ANÁLISIS DE VULNERABILIDADES A NIVEL DE SEGURIDAD EN REDES SDN PARA LOS PLANOS DE CONTROL Y PLANO DE DATOS

DAIRON JAVIER RAMOS SUAVITA

Trabajo de grado presentado como requisito para optar al título de:
INGENIERÍA EN TELECOMUNICACIONES

Directores:
ING. EDITH PAOLA ESTUPIÑAN CUESTA, M. Sc
ING. JUAN CARLOS MARTINEZ QUINTERO, M. Sc

UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES
BOGOTÁ D.C., 2021

DEDICATORIA

A Dios por ser mi guía en el camino de mi vida, bendiciéndome y dándome fuerzas para continuar con mis metas.

A mis padres por ser el pilar más importante, quienes con su amor y sacrificio en todos estos años me han permitido llegar hasta aquí por haber sido mi apoyo a lo largo de toda mi vida.

A mis amigos por apoyarme cuando más los necesite en este proceso.

AGRADECIMIENTOS

Agradezco a Dios por todas sus bendiciones que llena mi vida, a mis Padres por darme su ejemplo dándome el apoyo suficiente para seguir adelante.

A mis tutores la Ing. Edith Paola Estupiñán y el Ing. Juan Carlos Martínez quiero expresar mis agradecimientos quienes principalmente me colaboraron durante todo este proceso, aportando sus conocimientos y enseñanzas siendo una guía en todo el transcurso de la carrera y en el trabajo de grado.

Finalmente agradezco a la Universidad Militar Nueva Granada por abrirme las puertas y permitirme realizar mi proceso profesional adquiriendo conocimientos en estos años.

CONTENIDO

Lista de Figuras

Lista de tablas

RESUMEN

En el presente proyecto se realiza el estudio e implementación de una tecnología denominada Red Definida por Software con el objetivo de analizar el plano de control y datos mediante la captura de datos para identificar las posibles vulnerabilidades usando herramientas como Mininet que permite emular estas redes de forma sencilla con la integración del controlador OpenDayLight.

Como primer paso del estudio, se realizan el levantamiento de información correspondiente a SDN como las principales características de esta red, con el fin de comprender su funcionamiento como son las capas que conforman la arquitectura, los tipos de topologías que pueden ser implementadas, controlador comercial SDN y las principales características de los sistemas de seguridad comúnmente implementadas en redes SDN.

Por otra parte, se identifican los potenciales ataques que pueden afectar a la red, se selecciona el ataque de denegación de servicios para la realización de las pruebas de rendimiento mediante la ejecución de un ataque DoS para evaluar varios puntos de la red, usando herramientas como Wireshark y línea de comando, se realiza la captura de datos para realizar el análisis. Los resultados obtenidos evidencian el incremento del tiempo de ida y vuelta (RTT) de los mensajes que viajan por la red, además, el controlador presenta inundación de paquetes falsificados que obstaculizan el correcto funcionamiento del software. Este estudio permite comprender el comportamiento de las redes SDN para que investigaciones futuras puedan implementar herramientas para mitigar este tipo de ataques.

1. INTRODUCCIÓN

La tecnología ha mejorado en muchos sentidos especialmente las redes SDN se han hecho muy conocidas debido a las ventajas a nivel administrativo que pueden traer, en comparación con las redes convencionales que llevan más tiempo de uso. La red SDN se caracteriza por funcionar de manera centralizada debido a que su arquitectura lo permite porque sus planos se encuentran separados y se comunican por medio de un protocolo; con el propósito de poder gestionar el administrador el tráfico de datos de la red desde un único punto que es el controlador el cual “permite construir topologías dinámicas y programables para adaptar la red a las necesidades de manera rápida y optimizada”, eliminando la complejidad de administración de la red (UserDataCenter, 2020). En consecuencia, al presentar un diseño donde la separación expone los planos, generan nuevos retos en la actualidad con respeto a la seguridad en redes SDN, ya que, toda la red puede verse afectada, este es el peor de los casos donde se gestiona información valiosa.

La seguridad en SDN es un factor muy importante hoy en día, debido al gran manejo de información que viaja por la infraestructura de la red es de gran importancia contar con buenas prácticas de seguridad, es por esto por lo que si un administrador realiza una mala práctica en el desarrollo y diseño de una red SDN puede provocar una exposición a nivel de seguridad enfocado en los “tres pilares que son confidencialidad, integridad y disponibilidad” (Ramiro, 2017).

El propósito de este trabajo de grado es analizar la red en los planos de control y de datos mediante la implementación de un ataque de denegación de servicio en el escenario propuesto con la finalidad de identificar las posibles vulnerabilidades en la red. Este estudio permitirá comprender con mayor profundidad cómo se comporta la red frente a un ataque identificando los puntos débiles de la red, con el fin de informar de acciones de seguridad para investigaciones futuras que se enfoquen en mitigar el ataque.

Este documento está estructurado en cuatro capítulos fundamentales que cubren el estudio de la red SDN y están conformados de la siguiente forma: 1) Características de la arquitectura y principales métodos de seguridad en SDN, 2) Identificación de los diferentes tipos de ataques potenciales para redes SDN, 3) Establecimiento del escenario y pruebas para aplicar, y 4) análisis de resultados de posibles vulnerabilidades en la red SDN propuesta.

1.1 PLANTEAMIENTO DEL PROBLEMA

Las redes definidas por software “permite gestionar y controlar la red desde un enfoque centralizado para administrar cada uno de los diferentes componentes de hardware como host y switches con ayuda de software”, incrementando los beneficios de virtualización de la red (IONOS, 2019). Las principales ventajas de esta tecnología frente a las redes convencionales son: Reducción en los tiempos de implementación y administración de redes, igualmente para labores de mantenimiento de componentes y permite inspeccionar en tiempo real los recursos que está compuesta la red de forma dinámica mediante el controlador sin la necesidad de configurar cada dispositivo, logrando las empresas una disminución en los costes de los equipos (IONOS, 2019). Actualmente, las redes SDN están tomando un papel muy importante ya que se consideran más eficientes a comparación a las redes convencionales que se conoce hoy en día, SDN se caracteriza por administrar desde un único punto a la red de forma simple, permitiendo adaptar la red a las necesidades de manera optimizada. “Los problemas de seguridad se dirigen hacia los tres planos de la arquitectura: El plano de aplicación puede ser infectado con malware para robar información, el plano de control puede afectar todos los componentes de la red cuando atacan el Controlador SDN, y el plano de datos” se encuentran los dispositivos donde viaja el tráfico llamados conmutadores que tienen como consecuencia ser comprometidos causando que los paquetes que fluyen a través de él no se reenvíen correctamente (Singh & Behal, 2020). Por esta razón es importante detectar las vulnerabilidades antes de que sean aprovechadas por los atacantes para generar un impacto negativo en la red SDN, esto conlleva a realizar la siguiente pregunta problema.

¿Qué vulnerabilidades se pueden encontrar en una red SDN analizando las capas de control y datos, evaluadas mediante la emulación de un escenario usando Mininet con la implementación de un ataque?

1.2 OBJETIVOS

1.2.1 Objetivo General

El objetivo principal para la realización de este proyecto es realizar un análisis de vulnerabilidades a nivel de seguridad en redes SDN para los planos de control y plano de datos.

1.2.2 Objetivos Específicos

El proyecto se desarrollará a partir de los siguientes objetivos específicos que permitirán finalizar con el trabajo:

- Analizar e identificar las principales características de la arquitectura de redes SDN y las características de seguridad comúnmente implementadas en este tipo de redes.
- Identificar los mecanismos potenciales de ataques al sistema de seguridad para redes SDN y seleccionar uno que será implementado.
- Definir e implementar el escenario de redes SDN y el mecanismo de ataque seleccionado.
- Explotar las posibles vulnerabilidades de la red SDN propuesta y realizar la respectiva documentación.

1.3 JUSTIFICACIÓN

Este trabajo de grado se enfoca en el estudio de las redes definidas por software dando a conocer la exploración de la arquitectura y funcionamiento de la red, siendo una parte vital entender con mayor profundidad esta tecnología ya que entrega grandes beneficios a la hora de administrar redes reduciendo los tiempos de implementación por ser dinámica y adaptable.

Sin embargo, surgen diversos desafíos con respecto a la seguridad de la red SDN ya que se convierte en el objetivo de los atacantes aprovechando que la arquitectura separa los planos, por esta razón están en busca de vulnerabilidades para obstruir la red según lo investigado (Sidhu & Gupta, 2019).

Este desarrollo brinda un análisis de las posibles vulnerabilidades que se pueden encontrar a partir de la ejecución de un ataque de denegación de servicio en la topología implementada y emulada en Mininet para evaluar y dar una visión sobre los puntos más vulnerables en la red específicamente en la arquitectura analizando el plano de datos y de control.

1.4 METODOLOGÍA

Para la ejecución del proyecto se establecieron tres fases principales (ver Figura 1) con el fin de optimizar las tareas a desarrollar en el proceso del estudio de la tecnología SDN.

La primera fase consiste en la búsqueda de información de esta tecnología permitiendo identificar los diferentes componentes con los que está conformada la arquitectura con el objetivo de aprender su funcionamiento para proceder en la búsqueda de los mecanismos de ataque que son factibles a sufrir para este tipo de tecnología.

La segunda fase consiste en proponer el escenario de una red definida por software para ser llevada la implementación del ataque evaluando la red propuesta, se llevará a cabo la emulación por medio del software seleccionado a partir de la información adquirida en la primera fase.

La última fase se centra en explotar las posibles vulnerabilidades, teniendo un mayor peso en el estudio, ya que consiste en aplicar los conocimientos obtenidos en las fases anteriores, donde se realizarán diversas emulaciones para poner a prueba la red y llevarla al límite según lo permita el software.

Figura 1. Fases del Proyecto

2. CARACTERÍSTICAS DE LA ARQUITECTURA Y PRINCIPALES MÉTODOS DE SEGURIDAD EN SDN

En esta sección se describe con mayor detalle conceptos como: que es una red SDN, la arquitectura SDN, Protocolo OpenFlow, controladores, emulador y finalmente características de métodos de seguridad.

2.1 REDES DEFINIDAS POR SOFTWARE

Actualmente, la tecnología ha avanzado en muchos rangos y una de ellas es en el entorno de las redes donde encontramos la red SDN que se caracteriza por traer nuevas funciones para la administración debido a que su arquitectura separa los planos permitiendo aportar gran ventaja, lo que permite que la red sea “dinámica, rentable y adaptable a las diferentes necesidades que puede sufrir la red” obteniendo una mejor optimización al administrar este tipo de red (Byun et al., 2019).

La historia de las redes definidas por software está dividida en tres fases donde se desarrolla esta tecnología, todo comienza a surgir a partir de las redes activas en la década de 1990, cuando el internet estaba comenzado a despegar, este tipo de redes “introdujeron nuevas funciones para la red como la programación”, la segunda etapa correspondió a la separación de los planos de control y plano de datos dándose este evento en el año 2001 hasta 2007, surgió por el aumento de tráfico que se estaba presentando en ese tiempo lo cual optaron por esa solución para mejorar la administración de la red, llevándolos a explorar diferentes enfoques, y por último la tercer etapa consistió en la creación de la API OpenFlow desde 2007 hasta 2010, permitiendo a los dos planos la capacidad de ser escalables y práctica a la hora de implementar la red SDN (Feamster Nick et al., 2019).

La arquitectura de las redes SDN está conformada por tres planos como se observa en la Figura 2, aquí vamos a encontrar cómo funciona cada plano entendiendo que responsabilidades tiene en la red SDN.

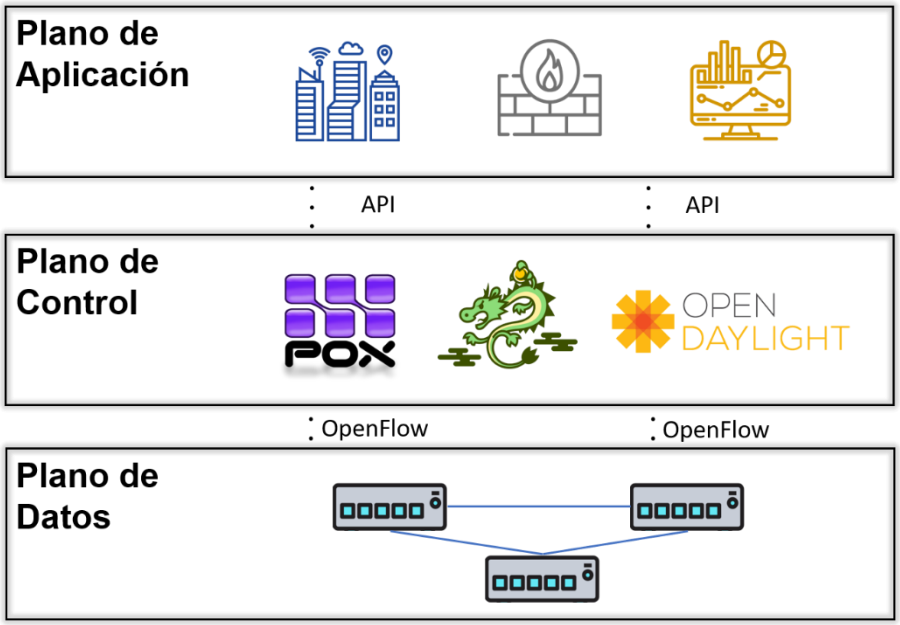


Figura 2. Arquitectura SDN

2.1.1 Plano de aplicación

Este plano se caracteriza por estar localizado en la “capa superior de la arquitectura SDN” (ver Figura 3), el cual tiene como función manejar una gran cantidad de aplicaciones enfocadas hacia el lado de la seguridad y comercial permitiendo interactuar con usuarios, ciudades inteligentes, industria, entre otros, para ofrecer servicios de software como mediciones, gestión de movilidad, implementación de firewall, etc. Esta se comunica con el plano de control con APIs (Singh & Behal, 2020).

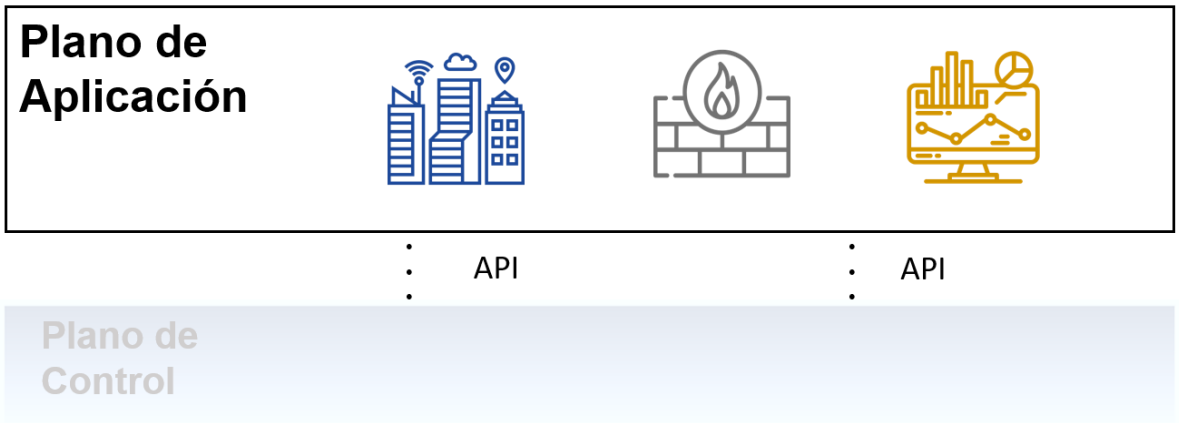


Figura 3. Plano de Aplicación

2.1.2 Plano de control

Este plano tiene un papel muy importante en la red SDN ya que se encarga de múltiples funciones por ser el cerebro de todas las operaciones (ver Figura 4), como “enrutamiento, reenvió y caída de paquetes a través de la programación”, este proceso lo realiza el controlador que se encuentra ubicado en medio del plano de aplicación y datos, envía flujos hacia los conmutadores mediante la API OpenFlow permitiéndole gestionar estos dispositivos de la red de forma remota (Singh & Behal, 2020).

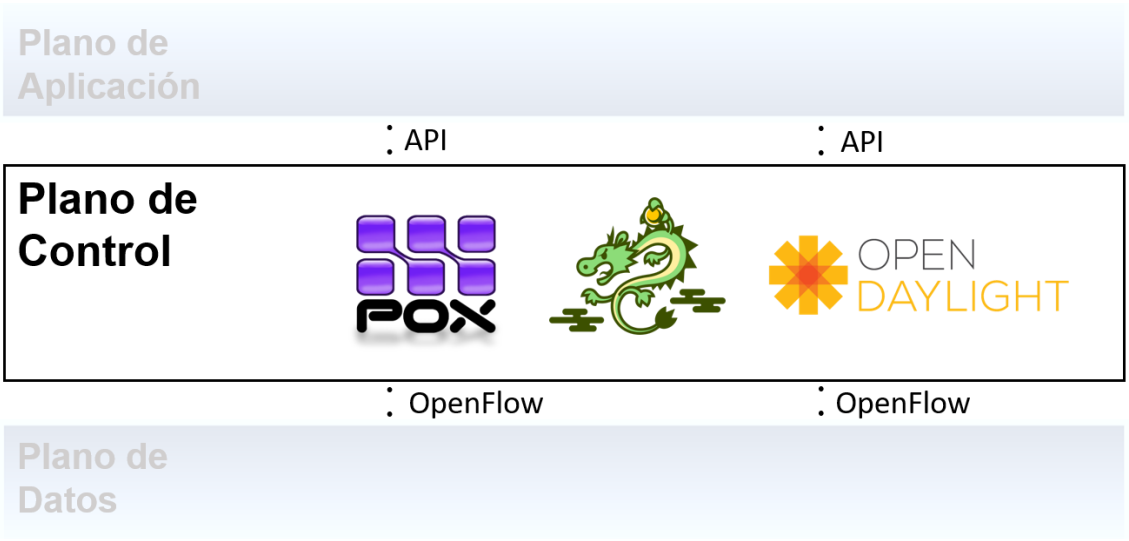


Figura 4. Plano de Control

2.1.3 Plano de datos

El plano de datos o infraestructura tiene como “función de reenviar el tráfico hacia los distintos dispositivos de la red” como conmutadores o enrutadores de acuerdo con las políticas entregadas por el controlador, cada vez que llega un nuevo paquete en los conmutadores se solicita una nueva regla de flujo para poder enviarse el paquete hacia el destino (Singh & Behal, 2020) (ver Figura 5).

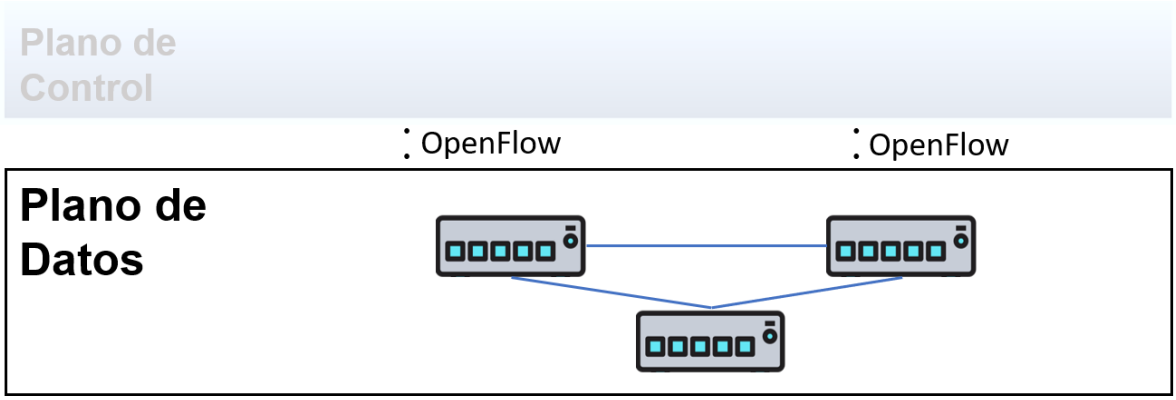


Figura 5. Plano de Datos

2.2 OPENFLOW

El protocolo OpenFlow es de código abierto que permite “gestionar de forma remota las tablas de reenvío en los conmutadores”, creando redes escalables con una mayor eficiencia. Con OpenFlow se puede obtener varios beneficios como (Brandon Heller, 2015):

- Ver toda la información de las tablas de flujo para el análisis.
- Simular una red de múltiples conmutadores y hosts con software de emulación enfocadas a redes SDN como la herramienta Mininet.
- Capturar los paquetes diseccionando el mensaje OpenFlow con el software Wireshark.

“OpenFlow es uno de los primeros estándares de comunicación en redes SDN para comunicar los planos de control y de datos de la arquitectura”, mediante la implementación de este protocolo se puede identificar el tráfico de red por medio de flujos, permitiendo que la red sea programable y adaptable según los requisitos que requiera, ofreciendo un mayor control y respuesta en tiempo real a partir de políticas predefinidas en la capa de control exactamente en el controlador. Actualmente, se caracteriza por ser un protocolo que tiene la función de manipular principalmente los datos que llegan a los conmutadores en el plano de datos y es uno de los pocos protocolos enfocados para SDN (Spera & Cone, 2013).

2.2.1 Tabla de flujo

Los conmutadores OpenFlow cuenta con una memoria TCAM encargada de realizar la gestión de la tabla de flujo, que dispone de los siguientes componentes (ONF, 2013):

- **Match Fields:** Corresponde a los encabezados de los paquetes y también al puerto de entrada, incluso metadatos específicos por la tabla anterior.
- **Priority:** Es la prioridad que se le asigna a la entrada del flujo.
- **Counters:** Es un valor que se actualiza cuando los paquetes son parecidos.
- **Instructions:** Esta instrucción consiste en cambiar el procesamiento de la canalización del paquete.
- **Timeouts:** Es la cantidad de tiempo máxima de un flujo antes que se expire la solicitud.
- **Cookie:** En este campo se encuentra un valor que es utilizado para diferentes funciones como el de filtrar flujos y procesar solicitudes que sean generadas.
- **Flags:** Las banderas tienen como función gestionar las entradas de flujo modificándose.

2.3 TOPOLOGÍAS EN SDN

En las redes definidas por software se pueden encontrar diferentes tipos de topologías que definen el diseño de la red según las conexiones de los nodos como se presentarán a continuación en cada una de las topologías.

2.3.1 Topología simple

En la figura 6, se observa la topología simple que se caracteriza por componer un switch conectado a una cantidad de hosts. Con el siguiente comando es posible crear la topología simple en Mininet:

mn --topo single, N. Donde N corresponde al número de hosts a crear en la topología.

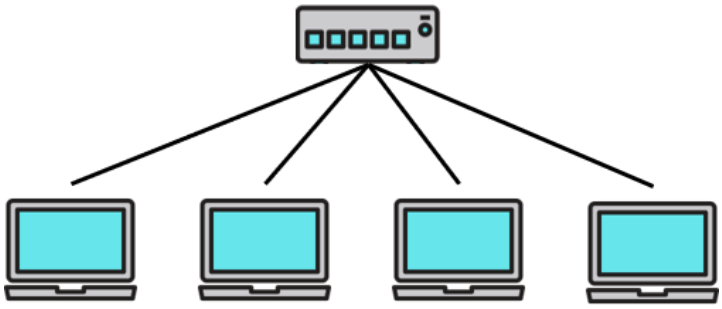


Figura 6. Topología Simple

2.3.2 Topología lineal

En la figura 7, se observa la topología lineal que se caracteriza porque cada switch se conecta a un host asignado. Para crear esta topología es necesario utilizar el siguiente comando en Mininet:

mn --topo linear, N.

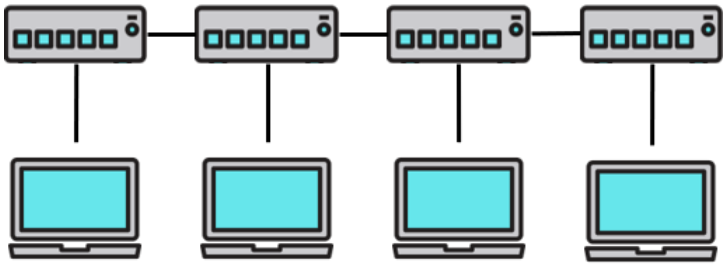


Figura 7. Topología Lineal

2.3.3 Topología de árbol

En la figura 8, se observa la topología que se caracteriza por su estilo de árbol debido a que tiene una profundidad de dispositivos igualmente una anchura que se extiende de forma horizontal, sus conexiones se establecen entre los nodos formando este peculiar diseño. A continuación, se tiene el siguiente comando para crear la topología de árbol en Mininet:

mn --topo tree, depth=M,fanout=N.

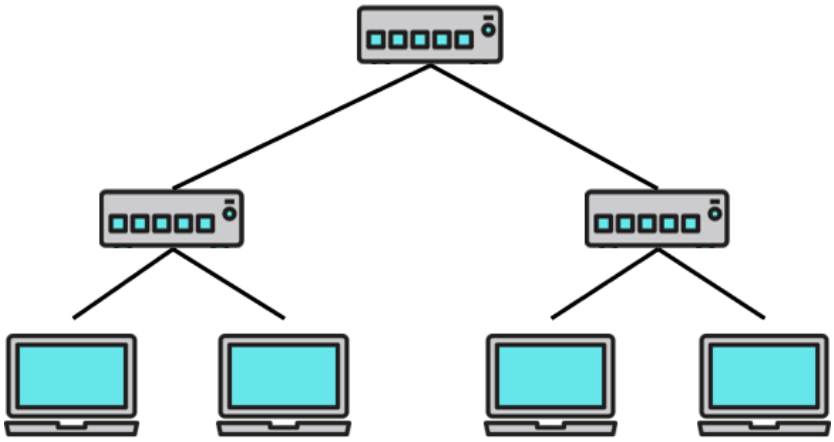


Figura 8. Topología de Árbol

2.3.4 Topología de malla

En la figura 9, se observa la topología de malla que se caracteriza por su estructura de malla X*Y formando una cuadrícula. Es una de las topologías más utilizadas en las redes convencionales ya que la información puede viajar por cualquier camino. Con el siguiente comando se crea la topología de malla:

mn --topo torus,x=M,y=N. Donde M y N tienen como requisito ser valores mayores o iguales a 3 para crear la topología de malla.

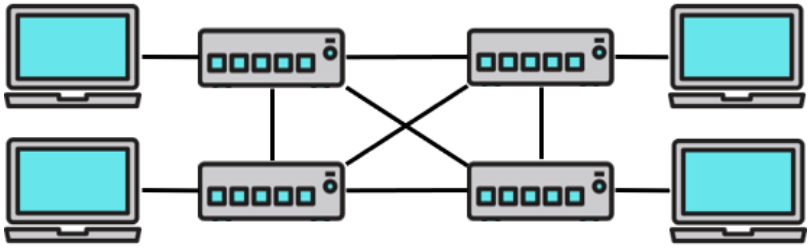


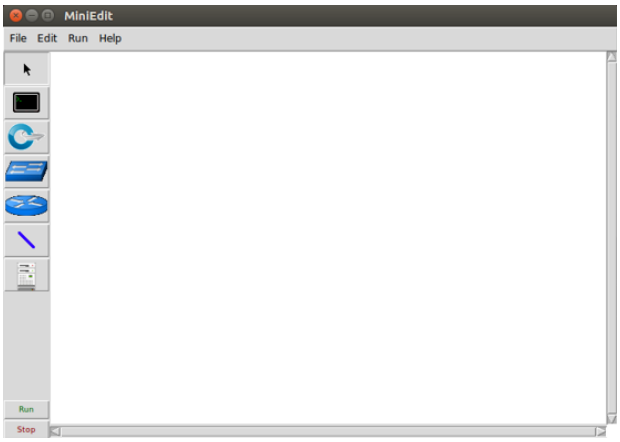
Figura 9. Topología de Malla

2.4 SOFTWARES PARA SDN

A continuación, se presentan los softwares necesarios para implementar una red SDN para realizar pruebas como es el controlador, el emulador y la herramienta para ejecutar el ataque como se observa en la Tabla 1.

Tabla 1. Softwares para implementar SDN

SOFTWARE	DESCRIPCIÓN
Controlador OpenDayLight	OpenDayLight (ODL) es un software enfocado para automatizar redes SDN y personalizar redes de cualquier tamaño de dispositivos. OpenDayLight es un proyecto que surgió a partir del nacimiento de SDN, diseñado para dar solución a la adaptabilidad de esta red con un enfoque claro en la programabilidad. OpenDayLight permite a los desarrolladores y usuarios los siguientes beneficios (OpenDayLight, 2020): ▪ Instale solo los protocolos y servicios que necesitan. ▪ Desarrolle rápidamente funciones personalizadas para casos de uso altamente especializados ▪ Combine múltiples protocolos y servicios para resolver problemas más complejos a medida que surjan las necesidades. ▪ Desarrollar de manera incremental y colaborativa las capacidades de la plataforma.
Mininet	Mininet es un software desarrollado para realizar pruebas especialmente con redes SDN permitiendo dispositivos como conmutadores, host y hasta la integración de controladores de terceros para poder experimentar con estas topologías personalizadas mediante software de manera fácil. El código de programación que utiliza Mininet es Python, el cual permite crear y configurar redes de forma sencilla, además cuenta con una gran base de datos para consultar (Mininet, 2021).
	El emulador Mininet cuenta con una herramienta gráfica para crear topologías llamada MiniEdit (ver Figura 10), es

Miniedit	un editor GUI simple experimental para ejecutar emulaciones de red de forma sencilla para el usuario con pocos conocimientos en programación Python.  *Figura 10. Interfaz gráfica Miniedit*
Hping	El software Hping es una herramienta que funciona en Linux, MacOs y Windows para realizar pruebas de seguridad ya que permite paquetes TCP/IP para probar redes y hosts. Hping es un programa de línea de comando compatible con protocolos como TCP, ICMP y UDP, las principales funciones son (Hping, 2021): ▪ Pruebas dirigidas a cortafuegos para evaluar su funcionamiento. ▪ Auditoría en TCP / IP. ▪ Pruebas de red, aplicando los distintos protocolos compatibles. ▪ Útil para el aprendizaje de los estudiantes en TCP/IP. ▪ Permite hacer escaneo de puertos a servicios.

2.5 MÉTODOS DE SEGURIDAD

Los métodos de seguridad tienen como función proteger la infraestructura de posibles amenazas, en esta sección se definen las principales características de métodos de seguridad que pueden aplicarse en una red SDN para asegurar el correcto funcionamiento.

2.5.1 Sistema de detección y prevención de intrusos

En seguridad estos son métodos muy eficientes donde encontramos “los sistemas de detección de intrusos que tienen como función de monitorear el tráfico entrante de la red para evaluar cada dato y así poder detectar alguna actividad sospechosa que ponga en peligro el sistema”, puede ser alguna anomalía entrante. Por otro lado, están los sistemas de prevención de intrusos, estos se encargan de bloquear esas amenazas para que no causen daños en la infraestructura por medio de asignación de reglas para que filtre el tráfico malicioso.

Estas dos herramientas son utilizadas en la ciberseguridad debido a las ventajas que ofrece como (HelpSystems, 2020):

- Proteger la información evitando que los intrusos se comprometan con los datos.
- Proporciona visibilidad de la seguridad de los equipos identificando posibles vulnerabilidades.
- Monitorea la red observando el tráfico y analizando patrones.
- Identifica las amenazas y responde de forma rápida.

2.5.2 Escaneo de vulnerabilidades

Otra forma para proteger los sistemas es el escaneo de vulnerabilidades el cual “permite identificar posibles amenazas en los elementos de la red” de una organización para luego reportar esas fallas y así evitar que una amenaza se convierta en un grave riesgo para todo el sistema. Los escaneos son realizados en los puertos, aplicaciones y servicios que pueden ser de tipo (Datasec-soft, 2020):

- **Interno:** se examina los elementos que se encuentran dentro de una organización como los equipos del personal que tienen acceso al sistema.
- **Externo:** se examina la seguridad de forma remota con el punto de vista de una persona que no tiene vínculos con la organización.
- **Mixto:** es la combinación entre las perspectivas de escaneo externas e internas para asegurar la seguridad en ambos sentidos.

2.5.3 Firewall

En redes SDN, los firewalls son otra solución de seguridad para aplicar ya que estos sistemas de seguridad implementan reglas ya establecidas para permitir o denegar tráfico específico agregando un muro de seguridad entre las redes internas y externas. En la actualidad se encuentran firewall de tipo software, hardware o ambos, algunos de ellos son (Cisco, 2021):

- Los firewall proxy están enfocados para una aplicación específica.
- Los firewall de próxima generación tienen un enfoque en eliminar amenazas modernas.
- Los firewall virtuales son utilizados en servicios virtuales como Cloud.
- Los firewall de inspección de estado se encarga de permitir o bloquear según unos parámetros de estado.

2.6 VULNERABILIDADES EN SDN

Los planos de la red SDN son vulnerables debido a que desacopla el plano de datos y plano de control para brindar flexibilidad, pero como resultado, a continuación se detallan algunos de los problemas de seguridad que surgen en la red:

2.6.1 Desbordamiento de la tabla de flujo

Los conmutadores OpenFlow tienen la función de reenviar los paquetes hacia su destino mediante las reglas de flujo que se almacena en la memoria TCAM, estas

reglas se generan por cada paquete es por eso que es limitada su capacidad por su costo. Esta es una vulnerabilidad que puede aprovechar el atacante generando paquetes falsos de forma masiva para llenar la tabla de flujo del conmutador, el controlador envía el flujo y se almacena, como consecuencia la memoria se llena y los usuarios legítimos que quieren establecer conexión con otro usuario no pueden porque no hay espacio (Singh & Behal, 2020).

2.6.2 Congestión del enlace de control y datos

Los conmutadores cuando reciben un nuevo paquete de un usuario que no está registrado en la tabla de flujo, va a solicitar al controlador una nueva regla por medio de un mensaje “Packet_in” que es enviado por la interfaz del sur de la arquitectura. El atacante aprovecha esta característica de los conmutadores para enviar paquetes con suplantación de IP, el dispositivo estará obligado a generar mensajes “Packet_in” por cada IP nueva que reciba. El atacante realiza esta acción para dejar a los usuarios autorizados sin servicio debido a la inundación de los mensajes se sobrecarga el controlador (Singh & Behal, 2020).

2.6.3 Saturación del controlador

La saturación del controlador es ocasionada cuando el atacante “envía una gran cantidad de paquetes falsos hacia el conmutador”, el cual este dispositivo enviará solicitudes al controlador para tenerlo abrumado resolviendo las peticiones en la creación de nuevas reglas de flujo, con el propósito de agotar el rendimiento y la potencia de procesamiento, como consecuencia toda la arquitectura SDN será afectada (Singh & Behal, 2020).

3. MECANISMOS POTENCIALES DE ATAQUE EN REDES SDN

En esta sección describe posibles mecanismos de ataque en redes SDN. Los ataques de denegación de servicio principalmente tienen un mayor impacto en la red SDN, porque se deben crear reglas de flujo para reenviar los paquetes a su destino.

3.1 ATAQUE DE DENEGACIÓN DE SERVICIO

Los ataques DoS están clasificados como los más reconocidos a nivel de seguridad ya que tienen una trayectoria de ser implementados para inhabilitar servicios mediante la generación de paquetes para sobrecargar los componentes de la infraestructura con el objetivo de que los sistemas operativos, servidores y equipos de red no sean capaces de procesar las solicitudes generadas de los usuarios legítimos, el atacante utiliza un solo equipo para generar los paquetes pero también existe un ataque que emplea varios equipos llamado denegación de servicio distribuido (DDoS) el cual se basa en extensas redes de bots o botnets, consiste en dirigir una gran cantidad de peticiones simultáneas de varios ordenadores hacia la víctima para que no pueda atender las solicitudes. Estos tipos de ataque de denegación se pueden dividir en tres categorías que se muestran a continuación (Ionos, 2019):

- Enfocados en la saturación del sistema mediante el consumo de los recursos.
- Enfocados en la sobrecarga del ancho de banda.
- Enfocados en aprovechar los errores de seguridad en los sistemas.

3.1.1 Sobrecarga del ancho de banda

Los ataques enfocados hacia el ancho de banda buscan deshabilitar la red bloqueando el funcionamiento de los dispositivos conectados con el fin de saturar toda la capacidad de los dispositivos que procesan datos para que otros usuarios ya no puedan acceder a servicios (Ionos, 2019).

Ataque Smurf: Una clasificación de este ataque es el smurf como se observa en la Figura 11, consiste en enviar una serie de paquetes ICMP falsificados

implementados en una red, estos paquetes son dirigidos hacia la dirección broadcast utilizada como medio de difusión con el uso de la dirección IP de la víctima como remitente con el fin de aprovechar el envío de ICMP para el intercambio de información y además de mensajes de error, cuando realiza la petición broadcast solicitara una respuesta a cada dispositivo conectado en la red a la dirección del remitente, afectando masivamente la disponibilidad del ancho de banda (Ionos, 2019).

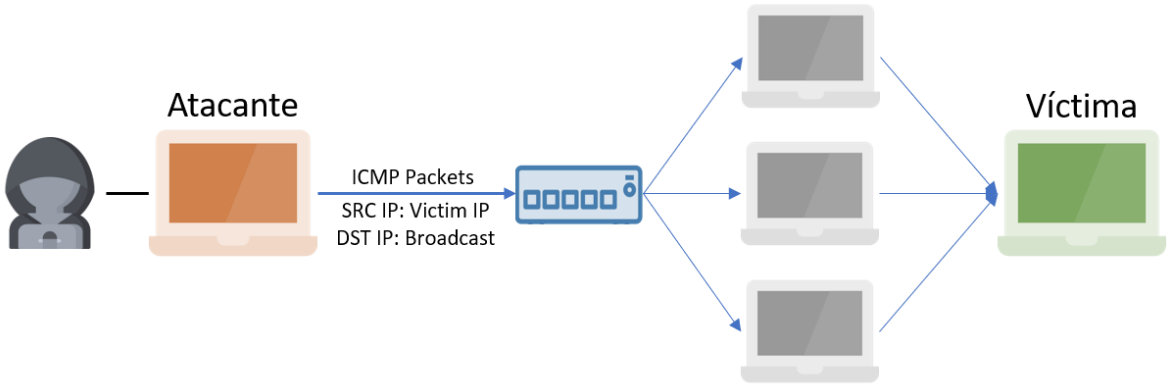


Figura 11. Esquema de Ataque Smurf

3.1.2 Saturación de los recursos del sistema

Los ataques de este tipo están dirigidos hacia los recursos del sistema siendo un punto de gran ventaja para los cibercriminales ya que los servidores suelen tener un número de conexiones establecidas según los recursos, que es aprovechado por los atacantes para saturar con peticiones falsas mediante una inundación para bloquear a los usuarios. Podemos encontrar varios tipos de inundaciones como (Ionos, 2019):

Inundación Ping: Una inundación Ping (ver Figura 12) es capaz de ralentizar los sistemas debido a que estos dispositivos tienen la obligación de responder a las solicitudes que reciba, es por eso, el atacante envía paquetes ICMP Request por medio de redes zombies llamadas “bots” para hacer a gran escala el ataque y saturar el sistema limitando procesamiento de solicitudes de usuarios legítimos de la red (Ionos, 2019).

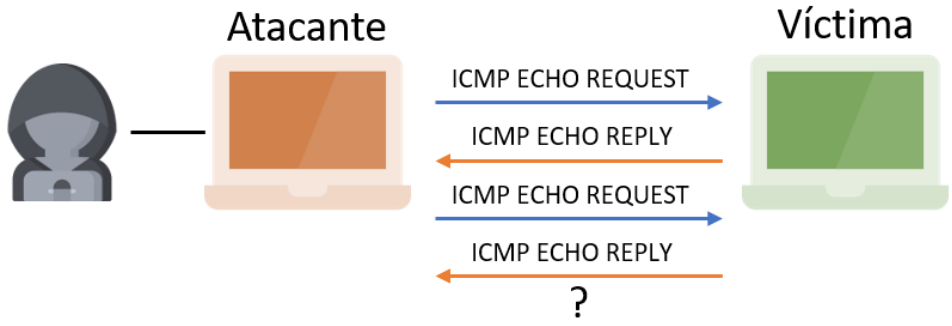


Figura 12. Esquema de Inundación Ping

Inundación UDP: Este tipo de ataque (ver Figura 13), consiste en generar una gran cantidad de paquetes de tipo UDP donde los puertos son asignados de modo random, con el fin de que el sistema atacado deba descubrir cuales son los usuarios o servicios que están en espera de los datos generados por el atacante para enviárselos, al transcurrir el tiempo el servidor al no identificar el destinatario como siguiente paso de respuesta el sistema envía un paquete ICMP el cual tiene el posterior mensaje “Dirección de destino no disponible” para lograr sobrecargar el sistema con varias peticiones generadas (Ionos, 2019).

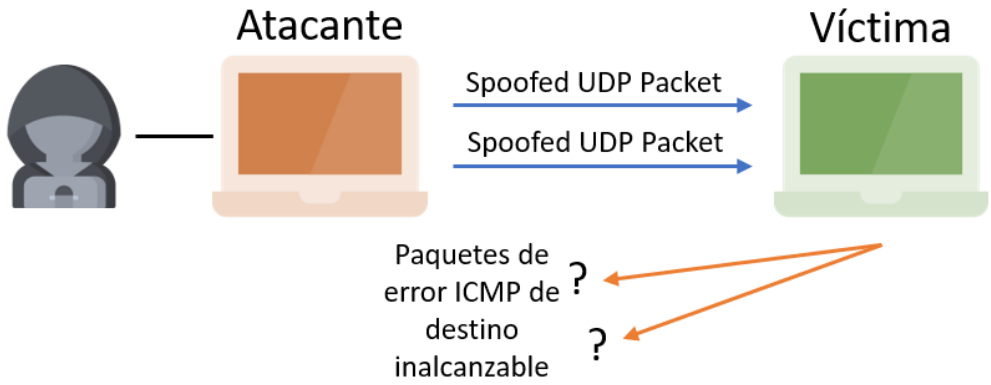


Figura 13. Esquema de Inundación UDP

3.1.3 Explotación de fallos en el software

Este tipo de ataques son una amenaza para programas o sistemas operativos en el cual el atacante examina las vulnerabilidades del sistema y a partir de los parámetros diseña un ataque de solicitudes dirigido a los servidores para generar errores y disminuir el desempeño. En esta categoría se pueden encontrar dos ataques, a continuación, se explica cada uno (Ionos, 2019):

Ping de la muerte: Este es uno de los ataques que busca aprovechar los errores del protocolo por el atacante mediante la producción de datos originando la caída del sistema a partir de esta vulnerabilidad como se observa en la Figura 14, los paquetes IP son enviados en fragmentos y no el paquete completo. El atacante busca engañar al sistema operativo para que genere paquetes IP excediendo el tamaño máximo del paquete de 65.535 bytes, en el momento a volver a montar el paquete se produzca un buffer overflow (Ionos, 2019).

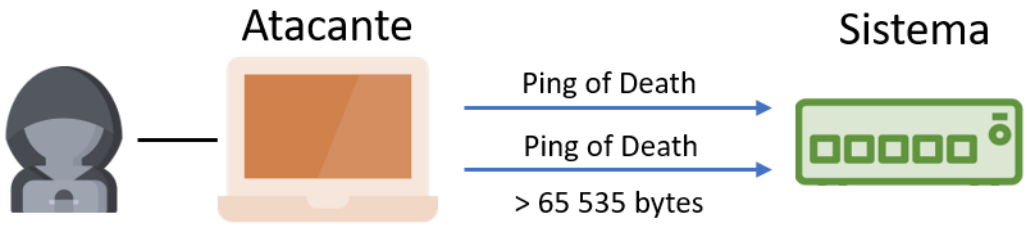


Figura 14. Esquema de Ping de la Muerte

Ataque LAND: El atacante envía paquetes tipo SYN como se observa en la Figura 15, con la dirección del remitente y destinatario correspondiente al servidor con el objetivo de que el mismo se envíe mensajes de paquetes de tipo SYN/ACK, creando como resultado un ciclo de peticiones y respuestas por parte del servidor, como consecuencia el sistema del servidor está sobrecargado por estos paquetes llegando al colapso (Ionos, 2019).

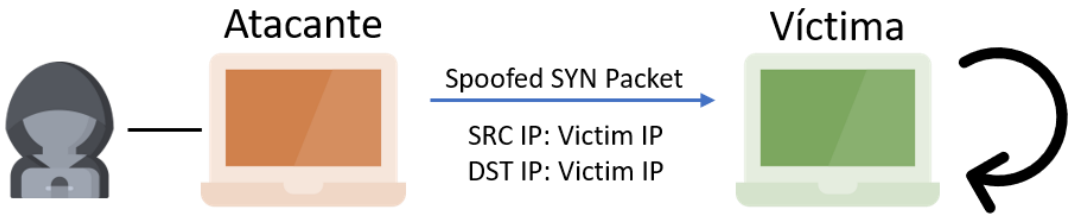


Figura 15. Esquema de Ataque LAND

3.2 ATAQUE MAN IN THE MIDDLE

Un ataque Man In The Middle (ver Figura 16) consiste en explotar los puntos más vulnerables para vigilar los mensajes que se envían entre dos dispositivos como un usuario y un servidor. El atacante tiene como objetivo interceptar la transmisión entre dos partes para capturar toda la información posible de forma ilegal en modo

oculto sin que ninguno perciba su presencia volviéndose uno de los más difíciles de detectar para usuarios (Infocyte, 2021).

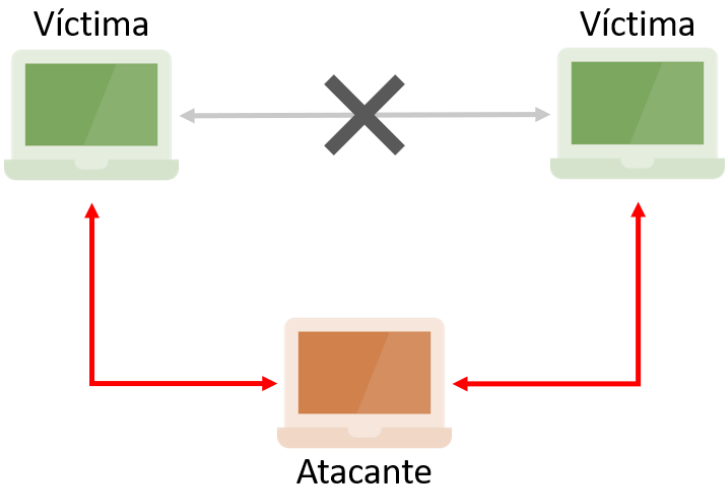


Figura 16. Esquema de Ataque MitM

3.3 ATAQUE SUPLANTACIÓN DE IDENTIDAD

Uno de los ataques más comunes es el Phishing, este consiste en enviar un número masivo de correos electrónicos a diferentes víctimas haciéndolos pensar que son de una fuente confiable. Tienen como objetivo robar información e introducir malware mediante la vinculación de un archivo o script malicioso que permite el acceso al dispositivo para recopilar información delicada para la víctima como información personal y financiera, que será utilizada para diferentes propósitos según cómo la utilice el atacante. Hay varios tipos de ataques de phishing como (Infocyte, 2021):

- **Spear Phishing:** Este ataque se enfoca en robar información importante de empresas o individuos específicos.
- **Pharming:** Radica en capturar credenciales de usuarios por medio del envenenamiento del DNS a través de un inicio de sesión falsa.

4. IMPLEMENTACIÓN DE ESCENARIO Y MECANISMO DE ATAQUE

A continuación, en esta sección se detalla con más precisión el escenario a implementar, la configuración y las pruebas a realizar el cual evaluaran las posibles vulnerabilidades de la red SDN.

4.1 ESPECIFICACIONES TÉCNICAS

En la implementación de la topología se realiza la emulación en la herramienta Mininet en un computador donde se encontrará el controlador y los usuarios en un mismo equipo. Las siguientes características técnicas corresponde al computador como se observa en la Tabla 2.

Tabla 2. Especificaciones técnicas del PC

Requerimientos	Datos
Sistema Operativo	Ubuntu 16.04 64 bits
Memoria RAM	6 GB
CPU	2 núcleos - 2.0GHz
Almacenamiento	400 GB

En la Tabla 3, se describen las versiones de los softwares utilizados para la implementación de la red SDN.

Tabla 3. Requisitos de Software

Requisitos	Versión
OpenDayLight	0.3.0 Lithium
Mininet	2.3.0
Hping	3.0.0 alpha-2

4.2 TOPOLOGÍA

Para el desarrollo del proyecto se planteó la siguiente topología según los requisitos anunciados previamente como se muestra en la Figura 17, el cual se seleccionó una topología de tipo malla ya que ofrece una mejor ventaja en cuanto a la redundancia de red ante un fallo, suceso que no ocurre con las demás topologías cuando sufre la desconexión de un enlace en la red, además, es una de las topologías más implementadas en las redes.

La topología propuesta consta de un controlador OpenDayLight, tres conmutadores Open vswitch y seis hosts, implementados en el emulador Mininet instalado en el equipo.

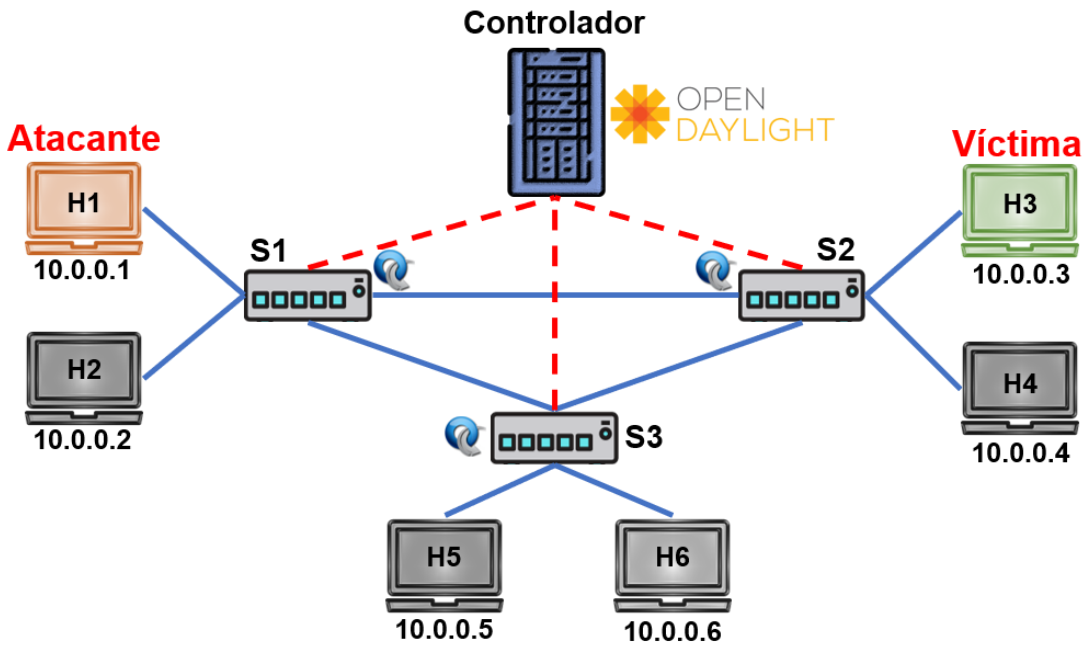


Figura 17. Topología Propuesta

La topología de la red SDN es elaborada en el software Mininet ya que esta herramienta ofrece grandes ventajas para la creación de la topología. Como primer paso se diseñó la topología en la interfaz gráfica llamada Miniedit por la sencilla razón que permite la construcción de la red según la necesidad como se observa en la Figura 18, además permite extraer el código Python para ejecutar la topología de forma rápida y sencilla mediante línea de comando y no tener la necesidad de ejecutar la herramienta Miniedit para realizar las pruebas, el código se puede visualizar en el Anexo C.

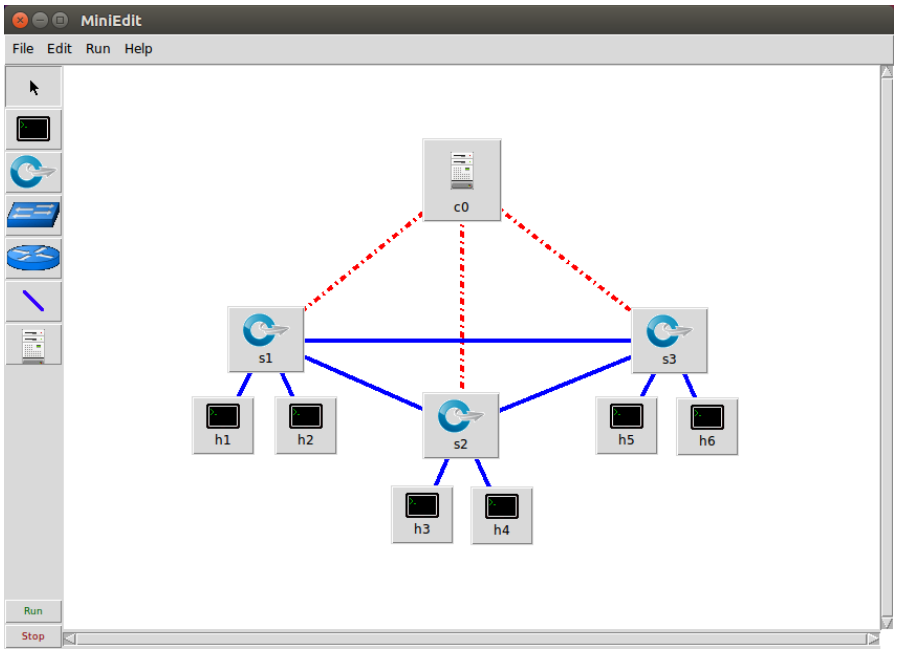


Figura 18. Topología en Miniedit

4.2.1 Prueba de conectividad

Para la validación del funcionamiento de la topología SDN implementada se realiza una prueba de conectividad para verificar la conexión y disponibilidad de todos los dispositivos en la red, para esto se procede a enviar un “pingall” desde la línea de comando de Mininet, en esta prueba de conexión se mandan solicitudes a cada dispositivo de la topología, como resultado se observa en la Figura 19, el cual se evidencia que existe conectividad en los equipos y no se presentaron pérdidas de paquetes en la prueba.

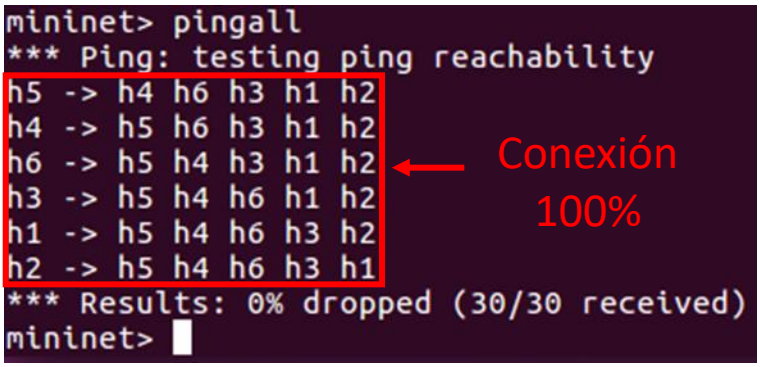


Figura 19. Ping de Conectividad

Igualmente, desde el controlador OpenDayLight se visualiza el estado de conexión de los componentes de la red desde el navegador de preferencia ya que este

software cuenta con una interfaz gráfica web con la capacidad de administrar la red observando cada elemento como se ve en la Figura 20, se evidencia que está conformado por tres dispositivos switch y seis hosts que fueron creados desde la herramienta Miniedit.

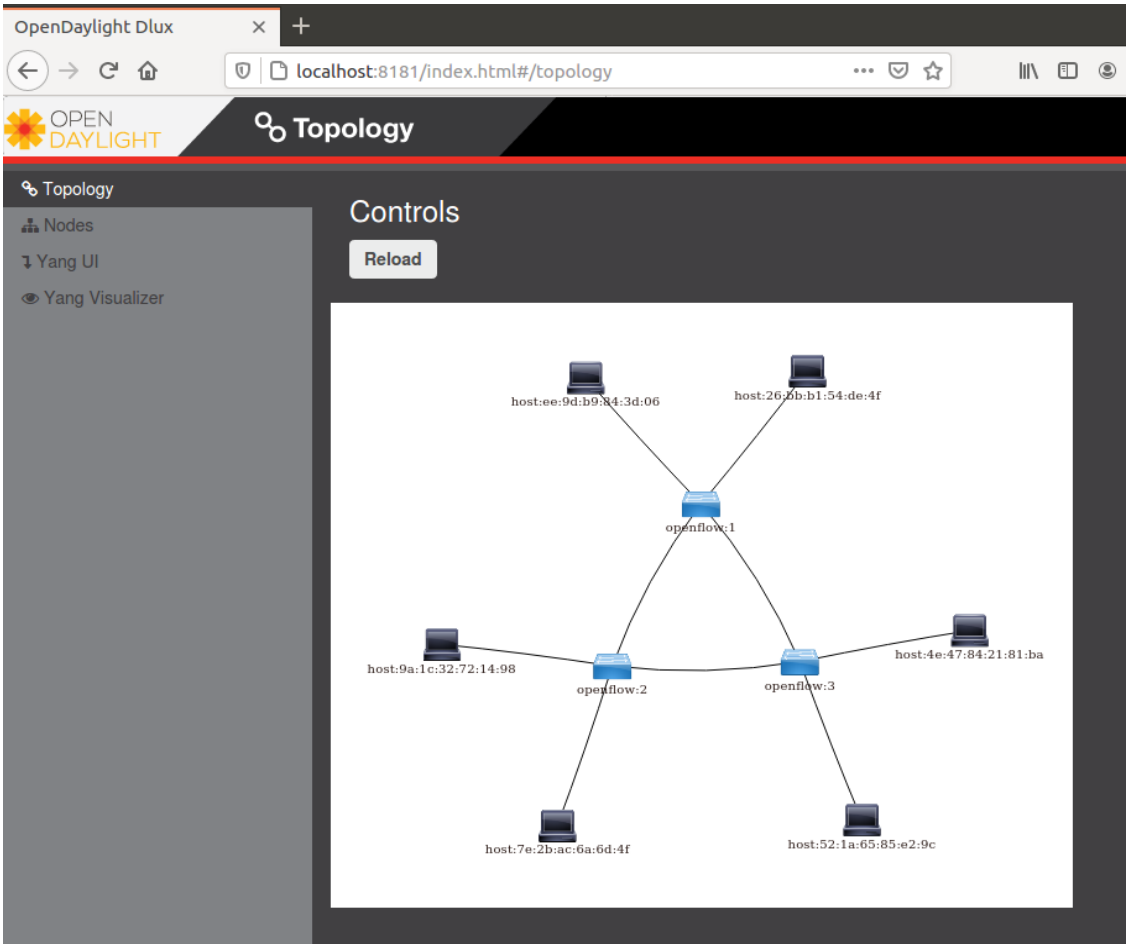


Figura 20. Validación de topología en el Controlador

4.3 ESCENARIOS

A continuación, se describe la metodología de la ejecución del ataque en la topología implementada el cual se establece el escenario para poner a prueba la red emulada donde se ejecutará el ataque en específico:

4.3.1 Escenario: Inundación ICMP

Este escenario consiste en la generación de paquetes mediante la variación del tamaño desde el usuario Host 1 con la IP 10.0.0.1 a la víctima Host 3 con la IP 10.0.0.3 por un periodo de tiempo de 40 segundos lo cual se envía paquetes de tipo ICMP Request mediante el software Hping, este ataque tiene como objetivo consumir el ancho de banda sobrecargando tanto la red como a la víctima y se captura los paquetes en el controlador que tiene como IP asignada la 127.0.0.1 para analizar los paquetes como se observa el escenario de ataque en la Figura 21. Para el análisis de la red se tendrá en cuenta la métrica de RTT y los Packets/sec, se realizan dos pruebas con tamaños diferentes (ver Tabla 4) para ver el comportamiento de la red.

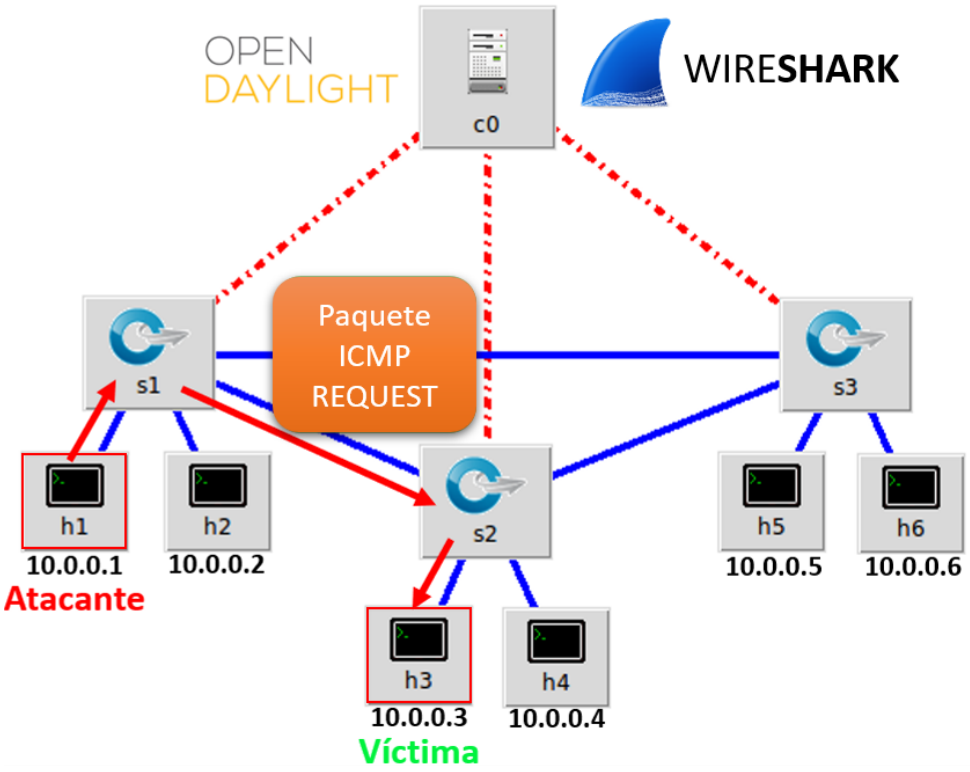


Figura 21. Escenario de Ataque ICMP

Tabla 4. Pruebas para el Ataque ICMP

Ataque	Tamaño de Paquetes	Métricas
Prueba #1	42 bytes	▪ RTT ▪ Packets/sec
Prueba #2	542 bytes	

5. EXPLOTACIÓN DE LAS POSIBLES VULNERABILIDADES DE LA RED SDN EMULADA

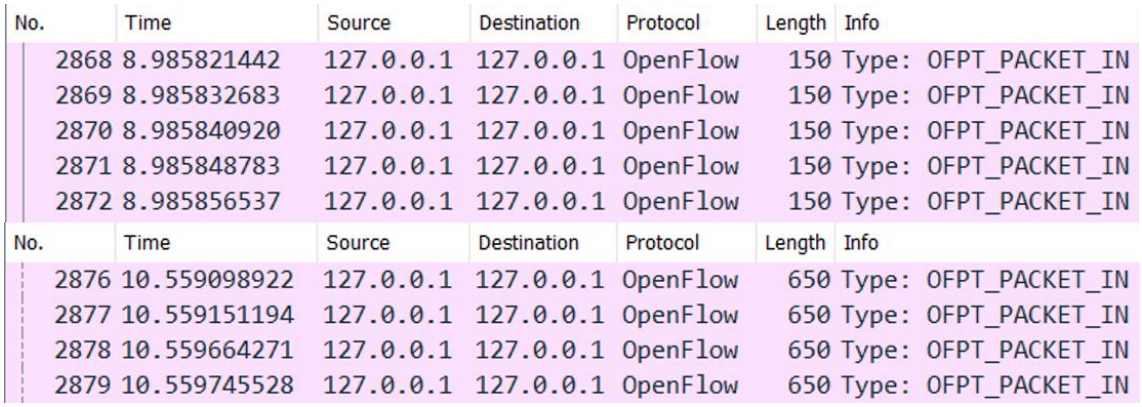
Esta sección analiza los resultados de las pruebas de rendimiento ejecutadas con la implementación del ataque para probar la red SDN identificando cómo es afectada.

5.1 RESULTADOS

A continuación, se muestran los resultados y análisis del escenario establecido en la sección 4.3.

5.1.1 Inundación ICMP

En el segundo escenario se realizó la inundación con el protocolo ICMP, igualmente se envió paquetes con la herramienta Hping hacia la víctima variando el tamaño del paquete entre 42 y 500 bytes. En la figura 22, se observa los mensajes “Packet_In” en el controlador con el color rosado en el que identifica Wireshark que los mensajes son de tipo ICMP.



No.	Time	Source	Destination	Protocol	Length	Info
2868	8.985821442	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_IN
2869	8.985832683	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_IN
2870	8.985840920	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_IN
2871	8.985848783	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_IN
2872	8.985856537	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_IN

No.	Time	Source	Destination	Protocol	Length	Info
2876	10.559098922	127.0.0.1	127.0.0.1	OpenFlow	650	Type: OFPT_PACKET_IN
2877	10.559151194	127.0.0.1	127.0.0.1	OpenFlow	650	Type: OFPT_PACKET_IN
2878	10.559664271	127.0.0.1	127.0.0.1	OpenFlow	650	Type: OFPT_PACKET_IN
2879	10.559745528	127.0.0.1	127.0.0.1	OpenFlow	650	Type: OFPT_PACKET_IN

Figura 22. Paquetes en el controlador ejecutando ICMP

Se procede con el análisis del tráfico en el controlador como se observa en la Figura 23, se identifica el mismo patrón de los paquetes en los dos casos, teniendo como máximo de 30.000 packets/sec para los paquetes ICMP con un tamaño de 42 bytes mientras para los paquetes que se agregaron 500 bytes el resultado fue de 4.100 packets/sec.

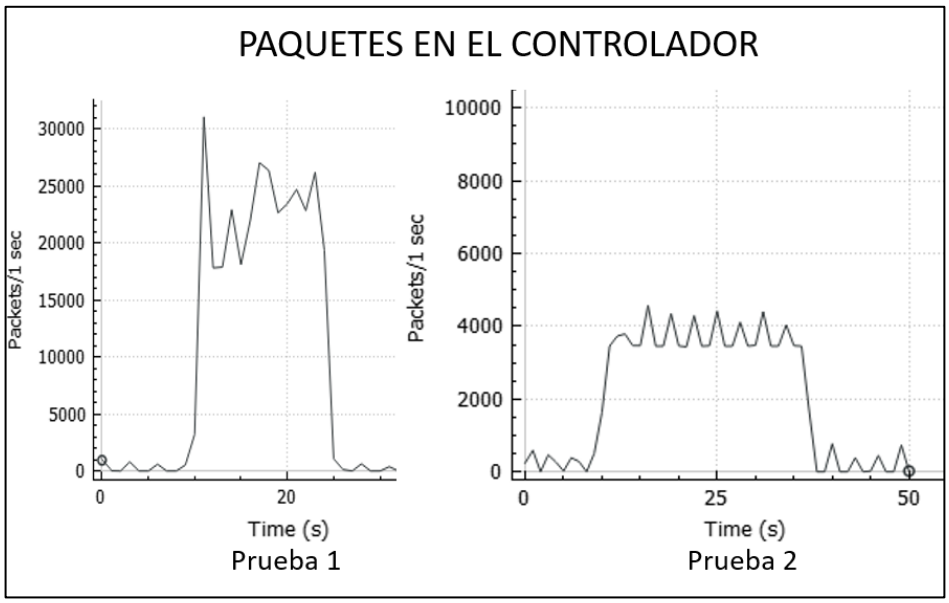


Figura 23. Paquetes en el controlador ejecutando ICMP

A continuación, se procede con analizar el tiempo de ida y vuelta de los paquetes como se observa en la Figura 24, para el escenario de la inundación ICMP, los datos son capturados desde el equipo Host 5 con la IP 10.0.0.5 en el que se establece conexión con los demás equipos de la red para evaluar qué puntos son más afectados por el ataque. Se identifica un comportamiento similar al escenario anterior, donde se obtiene un mayor impacto en los tiempos cuando se agrega datos al paquete, además, se evidencia que los equipos con una mayor afectación en cuanto a conexión son Host 2 y Host 4 presentando los mayores tiempos con los demás dispositivos.

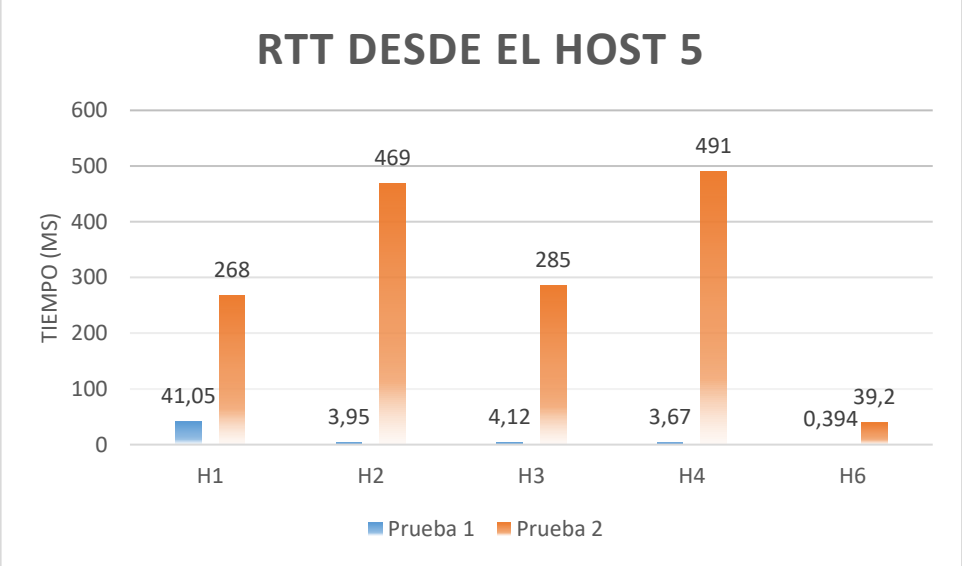


Figura 24. Tiempo de ida y vuelta ICMP

5.2 PUNTOS CRÍTICOS DE LA RED

Una de las principales vulnerabilidades identificadas en cada prueba, se encontró en el plano de control (ver Figura 25). Este punto el controlador se considera la mayor amenaza ya que cumple con una serie de funciones como es la de gestionar todos los equipos que se encuentran en la red mediante la generación de reglas de flujo de cada paquete originario de un usuario para que sean reenviados a su destino. A partir del resultado se identificó que estos paquetes que llegan al conmutador con un mayor tamaño de bytes enviados por el atacante provocan la dificultad en el funcionamiento del controlador para poder procesar cada dato recibido.

Adicionalmente se analiza el plano de datos para identificar otra vulnerabilidad obteniendo como resultado que este se ve afectado en el ataque ya que al generar la inundación de paquetes estos circulan por la red con el objetivo de provocar un desgaste o consumo en el ancho de banda afectando la conexión principalmente de usuarios que pretender establecer enlace. Otro punto de vista es el enlace entre el conmutador y el controlador que mantienen comunicación mediante el protocolo OpenFlow, por este medio pasan todos los paquetes de la inundación enviados por el atacante hacía el controlador viéndose involucrado el enlace frente al ataque.

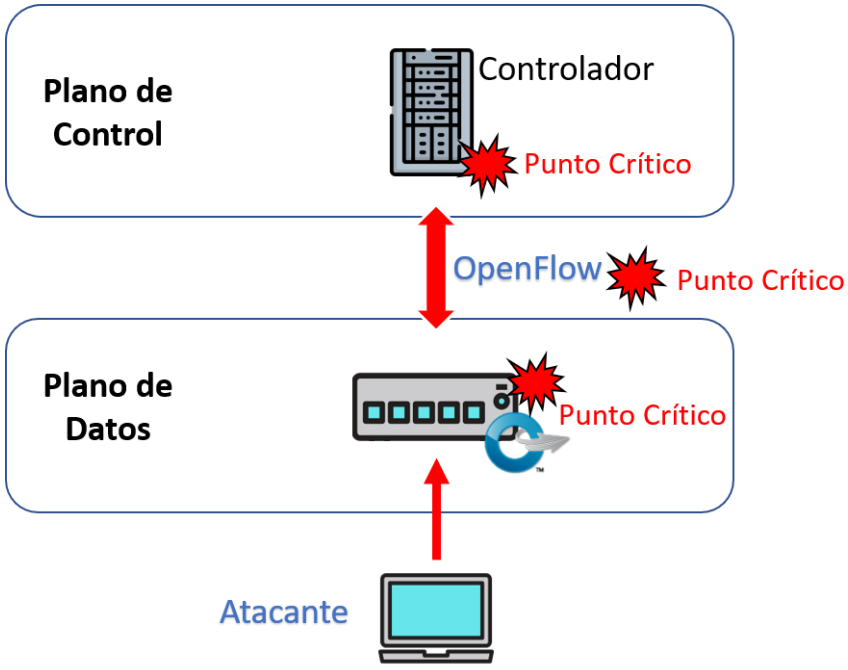


Figura 25. Puntos críticos en SDN

6. CONCLUSIONES

En el estudio de la arquitectura y funcionamiento de la red SDN se identifica que esta tecnología tiene gran potencial ya que ofrece una serie de beneficios enfocadas hacia la administración de la red debido a su estructura permitiendo que la red sea dinámica y programable teniendo un gran potencial para su implementación brindando ventajas en la optimización de tiempos en la configuración de los dispositivos de la red.

Con respecto a los ataques que se usan en estos tiempos, se encontró que uno de los más utilizados son los ataques DoS para inhabilitar diferentes tipos de servicios mediante el colapso de los sistemas por medio de la inundación de paquetes en la red. Analizando la arquitectura de la red SDN en cuanto a sus planos que están separados se puede identificar que tiene puntos vulnerables susceptibles a ser atacados por cibercriminales debido a que el controlador localizado en el plano de control tiene como función crear reglas de flujo para cada uno de los paquetes para poder ser reenviado hacia el destino convirtiéndose en una ventaja para el atacante.

Como resultados se obtiene que en la capa de control los tres ataques de inundación presentaron grandes envíos de paquetes el cual provocaron la generación de solicitudes de reglas de flujo hacia el controlador mediante mensajes “Packet_In”, en consecuencia, se ve sobrecargado el controlador por las peticiones generadas produciendo una disminución en el rendimiento de la red el cual se ve reflejado en la capa de datos donde se evidenció el incremento en los tiempos de respuesta de ida y vuelta como en el host 2 con 482ms y host 4 con 491ms, los dispositivos más afectados están ubicados en el entorno del atacante y víctima con los mayores tiempos.

BIBLIOGRAFÍA

Brandon Heller, Y. Y. (2015). *Home · mininet/openflow-tutorial Wiki · GitHub*. GitHub. https://github.com/mininet/openflow-tutorial/wiki#OpenFlow

Byun, M., Lee, Y., & Choi, J. Y. (2019). Risk and avoidance strategy for blocking mechanism of SDN-based security service. *International Conference on Advanced Communication Technology, ICACT*, *2019-Febru*, 187–190. https://doi.org/10.23919/ICACT.2019.8701887

Cisco. (2021). *What Is a Firewall? - Cisco*. CISCO. https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html

Datasec-soft. (2020). *Escaneo de Vulnerabilidades y Hackeo Ético | Datasec*. Datasec-Soft. https://www.datasec-soft.com/content/escaneo-de-vulnerabilidades-y-hackeo-ético

HelpSystems. (2020). *Sistemas de prevención y detección de intrusiones | HelpSystems*. HelpSystem. https://www.helpsystems.com/es/soluciones/seguridad-informatica/infraestructura/deteccion-y-prevencion-de-intrusiones

Hping. (2021). *Hping - Active Network Security Tool*. Hping. http://www.hping.org/

Infocyte. (2021). *Cybersecurity 101: Intro to the Top 10 Common Types of Cybersecurity Attacks - Infocyte*. Infocyte. https://www.infocyte.com/blog/2019/05/01/cybersecurity-101-intro-to-the-top-10-common-types-of-cyber-security-attacks/

Ionos. (2019). *DoS y DDoS: patrones de ataque y medidas de defensa - IONOS*. IONOS. https://www.ionos.es/digitalguide/servidores/know-how/dos-y-ddos-un-vistazo-a-ambos-patrones-de-ataque/

IONOS. (2019). *SDN: el centro de datos del futuro ya está aquí - IONOS*. IONOS. https://www.ionos.es/digitalguide/servidores/know-how/software-defined-network/

Feamster Nick, Jennifer Rexford, & Zegura Ellen. (2019). The road to SDN. In *The Technology Trap* (Vol. 150, Issue 10, pp. 342–366). https://queue.acm.org/detail.cfm?id=2560327

Mininet. (2021). *Mininet Overview - Mininet*. Mininet. http://mininet.org/overview/

ONF. (2013). OpenFlow Switch Specification 1.4.0. *Current*, *0*, 1–3205.

OpenDayLight. (2020). *Platform Overview - OpenDaylight*. OpenDayLight. https://www.opendaylight.org/about/platform-overview

Ramiro, R. (2017). *Seguridad en las redes definidas por Software (SDN)*. https://ciberseguridad.blog/seguridad-en-las-redes-definidas-por-software-sdn/

Sidhu, S., & Gupta, H. (2019). A Security Mechanism for Software Defined Vulnerabilities. *2019 4th International Conference on Information Systems and Computer Networks, ISCON 2019*, 59–62. https://doi.org/10.1109/ISCON47742.2019.9036247

Singh, J., & Behal, S. (2020). Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. *Computer Science Review*, *37*, 100279. https://doi.org/10.1016/j.cosrev.2020.100279

Spera, C., & Cone, B. D. M. L. S. (2013). *Software Defined Network: el futuro de las arquitecturas de red*. 42–45.

UserDataCenter. (2020). *Redes Tradicionales vs SDN Definidas por Software | Blogs La Salle | Campus Barcelona*. https://blogs.salleurl.edu/es/redes-tradicionales-vs-sdn-definidas-por-software

ANEXOS

A. INSTALACIÓN MININET

La instalación de Mininet se realiza de forma sencilla, como primer paso se obtiene el código fuente con la siguiente línea comando:

```
$ git clone git://github.com/mininet/mininet
```

Una vez copiado los paquetes se procede a acceder a la carpeta e instalar Mininet con los siguientes comandos:

```
$ cd mininet
$ mininet/util/install.sh
```

Con estos sencillos pasos se instala Mininet además viene por defecto herramientas preinstaladas de OpenFlow.

B. INSTALACIÓN OPENDAYLIGHT

El primer paso es descargar la compilación de OpenDayLight en la página oficial con el siguiente link: https://docs.opendaylight.org/en/latest/downloads.html. Una vez descargado los archivos se procede hacer los siguientes pasos en la terminal de Ubuntu:

```
sudo apt update
sudo apt install openjdk
sudo update-alternatives --config java
echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre' >> ~/.bashrc
source ~/.bashrc
echo $JAVA_HOME
unzip karaf-0.8.4.zip
```

Cuando esté descomprimido el archivo se inicia el controlador como se observa en la Figura 26, con la siguiente línea de comando se enciende el software OpenDayLight:

```
$ sudo ./bin/karaf
```

Figura 26. Arranque del controlador OpenDayLight

Para el correcto funcionamiento del controlador con Mininet es necesario instalar unas extensiones para agregar nuevas herramientas mediante el siguiente comando:

```
feature:install odl-restconf-all odl-openflowplugin-all odl-l2switch-all odl-mdsal-all
odl-yangtools-common odl-dlux-all
```

C. CÓDIGO DE TOPOLOGÍA SDN

A continuación, se encuentra el código Python de la topología implementada en Mininet el cual se trabajó en la investigación.

```
#!/usr/bin/env python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():
    net = Mininet( topo=None,
                   build=False,
                      host=CPULimitedHost,
                      link=TCLink,
                   ipBase='10.0.0.0/8')

    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                      controller=RemoteController,
                      ip='127.0.0.1',
                      protocol='tcp',
                      port=6633)

    info( '*** Add switches\n')
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch)
    s1 = net.addSwitch('s1', cls=OVSKernelSwitch)
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch)

    info('*** Add hosts\n')
    h5 = net.addHost('h5', cls=Host, ip='10.0.0.5', defaultRoute=None)
    h4 = net.addHost('h4', cls=Host, ip='10.0.0.4', defaultRoute=None)
    h6 = net.addHost('h6', cls=Host, ip='10.0.0.6', defaultRoute=None)
    h3 = net.addHost('h3', cls=Host, ip='10.0.0.3', defaultRoute=None)
    h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
```

```
    info( '*** Add links\n')
    h1s1 = {'bw':10,'max_queue_size':1000}
    net.addLink(h1, s1, cls=TCLink , **h1s1)
    h2s1 = {'bw':10,'max_queue_size':1000}
    net.addLink(h2, s1, cls=TCLink , **h2s1)
    h3s2 = {'bw':10,'max_queue_size':1000}
    net.addLink(h3, s2, cls=TCLink , **h3s2)
    h4s2 = {'bw':10,'max_queue_size':1000}
    net.addLink(h4, s2, cls=TCLink , **h4s2)
    s3h6 = {'bw':10,'max_queue_size':1000}
    net.addLink(s3, h6, cls=TCLink , **s3h6)
    s3h5 = {'bw':10,'max_queue_size':1000}
    net.addLink(s3, h5, cls=TCLink , **s3h5)
    s1s2 = {'bw':10,'max_queue_size':1000}
    net.addLink(s1, s2, cls=TCLink , **s1s2)
    s2s3 = {'bw':10,'max_queue_size':1000}
    net.addLink(s2, s3, cls=TCLink , **s2s3)
    s3s1 = {'bw':10,'max_queue_size':1000}
    net.addLink(s3, s1, cls=TCLink , **s3s1)

    info( '*** Starting network\n')
    net.build()
    info( '*** Starting controllers\n')
    for controller in net.controllers:
        controller.start()

    info( '*** Starting switches\n')
    net.get('s3').start([c0])
    net.get('s1').start([c0])
    net.get('s2').start([c0])

    info( '*** Post configure switches and hosts\n')

    CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    myNetwork()
```