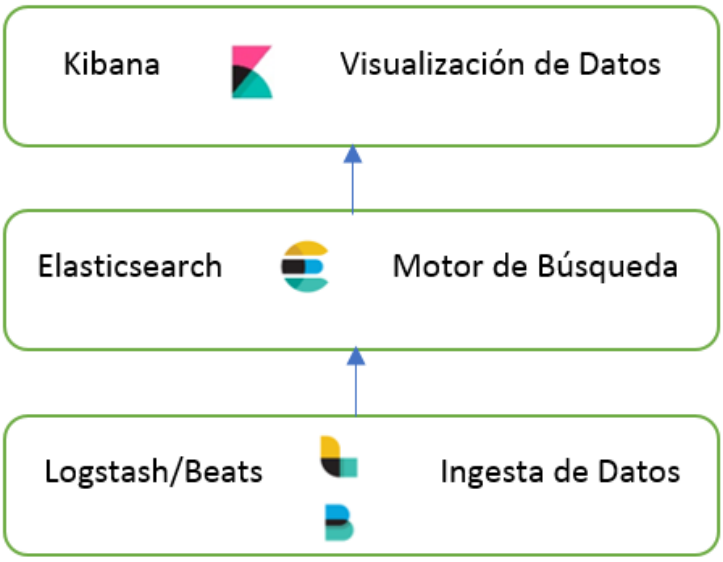
Con la ayuda de Elasticsearch se puede realizar todo lo que implica el almacenamiento y búsqueda de los datos, pero para lograr la visualización de estos datos es necesario Kibana. Kibana es una interfaz que se utilizar para visualizar todos los datos almacenados en Elasticsearch y permite a los desarrolladores tener una visión rápida de los mismos (Gormley & Tong, 2015). Su interfaz web ofrece varias herramientas: diagramas interactivos, datos geoespaciales, gráficos para visualizar datos complejos, herramientas de machine learning, diagramas y tablas del tipo estadístico y un motor de búsqueda. Mediante Kibana se pueden

realizar búsquedas e interactuar con los datos almacenados en Elasticsearch (Gormley & Tong, 2015). Las principales características de Kibana son:

- Búsqueda y visualización de los datos en tiempo real
- Ejecuta consultas sobre los datos y permite visualizarlos en tablas, gráficos y mapas
- Paneles que se pueden configurar de acuerdo con la necesidad
- Fácil visualización
- Interfaz web intuitiva y fácil de usar

Figura 11

Sistema ELK



Capítulo III

Diseño e Implementación

En el presente capitulo se describirá el procedimiento que se utilizó para el diseño e implementación del sistema de detección de intrusos en tiempo real. Se describirá como se diseñó el sistema, que función cumple cada herramienta utilizada y al final se presentará la topología final del sistema. Además se presentan también todos aspectos importantes, que si bien no son parte directamente del sistema, pero fueron fundamentales para la implementación de este, como por ejemplo se describirá el conjunto de datos que se utilizó para el entrenamiento del modelo de machine learning.

Conjunto de Datos

AWID es una base de datos publica que contiene tráfico normal y anormal de una red Wifi. Este conjunto de datos fue recolectado en un laboratorio que simulaba una típica infraestructura SOHO (Small Office Home Office, por sus siglas en inglés). Esta red estaba compuesta por diversos clientes, tantos dispositivos móviles: 2 laptops, 2 smartphones, 1 Tablet y, dispositivos estáticos: 1 Smart tv y 1 computadora de escritorio, además de contar con un dispositivo que cumplía el rol de atacante (Kolias, Kambourakis, Stavrou, & Gritzalis, 2016). El dispositivo del atacante se lo llevo a cabo mediante la distribución de Unix/Linux, Kali Linux, el mismo que permitió realizar diversos ataques a la red. La recolección de los datos lo realizó un dispositivo que no estaba conectado a la red y se encontraba en modo monitor, lo que le permitía recolectar el tráfico gracias a la ayuda de Tshark. Esta base datos consta con dos conjuntos de datos, una dedicado para el entramiento y otra que permite la validación del modelo. La base de datos de entrenamiento y la de validación contiene paquetes de diferentes

ataques que han sido agrupados en tres categorías: normal, denegación de servicio e Impersonificación. El conjunto de datos para el entrenamiento contiene 1,747,053 paquetes. El conjunto de validación está conformado por 555,564 paquetes.

Tabla 5

Clasificación de las tramas de la base de datos AWID-Training

Clase	**AWID – Training**	**AWID – Test**
Normal	1,633,190	530,785
Impersonificación	48,484	8097
DoS	65,379	16,682

Cada paquete dentro de este conjunto de datos está conformado por 153 características o campos, entre las cuales incluye la clase.

Selección de Características

Cada paquete contiene diferentes características de datos que corresponden a marcas de tiempo, números en hexadecimales, campos netamente que contienen letras, como por ejemplo las direcciones MAC, y valores numéricos. Para poder entrenar el algoritmo de Machine Learning es necesario realizar una limpieza de los datos como un primer paso. Existen características que tienen el mismo valor para casi todos los paquetes del conjunto, por lo cual estas características fueron removidas (Reyes, Vaca, Aguayo, Niyaz, & Devabhaktuni, 2020). Además, se eliminó las características que tenían valores nulos; el criterio que se eligió fue el de

eliminar aquellas características que contenían más del 50% de valores nulos. De las 154 características que contiene los paquetes en la base de datos, en el artículo titulado: “ROLE OF FEATURE SELECTION IN INTRUSION DETECTION SYSTEMS FOR 802.11 NETWORKS” , presenta las características que se debe utilizar para analizar cada paquete, estas son:

- Tipo de trama: Valor que indica que tipo de trama es: control, gestión o datos.
- Subtipo de trama: Dependiendo del tipo de trama existen subtipos de tramas, por ejemplo dentro de las tramas de gestión tenemos: desautenticación, disociación, autenticación, probe request, probe response, etc.
- ToDs: Este campo indica si la trama es enviada al sistema de distribución.
- FromDs: Indica si la trama proviene del sistema de distribución.
- More Fragment: Valor que indica si la trama es la última o si existe más tramas relacionadas a la misma.
- Retry: Indica si la trama es retransmitida.
- Power Managment: Indica si el dispositivo esta activo o en modo de ahorro de energía.
- Protected: Indica si los datos están encriptados.
- Order: Indica si las tramas deben ser reensambladas estrictamente según el orden de llegada.
- Duration: Tiempo en el cual el medio va a estar ocupado durante una transmisión.
- RA: Dirección MAC de la estación que recibe la trama.

- TA: Dirección MAC de la estación que transite la trama.

Para complementar el estudio previo, se utilizó otra investigación “A MACHINE LEARNING BASED TWO-STAGE WI-FI NETWORK INTRUSION DETECTION SYSTEM”**.** En este artículo se realizó un análisis de las características más importantes para detectar ataques a la red usando la base de datos AWID y el modelo de machine learning, Random Forest. Las características más importantes, según los autores de este artículo que permiten al modelo tener la mayor exactitud son las siguientes:

Tabla 6

Características seleccionadas para el modelo Machine Learning

Número	**Característica**
0	frame.len
1	radiotap.present.flags
2	radiotap.datarate
3	radiotap.channel.type.cck
4	wlan.fc.type
5	wlan.fc.subtype
6	wlan.fc.ds
7	wlan.fc.frag

8	wlan.fc.retry
9	wlan.fc.pwrmgt
10	wlan.fc.moredata
11	wlan.fc.protected
12	wlan.duration
13	wlan.ra
14	wlan.da
15	wlan.frag
16	wlan.seq

Al analizar las dos investigaciones, se observa que las características de los dos son las mismas y aún más completas en el caso del segundo estudio. Es por ello, y dado que en la investigación: "A MACHINE LEARNING BASED TWO-STAGE WI-FI NETWORK INTRUSION DETECTION SYSTEM" , su modelo de machine learning logra una exactitud del 99,85%, se decidió elegir este set de características para entrenar el modelo y de esta manera poder optimizar el modelo el cual deberá clasificar en tiempo real.

Modelo de Machine Learning

El modelo que se implementó para el sistema de detección de intrusos fue el de Random Forest, ya que este modelo es el que mayor precisión tiene para detectar anomalías en el tráfico de una red inalámbrica Wifi (Le, Park, Cho, & Kim, 2018)**.** Además Spark tiene este modelo disponible en su librería de Machine Learning, por lo cual su implementación no

demandara mayor esfuerzo. Random forest es un conjunto de árboles de decisión, los cuales se combinan para tener un mejor desempeño. En Spark, admite la implementación de este modelo para llevar a cabo una clasificación binaria, multiclase o de regresión. Este modelo es capaz de manejar características categóricas, numéricas y no requiere normalización o escalado de las características y puede también inferir no linealidades e interacciones de características (Neelakantan & Nagesh, 2011). La implantación de Random Forest en Spark permite que durante el entrenamiento de los árboles de decisión, estos se entren por separado y con aleatoriedad para que cada árbol de decisión sea diferente entre sí. En el caso de clasificación, la predicción se realiza por voto mayoritario. Cada predicción de los árboles se cuenta como un voto y la predicción final será la clase con mayores votos.

Una de las primeras acciones que se deben realizar sobre los datos que se usarán para entrenar y validar el modelo de Machine Learning es transformarlos todos a valores numéricos, ya que los modelos de Machine Learning en Spark solo admiten datos de este tipo de entrada. Por lo cual, al tener variables categóricas es preciso primero transformarlas a variables numéricas. En la librería de Machine Learning de Apache Spark vienen incluidas herramientas que permiten realizar el procesamiento necesario sobre variables categóricas. Estas herramientas son: StringIndexer y VerctorIndexer. Una de las características no tiene un formato numérico (tampoco booleano) por lo cual es necesario mapear su contenido a un valor numérico. Stringindexer permite convertir variables categóricas a diferentes índices cuando estas no están en un formato numérico (Reyes, Vaca, Aguayo, Niyaz, & Devabhaktuni, 2020). Una vez que ya se tiene todas las características en un formato numérico es necesario agruparlas en un solo vector, pues los modelos de machine learning solo admiten como entrada un vector, con valores netamente del tipo continuo, el cual contiene todas las características.

Tabla 7

Características con valores categóricos

Característica
radiotap.present.flags
radiotap.channel.type.cck
wlan.fc.type
wlan.fc.ds
wlan.fc.frag
wlan.fc.retry
wlan.fc.pwrmgt
wlan.fc.moredata
wlan.fc.protected
wlan.ra
wlan.da

Es necesario distinguir que características posean una cualidad categórica, a pesar de tener valores numéricos. Para lo cual, VectorIndexer es una herramienta útil. Esta herramienta permite encontrar variables categóricas en el vector de la característica basada en el número de

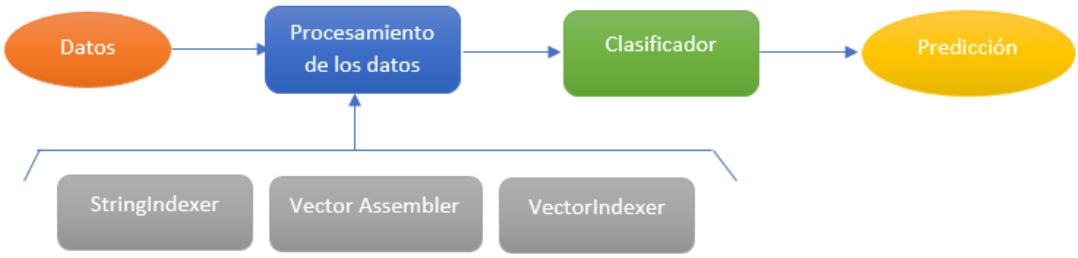
distintos valores y calcula índices según el esquema base cero. Después de haber realizado este proceso, se puede ingresar los datos al algoritmo para entrenar el modelo.

Tubería de Datos

Una vez definido todo el procesamiento que se debe aplicar a los datos, Spark permite crear una tubería de datos que optimiza el procesamiento de los datos. Mediante una tubería de datos (Pipeline por sus siglas en inglés), los datos provenientes del streaming se adecuaran al formato necesario para poder ingresar al modelo de machine learning que a su vez realizara la clasificación y por último se obtendrá el resultado.

Figura 12

Tubería de datos aplicado a los datos del streaming



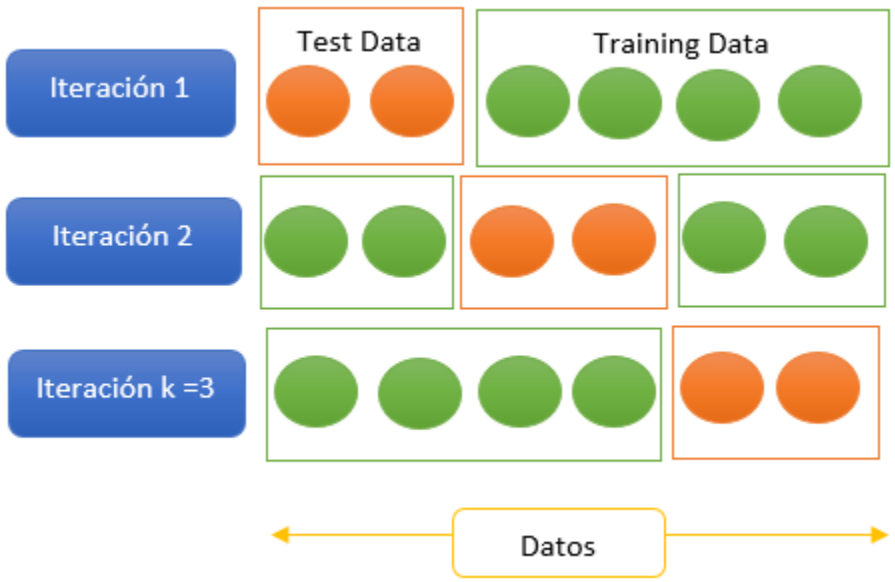
Entrenamiento del Modelo

Para la selección de los parámetros del modelo, la librería de machine learning incluye una herramienta que facilita la selección de los parámetros. Cross-Validation permite dividir el conjunto de datos en k subconjuntos, los cuales son usados para el entrenamiento y la validación del modelo. En este caso se utilizó una división de k = 3, con lo cual se generarán 3

subconjuntos de datos entrenamiento y de validación, los cuales son diferentes entre sí. Para conformar el subconjunto de entrenamiento se utiliza 2/3 del conjunto de datos total y el restante 1/3 es usado para conformar el conjunto de validación.

Figura 13

Descripción del método Cross-Validation



Cross-Validation realiza una iteración sobre cada subconjunto evaluando el desempeño de cada modelo que se generó usando los diferentes subconjuntos y después de evaluar cada modelo regresa el modelo que mejor desempeño tuvo. Para lo cual es posible decidir qué parámetros Cross-Validation debe cambiar en los modelos y la métrica que se usa para evaluarlos es la de exactitud. La profundidad de los árboles y el número de estos son los dos parámetros que Cross-Validation valido en cada iteración y los que cambio para verificar con que valores el modelo tiene un mejor desempeño. Cabe recalcar que realizar esta operación demanda un gran costo computacional, por lo cual solo se eligió estos dos parámetros para que

Cross-Validation utilice en su proceso. Además estos dos parámetros son los que más influencia tienen sobre el desempeño del modelo Random Forest (Chio & Freeman, 2018).

Evaluación del Modelo

La librería de Machine Learning de Apache Spark no solo tiene incluida modelos de machine learning para clasificación y regresión y herramientas que facilitan la selección de los parámetros de cada modelo, sino que además cuenta con herramientas para evaluar el modelo. En este caso MulticlassClassificationEvaluator, permite obtener métricas del desempeño del modelo. Entre las métricas que calcula esta herramienta están: F1-score, weightedPrecision, weightedRecall y accuracy (exactitud, por sus siglas en inglés). Después de obtener el modelo que Cross-Validation generó, se pone a prueba ese modelo con el conjunto de validación, conjunto de datos con el cual el modelo no ha tenido contacto alguno y que es parte del conjunto original AWID. A continuación se reflejan las métricas de este modelo.

Tabla 8

Métricas de evaluación del modelo Random Forest

Métrica	**Valor**
F1	0.9929501428502706
Accuracy	0.9934534
weightedPrecision	0.9934070932251021
weightedRecall	0.9934533735760149

Las métricas son bastantes similares a los obtenidos en : “A MACHINE LEARNING BASED TWO-STAGE WI-FI NETWORK INTRUSION DETECTION SYSTEM”.

Arquitectura del sistema

Como primer paso, se debe realizar la recolección del tráfico de la red inalámbrica, para lo cual se utilizó una herramienta de software libre; en este caso se usó Tshark. Esta herramienta permite analizar el tráfico de red, capturando los paquetes de la red y desplegándolos en el formato que el usuario deseé. Tshark permite no solo realizar una captura activa, sino que además cuenta con opciones que permiten modificar la captura, de tal manera que el usuario puede seleccionar solo los datos que desea observar, lo cual lo puede realizar mediante el uso de filtros. De esta manera, la captura del tráfico de la red se realiza fácilmente, pues mediante el uso de filtros se seleccionó solo aquellos campos de interés para el modelo

como se especificó anteriormente. El formato de salida que se elegio es el JSON. Sin embargo, Tshark solo permite capturar el tráfico de la red, lo cual desplegaría en la consola de comandos. Pero, y como se verá más adelante, lo que realmente se necesita para el diseño del sistema de detección de intrusos, es que todos los datos capturados sean enviados al bróker de Kafka. Para este último propósito, se utilizó Kafkacat, la cual es una herramienta de línea de comandos que permite producir o consumir mensajes de un tópico. El funcionamiento de estas dos herramientas en conjunto permite capturar el tráfico y enviarlo al bróker, en el cual se podrán consumir todos los paquetes capturados para posteriormente procesarlos y analizarlos con el modelo de machine learning. Es evidente, pero a la vez necesario mencionar, que para la captura del tráfico es necesario utilizar una tarjeta USB Wifi, la cual se le configura en modo monitor para obtener todos los paquetes de la red. En este caso se utilizó la tarjeta: Metageek AirPcap Nx: USB 802.11a/b/g/n Adapter.

Figura 14

Captura de tráfico del sistema de detección

Una vez capturado el tráfico de la red, este debe ser enviado al bróker de Kafka donde se almacenarán todos los paquetes recolectados en espera para que sean consumidos por Apache Spark, herramienta que es en donde se realizará todo el procesamiento de los datos recolectados y en donde se analizaran estos datos procesados juntamente con el modelo de Machine Learning. Dicho de otra forma esta última parte, el productor de mensajes seria en este caso el dispositivo USB Wifi, el cual recolecta el tráfico, mediante Tshark y envía los paquetes usando Kafkacat, la cual está configurada en modo productor. Todos estos datos son enviados al bróker de Kafka, el cual está configurado previamente con un solo tópico y una sola partición. Cabe mencionar que el bróker de Kafka se ejecutó en una computadora con las siguientes características: Intel Core i5-2450M GPU 2,50 GHz y con una memoria RAM de 8 GB en un sistema operativo Linux mediante la distribución de Ubuntu 18.04. El consumidor es una computadora, la cual es encarga de realizar el procesamiento y análisis de los datos usando Apache Spark.

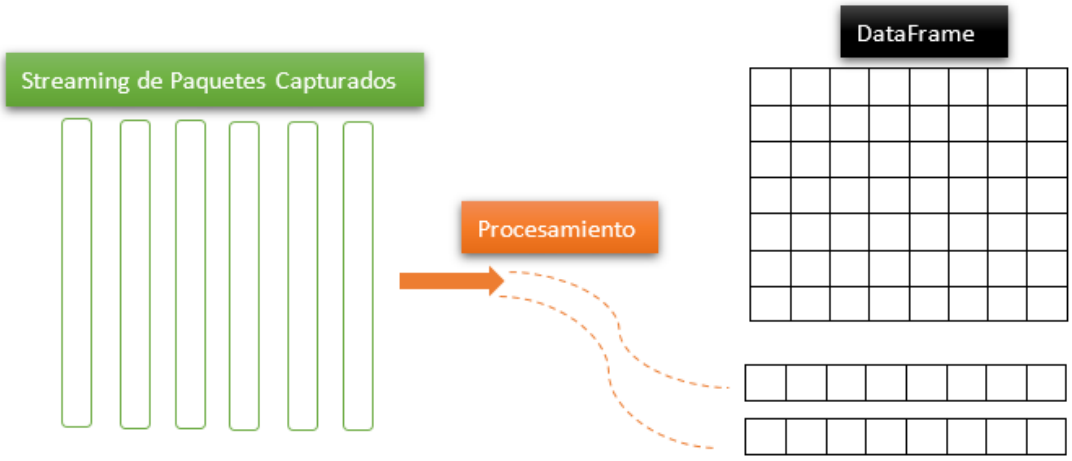
Figura 15

Descripción grafica de la captura del tráfico

Apache Spark se ejecutó de modo local, lo que quiere decir que tanto el nodo maestro como los nodos que realizan las tareas se ejecutaron en la misma computadora, la cual tiene las siguientes características: Intel Core i7-8750H de 6 núcleos, una memoria RAM de 16 GB y disco duro de 512 SSD. La aplicación de Spark se desarrolló en el lenguaje de programación Scala y las principales librerías que se utilizaron fueron: Machine Learning y Structured Streaming. Esta última librería permite constantemente leer todos los paquetes almacenados en el bróker de Kafka. Los paquetes que se leen directamente del tópico de Kafka están en formato JSON, por lo cual lo primero que se realiza en Spark es transformar todos los paquetes de ese formato y colocarlos en un formato adecuado para el procesamiento de datos. Structured Streaming soporta el manejo de DataFrame que es un conjunto de datos organizados en columnas, que conceptualmente son similares a una tabla, las cuales se usan en bases de datos relacionales.

Figura 16

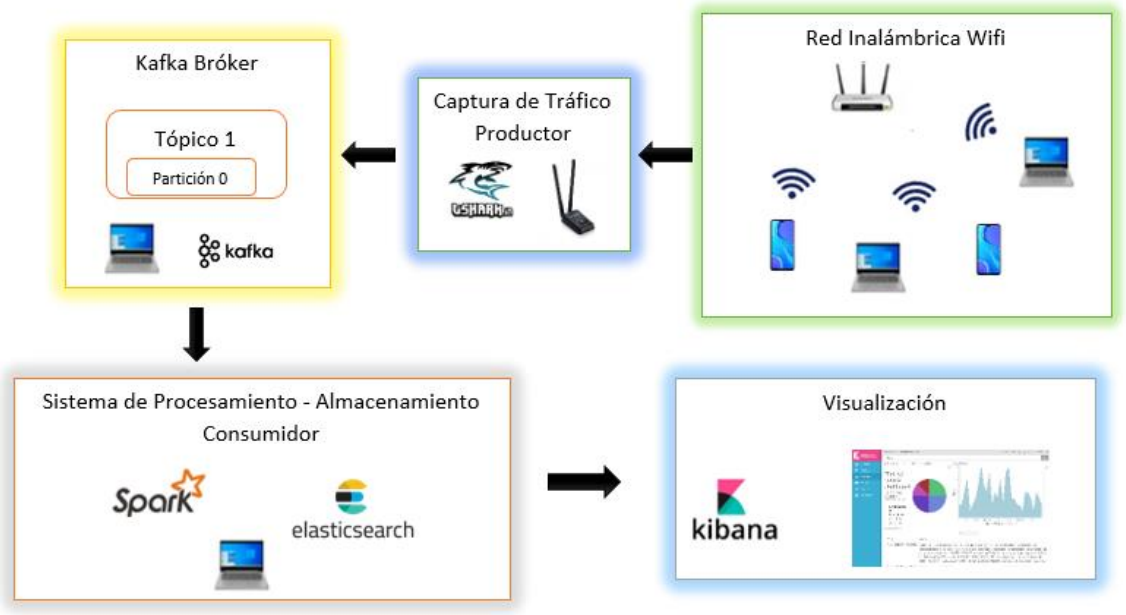
Descripción del procesamiento en tiempo real



Después de realizar todo el proceso y análisis de los paquetes con el modelo, los resultados deben ser enviados a Elasticsearch, en donde se almacenarán y además permitirá visualizar los resultados. Elasticsearch fue utilizado de igual forma de modo local, lo que significa que solo se utilizó un solo nodo (comúnmente conocido como Standalone clúster). Lo ideal sería configurar Elasticsearch, que incluye Kibana por defecto, en otra computadora ya que dependiendo de la aplicación puede consumir grandes recursos, pero debido a que se trata del diseño de un sistema y el objetivo primordial es solo probar de manera general el sistema, se ejecutó Elasticsearch en la misma computadora que Spark. De esta forma el sistema de detección de intrusos tiene la siguiente topología final:

Figura 17

Arquitectura final del sistema de detección de intrusiones



Capítulo IV

Análisis del sistema de detección de intrusiones

Para probar el sistema de detección de intrusiones se ha creado seis diferentes escenarios que son los más comunes que se producen en la vida real. En cada escenario se prueba diferentes ataques, ya sea atentando contra el cliente, el AP o ambos. En cada escenario se usa una herramienta de penetración diferente para poder comprobar si detecta o no el sistema dicho ataque. En cada ataque siempre en los primeros minutos se genera tráfico normal para poder comprobar posteriormente que el sistema puede diferenciar entre tráfico normal o anómalo.

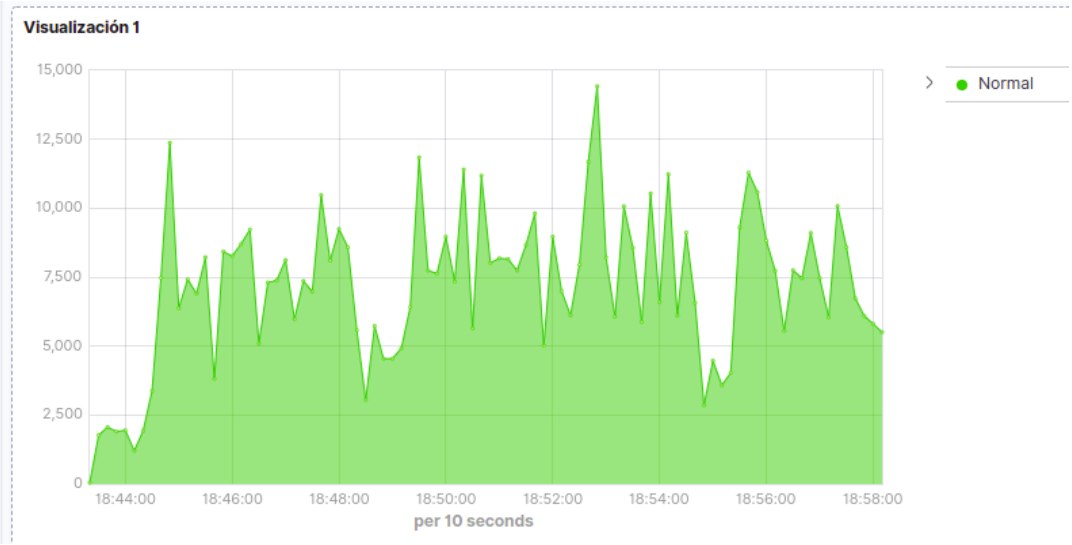
Características del Sistema de Detección de Intrusiones

El sistema de detección de intrusiones captura el tráfico y lo presenta en el Dashboard en Kibana de tal manera que permita a un administrador poder analizar el canal o la red que se desea observar. Para esto, se ha implementado en el Dashboard diferentes herramientas que, según el caso o la prueba de vulnerabilidad, permiten observar cómo se comporta el tráfico de la red, sino que además permite fácilmente detectar tramas maliciosas generadas por un atacante. A continuación se describen todas las herramientas implementadas en el sistema de detección de intrusiones:

- En la visualización 1 se puede observar la cantidad de paquetes que el sistema detecta. Estos además se agrupan según su tipo: normal, denegación de servicio o Impersonificación. Esta visualización se actualiza cada segundo y permite filtrar el resultado para mostrar solo las tramas de interés, ya sea normal o cualquier otra.

Figura 18

Tramas clasificadas por cantidad y tiempo de llegada



- En la visualización 2 es muy similar a la visualización 1. La diferencia radica en que esta permite observar la llegada de paquetes según el tipo de estos: gestión, datos o control. De igual forma se actualiza cada segundo y se puede seleccionar que tipos mostrar.

Figura 19

Tipos de tramas graficadas según la cantidad y tiempo de llegada

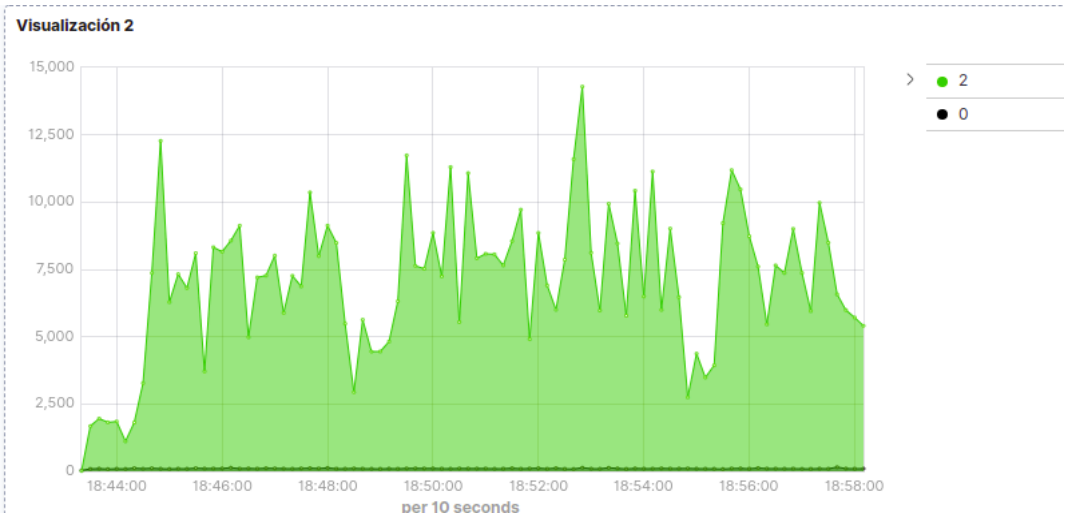
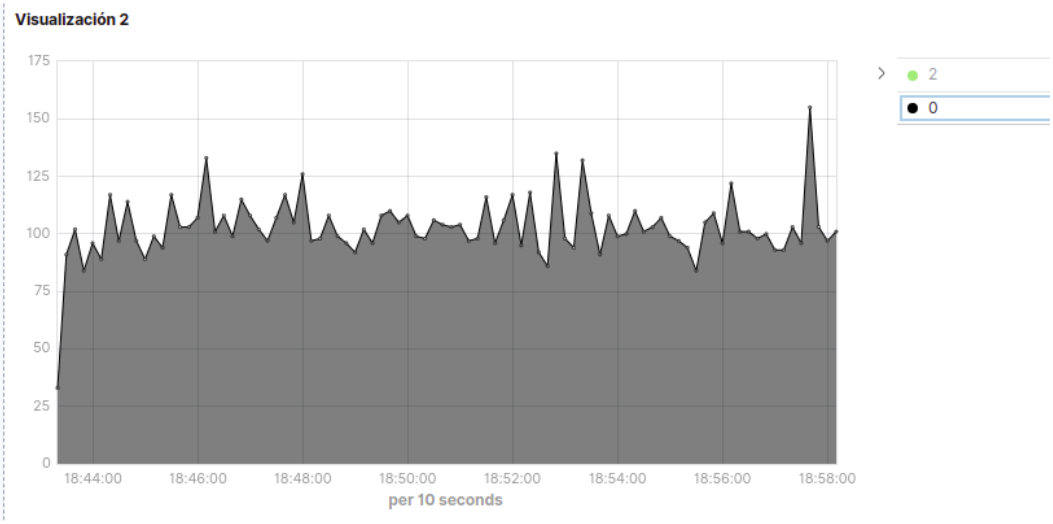


Figura 20

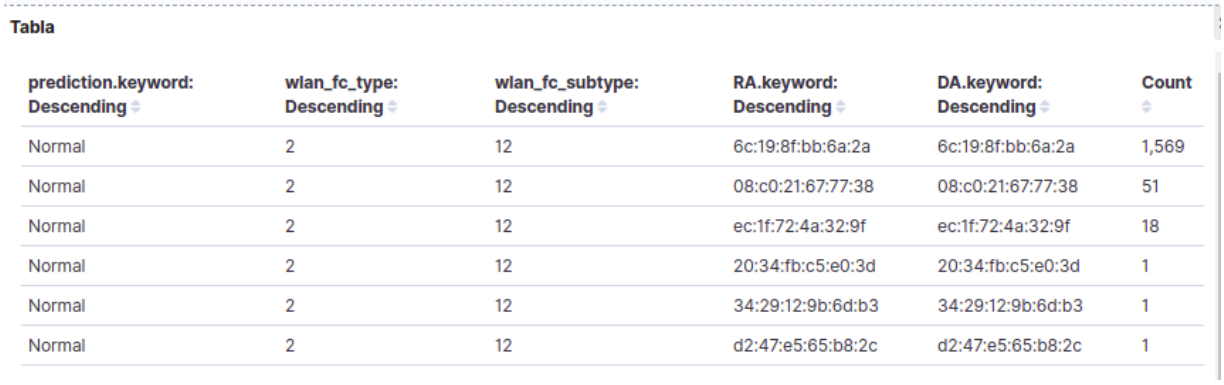
Visualización 2 con un filtro para mostrar solo tramas de gestión



- En la siguiente tabla se agrupan todas las tramas recibidas según el tipo y el subtipo. Además de que se muestra la dirección de destino y del receptor. Esta tabla es útil para detectar a que cliente se ataca o para visualizar si se trata de un ataque en Broadcast. Se actualiza cada segundo y se puede filtrar los resultados, ya sea por el tipo, subtipo, clase o dirección.

Figura 21

Tabla con datos de las tramas recibidas en tiempo real



Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
Normal	2	12	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	1,569
Normal	2	12	08:c0:21:67:77:38	08:c0:21:67:77:38	51
Normal	2	12	ec:1f:72:4a:32:9f	ec:1f:72:4a:32:9f	18
Normal	2	12	20:34:fb:c5:e0:3d	20:34:fb:c5:e0:3d	1
Normal	2	12	34:29:12:9b:6d:b3	34:29:12:9b:6d:b3	1
Normal	2	12	d2:47:e5:65:b8:2c	d2:47:e5:65:b8:2c	1

- La tabla 1 es muy similar a la tabla anterior, con la diferencia que esta agrupa las tramas según el tipo y subtipo y muestra la duración de esta. De igual forma se puede personalizar la visualización de los resultados y se actualiza en tiempo real.

Figura 22

Tabla según la duración de cada trama según el tipo y subtipo

Tabla 1

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	wlan_duration: Descending	Count
Normal	2	8	36	505,822
Normal	2	8	48	55,737
Normal	2	8	44	33,131
Normal	2	8	84	4,194
Normal	2	8	202	69
Normal	2	8	49	46
Normal	2	8	60	40
Normal	2	8	314	27
Normal	2	8	223	3
Normal	2	8	213	1

- La tabla 2, por otro lado, al igual que las tablas inferiores permite visualizar cada trama según la secuencia de cada trama.

Figura 23

Tabla según la secuencia de cada trama según el tipo y subtipo

Tabla 2

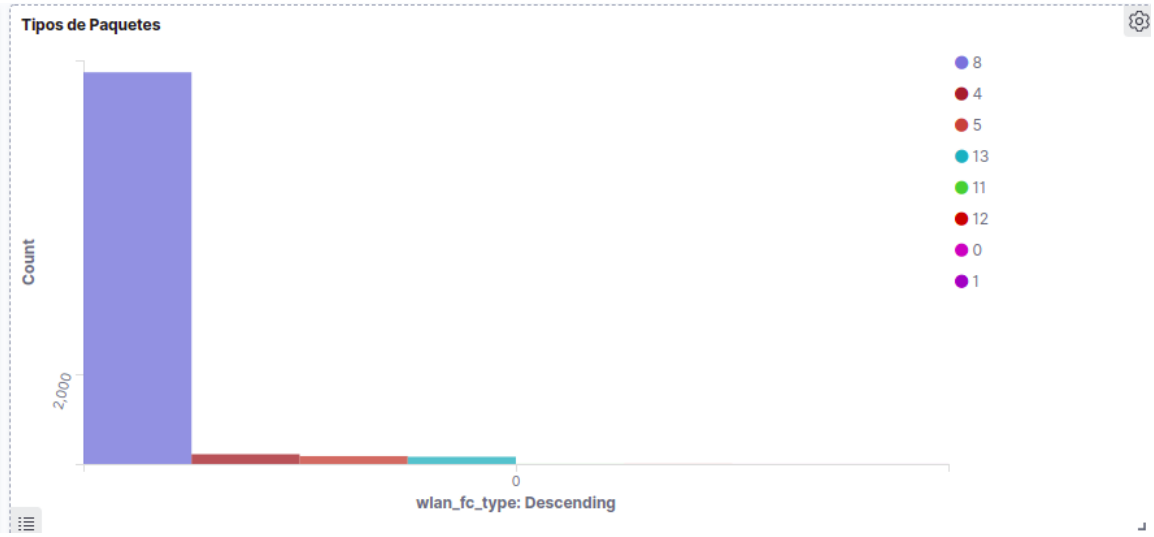
prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	wlan_seq: Descending	Count
Normal	0	8	868	4
Normal	0	8	2,075	4
Normal	0	8	31	3
Normal	0	8	105	3
Normal	0	8	208	3
Normal	0	8	421	3
Normal	0	8	422	3
Normal	0	8	423	3
Normal	0	8	424	3
Normal	0	8	425	3

Export: Raw Formatted

- El siguiente diagrama de barras permite visualizar la cantidad de tramas según el tipo y subtipo de paquetes que detecta el sistema. Este diagrama permite obtener una vista rápida para detectar inundación de tramas que pueden significar en un ataque de denegación de servicio. Se puede filtrar el tipo de tramas que se desea observar y se actualiza cada segundo.

Figura 24

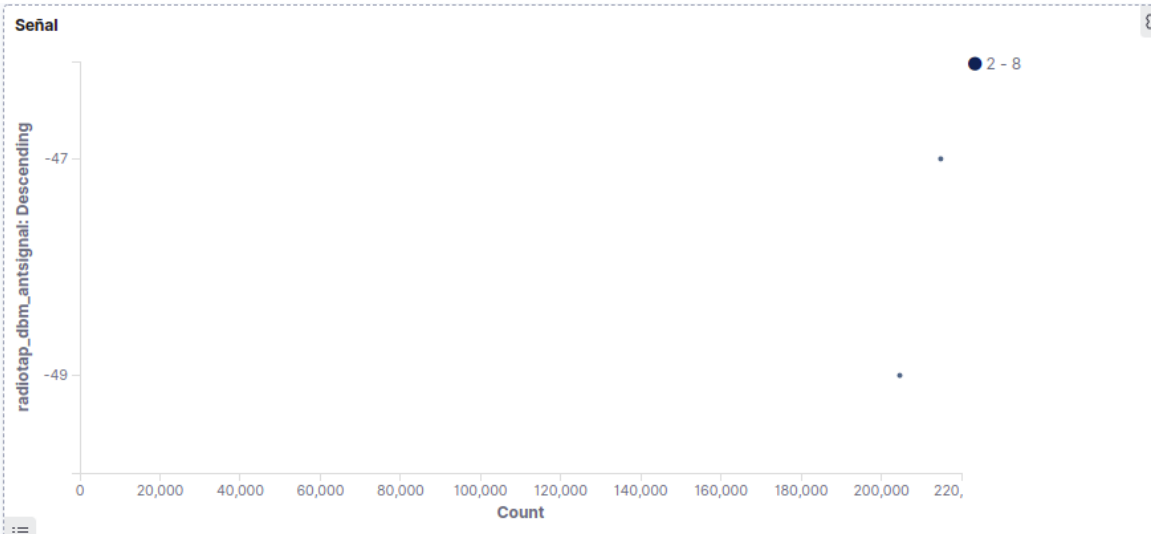
cantidad de paquetes recibidos según el tipo y subtipo



- La siguiente grafica utiliza como parámetro la potencia de la señal recibida de cada trama y la gráfica en función de la cantidad de tramas. Esta grafica juega un papel fundamental en ataques de Impersonificación ya que, filtrando solamente tramas Beacon se puede detectar la cercanía del atacante en función de la potencia de la señal. Se actualiza cada segundo.

Figura 25

Grafica según la potencia de señal recibida de cada trama



- Esta visualización permite tener un conteo de todas las tramas clasificadas que el sistema detecta. Esta se actualiza cada segunda y muestra los tres tipos de tramas que el sistema puede detectar: normal, DoS (denegación de servicio) e Impersonificación. Además de que muestra también un diagrama de pastel en el que muestra el porcentaje del total de las tramas recibidas. Se actualiza cada segundo.

Figura 26

Cantidad de paquetes clasificados por el sistema

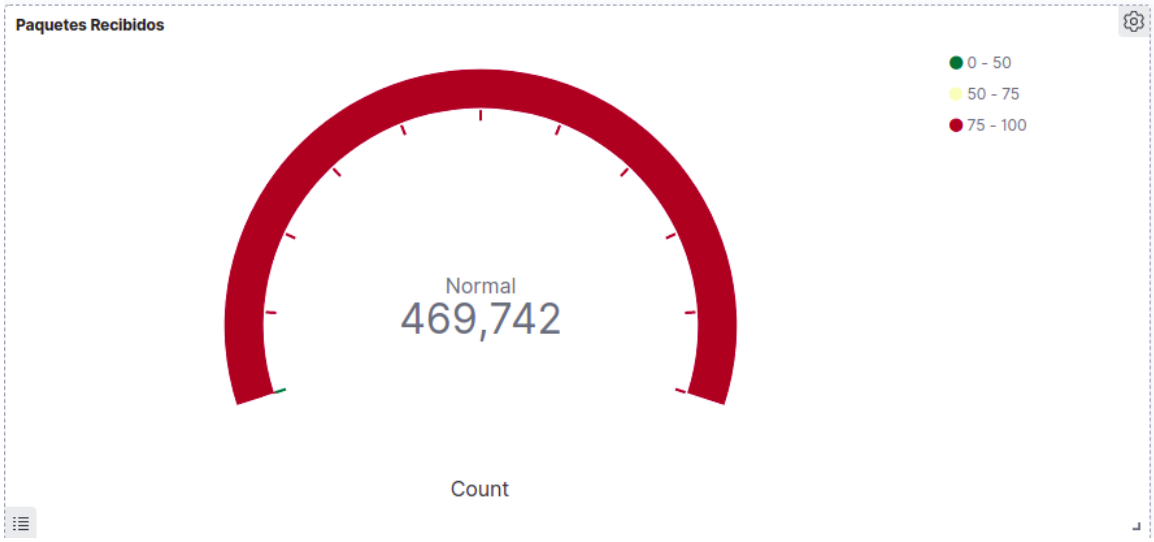
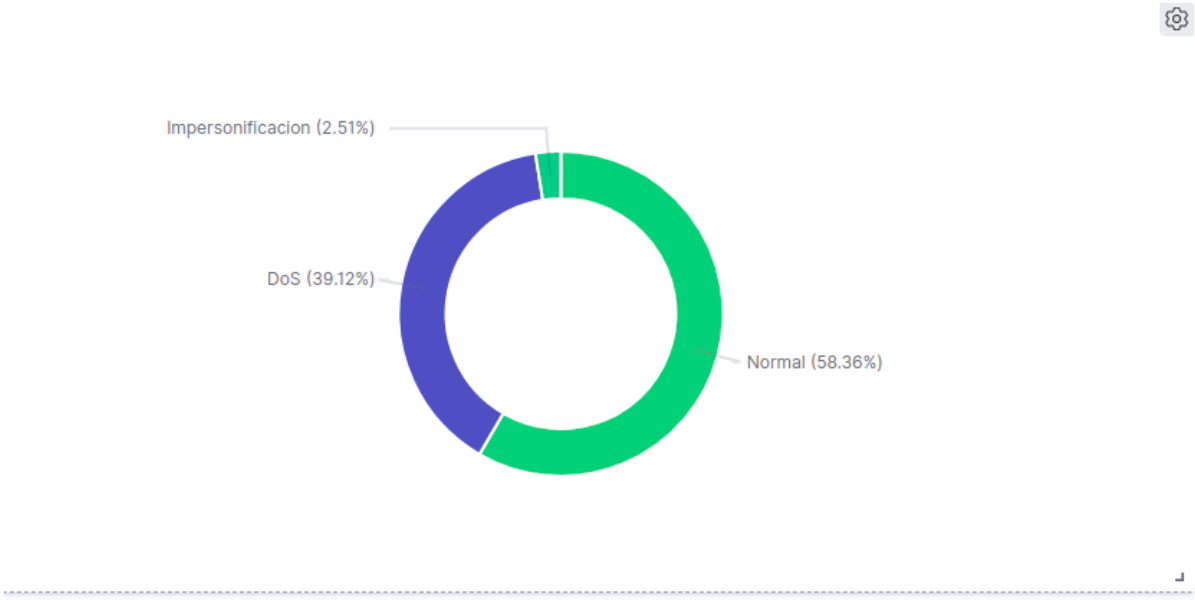


Figura 27

Cantidad recibida de paquetes clasificados y en porcentaje



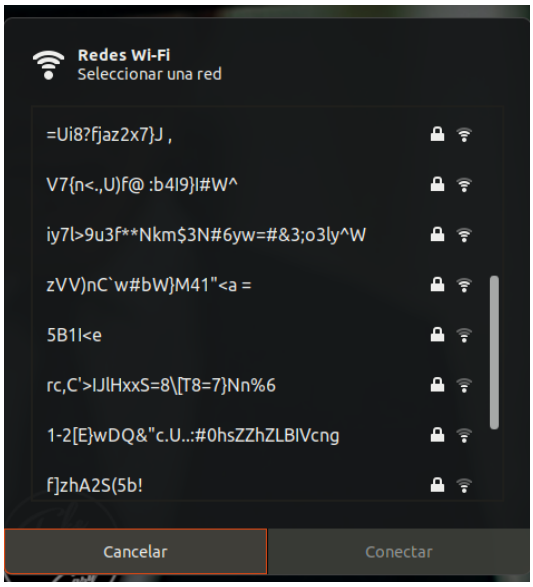
Pruebas de Penetración

Inundación de Tramas Beacon

En este tipo de denegación de servicio, el atacante emitirá tramas Beacon falsas de puntos de acceso que no existen. El objetivo de enviar una gran cantidad de este tipo de tramas Beacon, las cuales representan redes inalámbricas que no existen, es saturar los dispositivos de los clientes, ya que estos dispositivos recibirán tantas tramas que imposibilitara al cliente encontrar la red inalámbrica a la cual desea conectarse. Es común, que los dispositivos, tanto celulares como computadoras, muestren una limitada cantidad de redes disponibles a la cual conectarse, dicho de otra forma, el nombre de la red a la que el usuario quiere conectarse podría no aparecer en esta lista. Por ende, este ataque puede impedir la conexión a la red.

Figura 28

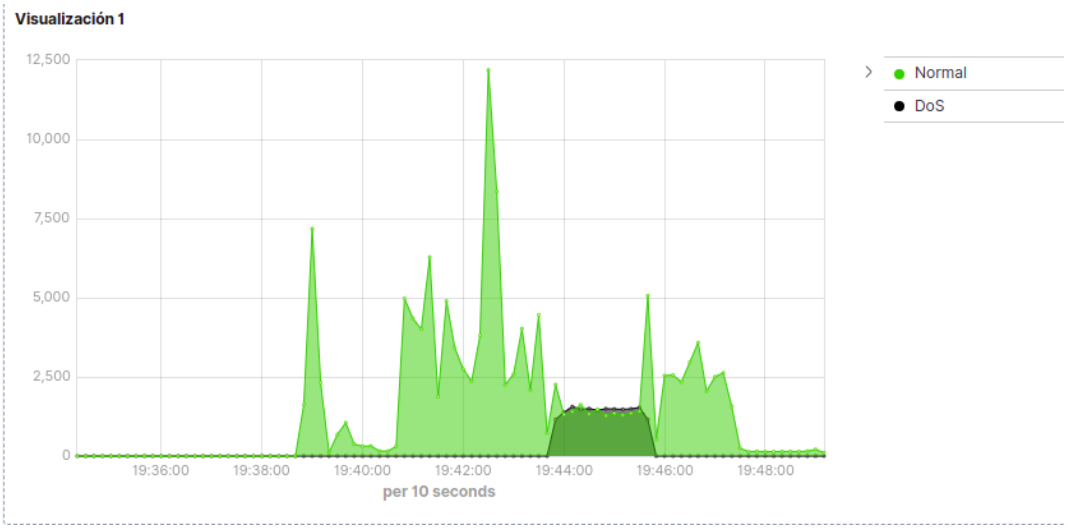
Redes falsas creadas detectadas en el dispositivo del usuario



Durante los primeros cinco minutos de la prueba, el sistema de detección de intrusos monitorea la red, la misma que no sufre ningún ataque y se encuentra bajo condiciones normales. Durante este periodo de cinco minutos, dos usuarios conectados a la red generan tráfico normal. Posterior a los cinco minutos, se utiliza la herramienta mdk3, la cual sirve para realizar pruebas de penetración en redes inalámbricas, para generar tramas Beacon falsas de redes inexistentes las cuales inundarán el tráfico e imposibilitarán a un tercer usuario conectarse a la red.

Figura 29

Tramas recibidas y clasificadas en tiempo real durante el ataque

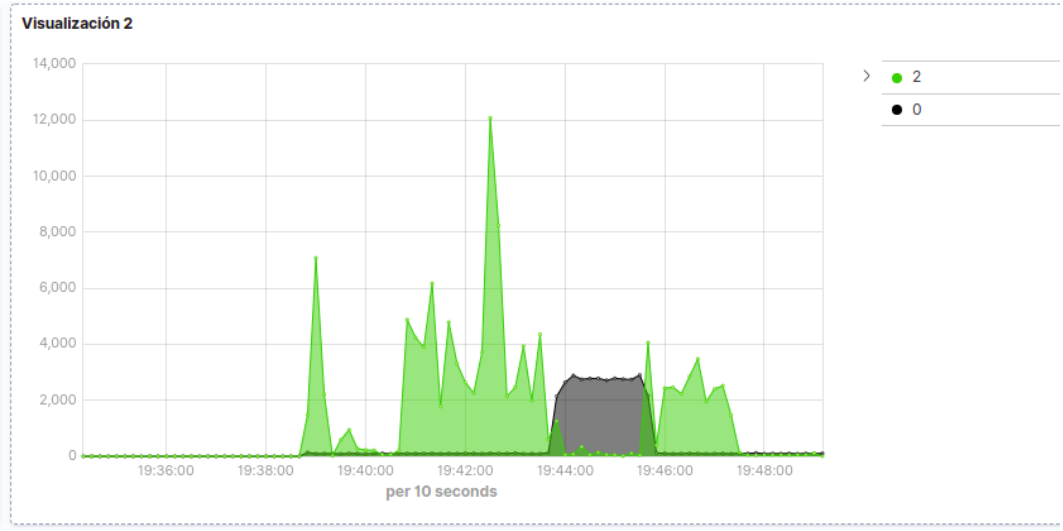


De la figura 29**,** se puede observar el monitoreo completo durante la prueba. La figura muestra claramente en el cual la red comienza a sufrir un ataque de denegación de servicio. Durante los primeros cinco minutos, el sistema detecta un comportamiento normal como se esperaba, por lo cual el área sombreada es de color verde. Posterior a eso, el sistema empieza a recibir tramas Beacon con la intención de comprometer la disponibilidad de la red. El sistema

detecta este ataque (área color negro) que dura aproximadamente dos minutos. Un administrador de red puede verificar este comportamiento anómalo y con la ayuda del sistema de detección de intrusiones, realizar un análisis de todas las tramas que recibió; todo esto en tiempo real.

Figura 30

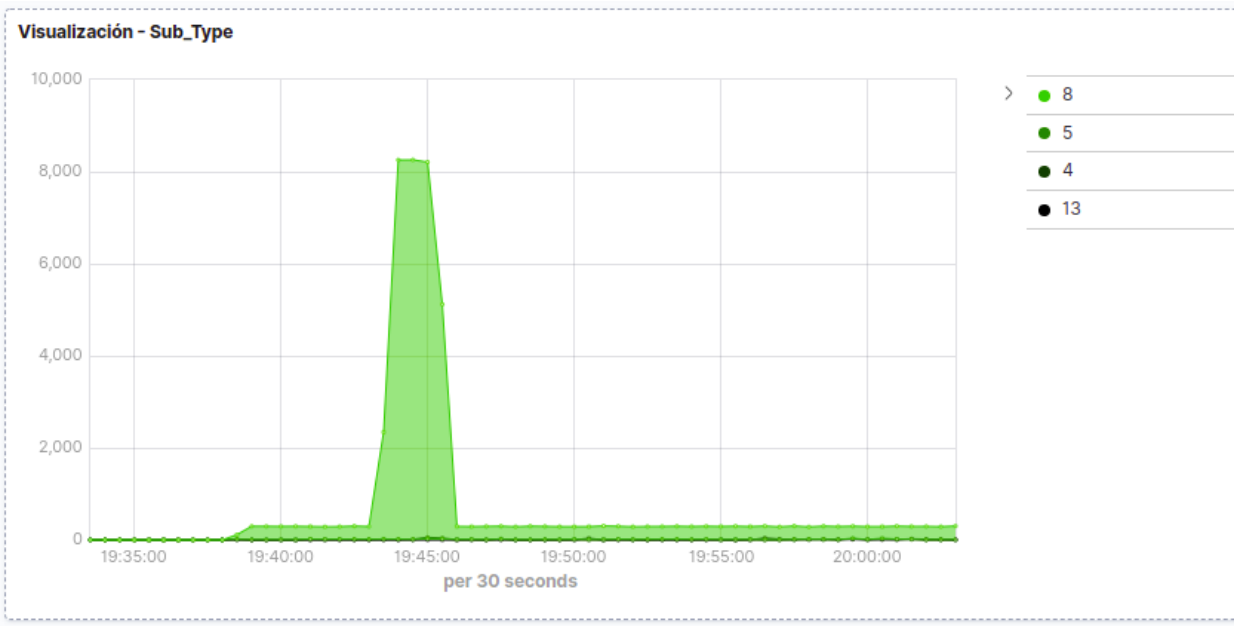
Tramas recibidas según el tipo y subtipo durante el ataque



La figura 30**,** muestra la visualización según el tipo de tramas. En este caso muestra dos tipos de tramas: datos (tipo 2) y de gestión (0). Es evidente que en un lapso, las tramas de gestión aumentan abruptamente. Es una clara evidencia de que la red, durante este tiempo, tiene un comportamiento anómalo. El sistema de detección de intrusiones permite indagar aún más, pues es necesario verificar que tipos de tramas de gestión aumentaron considerablemente para hallar el problema.

Figura 31

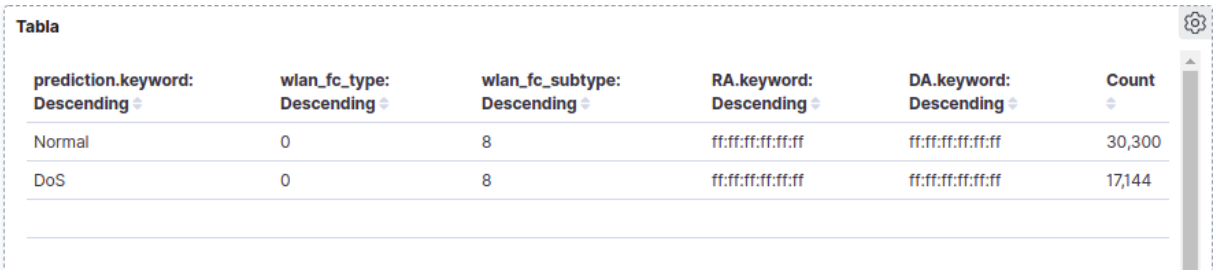
Tipos de tramas de Gestión recibidas durante el ataque



La gráfica 31**,** es una visualización en tiempo real del tipo de tramas de gestión que recibió el sistema de detección de intrusiones. Durante este tiempo el sistema recibió las siguientes tramas de gestión: Beacon, Probe Respone, Probe Request y Action. Tomando en cuenta el tiempo en el cual se detectó el inicio del ataque, en ese momento de igual forma se registra un aumento considerablemente alto de tramas Beacon, que dura durante todo el tiempo del ataque. Por lo cual, se puede concluir fácilmente que la red está sufriendo un ataque de inundación de tramas Beacon con la intención de denegar la disponibilidad de la red para cualquier usuario que desee conectarse a la red. El sistema de detección de intrusos permite filtrar el tipo y subtipo de tramas que se visualizaran en el dashboard. En este caso, se filtrará y buscará el tipo de tramas de gestión del subtipo Beacon para observar sus características relevantes.

Figura 32

Tramas clasificadas por el sistema durante el ataque



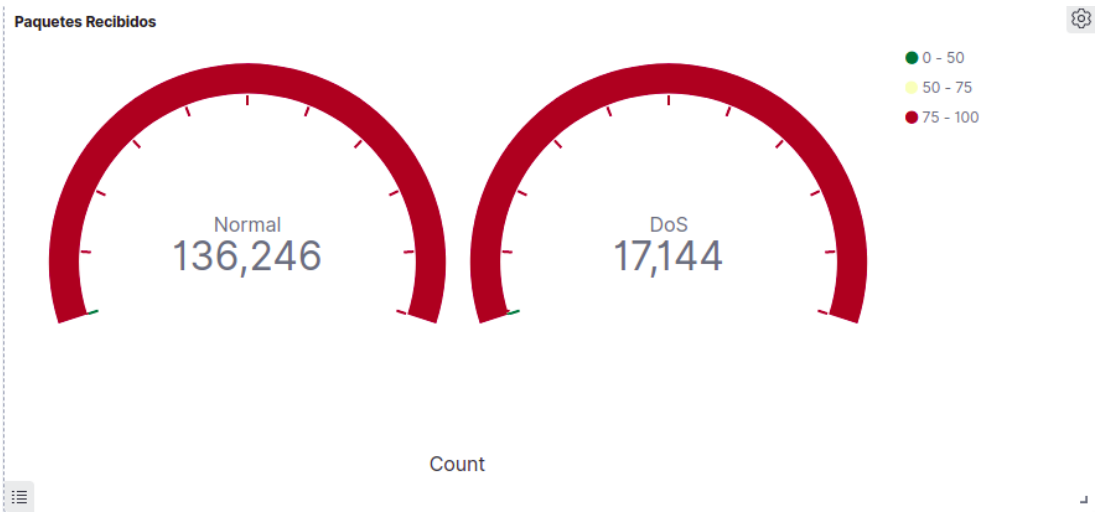
Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
Normal	0	8	ff:ff:ff:ff:ff:ff	ff:ff:ff:ff:ff:ff	30,300
DoS	0	8	ff:ff:ff:ff:ff:ff	ff:ff:ff:ff:ff:ff	17,144

De la gráfica anterior se puede observar la cantidad total de tramas de gestión del tipo Beacon que recibió durante los primeros cinco minutos de actividad normal, sumando 4 minutos más, dentro de los que se generó el ataque. Esto permite tener un balance de todas las tramas recibidas.

Figura 33

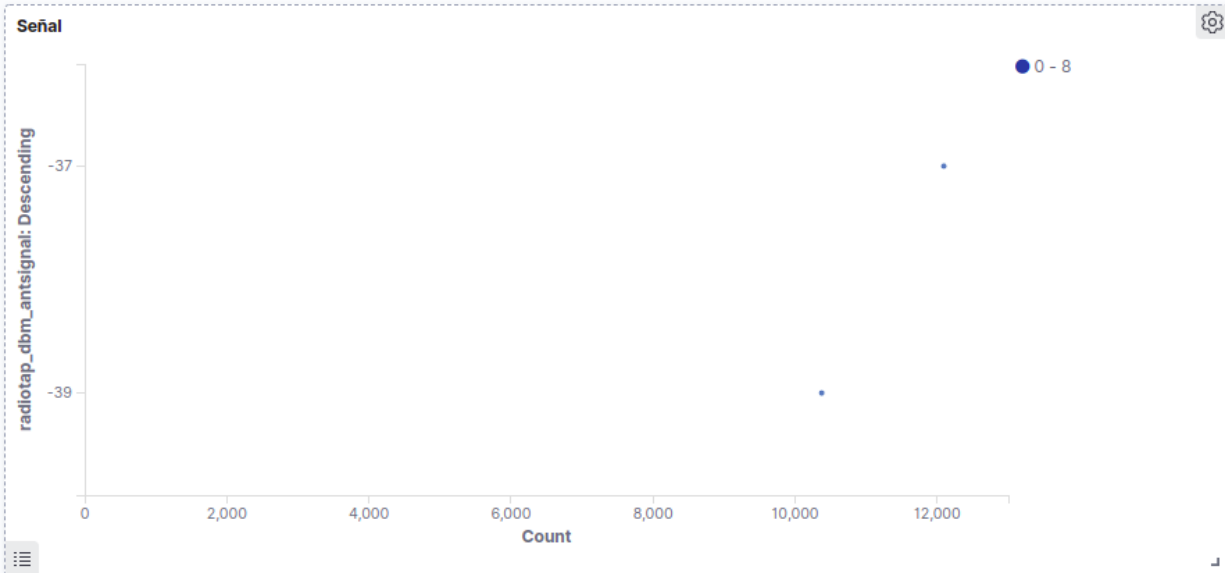
Total de tramas recibidas y clasificadas



Una herramienta útil del Dashboard, es que muestra una gráfica en función de la señal recibida de cada trama. Una opción bastante útil que permite determinar la cercanía del atacante. Como se observa en la siguiente figura 34**,** existen dos dispositivos que envían tramas Beacon constantemente: una pertenece al AP de la red autentica y el otro dispositivo corresponde al atacante enviando tramas Beacon de redes inexistentes.

Figura 34

Potencia recibida de tramas generadas por el AP y el AP falso



Inundación de tramas de Autenticación

Cualquier dispositivo que desea pertenecer a una red inalámbrica debe primero autenticarse con el AP antes de poder establecer una comunicación con este y ser parte de la red. Dado que el AP puede dar servicio a un número limitado de usuarios según los recursos de memoria que posea. Cuando el atacante envía gran cantidad de tramas de autenticación, el AP no puede procesar dichas tramas y por ende entorpece el servicio que brinda a los clientes

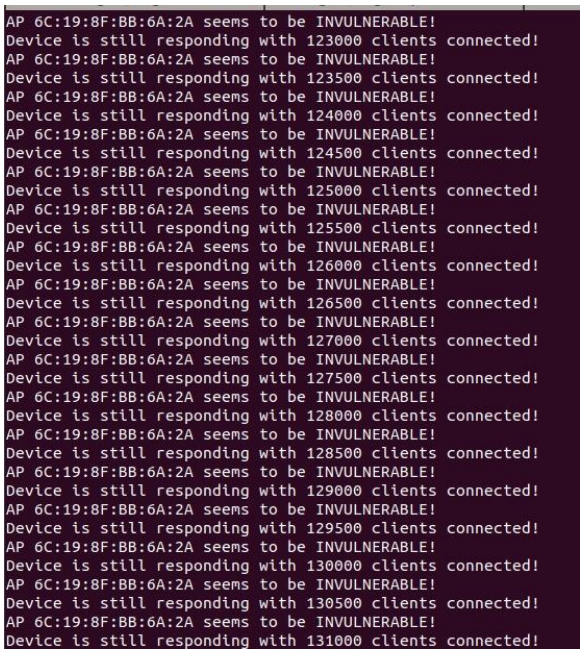
miembros de la red. Además de que durante el ataque, es muy poco probable que cualquier usuario legitimo pueda acceder al servicio.

El procedimiento para evaluar al sistema de detección de intrusiones durante este ataque es similar al anterior: durante los primeros cinco minutos el sistema solo se lo pone a prueba de tráfico normal y posterior a ese tiempo se realiza el ataque a la red. El ataque se lo realiza usando la herramienta mdk3.

En este caso, durante el ataque se intentó autenticar falsamente a 131000 clientes para interrumpir el servicio que brinda el AP. A pesar de que el AP podía seguir brindando servicio a los usuarios que previamente estaban conectados; durante el ataque ningún usuario podía acceder a la red.

Figura 35

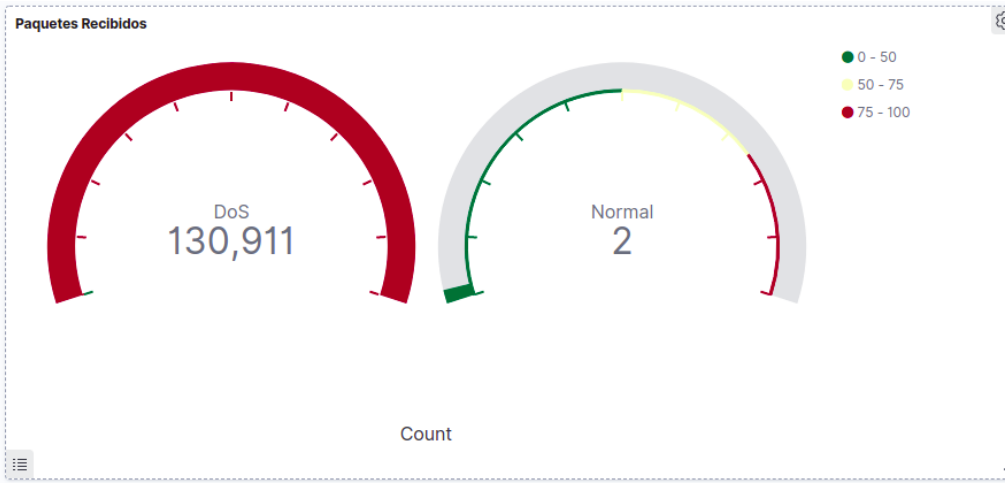
Tramas falsas de autenticación enviadas por el atacante



El sistema de detección de intrusiones reconoció el ataque y todas las tramas con las que el atacante inundó el canal en el que estaba trabajando del AP.

Figura 36

Tramas de Autenticación clasificadas por el sistema



Además de todas las tramas del atacante, el sistema reconoció dos tramas de autenticación que genero un usuario, el cual buscaba poder acceder a la red.

Figura 37

Tramas de autenticación recibidas y clasificadas durante el ataque

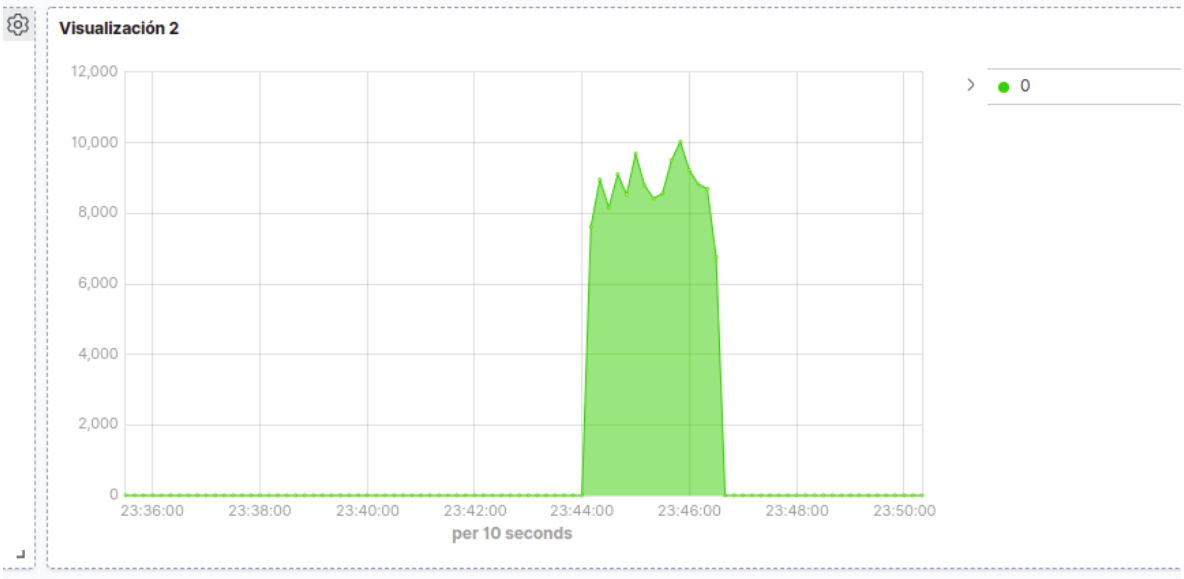
Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
DoS	0	11	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	130,911
Normal	0	11	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	2

Similar al anterior ataque, existe una clara inundación del canal con tramas de gestión, las cuales se puede fácilmente filtrar y visualizar en el Dashboard. Esto permite poder identificar la hora exacta en la cual se produce esta anomalía en el tráfico del canal, producto del ataque.

Figura 38

Inundación del canal con tramas Beacon



En la siguiente figura 39 se puede observar claramente las tramas de tipo gestión (type 0) que pertenece al subtipo de autenticación (subtype 11). Como se puede observar el sistema es capaz de diferenciar las tramas originales que genera un usuario y las tramas falsas de un atacante.

Figura 39

Tramas de autenticación detectadas por el sistema

Tabla 3

prediction.keyword: Descending	frame_len: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	Count
DoS	48	0	11	130,911
Normal	48	0	11	2

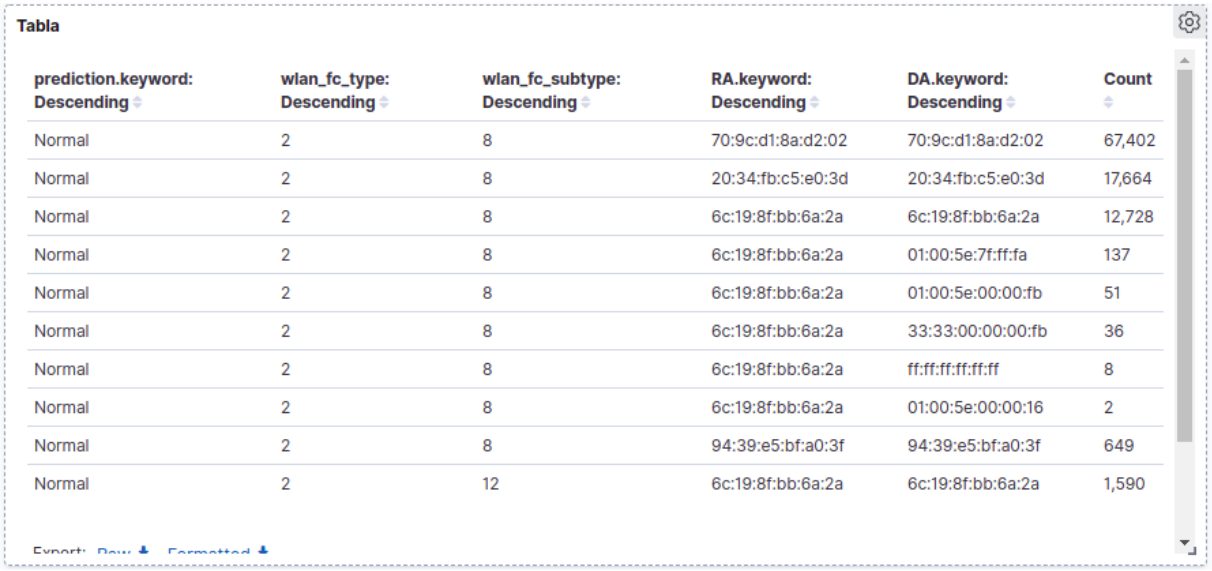
Ataque de desautenticación a un cliente

El atacante envía tramas de desautenticación falsas para cortar la comunicación entre el AP y el cliente. El atacante finge ser el cliente y envía tramas hacia el para desautenticarse de la red, y de igual forma finge ser el AP y envía tramas de desautenticación hacia el cliente para avisarle que se va a desconectar de la red. Durante el ataque, el cliente no puede acceder a la red mientras el atacante envié este tipo de tramas. Este tipo de ataque se lo llevo a cabo con una herramienta de penetración llamada aireplay que pertenece a la suite de software de seguridad inalámbrica.

El escenario de este ataque simula ser un caso práctico. En primera instancia se tiene a tres usuarios conectados a la red, los cuales generan tráfico normal. El atacante espía la comunicación entre los usuarios y el AP, y mediante herramientas como airodump-ng puede conocer las direcciones MAC de cada dispositivo, tanto del AP como de los usuarios y el canal en el cual se están comunicando. Con esta información, el atacante tiene todo lo necesario para llevar a cabo el ataque.

Figura 40

Tramas recibidas por el sistema y clasificadas como normales



Tabla

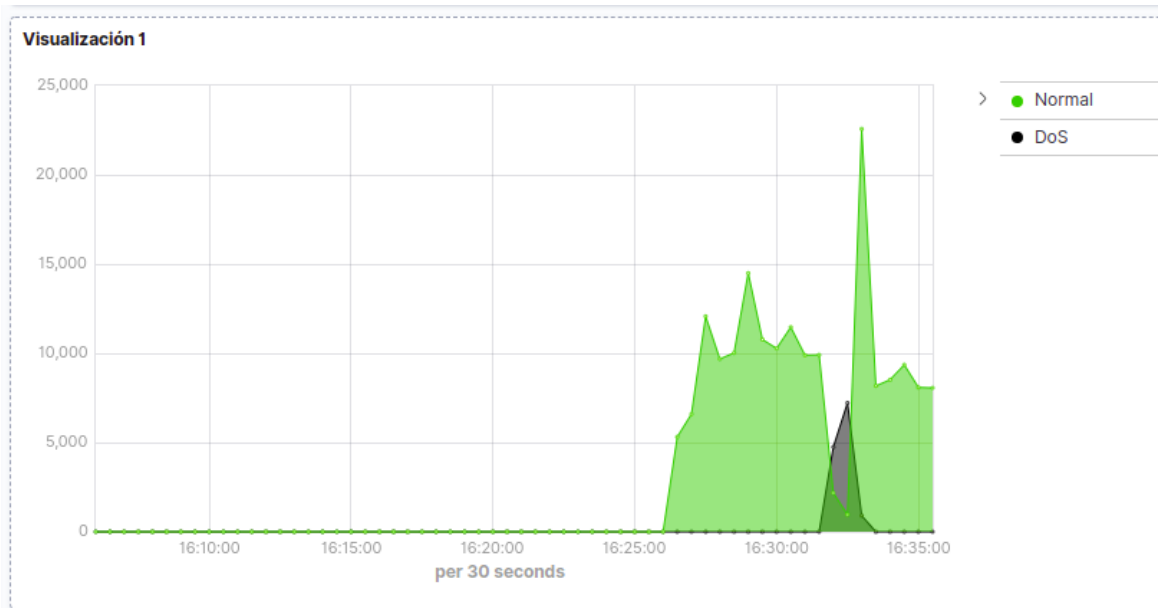
prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
Normal	2	8	70:9c:d1:8a:d2:02	70:9c:d1:8a:d2:02	67,402
Normal	2	8	20:34:fb:c5:e0:3d	20:34:fb:c5:e0:3d	17,664
Normal	2	8	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	12,728
Normal	2	8	6c:19:8f:bb:6a:2a	01:00:5e:7f:ff:fa	137
Normal	2	8	6c:19:8f:bb:6a:2a	01:00:5e:00:00:fb	51
Normal	2	8	6c:19:8f:bb:6a:2a	33:33:00:00:00:fb	36
Normal	2	8	6c:19:8f:bb:6a:2a	ff:ff:ff:ff:ff:ff	8
Normal	2	8	6c:19:8f:bb:6a:2a	01:00:5e:00:00:16	2
Normal	2	8	94:39:e5:bf:a0:3f	94:39:e5:bf:a0:3f	649
Normal	2	12	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	1,590

Export: Raw Formatted

El sistema de detección de intrusiones fácilmente reconoce el momento exacto en el que se lleva a cabo el ataque, el mismo que se llevó a cabo de igual forma a la metodología empleado en los anteriores ataques; después de cinco minutos en los cuales las condiciones de la red son normales.

Figura 41

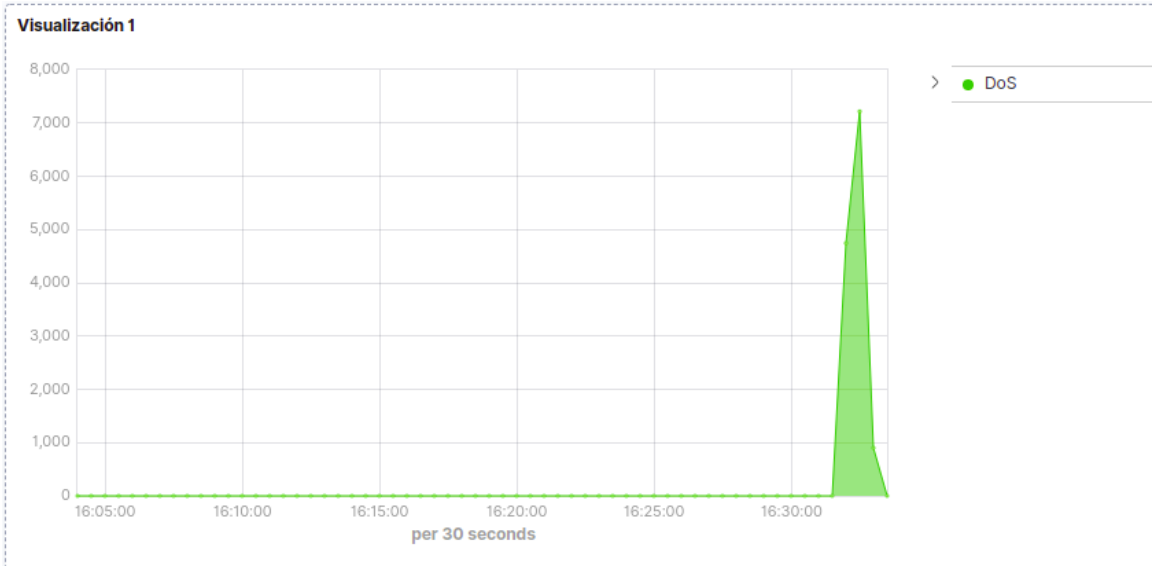
Tramas clasificadas por el orden de llegada y la cantidad



La grafica 41, permite observar que la red está bajo un ataque de denegación de servicio. Indagando aún más para conocer a que se debe este tipo de ataque, se puede filtrar el tipo de tramas que recibe. Como se sabe que la principal vulnerabilidad del protocolo 802.11 está en la no encriptación de las tramas de gestión, son estas tramas las que siempre deben ser analizadas.

Figura 42

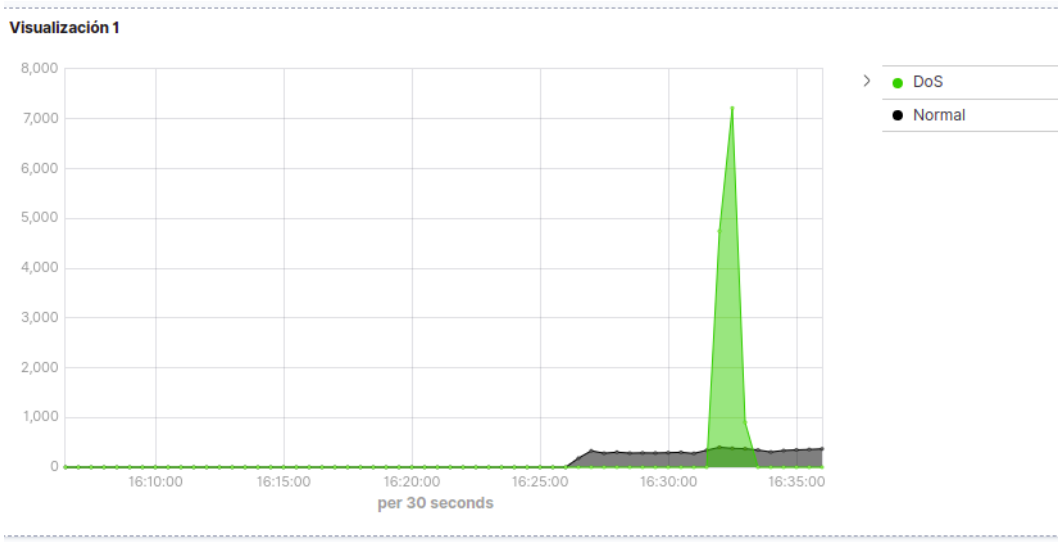
Tramas de desautenticación según la hora de llegada



De la figura anterior se puede resaltar las siguientes observaciones: en primero lugar se registra un incremento poco común de tramas de gestión que no se dan durante los primeros cinco minutos, y el tipo de tramas de gestión que incrementan son el de tipo desautenticación. Por lo cual es evidente que el canal está siendo inundado con este tipo de tramas si se compara con los cinco minutos anteriores en los cuales la cantidad de tramas de este tipo es casi nula.

Figura 43

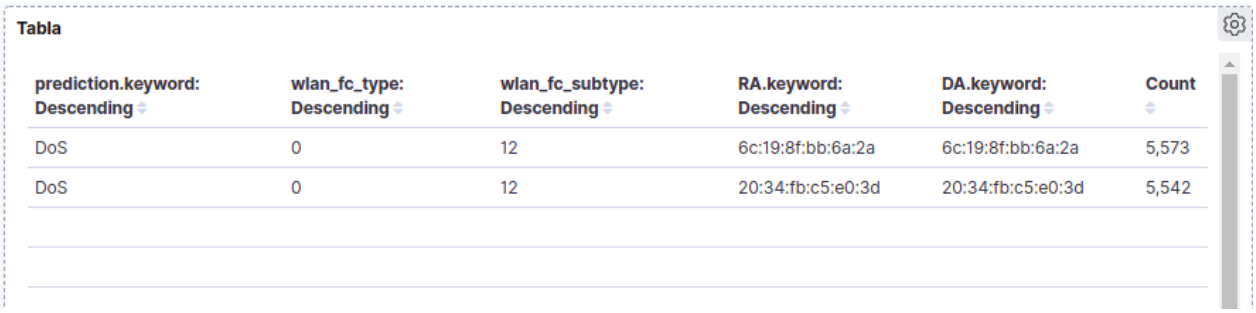
Visualización de tramas de gestión durante la prueba



Se puede filtrar todas las tramas recibidas y buscar a quien se está atacando.

Figura 44

Tramas clasificadas como de denegación de servicio



Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
DoS	0	12	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	5,573
DoS	0	12	20:34:fb:c5:e0:3d	20:34:fb:c5:e0:3d	5,542

En este caso se puede identificar las dos tramas que en un inicio se esperan cuando se intenta desautenticar a un usuario: tramas dirigidas al AP (6c:19:8f:bb:6a:2a) y al cliente (20:34:fb:c5:e0:3d). Finalmente el Dashboard muestra todas las tramas recibidas y las clasifica.

Figura 45

Total de tramas recibidas durante la prueba penetración

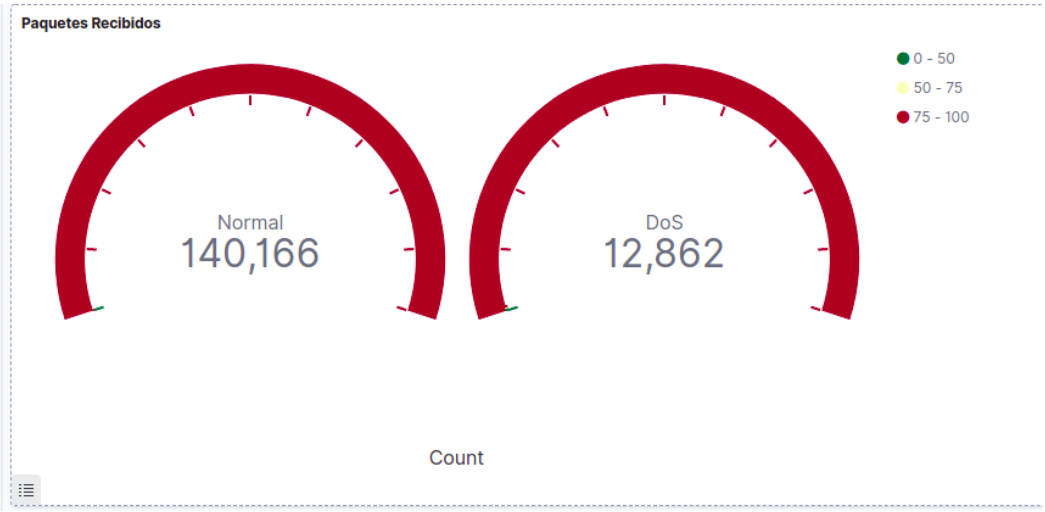
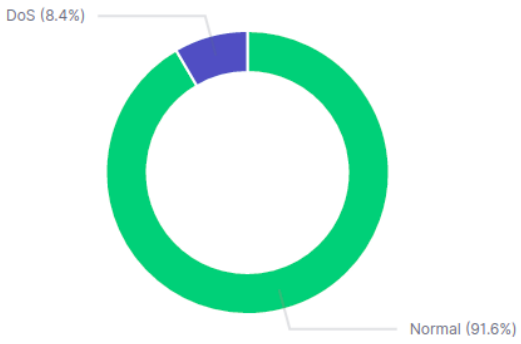


Figura 46

Tramas clasificadas por el sistema y mostradas en porcentaje

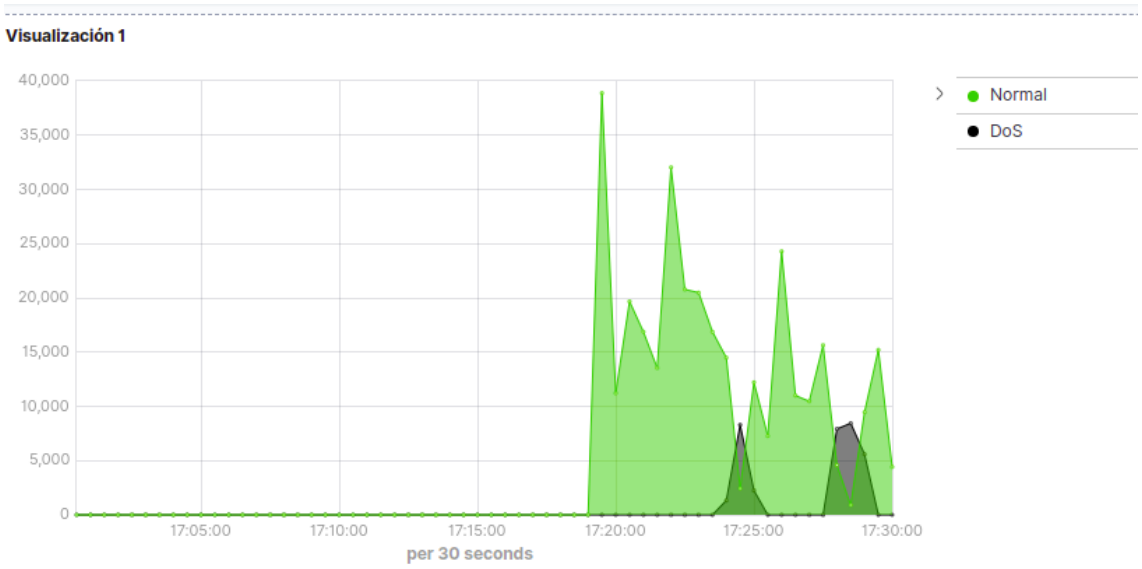


Ataque de desautenticación a todos los clientes

Este ataca tipo de desautenticación busca desconectar a todos los clientes de la red. Para ello, el atacante finge ser el AP y envía tramas del tipo de desautenticación con dirección *Broadcast* con la intención de que todos los clientes que pertenecen a la red reciban esa trama y se desconecten de la misma. Este ataque se llevó a cabo desarrollando un script en Python usando la librería Scapy. El escenario de este ataque es similar a los otros, en los cuales los cinco primeros minutos solo se produce tráfico normal y posterior a ese tiempo se realiza dos ataques.

Figura 47

Tramas recibidas clasificadas durante la prueba de penetración

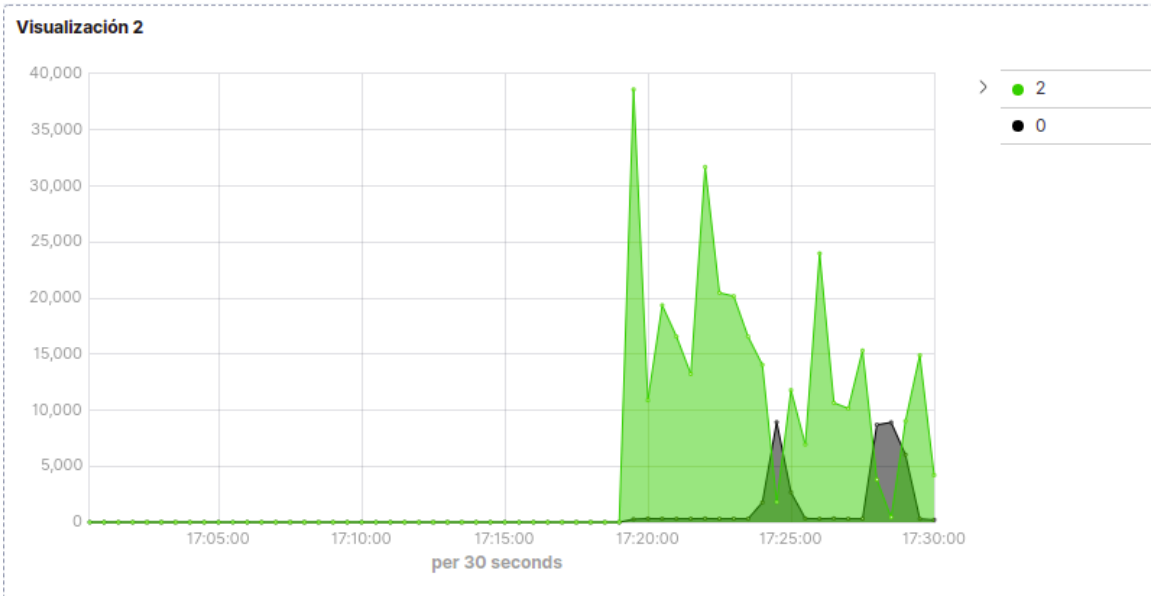


De la figura 47 se puede observar el momento exacto en los cuales se llevó a cabo los ataques. El sistema de detección de intrusiones detecta los ataques de denegación de servicio. Es evidente que durante el tiempo del ataque las tramas que

pertenecen al tráfico normal disminuyen considerablemente, lo cual es lógico ya que durante los ataques ningún cliente tiene acceso a la red y por lo tanto no tiene comunicación con el AP.

Figura 48

Tramas de Gestión y Datos



De igual forma que en el caso en el cual se ataca a un cliente, el aumento de tramas de gestión es un claro indicador que la red sufre un ataque. En la figura 48, se evidencia como las tramas de gestión durante dos lapsos de tiempo aumentan repentinamente. Es una anomalía en el tráfico de la red que indica un ataque a la red.

Figura 49

Tramas de gestión clasificadas

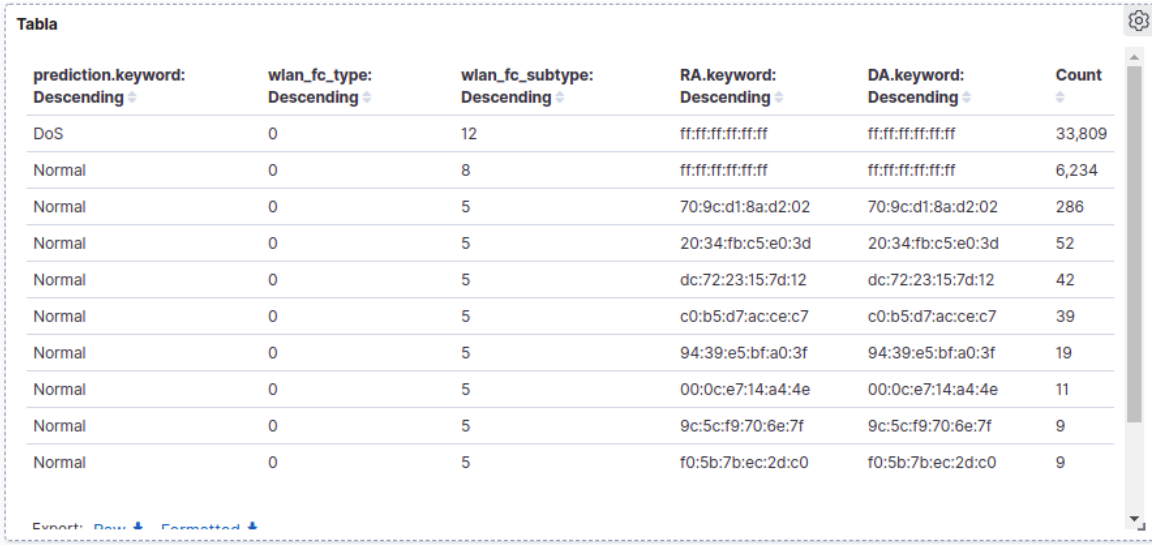
Tabla 3

prediction.keyword: Descending	frame_len: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	Count
DoS	44	0	12	33,809
Normal	310	0	8	6,234
Normal	395	0	5	562
Normal	51	0	13	379
Normal	48	0	13	179
Normal	48	0	11	130
Normal	275	0	4	102

Si se filtra únicamente las tramas de gestión, se puede observar todos los tipos de tramas que se han recibido y como se han clasificado. En este caso el sistema detectó una gran cantidad de tramas de desautenticación que clasifico como denegación de servicio. Adicional se detectó tramas de: Beacon, Probe Response, Action, Autenticación, las cuales el sistema detecto como de tráfico normal.

Figura 50

Tramas de gestión y los subtipos detectados



Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
DoS	0	12	ff:ff:ff:ff:ff:ff	ff:ff:ff:ff:ff:ff	33,809
Normal	0	8	ff:ff:ff:ff:ff:ff	ff:ff:ff:ff:ff:ff	6,234
Normal	0	5	70:9c:d1:8a:d2:02	70:9c:d1:8a:d2:02	286
Normal	0	5	20:34:fb:c5:e0:3d	20:34:fb:c5:e0:3d	52
Normal	0	5	dc:72:23:15:7d:12	dc:72:23:15:7d:12	42
Normal	0	5	c0:b5:d7:ac:ce:c7	c0:b5:d7:ac:ce:c7	39
Normal	0	5	94:39:e5:bf:a0:3f	94:39:e5:bf:a0:3f	19
Normal	0	5	00:0c:e7:14:a4:4e	00:0c:e7:14:a4:4e	11
Normal	0	5	9c:5c:f9:70:6e:7f	9c:5c:f9:70:6e:7f	9
Normal	0	5	f0:5b:7b:ec:2d:c0	f0:5b:7b:ec:2d:c0	9

Export: Raw Formatted

La figura 50 muestra el resultado que se esperaba obtener al tratarse de un ataque de desautenticación en Broadcast. La dirección de las tramas de denegación de servicio tiene dirección MAC ff:ff:ff:ff:ff:ff , las mismas que el atacante envía para que todos los clientes conectados a un determinado AP reciban y sepan que la comunicación con este último ha cesado.

Figura 51

Total de tramas clasificadas

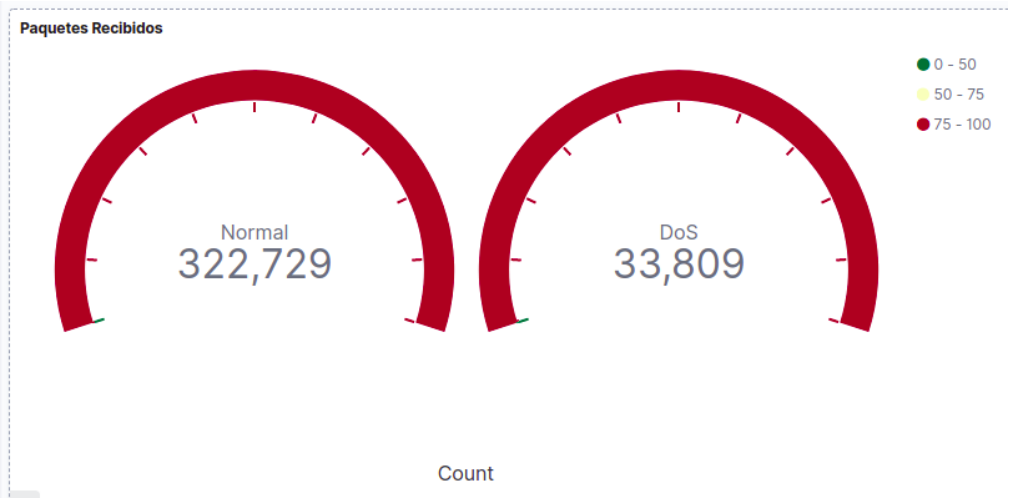
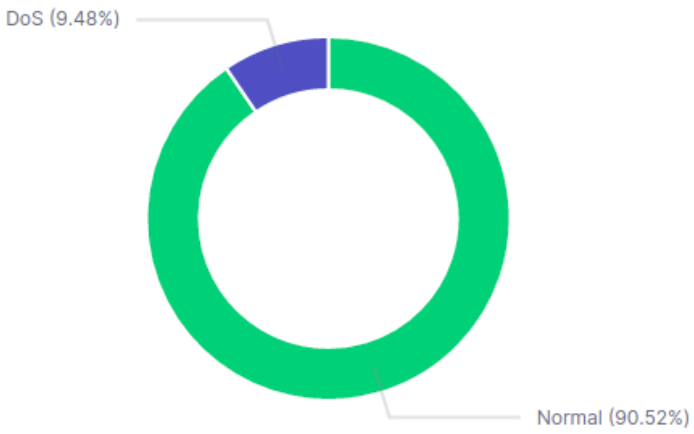


Figura 52

Tramas clasificadas y mostradas en porcentaje



Ataque de disociación a todos los clientes

Este ataque de denegación de servicio utiliza las tramas de disociación para corromper la comunicación entre los clientes y el AP. El atacante en este caso ya no envía tramas de desautenticación sino más bien de disociación para denegar el servicio. En este caso el atacante puede enviar tramas con dirección en Broadcast o a los clientes que desea atacar. Para la realización de este ataque se utilizó la herramienta mdk3, la cual realiza un ataque inteligente, en el cual analiza y recopila información de todos los clientes conectados a un AP y envía tramas de desautenticación y disociación a todos los clientes conectados a la red. La metodología que se utiliza es igual a los ataques anteriores.

Figura 53

Tramas de Gestión y Control según la cantidad y hora de llegada

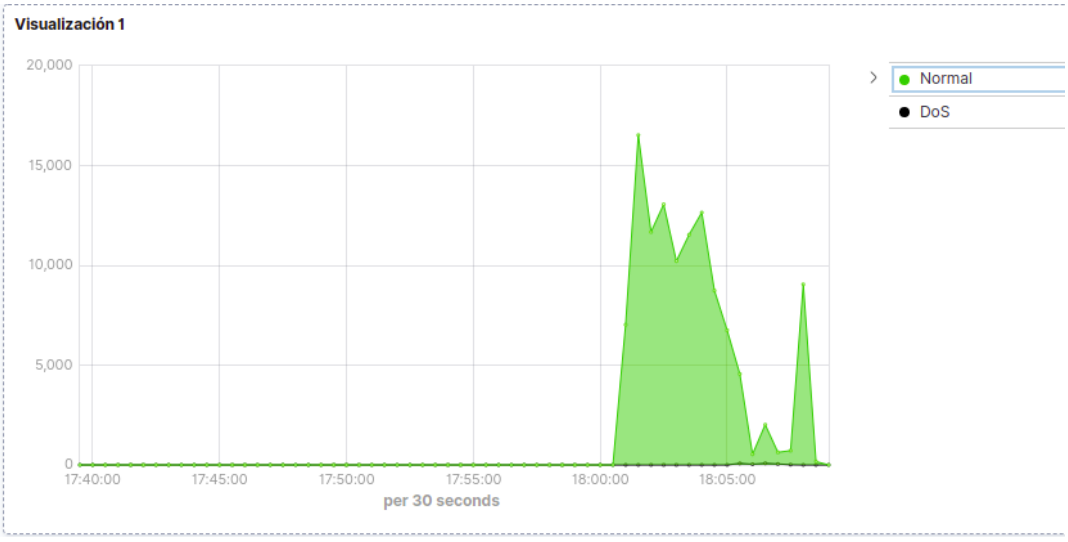
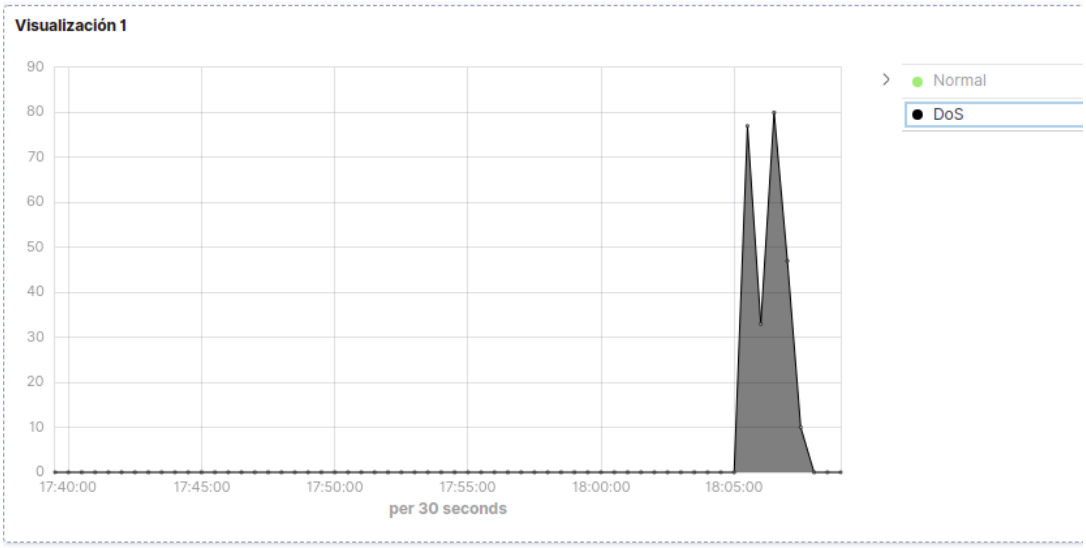


Figura 54

Tramas clasificadas como denegación de servicio

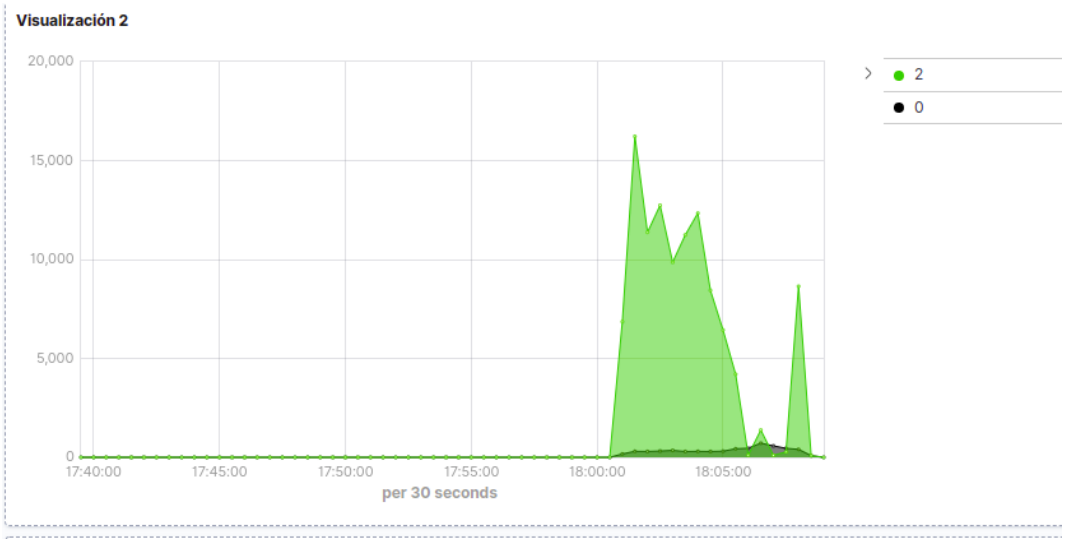


Este ataque es difícil de visualizar a simple vista ya que la cantidad de tramas que el atacante envía son muy bajas, por lo cual es necesario desplegar por separado el análisis de las tramas en tiempo real, como se muestra en las dos figuras anteriores.

Una vez que se detecta el ataque, se procede a indagar a que se debe, para lo cual, y como se realizó en las ocasiones anteriores siempre se revisa el tipo de tramas que recibe el sistema para identificar alguna anomalía en el sistema.

Figura 55

Tramas de gestión y datos



La figura 55 muestra aparentemente que no existe una inundación de tramas del tipo de gestión, lo que sucedía en los anteriores casos. Como se puede observar las tramas de gestión no presentan cambios abruptos muy notorios.

Figura 56

Tramas de denegación de servicio y sus campos

Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
DoS	0	10	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	63
DoS	0	10	04:f0:21:02:45:52	04:f0:21:02:45:52	29
DoS	0	10	70:9c:d1:8a:d2:02	70:9c:d1:8a:d2:02	20
DoS	0	10	20:34:fb:c5:e0:3d	20:34:fb:c5:e0:3d	12
DoS	0	12	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	59
DoS	0	12	04:f0:21:02:45:52	04:f0:21:02:45:52	32
DoS	0	12	70:9c:d1:8a:d2:02	70:9c:d1:8a:d2:02	20
DoS	0	12	20:34:fb:c5:e0:3d	20:34:fb:c5:e0:3d	12

Sin embargo, si filtramos las tramas según el tipo de clasificación, podemos observar que las tramas clasificadas como denegación de servicio son del tipo de gestión; específicamente se detectaron tramas del subtipo de desautenticación y de disociación, lo cual era el resultado esperado. Además se puede observar a quien están dirigidas, pues se observa que envía tramas tanto al AP (6c:19:8f:bb:6a:2a) como a los clientes. Es interesante notar que también la cantidad de tramas en comparación a las tramas de tráfico normal es totalmente inferior, debido a que el atacante no envía tramas constantemente sino que envía tramas solo cuando el cliente desea reestablecer la conexión.

Figura 57

Total de tramas recibidas

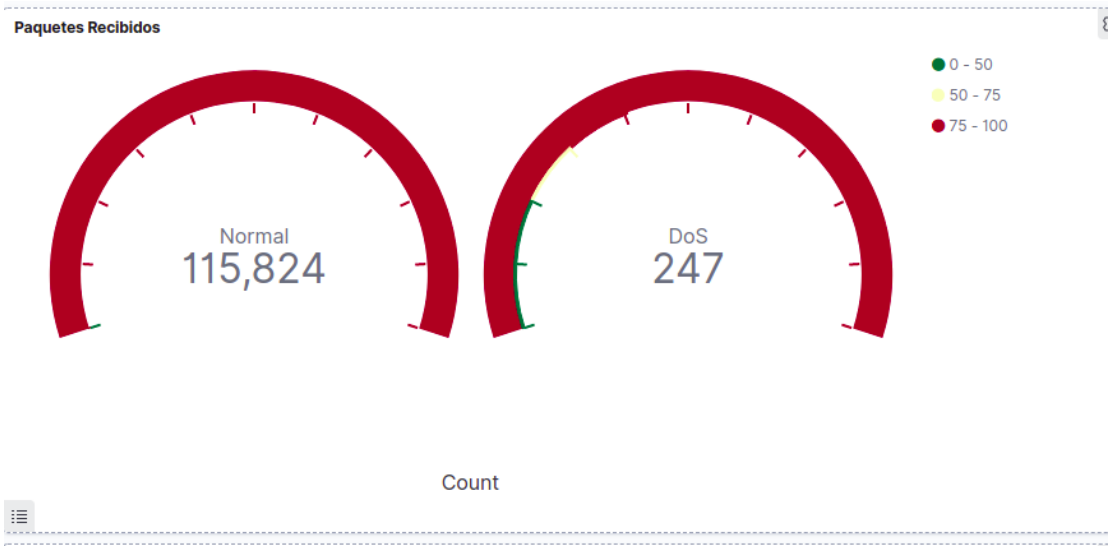
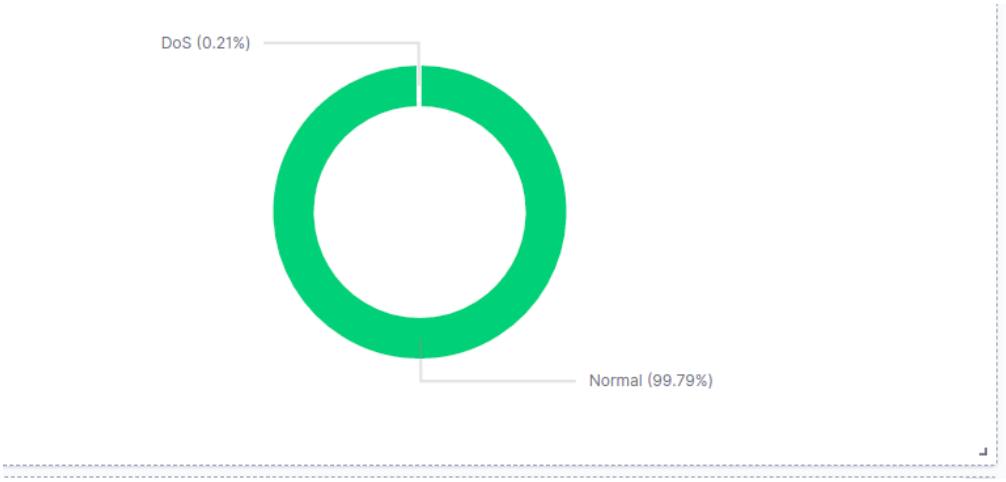


Figura 58

Total de tramas recibidas y mostradas en porcentaje

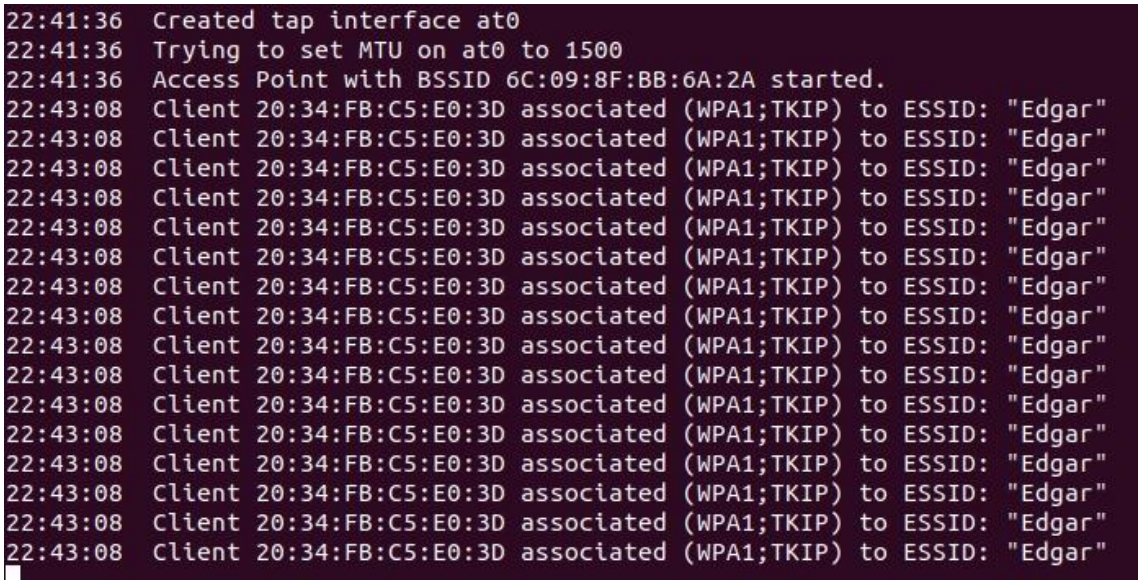


Ataque de desautenticación e Impersonificación

El objetivo de este ataque es que el cliente se conecte a un AP falso (*Evil Twin,* por sus siglas en inglés), para de esta forma lograr capturar todo el tráfico que el usuario genere, además de poder recuperar información sensible como cuentas bancarias o correos electrónicos con sus respectivas contraseñas a través de un portal cautivo. En primera instancia el atacante, crea una red con el SSID idéntico de la red que se va a atacar. Después el atacante envía tramas de desautenticación para lograr que el cliente pierda conexión de la red original. Posterior a eso, el dispositivo del cliente intentará conectarse a la red falsa con el mismo SSID (nombre), logrando así que el cliente no se percate de lo sucedido.

Figura 59

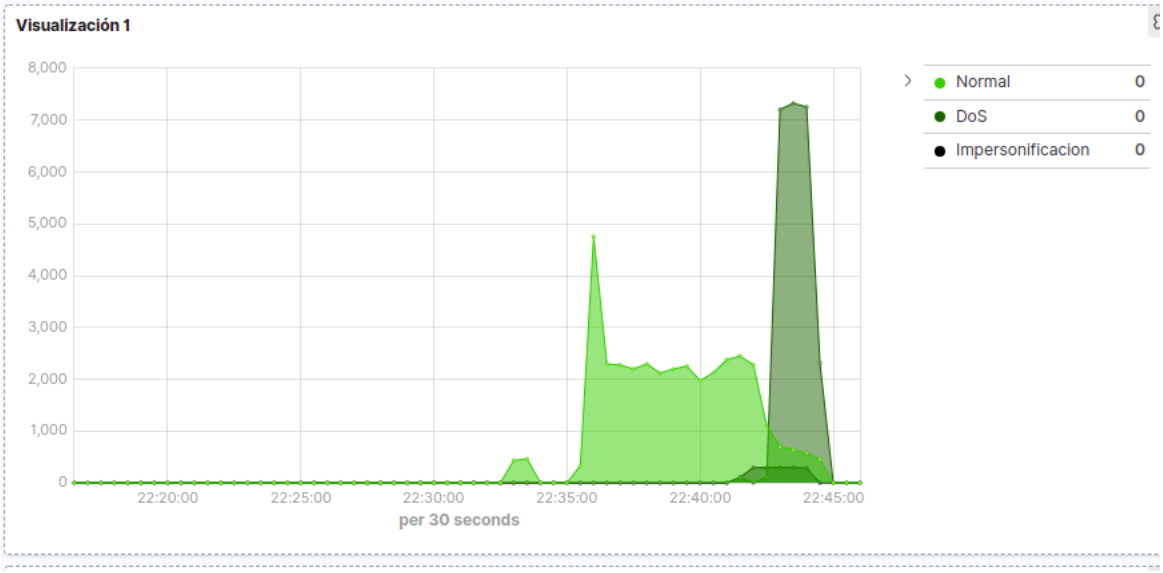
Cliente asociado a una red maliciosa creada por un atacante



La metodología que se llevó a cabo para este ataque es una que se asemeja a un caso real. En primera instancia el tráfico de la red es normal, después el atacante establece la red falsa y luego realiza el ataca de desautenticación. Las herramientas utilizadas para este ataque son: airbase-ng para la creación de la red falsa y aireplay-ng para la desautenticación.

Figura 60

Tramas recibidas y clasificadas durante la prueba de penetración



En la figura 60 se puede distinguir todas las etapas. En primera instancia, el sistema solo detecta tráfico normal. Después de unos minutos (6 aproximadamente) se detecta un ataque de Impersonificación, el cual no se caracteriza por tener una cantidad grande de tramas que permitan su detección, más bien solo son tramas Beacon

enviadas en forma muy similar a cualquier AP normal. Posterior al inicio del ataque de Impersonificación, el ataque de desautenticación se lleva a cabo.

Figura 61

Tramas maliciosas clasificadas por el sistema

Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
Normal	0	8	ff:ff:ff:ff:ff:ff	ff:ff:ff:ff:ff:ff	5,860
Impersonificacion	0	8	ff:ff:ff:ff:ff:ff	ff:ff:ff:ff:ff:ff	1,562

De la figura 61 se puede identificar las tramas Beacon que recibe el sistema; las normales pertenecen a las tramas enviadas por el AP autentico y las de Impersonificación pertenecen a la red falsa que simula ser la original. Evidentemente este tipo de tramas tienen dirección de Broadcast.

Figura 62

Tramas de Gestión

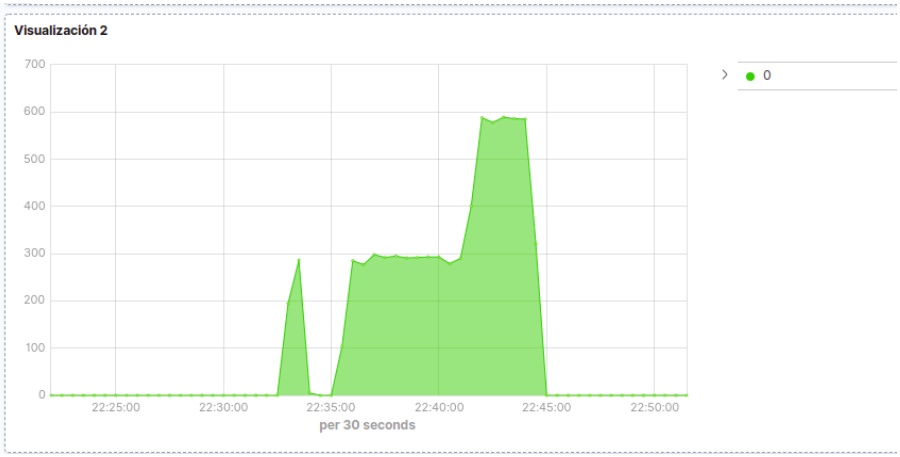
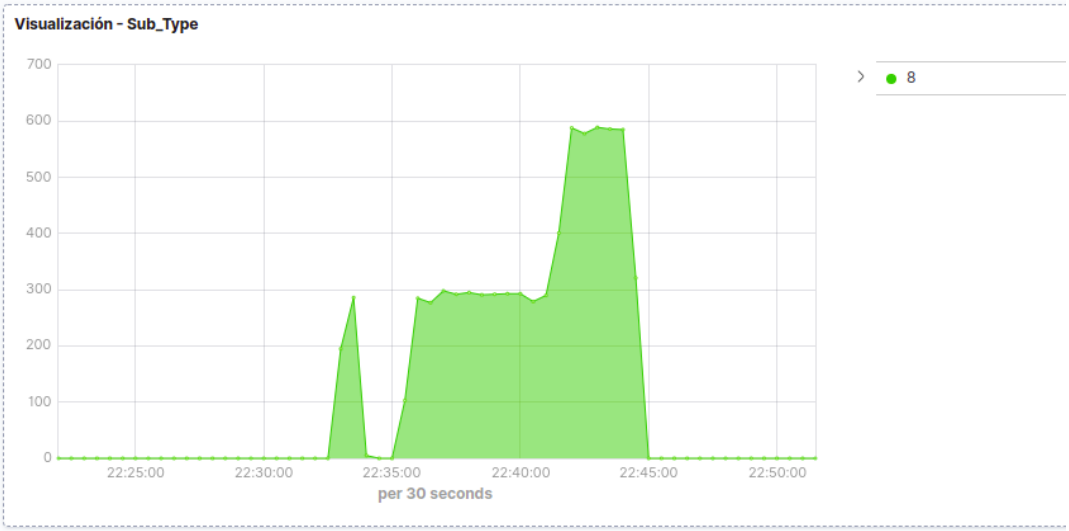


Figura 63

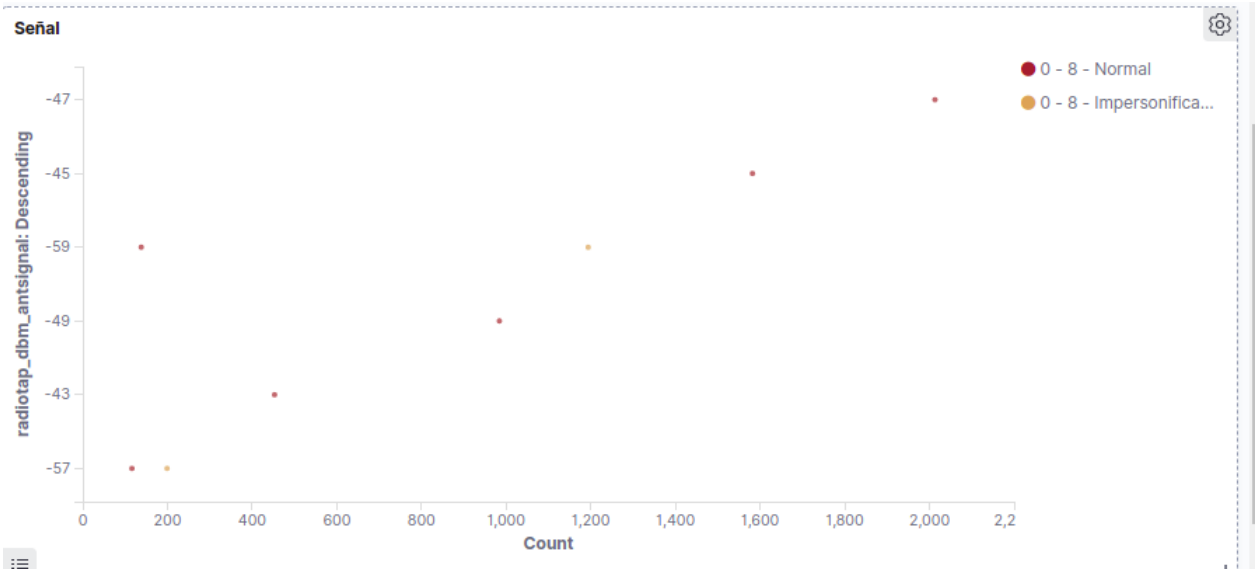
Tramas Beacon detectadas durante la prueba de penetración



Las gráficas anteriores muestran como hay un cambio anormal de la cantidad de tramas Beacon en el canal. En el lapso en el que solo se registra tráfico normal, la cantidad de tramas Beacon recibidas es casi constante. Sin embargo, cuando el ataque se detecta, las tramas Beacon aumentan considerablemente.

Figura 64

Potencia de la señal detectada en cada trama y clasificadas



La figura 64 es muy útil porque permite detectar la cercanía del atacante, que por lo general suele ser muy cercana al cliente para que este fácilmente se asocie a la red falsa. Esta gráfica permite identificar la señal de potencia de las tramas recibidas clasificas como ataque de Impersonificación.

Figura 65

Tramas de denegación de servicio

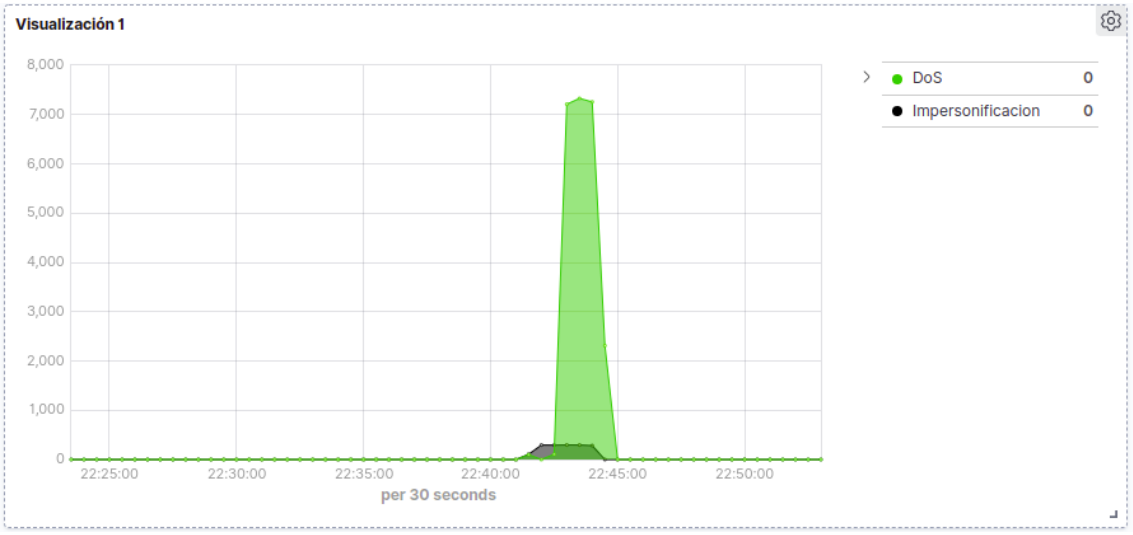
Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
DoS	0	12	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	12,112
DoS	0	12	20:34:fb:c5:e0:3d	20:34:fb:c5:e0:3d	12,099

Además de las tramas de Impersonificación, se detectan las tramas de denegación de servicio. En este caso como se puede observar de la figura anterior se identifican tramas de desautenticación dirigidas hacia el AP y al cliente.

Figura 66

Tramas de DoS e Impersonificación según la hora de detección



Como se muestra en la figura anterior, las tramas de desautenticación se envían consta mente para evitar que el usuario vuelva a conectarse a la red original. El sistema detecto los tres tipos de ataques al cual fue sometida la red y pudo identificar las tramas que caracterizan a cada ataque

Figura 67

Total de tramas

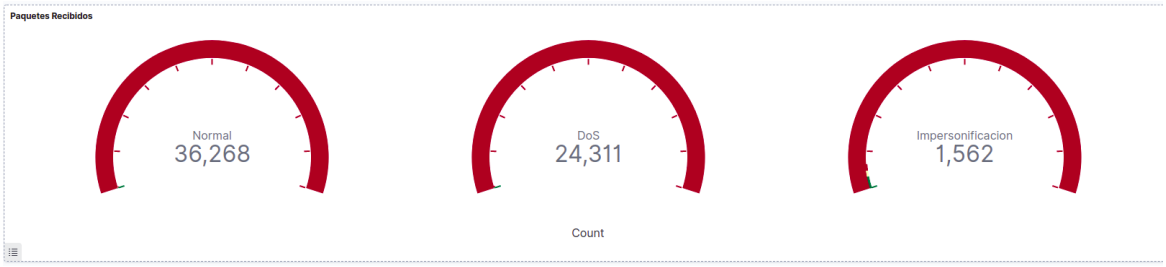
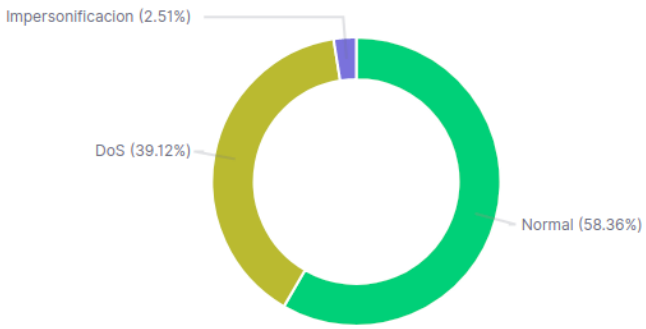


Figura 68

Tramas clasificadas y mostradas en porcentaje



Inyección de una Trama Original

El atacante en este caso no genera ninguna trama falsa, sino más bien lo que busca el atacante es escuchar todo el tráfico de la red y esperar capturar una trama de desautenticación correspondiente a un cliente original para poder utilizarla posteriormente. Una vez capturada la trama original de desautenticación, el atacante puede inyectarla en el canal para buscar denegar el servicio al cliente cuando este vuelva a conectarse. Este método puede pasar desprevenido por el sistema de detección de intrusiones ya que al tratarse de una trama original, el modelo de machine learning no puede identificar que se trata de un ataque de denegación de servicio. Para comprobar esta hipótesis, se evalúa al sistema de una manera similar a los ataques anteriores. La captura de la trama de desautenticación se la realizo previamente antes de iniciar la prueba.

Figura 69

Tramas de gestión detectadas

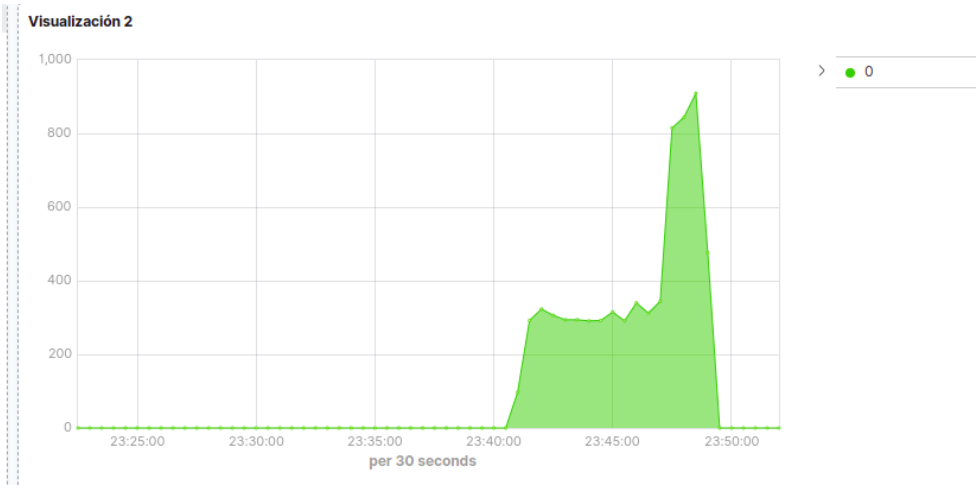
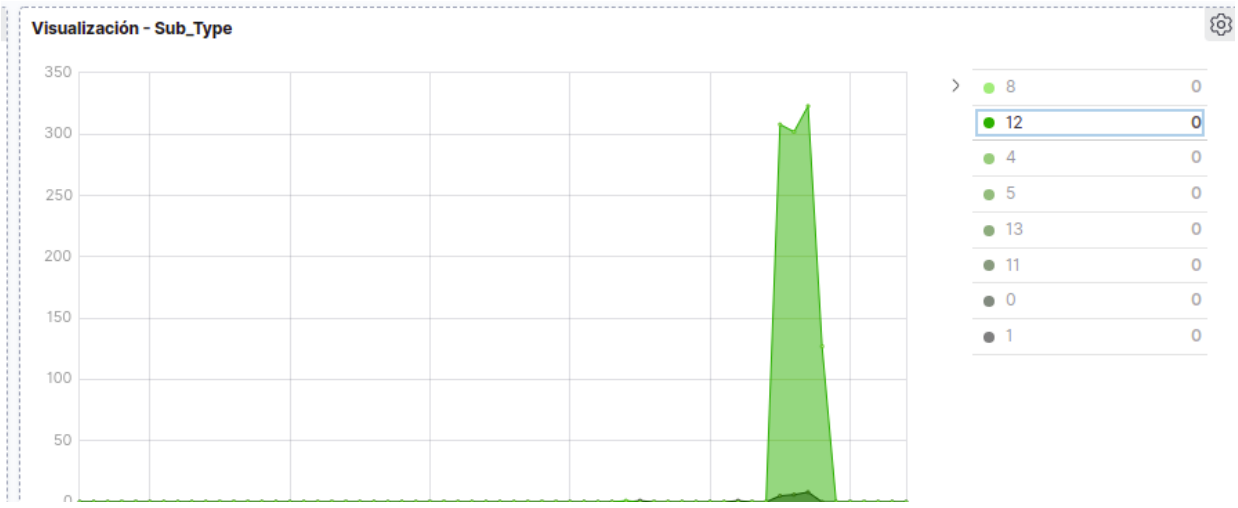


Figura 70

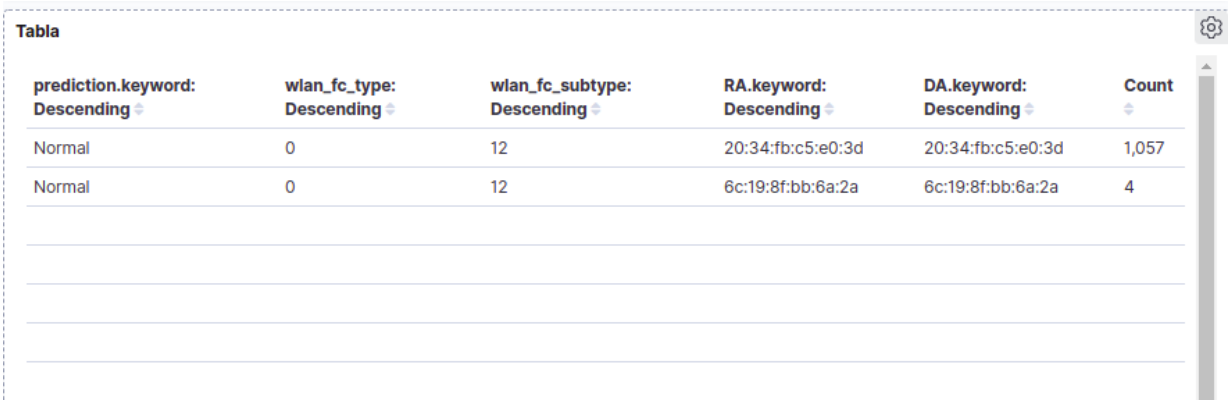
Tipos de tramas de gestión



En este análisis solo se tomará en cuenta las tramas de gestión, ya que no las tramas de datos y control no presentan ningún interés. Por lo cual, las dos figuras anteriores muestran solo las tramas de gestión generadas durante todo el tiempo de la prueba. En la figura 69, se puede observar que durante los primeros cinco minutos las tramas de gestión se mantienen durante un rango bastante constante no muy elevado, sin embargo posteriormente a los cinco minutos las tramas de gestión aumentan significativamente. En la figura 70 , se puede observar que ese aumento corresponde a tramas de desautenticación. Comparando con la primera figura, coinciden en el tiempo en el que aparecen.

Figura 71

Tramas clasificadas como normales



Tabla

prediction.keyword: Descending	wlan_fc_type: Descending	wlan_fc_subtype: Descending	RA.keyword: Descending	DA.keyword: Descending	Count
Normal	0	12	20:34:fb:c5:e0:3d	20:34:fb:c5:e0:3d	1,057
Normal	0	12	6c:19:8f:bb:6a:2a	6c:19:8f:bb:6a:2a	4

La figura 71 se muestra la cantidad de tramas de desautenticación que el sistema detecta. Estas tramas tienen como objetivo atacar a un cliente en específico. El atacante reenvía solo tramas de desautenticación fingiendo ser el AP, ya que la dirección MAC de destino es la del cliente (20:34:fb:c5:e0:3d). Además, como se muestra en la figura, estas tramas están clasificadas como tráfico normal. De esta manera se observa que este ataque no puede ser detectado por el sistema de detección de intrusiones, ya sea que el atacante inyecte tramas de desautenticación o disociación.

Análisis del flujo de Trafico

En vista de que el reenvió de tramas originales de desautenticación o disociación, una de las alternativas para poder corregir esta deficiencia del sistema seria analizar el flujo de tráfico. El sistema de detección de intrusiones presentado en este trabajo está basado en el análisis de los campos de la cabecera MAC de las tramas

802.11 . Es decir, analiza cada una y en base a las características de cada campo, el sistema clasifica las tramas, ya sea: normal, DoS (denegación de servicio) o Impersonificación. Entre los aspectos más importantes que el sistema tiene en cuenta para diferenciar o clasificar las tramas son:

- En el campo de secuencia de cada trama, las falsificadas o tramas maliciosas creadas por un atacante siempre tiene este valor fijo. Por lo general siempre se encuentra en un valor de 0. Esto se debe a que este parámetro se autogestiona mediante el controlador, por lo cual este parámetro no está al alcance de modificación por parte del atacante.

- La longitud de cada trama es otro parámetro que puede ser un indicativo de una señal. En el caso de las tramas Beacon, las tramas maliciosas siempre tiene un valor pequeño en comparación a las tramas originales, ya que en las originales se encuentran todos los parámetros que el usuario necesita conocer para poder establecer una conexión con el AP.

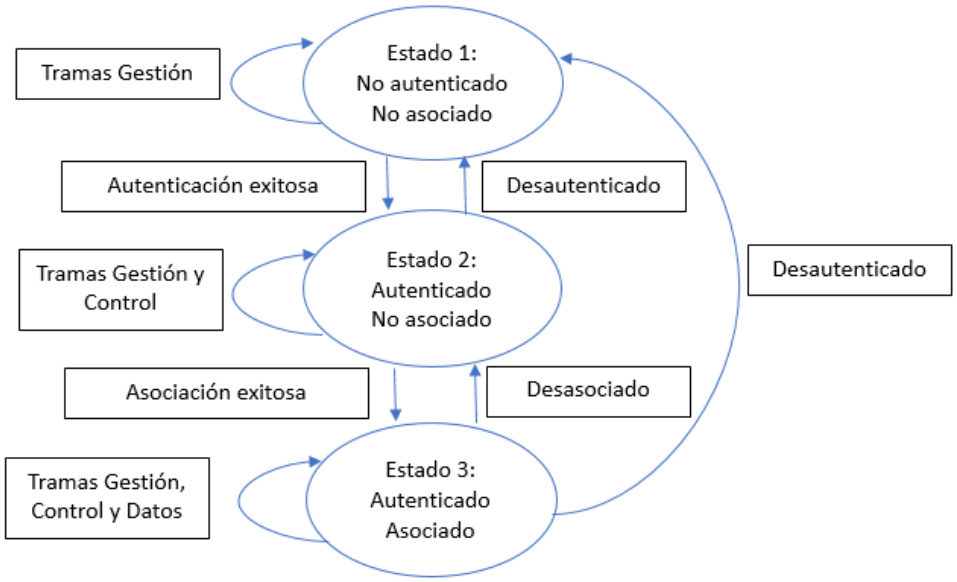
- El parámetro de duración es otro campo que por lo general no presenta valores adecuados o normales. Este parámetro en tramas maliciosas suele tener por defecto un valor de 0, el cual es insertado automáticamente por el controlador y en muchas ocasiones es muy difícil de cambiar debido a la compatibilidad de los controladores con las tarjetas USB Wifi.

- En el campo del Reason Code todas las tramas maliciosas siempre tienen un valor por defecto, el cual esta inicializado a 0. Esto especialmente se evidencia en todas las tramas generadas mediante las herramientas de Kali Linux.

El problema surge cuando el atacante maliciosamente reenvía tramas originales de desautenticación o disociación, las cuales al no ser falsificadas por el atacante, estas pasan desapercibidas por el detector. Sin embargo, y gracias a las gráficas del tráfico que el sistema muestra a través del Dashboard, se puede observar que cuando existe un ataque de denegación de servicio, el flujo de tráfico, especialmente con las tramas de control, empieza a aumentar de manera poco usual. Este aumento de tráfico de tramas de gestión indica que se inunda el canal con tramas de desautenticación o disociación. Precisamente este flujo de tráfico puede llegar a ser muy útil para detectar estos ataques de denegación de servicio.

Figura 72

Estados de un cliente en una red inalámbrica



Tomando en cuenta la figura anterior, es claro que existen patrones muy claros que definen el tráfico en todas las redes Wifi. Es poco común, que el usuario que se encuentre en el estado 3 empiece a generar gran cantidad de tramas de desautenticación dirigidas hacia el AP, o viceversa. Analizar y caracterizar el flujo de tráfico permitiría detectar anomalías comunes o ataques derivados existentes, lo cual haría a un sistema de detección de intrusiones más robusto. Los patrones que se deberían tomar en cuenta son:

- Cantidad de tramas Beacon

- Tipos de tramas que se generan ya que en base a esto se puede determinar el estado en el que se encuentra el usuario
- Estados en el que se encuentra el usuario y el tiempo que permanece allí
- Cantidad de tramas de desautenticación o disociación
- Dirección MAC de destino de las tramas
- Cantidad de tramas de gestión y control generadas por el usuario
- Cantidad de tramas de gestión y control generadas por el AP
- Potencia de la señal de las tramas pertenecientes al AP recibidas por el usuario

Estas son algunas de las características que se deberían tomar en cuenta para poder crear, o en su defecto, complementar el sistema de detección de intrusiones para poder aumentar la robustez de este.

Capítulo V

Conclusiones, Recomendaciones y Trabajos Futuros

Conclusiones

En este trabajo se desarrolló el diseño e implementación de un sistema de detección de intrusiones para detectar ataques a redes Wifi que utilizan el protocolo 802.11 en tiempo real. El diseño e implementación solo utilizo herramientas de código abierto, ya que el objetivo es que sea implementado en pequeñas oficinas , hogares o pequeñas y medianas empresas que siempre necesitan implementar mecanismos para proteger la seguridad de su red y las cuales no se permiten costear sistemas costos de seguridad informática. Al usar herramientas de código abierto la implementación del sistema prácticamente no conllevaría ningún costo. Uno de los objetivos implícitos al diseñar este sistema de detección de intrusiones es, en cierta forma, proteger a redes Wifi de vulnerabilidades producto de fallas en el diseño del protocolo 802.11, para lo cual se ha utilizado un modelo de aprendizaje supervisado. Es por ello, que no para remediar estas fallas no ha sido necesario modificar el protocolo, lo que conllevaría un enorme esfuerzo. Al implementar este sistema, se puede brindar mayor seguridad a las redes inalámbricas Wifi y facilita a un administrador de red detectar anomalías en el tráfico. El sistema además es fácil de implementar ya que no es necesario contar con hardware especializado para su implementación, pues solamente es necesario contar con una tarjeta de red Wifi USB que se puede encontrar en el mercado por un precio accesible.

El diseño del sistema es fácilmente escalable, pues este está diseñado para procesar grandes cantidades de datos provenientes de varias redes Wifi, inclusos de varios dispositivos capturadores de tráfico. Todo el diseño del sistema de detección de intrusiones fácilmente puede ser desplegado en la nube para poder garantizar el rendimiento. Al desplegarlo en la

nube se podría aprovechar los grandes recursos de esta para poder utilizar el sistema de detección de intrusiones y analizar varias redes wifi o canales a la vez, todo esto en tiempo real.

El sistema no necesita modificaciones, tanto en software como en hardware, en la red que se va a analizar. El funcionamiento del sistema es independiente de los dispositivos y sus características que conforman la misma.

El sistema es fácilmente modificable. Es decir, se puede actualizar, modificar o cambiar el modelo de machine learning rápidamente sin alterar la arquitectura completa del sistema. Además de que se le puede agregar modificaciones al procesamiento de datos según las necesidades del administrador de red, como por ejemplo: extraer más características de las tramas.

El sistema fue capaz de diferenciar tráfico normal de tráfico malicioso generado por atacantes utilizando diferentes herramientas de penetración.

Todo el diseño esta implementado para ser a prueba de fallas, ya que todos los clústeres permiten crear nodos de respaldo.

La visualización creada en la interfaz de Kibana es flexible y modificable. Además esta permite la búsqueda en tiempo real de las diferentes tramas que se desee analizar.

Recomendaciones

Uno de los puntos clave para que el rendimiento del sistema de detección de intrusiones sea optimo, es el de asegurarse de configurar adecuadamente los clústeres, tanto el de Kafka, ELasticsearch y Apache Spark. Es deseable tener los recursos necesarios para poder configurar nodos de respaldo que permitan poder almacenar todas las tramas para que, en caso de falla de los nodos principales, el sistema no deje de funcionar. El cluster de Apache Spark es

necesario que se lo despliegue en un servidor o computadora con suficientes recursos de hardware, ya que este nodo es la encargada de realizar todo el procesamiento de datos y analizar el tráfico con el modelo de Machine Learning. Es necesario desplegar todo el sistema bajo el sistema operativo Linux para asegurar el correcto funcionamiento del sistema.

Trabajos Futuros

Implementar el sistema de detección de intrusiones en la nube y realizar pruebas en un entorno más grande en comparación al que se utilizó en este trabajo para poder evaluar el desempeño del sistema.

Modificar el modelo de Machine Learning para cambiar el enfoque del análisis para analizar las tramas de la red. Es decir, en este trabajo se analizó las características de cada trama generada y en base a esto se pudo determinar si las tramas son normales o maliciosas. Pero, como se presentó en este trabajo, el atacante puede crear ataques que pasen desapercibidos por el sistema, como sucedió en el caso en el que, el atacante utilizo una trama de desautenticación original para realizar un ataque de denegación de servicio. Por ende, una de las soluciones que puede detectar este tipo de debilidades del sistema, seria cambiar el enfoque de análisis, y en lugar de analizar las tramas, analizar el tráfico en base a los estados (estado 1, estado 2 o estado 3) en los cuales un cliente se encuentra. Esto no solo permitiría detectar nuevos ataques de denegación de servicio sino también cualquier nuevo ataque que puede encontrarse a futuro en la capa 2.

Fuentes Bibliográficas

Allahdadi, A., & Morla, R. (2016). Anomaly Detection and Modeling in 802.11 Wireless Networks. *Springer* .

Chambers, B., & Zaharia, M. (2018). *Spark The Definitive Guide .* Sebastopol: O'Reilly Media.

Chio, C., & Freeman, D. (2018). *Machine Learning & Security .* Sebastopol : O'Reilly Media.

Dasari, M., & Univeristy, S. B. (2017). Real Time Detection of MAC Layer DoS Attacks in IEEE 802.11 Wireless Networks. *IEEE*. Obtenido de: http://doi.org/ 10.1109/SCOReD50371.2020.9250990

Elhigazi, A., Hamdan, M., Razak, S., Mohammed, B., Abaker, I., & Elsafi, A. (2020). Authentication Flooding DOS Attack Detection and Prevention in 802.11. *IEEE*.

Fleck, B., & Potter, B. (2002). *802.11 Security .* O'Reilly.

Gaitán, V. (2017). *Monitorización y Análisis.* Madrid.

Garg, N. (2015). *Learning Apache Kafka.* Birmingham : Packt Publishing Ltd.

Gast, M. (2002). *802.11 Wireless Networks: The Definitive Guide .* O'Reilly.

Gormley, C., & Tong, Z. (2015). *Elasticsearch The Definitive Guide .* Sebastopol: O'Reilly Media.

Hiertz, G. R., & Denteneer, D. (2010). *The IEEE 802.11 Universe.* IEEE. Obtenido de: http://doi.org/ 10.1109/MCOM.2010.5394032

Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2016). Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. *IEEE*.

Le, T.-T.-H., Park, T., Cho, D., & Kim, H. (2018). An Effective Classification for DoS Attacks in Wireless Sensor Networks. *IEEE*. Obtenido de: http://doi.org/ 10.1109/ICUFN.2018.8436999

Narkhede, N., Shapira, G., & Palino, T. (2017). *Kafka The Definitive Guide .* Sebastopol: O'Reilly Media.

Neelakantan, P., & Nagesh, C. (2011). Role of Feature Selection in Intrusion Detection Systems for 802.11 Networks. *IRNET International Journal of Smart Sensor and Adhoc Network*. Obtenido de: https://www.interscience.in/ijssan/vol1/iss2/14

Osterhage, W. (2018). *Wireless Network Security.* Frankfurt: CRC Press.

Póser, V., & Kozlovszky, M. (2019). WiFi vulnerability caused by SSID forgery in the IEEE protocol 802.11 . *IEEE*. Obtenido de: http://doi.org/ 10.1109/SAMI.2019.8782775

Rackley, S. (2007). *Wireless Networking Technology .* Oxford: Elsevier.

Ramachandran, V., & Buchanan, C. (2015). *Kali Linux Wireless Penetration.* BIRMINGHAM: Packt Publishing Ltd.

Reyes, A., Vaca, F., Aguayo, G., Niyaz, Q., & Devabhaktuni, V. (2020). A Machine Learning Based Two-Stage Wi-Fi Network. *Electronics*. Obtenido de: https://doi.org/10.3390/electronics9101689

Salloum, S., Dautov, R., & Chen, X. (2016). Big data analytics on Apache Spark. *Springer*.

Thing, V. L., & Cluster, C. S. (2017). IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach. *IEEE*. Obtenido de: http://doi.org/10.1109/WCNC.2017.7925567

Vallejo, M. A. (2016). *Evaluación del Desempeño Ddel Estándar IEEE 802.11N en un ambiente de Laboratorio. Caso de Estudio ESPE.* Quito. Obtenigo de: http://repositorio.puce.edu.ec/handle/22000/13137