UNIVERSIDAD TÉCNICA DEL NORTE

INSTITUTO DE POSTGRADO

MAESTRÍA EN TELECOMUNICACIONES

"METODOLOGÍA DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DDOS EN ENTORNOS SDN BASADO EN LA NORMA ISO/IEC 27001 PARA MEJORAR LA SEGURIDAD EN EL PLANO DE CONTROL"

Trabajo de Investigación previo a la obtención del Título de Magíster en Telecomunicaciones

DIRECTOR:

Msc. Fabián Cuzme

AUTORA:

Jessica Estefanía Báez Cheza

IBARRA - ECUADOR

2021

APROBACIÓN DEL TUTOR

Yo, Fabián Geovanny Cuzme Rodríguez, certifico que la estudiante Jessica Estefanía Báez Cheza con cédula N° 1003601182 ha elaborado bajo mi tutoría la sustentación del trabajo de grado titulado "METODOLOGÍA DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DDOS EN ENTORNOS SDN BASADO EN LA NORMA ISO/IEC 27001 PARA MEJORAR LA SEGURIDAD EN EL PLANO DE CONTROL"

Este trabajo se sujeta a las normas y metodologías dispuestas en el reglamento del título a obtener, por lo tanto, autorizo la presentación a la sustentación para la calificación respectiva.

Ibarra, 21 de julio de 2021

MSc. Fabián Cuzme Rodríguez
Tutor
CI: 1311527012

APROBACIÓN DEL TRIBUNAL

El presente trabajo de grado titulado "METODOLOGÍA DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DDOS EN ENTORNOS SDN BASADO EN LA NORMA ISO/IEC 27001 PARA MEJORAR LA SEGURIDAD EN EL PLANO DE CONTROL", constituye requisito previo para la obtención del título de Magíster en Telecomunicaciones del Instituto de Posgrado de la Universidad Técnica del Norte.

Autora: Jessica Estefanía Báez Cheza

Trabajo de grado, aprobado en nombre de la Universidad Técnica del Norte, por el siguiente jurado: MSc. Edwin Marcelo Jurado Ávila, MSc. Fabián Cuzme Rodríguez y MSc. Mauricio Domínguez Limaico, a los 17 días del mes de agosto de 2021.

MSc. Marcelo Jurado
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
FABIAN GEOVANNY CUZME RODRIGUEZ

hmdominguez@utn.edu.ec

Firmado digitalmente por hmdominguez@utn.edu.ec
DN: cn=hmdominguez@utn.edu.ec
Motivo:Asentamiento de nota de grado
Ubicación:OficinaVirtual
Fecha:2021-07-29

MSc. Fabián Geovanny Cuzme
TUTOR

MSc. Mauricio Domínguez
ASESOR

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1.- IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL CONTACTO	
Cédula de Identidad	100360118-2
Apellidos y Nombres	Báez Cheza Jessica Estefanía
Dirección	Av. 17 de Julio 5-26
E-mail	jebaezc@utn.edu.ec
Teléfono Fijo	062602997
Teléfono Móvil	0939374645
DATOS DE LA OBRA	
Título	METODOLOGÍA DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DDOS EN ENTORNOS SDN BASADO EN LA NORMA ISO/IEC 27001 PARA MEJORAR LA SEGURIDAD EN EL PLANO DE CONTROL
Autora	Báez Cheza Jessica Estefanía
Fecha: DD/MM/AA	20/08/2021
Programa de posgrado	Maestría en Telecomunicaciones
Título por el que opta:	Magíster en Telecomunicaciones
Tutor	MSc. Fabián Geovanny Cuzme Rodríguez

2. CONSTANCIAS

La autora Jessica Estefanía Báez Cheza, manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 20 días de agosto de 2021

Jessica Estefanía Báez Cheza
CI: 1003601182

DEDICATORIA

Este proyecto lo dedico a mis padres Raúl y María y a mi hermano Wilmer, quienes me han sabido apoyar en cada etapa de mi vida. Con su esfuerzo, consejos, valores, su buen ejemplo y comprensión me han guiado y enseñado a nunca dejarme vencer por ninguna adversidad, para ellos con todo mi amor y cariño.

Jessica Báez

RECONOCIMIENTO

Mis sinceros agradecimientos al MSc. Fabián Cuzme por aceptar mi proyecto de titulación y guiarlo de la mejor manera posible, atendiendo siempre a mis inquietudes en cada etapa que debía realizar.

Al MSc. Mauricio Domínguez por compartir sus conocimientos durante el desarrollo del proyecto de titulación y realizar observaciones acertadas que contribuyeron a mejorar mi trabajo.

INDICE DE CONTENIDOS

INDICE DE TABLAS

INDICE DE FIGURAS

UNIVERSIDAD TÉCNICA DEL NORTE INSTITUTO DE POSGRADO
PROGRAMA DE MAESTRÍA

"METODOLOGÍA DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DDOS EN ENTORNOS SDN BASADO EN LA NORMA ISO/IEC 27001 PARA MEJORAR LA SEGURIDAD EN EL PLANO DE CONTROL"

Autor: Jessica Estefanía Báez Cheza

Tutor: MSc. Fabián Cuzme Rodríguez

Año: 2021

RESUMEN

El presente trabajo se realizó con el objetivo de desarrollar una Metodología para la implementación de una solución de seguridad relacionada a la detección y mitigación de ataques DDoS en el plano de control de SDN, capaz de ser utilizada como guía para los profesionales de la rama y demás interesados en la seguridad de la información. La metodología se desarrolló en base a la norma ISO 27001 y su alineación con el ciclo PDCA, de donde se tomaron las directrices generales para la realización de cada uno de los subprocesos de la metodología planteada: Identificación de riesgos, Planificación, Selección del mecanismo, Pruebas, Implementación, Monitoreo y Mejora. La validación de la metodología se realizó mediante la aplicación de todas las actividades definidas dentro de los subprocesos mencionados en un escenario de simulación controlado, para ello se utilizó la herramienta Mininet y el controlador POX. Como resultado de la aplicación exitosa de la metodología se obtuvo la solución a la problemática planteada, a través de la implementación de un mecanismo de detección de ataques DDoS basado en el cálculo de la entropía de la red y mitigación mediante el bloqueo del puerto del switch desde donde se genera el tráfico malicioso. La solución resultó ser rápida y efectiva frente a distintos tipos de ataques DDoS: TCP, UDP e ICMP flood.

Palabras clave: DDoS, SDN, ISO27001, POX.

UNIVERSIDAD TÉCNICA DEL NORTE

INSTITUTO DE POSGRADO PROGRAMA DE MAESTRÍA

"METODOLOGÍA DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DDOS EN ENTORNOS SDN BASADO EN LA NORMA ISO/IEC 27001 PARA MEJORAR LA SEGURIDAD EN EL PLANO DE CONTROL"

Autor: Jessica Estefanía Báez Cheza

Tutor: MSc. Fabián Cuzme

Año: 2021

ABSTRACT

This work was carried out with the objective of developing a Methodology for the implementation of a security solution related to the detection and mitigation of DDoS attacks in the SDN control plane, capable of being used as a guide for professionals in the field and others interested in information security. The methodology was developed based on the ISO 27001 standard and its alignment with the PDCA cycle, from which the general guidelines were taken to carry out each of the sub-processes of the proposed methodology: Risk identification, Planning, Selection of the mechanism, Testing, Implementation, Monitoring and Improvement. The validation of the methodology was carried out by applying all the activities defined within the aforementioned sub-processes in a controlled simulation scenario, for which the Mininet tool and the POX controller were used. As a result of the successful application of the methodology, the solution to the problem was obtained, through the implementation of a DDoS attack detection mechanism based on the calculation of the entropy of the network and mitigation by blocking the switch port from where the malicious traffic is generated. The solution turned out to be fast and effective against different types of DDoS attacks: TCP, UDP and ICMP flood.

Keywords: DDoS, SDN, ISO27001, POX.

CAPITULO I
INTRODUCCIÓN

En este capítulo se realiza una reseña de la temática del presente trabajo, sus propósitos principales y aportes más relevantes.

1.1.Problema de Investigación

Las Redes Definidas por Software (SDN) son una tecnología de red emergente que proveen un nuevo tipo de arquitectura de red más flexible y escalable, capaz de responder rápidamente a los cambios en las necesidades del negocio y del usuario final a través de una gestión de red simplificada (Mladenov, 2019). El poder de SDN se ha demostrado día a día, se extiende en varias áreas, desde pequeñas redes de área local hasta arquitecturas de nube pública. En la mayoría de los casos, el SDN muestra su gran éxito al proporcionar confiabilidad, efectividad, simplicidad y flexibilidad con un menor costo (Dao et al., 2015). Sin embargo, a pesar de los numerosos beneficios, la seguridad SDN sigue siendo motivo de preocupación entre las comunidades de investigación.

La perspectiva SDN se basa en la separación del plano de control del plano de datos, en donde, la inteligencia de la red se concentra en un solo punto, denominado controlador. La naturaleza centralizada del controlador lo convierte en un elemento vulnerable a ataques de inundación que pueden provocar la interrupción del servicio de toda la red (Deepa et al., 2018). Uno de los desafíos críticos es el impacto de los ataques de Denegación de Servicio Distribuido (DDoS) en las redes SDN. Un ataque de DDoS dirigido hacia el controlador SDN podría agotar sus recursos de procesamiento, volviéndolo inaccesible para los paquetes legítimos, lo cual afectaría a la disponibilidad de servicio (Thomas & James, 2017).

La efectividad de los ataques DDoS se verá a un ritmo mucho más rápido y con un mayor daño en las redes SDN en comparación con las redes tradicionales (Mousavi & St-hilaire, 2016). Actualmente existen varios trabajos de investigación que proponen distintas formas de detección y mitigación de ataques DDoS en entornos SDN, sin embargo, no existe una metodología que sirva como guía para la implementación de estas soluciones.

1.2. Objetivos de la Investigación

Se define el objetivo general y los objetivos específicos relacionados con la presente investigación.

1.2.1. Objetivo General

Desarrollar una metodología de detección y mitigación de ataques DDoS en entornos SDN basado en la norma ISO/IEC 27001 para mejorar la seguridad en el plano de control.

1.2.2. Objetivos Específicos

- Analizar la literatura asociada a las redes definidas por software y su aplicabilidad en la presente investigación.
- Estudiar los diferentes tipos de ataques DDoS en SDN y sus mecanismos de detección y mitigación.
- Diseñar la metodología basada en la norma ISO/IEC 27001 en el dominio de Seguridad Lógica, para la detección y mitigación de ataques de inundación DDoS en SDN.
- Evaluar la metodología planteada en un escenario de simulación controlado.

1.3.Justificación

La seguridad en una red de datos es un tema que debe ser prioritario, pues de otro modo la organización se expone a una serie de amenazas que pueden comprometer la disponibilidad del servicio. Los ataques DDoS son uno de los principales problemas en la seguridad de la red actual, debido a la interrupción masiva que puede causar en cualquier tipo de infraestructura de red. (Kia, 2015). Los sitios de comercio electrónico, sitios de blogs y sectores financieros sol los principales objetivos de ataques DDoS (Sahoo, 2017). También se entiende que la estructura de SDN es vulnerable a tales ataques.

Un ataque de DDoS es el ataque más potencial en el entorno SDN (Deepa et al., 2018). Este tipo de amenazas están orientadas a que las funciones del controlador, como los servicios en línea o las aplicaciones web, no estén disponibles; ya que el controlador tendrá todos sus

recursos destinados a procesar un gran número paquetes maliciosos, haciéndolo inaccesible para el tráfico legítimo (Dharma et al., 2015).

Debido a que el plano de control es el punto más débil dentro de la seguridad de SDN, se considera la viabilidad de desarrollar una metodología que reúna las bondades de las propuestas para la detección y mitigación de ataques de DDoS de inundación dirigidos al plano de control con el objetivo de mejorar la seguridad dentro de las SDN. La metodología que se va a desarrollar será una guía específica para la implementación de las soluciones referentes al tema de la presente propuesta, difiriendo así de la norma ISO/IEC 27001 que presenta directrices generales en el área de gestión de la seguridad de la información.

Esta investigación se enmarca en el objetivo 5 del Plan Nacional de Desarrollo 2017-2021, que en su política 5.6 cita: "Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las universidades"(Plan Nacional de Desarrollo 2017-2021-Toda Una Vida, 2017).

La presente investigación contribuye a la línea de investigación institucional: Desarrollo, aplicación de software, cyber security (seguridad cibernética).

CAPITULO II

2. MARCO REFERENCIAL

En este capítulo se realiza una revisión de los trabajos previos realizados sobre el tema de estudio, además se realiza un desarrollo amplio de los conceptos necesarios para sustentar el problema planteado.

2.1.Antecedentes

Se visitaron bases de datos de investigación académica y se tomó como referencia los siguientes trabajos de investigación:

En el artículo científico titulado **"A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments"**, se presenta el estado del arte de ataques DDoS en las SDN y escenarios de computación en la nube. Se realiza un resumen de los trabajos de investigación respecto a este tipo de ataques y problemas abiertos para identificar (Dong et al., 2019).

En el artículo titulado **"Denial-of-Service Attacks in OpenFlow SDN Networks"** se emularon ataques en el marco de Mininet y se analizaron los efectos de los ataques. Además, se proponen algunos mecanismos para mitigar el impacto de estos ataques. También se discuten las configuraciones (valor de tiempo de espera y ancho de banda del plano de control) que deben ajustarse de acuerdo con los requisitos de la red para evitar ataques DoS (Kandoi & Antikainen, 2015).

En este sentido, Kandoi & Antikainen (2015) en su artículo titulado **"Denial-of-Service Attacks in OpenFlow SDN Networks"** se emularon ataques en el marco de Mininet y se analizaron los efectos de los ataques. Además, se proponen algunos mecanismos para mitigar el impacto de estos ataques. También se discuten las configuraciones (valor de tiempo de espera y ancho de banda del plano de control) que deben ajustarse de acuerdo con los requisitos de la red para evitar ataques DoS.

Por otra parte, Sahoo (2017) propone el mecanismo de detección de ataques DDoS basado en General Entropy (GE). Los resultados experimentales muestran que este mecanismo de

detección puede detectar el ataque rápidamente y lograr una alta precisión de detección con una baja tasa de falsos positivos.

El trabajo de investigación titulado **"Early Detection of DDoS Attacks against SDN Controllers"** propone utilizar el control centralizado de SDN para la detección de ataques e introduce una solución efectiva y liviana en términos de los recursos. Se muestra cómo los ataques DDoS pueden agotar los recursos del controlador y proporciona una solución para detectar tales ataques en función de la variación de entropía de la dirección IP de destino (Mousavi & St-hilaire, 2016).

Finalmente, en el trabajo de investigación realizado por Dao et al., (2015) se propone un método basado en la técnica de filtrado de IP para mitigar el ataque DDoS. El método presentado en este artículo funciona en el protocolo OpenFlow, analiza el tráfico del usuario para detectar y prevenir el ataque.

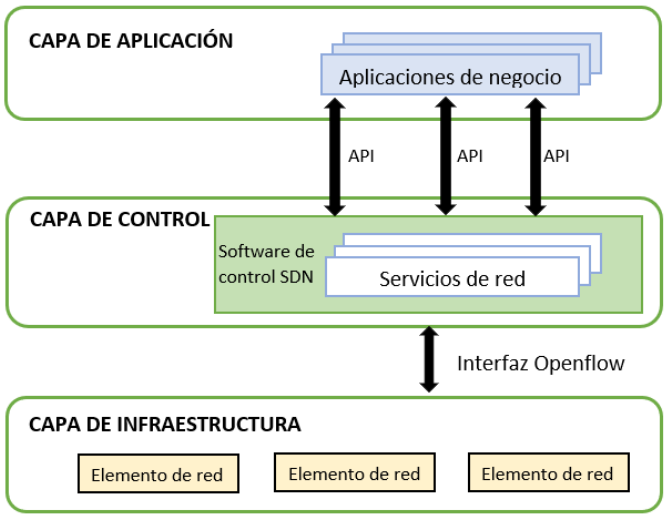
2.2.Referentes Teóricos SDN

La Red Definida por Software (SDN) es un enfoque de red que permite a los administradores de red inicializar, controlar, cambiar y administrar el comportamiento de la red de manera dinámica mediante interfaces abiertas como el protocolo OpenFlow. SDN está cambiando la forma en que se controlan, gestionan y configuran las infraestructuras de redes de TI (Lawal & At, 2018). La perspectiva SDN se basa en la separación del plano de control del plano de datos, en donde, uno toma las decisiones de reenvío de datos y el otro las ejecuta.

En cuanto con la arquitectura SDN, el plano de control está bajo la responsabilidad de un controlador centralizado que toma todas las decisiones de reenvío de flujo en la red. La comunicación entre los dos planos se logra a través del protocolo OpenFlow especificado por la Open Networking Foundation (ONF) (Lawal & At, 2018). La Figura 1 muestra la arquitectura de una SDN.

Figura 1

Arquitectura SDN



Nota. Adaptado de Dong et al. (2019)

En lo referente al plano de aplicación SDN, Bannour et al. (2017) indica que este plano incluye aplicaciones SDN o programas diseñados para implementar la lógica y las estrategias de control de la red. Este plano de nivel superior se comunica con el plano de control mediante una API Northbound. Las aplicaciones SDN comunican sus requisitos de red al controlador SDN, quien los traduce en comandos específicos de Southbound y reglas de reenvío que definen el comportamiento de los dispositivos individuales del plano de datos. Algunas aplicaciones SDN comunes son: enrutamiento, ingeniería de tráfico (TE, por sus siglas en inglés), firewalls y equilibrio de carga, las cuales se ejecutan sobre las plataformas de controladores existentes.

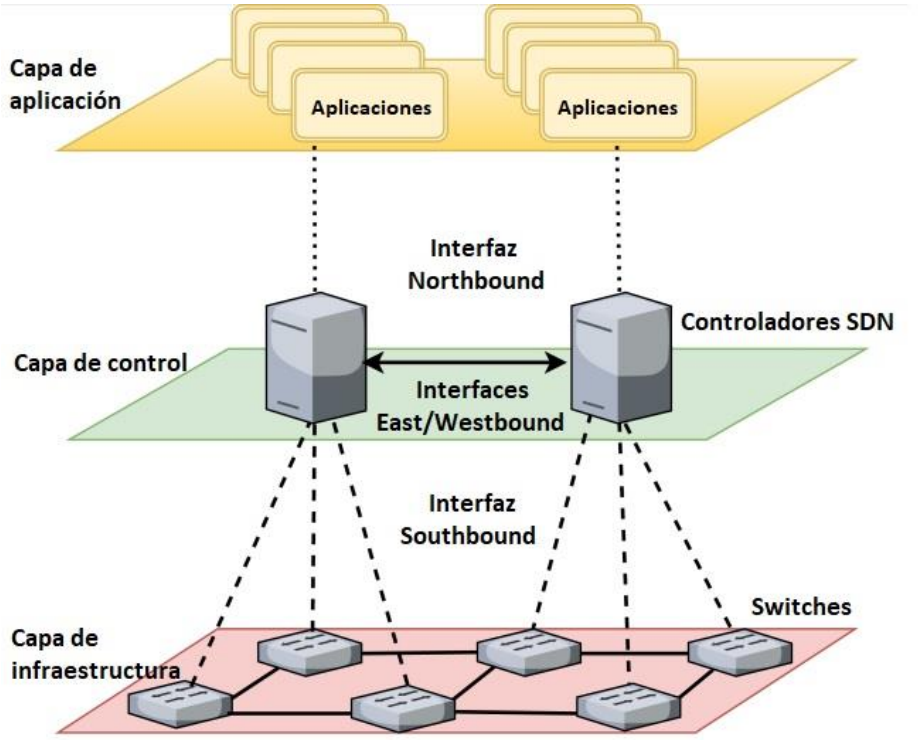
Por otra parte, el plano de control se considera la principal capa en la arquitectura SDN, consiste en un controlador de software centralizado que tiene la función de manejar las comunicaciones entre las aplicaciones y dispositivos de red a través de interfaces abiertas (Bannour et al., 2017). También proporciona funciones de gestión centralizada, por ejemplo, descubrimiento de topología, sincronización de estado, gestión de dispositivos (Yu et al., 2018). Adicionalmente permite la instalación de programas externos o aplicaciones.

La capa de control SDN es también denominada como Sistema Operativo de Red (NOS, por sus siglas en Inglés), ya que admite la lógica de control de la red y proporciona una vista abstracta de la red global a la capa de aplicación, la cual contiene la información necesaria para especificar políticas al tiempo que oculta todos los detalles de implementación (Bannour et al., 2017).

Los controladores SDN se comunican a través de cuatro interfaces: southbound, northbound, eastbound y westbound como se ilustra en la Figura 2. El controlador se comunica con la capa de aplicación usando el northbound y con la capa de infraestructura usando el southbound. Para permitir la comunicación entre controladores se usa las interfaces eastbound y westbound (Haque et al., 2017).

Figura 2

Interfaces de Comunicación del Controlador SDN



Nota. Adaptado de Latif et al. (2020)

Finalmente, el plano de datos o también conocido como la capa de infraestructura, es un conjunto distribuido de elementos de red (principalmente switches) encargados del reenvío de paquetes. El plano de datos es accesible remotamente para el controlador a través de una interfaz Southbound independiente del operador (Bannour et al., 2017).

i. OpenFlow

OpenFlow es una tecnología ampliamente aceptada y utilizada que permite implementar y desplegar la tecnología SDN en las redes actuales. OpenFlow define la manera en que un controlador de software centralizado se comunica con un dispositivo de reenvío de red, además proporciona una vista clara y unificada de la red (Ahmad et al., 2015). Las políticas y servicios de red se implementan como aplicaciones OpenFlow e interactúan con el plano de control a través de la API Northbound del plano de control.

La comunicación entre el conmutador y el controlador se realiza mediante un canal habilitado para la Seguridad de la capa de transporte (TLS, por sus siglas en inglés). De acuerdo con la especificación, cada switch debe contener una o más tablas de flujo que mantendrán las entradas de flujo especificadas por el controlador.

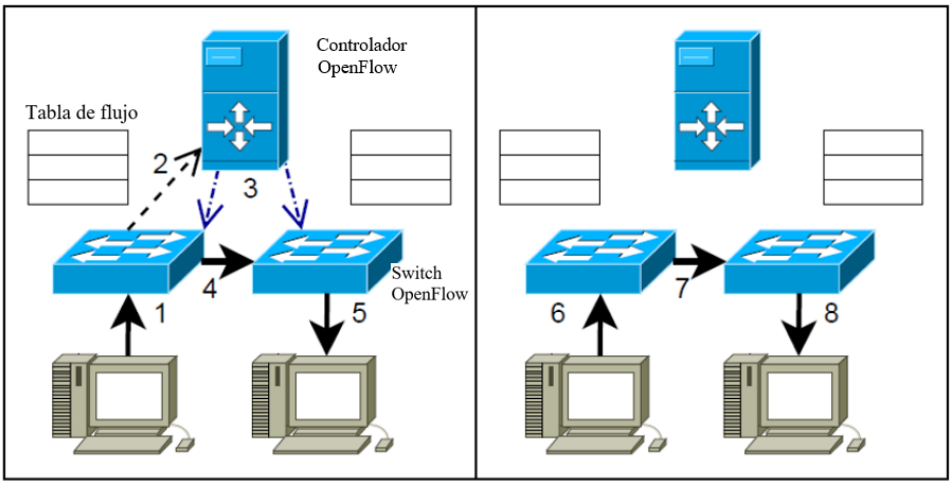
Una entrada de flujo posee un encabezado que identifica el flujo individual con el que coinciden los paquetes y la acción que debe realizar el switch para los paquetes coincidentes (Kia, 2015). Las acciones que se pueden tomar son: reenvío de paquetes, descarte, búsquedas adicionales en otras tablas de flujo, reescritura de los campos de encabezado, etc.

En la Figura 3 se describen los pasos para enrutar un flujo entre dos hosts a través de dos conmutadores en una red SDN. Inicialmente las tablas de flujo del conmutador están vacías. Cuando llega un nuevo paquete en el paso 1, este se reenvía al controlador (paso 2), debido a que no existe ninguna coincidencia en la tabla de flujo del switch. El controlador analiza el paquete y elige la acción que debe tomarse (hacia adelante o hacia abajo) y crea la entrada de flujo correspondiente. La nueva entrada de flujo se envía a los conmutadores que estarán en la ruta del paquete (paso 3). Luego, el paquete se envía al host receptor en los pasos 4 y 5. Finalmente, todos los paquetes nuevos pertenecientes al mismo flujo se enrutan

directamente, debido a que coincidirían con la nueva entrada en las tablas de flujo (pasos 6, 7 y 8).

Figura 3

Procesamiento de Flujo en OpenFlow



Nota. Tomado y traducido de Kia (2015)

ii. Controladores SDN

Los controladores constituyen el cerebro de las SDN. El controlador realiza diferentes acciones como: identificación de los dispositivos dentro de la red y las capacidades de cada uno, la recopilación de estadísticas de la red, etc. Las funcionalidades del controlador se pueden ampliar y mejorar instalando complementos como el monitoreo de la red y las detecciones de anomalías de tráfico (Centeno et al., 2016).

Actualmente existen varias implementaciones de controladores de código abierto y están escritas con diferentes lenguajes de programación como Python, C ++ y Java. Algunos de los controladores de código abierto y sus características se presentan en la Tabla 1.

Tabla 1

Características de Controladores SDN de Código Abierto

Características	Controladores SDN de Código Abierto					
	Beacon	Floodlight	NOX	POX	Ryu	ODL
Soporte OpenFlow	OF v1.0	OF v1.0	OF v1.0	OF v1.0	OF v1.0, v1.2, v1.3 y extensiones Nicira	OF v1.0
Virtualización	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch
Lenguaje de desarrollo	Java	Java	C++	Python	Python	Java
Interfaz Gráfica	Web	Web	Python+, QT4	Python+, QT4, Web	Web	Web
Soporte de plataformas	Linux, Mac OS, Windows y Android	Linux, Mac OS, Windows	Linux	Linux, Mac OS, Windows	Linux	Linux, Mac OS, Windows
Soporte de OpenStack	No	Si	Si	No	No	Si
Multiprocesos	Si	Si	Si	No	No	Si
Tiempo en el mercado	4 años	2 años	6 años	1 año	1 año	5 meses
Documentación	Buena	Buena	Media	Pobre	Media	Media

Nota. Adaptado de Centeno (2015)

El controlador POX es considerado como una plataforma adecuada para la implementación de SDN en el campo académico y de investigación (Kia, 2015). POX tiene un diseño sencillo

y ligero, en el cual se han realizado varias investigaciones académicas y prototipos de proyectos SDN.

iii. Seguridad en SDN

El esquema SDN puede generar nuevos desafíos que no se ven en las redes tradicionales. Se espera que conforme al despliegue gradual de tecnologías SDN aumente la lista de desafíos de seguridad que se deban mitigar. Las vulnerabilidades de seguridad en los SDN se concentran en los tres planos o capas (aplicación, control y datos). De acuerdo a Ahmad (2015), las principales amenazas en los planos o capas de SDN son las que se presentan en la Tabla 2.

Tabla 2

Principales Amenazas en los Planos de SDN

Plano	Tipo de amenaza	Descripción
Aplicación	Falta de autenticación y autorización	No hay mecanismos robustos de autenticación y autorización convincentes para las aplicaciones.
	Inserción de reglas de flujo fraudulentas	Aplicaciones malintencionadas pueden generar reglas de flujo falsas.
	Falta de control de acceso	Difícil de implementar el control de acceso en aplicaciones de terceros.
Control	Ataques DDoS	La naturaleza visible, la inteligencia centralizada y los recursos limitados del plano de control pueden atraer ataques DoS.
	Acceso no autorizado al controlador	Falta de mecanismos convincentes para imponer el control de acceso.
	Escalabilidad y disponibilidad	La centralización de la inteligencia tendrá desafíos de escalabilidad y disponibilidad.

Datos	Reglas de flujo fraudulentas	El plano de datos es más susceptible a las reglas de flujo fraudulento.
	Ataques de inundación	Las tablas de flujo de los conmutadores almacenan un número limitado de reglas de flujo.
	Secuestro del controlador	El plano de datos depende de la seguridad del controlador.
	Ataques TCP-Level	TLS (Transport Layer Security) es susceptible a ataques de nivel TCP.
	Ataque Man in the middle	Debido al uso opcional y complejidad de TLS.

Nota. Adaptado de Ahmad (2015)

2.3.Ataques DDoS en SDN

El ataque DDoS es un ataque dirigido por múltiples computadoras llamadas "bots" o "zombies", que son redes de computadoras controladas de forma remota por un atacante para realizar ataques masivos a un objetivo específico (Dong et al., 2019). Su motivación es provocar el agotamiento de los recursos de la red, con el objetivo de que el servicio se vea obstaculizado o detenido, lo que lleva a la falta de disponibilidad de servicio

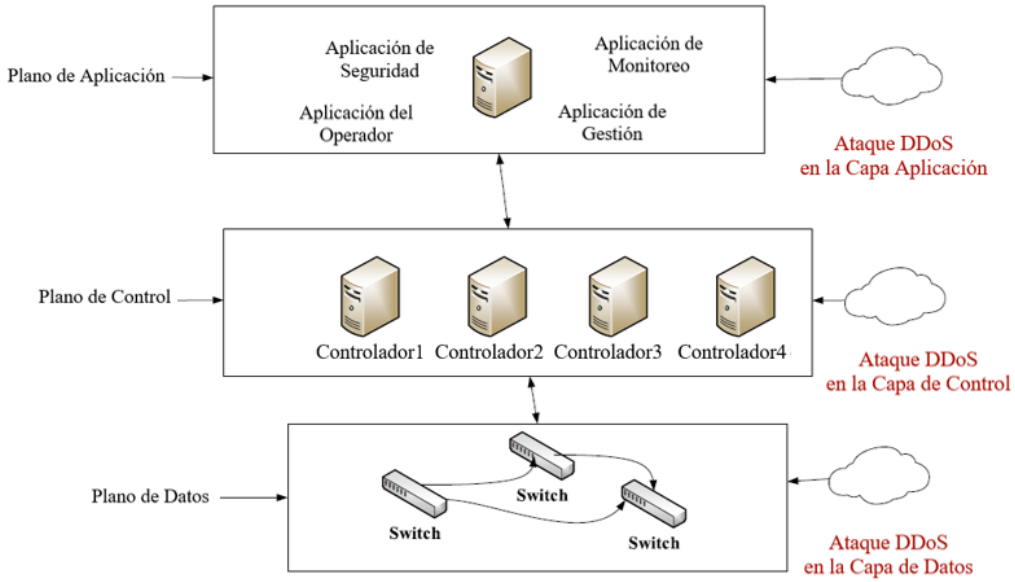
Durante el ataque se envía una gran cantidad de paquetes a uno o varios hosts de la red. Si las direcciones IP de origen de los paquetes entrantes son falsificadas (como suele ser el caso), el conmutador no encontrará una coincidencia en su tabla de flujo y tendrá que reenviar el paquete al controlador. La acumulación de paquetes de usuarios legítimos y paquetes generados por el atacante puede comprometer los recursos disponibles relacionados con la comunicación, el cálculo y el almacenamiento del controlador SDN llegando a agotarlos por completo en el peor caso. Incluso si hay un controlador de respaldo, tiene que enfrentar el mismo desafío (Mousavi & St-Hilaire, 2018).

Los ataques DDoS se pueden ejecutar en los tres planos de la arquitectura SDN como se ilustra en la Figura 4. En función de los posibles objetivos, los ataques DDoS se dividen en

tres categorías: ataques DDoS de la capa de aplicación, ataques DDoS de la capa de control y ataques DDoS de la capa de datos. Independientemente del objetivo, todos estos ataques tienen la característica común de inundar la red con enormes cantidades de paquetes, generalmente Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP) o paquetes User Datagram Protocol (UDP) (Lawal & At, 2018).

Figura 4

Ataques DDoS en SDN



Nota. Tomado y traducido de Dong et al. (2019)

En SDN el ataque DDoS se puede propagar verticalmente a medida que una capa se comunica con otra y horizontalmente debido a la comunicación entre controladores y entre aplicaciones. El atacante DDoS puede enviar una gran cantidad de tráfico malicioso a cualquier capa de SDN a través de las interfaces northbound, southbound, eastbound y westbound. El ataque se propaga a la capa de control mediante la interfaz southbound. En la capa de control, el ataque se propaga de un controlador a otro a través de la API westbound y eastbound. Por último, el tráfico se dirige a la capa de aplicación para desactivar los servicios SDN (Haque et al., 2017).

Uno de los puntos más atractivos para los ataques DoS y DDoS es el plano de control. Debido a la visibilidad de los recursos de red el controlador SDN puede convertirse en un único punto de falla, mismo que ocasionaría que toda la red se caiga ante un ataque de seguridad (Ahmad et al., 2015).

i. Ataques DDoS en la capa aplicación de SDN

Los ataques de DDoS hacia la capa de aplicación de SDN tienen como objetivo a las aplicaciones y la API northbound de SDN. Debido a que la separación entre aplicaciones SDN es difícil de reconocer, es posible que un ataque DDoS dirigido a la capa aplicación influya en otra aplicación que no sea el objetivo del atacante (Dong et al., 2019). Este tipo de ataques establecen conexiones TCP completas con el servidor de destino y luego comienzan a inundarlo con una gran cantidad de solicitudes HTTP con el objetivo de saturar el ancho de banda disponible a través del tráfico ilegítimo generado por el atacante. Al tratarse de ataques lentos y bajos, es una tarea difícil el distinguirlos del tráfico legítimo. Por lo tanto, los ataques a la capa de aplicación de SDN son herramientas que pueden resultar muy exitosas para que los atacantes dañen a las víctimas en los tiempos actuales (Bawany & Shamsi, 2016).

El principal desafío que se debe abordar en torno a este problema de seguridad, es diferenciar entre un ataque y una multitud repentina de paquetes de usuarios legítimos.

ii. Ataques DDoS en la capa de datos de SDN

El conmutador OpenFlow y la tabla de flujo pueden ser los objetivos para un ataque de DDoS, debido a que incluyen información administrativa, de control de acceso y de transmisión. El atacante tiene como objetivo romper la funcionalidad de la red mediante el acceso no autorizado físico o virtual a la red. Para el conmutador OpenFlow es imposible almacenar todas las reglas de flujo, ya que su capacidad de almacenamiento es limitada. Si el atacante envía una gran cantidad de paquetes desde una dirección desconocida en poco tiempo, el controlador escribe nuevas reglas para estos paquetes y las reenvía a la tabla de flujo, la cual se llena en poco tiempo y se queda sin espacio para escribir una nueva regla (Dong et al., 2019). En consecuencia, se detiene la transmisión del tráfico legítimo.

La memoria caché de flujo también es un objetivo para los ataques DDoS. Cuando el conmutador recibe el paquete desde el puerto de entrada se busca una coincidencia para los flujos en una tabla de flujo. Si existe una coincidencia, el paquete se reenvía desde la memoria caché hacia puerto de salida. Cuando no se encuentra una coincidencia, el paquete se reenvía al controlador con un mensaje Packet_In. El controlador responde con un mensaje OFPT_FLOW_MOD asignando el hard_timeout e idle_timeout, que definen las reglas necesarias para el paquete y cuánto tiempo permanecerán las reglas. Cuando el conmutador recibe la regla del controlador, el paquete se procesa y la regla se almacena en la memoria caché de la tabla de transmisión para procesar directamente los paquetes entrantes. Durante un ataque de DDoS una gran cantidad de paquetes reenviados al conmutador por nodos maliciosos se llevan a la memoria caché y esperan la respuesta del controlador, que incluye la información sobre las reglas de flujo (Polat et al., 2020). Los paquetes que provienen de nodos atacantes llenan el búfer del conmutador provocando que los paquetes legítimos no sean transmitidos.

iii. Ataques DDoS en la capa de control de SDN

El objetivo de los ataques DDoS dirigidos a la capa de control de SDN es provocar la indisponibilidad de la red al sobrecargar al controlador con grandes cantidades de tráfico proveniente de múltiples fuentes.

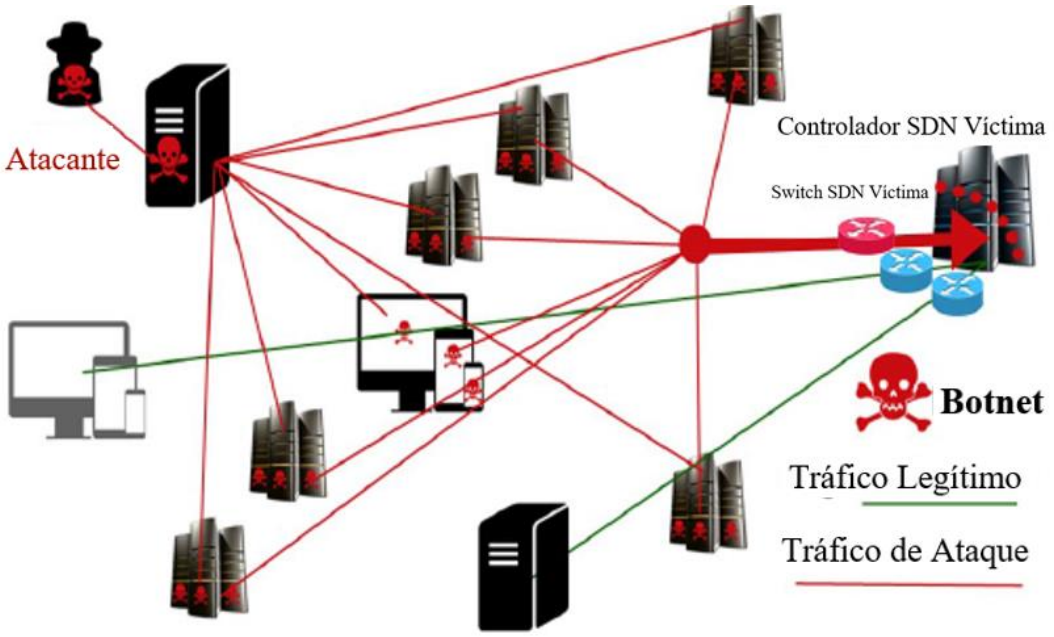
El controlador toma decisiones para reenviar paquetes de acuerdo con las reglas de flujo. Cuando el conmutador encuentra un nuevo paquete de red en el plano de datos, y no hay reglas de flujo para hacer coincidir la información de flujo existente en la tabla de flujo el paquete completo o parte del encabezado se envía al controlador para resolver la consulta. Con un gran volumen de tráfico de red, enviar un paquete total al controlador puede gastar un gran ancho de banda de datos (Dong et al., 2019).

El primer paso para el atacante es identificar la red SDN. Se debe tener en cuenta que las redes tradicionales suelen tener una tabla de reenvío preconfigurada. Por lo tanto, no se necesita tiempo adicional para procesar y crear una entrada de flujo para un nuevo paquete entrante. Por el contrario, en las SDN el controlador necesita un poco de tiempo para crear una nueva entrada de flujo para un nuevo paquete entrante. Además, el controlador agrega

más tiempo para procesar el primer paquete en comparación con los siguientes paquetes (Dao et al., 2015). Los atacantes se basan en este conocimiento para identificar si una red es SDN o no, para ello utilizan herramientas de escaneo de red para verificar si existe una diferencia entre los tiempos de respuesta del primer paquete y los siguientes (Ahmad et al., 2015). Una vez que se ha encontrado que la red es SDN se procede a ejecutar el ataque.

Figura 5

Ataque DDoS en SDN Usando Una Botnet



Nota. Tomado y traducido de Haque (2017)

La Figura 5 ilustra cómo se lleva a cabo un ataque DDoS dirigido hacia el controlador SDN. Se observa como un atacante infecta a otros dispositivos para que sean parte de la botnet. En el momento que los bots están listos para atacar reciben un comando para atacar durante un lapso definido. Cuando un paquete de datos no coincide con la tabla de flujo, el conmutador OpenFlow envía el paquete en un mensaje al controlador y carga mensajes Packet_in superando la capacidad de procesamiento del controlador (Haque et al., 2017). Dos consecuencias de este tipo de ataques en el controlador son el agotamiento de los recursos del controlador y la limitación del acceso del tráfico de legítimo al controlador.

iv. Detección y Mitigación de Ataques DDoS en la Capa de Control de SDN

De acuerdo al trabajo realizado por Dong (2019), los principales trabajos de investigación que abordan los ataques DDoS en SDN son los citados en Mousavi (2015), Dao (2015) y Dharma (2015). A continuación, se mencionan las distintas técnicas empleadas en dichos artículos científicos.

- **Método basado en entropía**

Este método analiza el nivel de aleatoriedad del tráfico que llega al controlador (cada host tiene una cantidad promedio de paquetes que intentan alcanzar por unidad de tiempo). La entropía puede ser usada para medir la aleatoriedad en los paquetes que llegan a una red. A mayor aleatoriedad, mayor será la entropía y viceversa. Cualquier patrón se desviará de este principio y dictará un cierto orden. Un patrón se reconoce por dos componentes: tamaño de ventana y umbral. El tamaño de la ventana permite cuantificar el tráfico entrante, se basa en un período de tiempo o en una cantidad de paquetes por ventana. En cada ventana, se mide el parámetro de destino en el campo de encabezado. El umbral corresponde al valor mínimo de entropía medido durante el flujo de tráfico legítimo. La entropía se calcula dentro de la ventana para medir la aleatoriedad en los próximos paquetes. Cuando el valor de la entropía está por debajo del umbral significa que hay un ataque en curso.

La entropía (*H*) se calcula con la fórmula 1, en dónde *n* es el número de paquetes en una ventana y *pi* es la probabilidad de cada elemento en la ventana (Mousavi & St-hilaire, 2016). En este caso se entiende por elemento a la dirección IP de destino de nuevos paquetes entrantes.

$$H = \sum_{i=1}^{n} p_i \log p_i \qquad (1)$$

La entropía está en su máximo valor si todos los elementos tienen probabilidades iguales. Si un elemento aparece más que otros, la entropía decrece. Si hay un flujo continuo de datos entrantes, estos se dividen en conjuntos iguales que se denominan ventanas. En la ventana, se cuentan cada elemento y su ocurrencia. Por ejemplo, si la ventana tiene 64 elementos y

todos los elementos aparecen solo una vez, la entropía tiene un valor de 1,80. Si un elemento aparece 10 veces, la entropía disminuye a 1,64 (Mousavi & St-hilaire, 2016).

El trabajo de Mousavi (2015) utiliza la entropía basada de la dirección IP de destino para los nuevos paquetes. Cuando ocurre un ataque, la entropía disminuirá ya que la dirección IP del host o hosts bajo ataque aparecerá con mayor frecuencia. Este trabajo está diseñado específicamente para el entorno SDN ya que la detección ocurre dentro del controlador. En esta investigación se obtuvo una tasa de detección del 96%. La flexibilidad es una ventaja de este método, ya que se puede variar cualquier parámetro incluso en tiempo real para ajustarse a los requisitos del controlador. La principal limitación es que un ataque contra una red completa puede no ser detectado. Además, la solución propuesta fue diseñada para un entorno de controlador único.

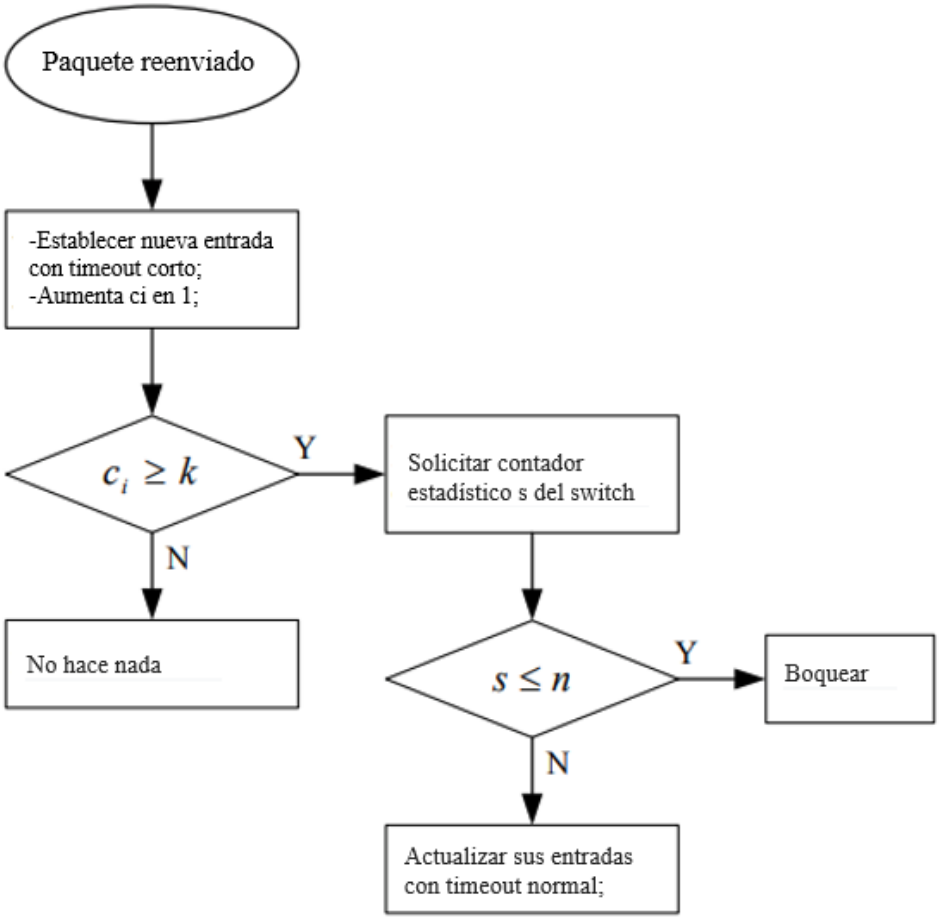
- **Método basado en el análisis de tráfico del usuario**

Este método consiste en diferenciar a los usuarios frecuentes de los usuarios "anormales" (paquetes de error o DDoS) según la cantidad de paquetes recibidos.

El trabajo realizado por Dao (2015) se basa en un estudio realizado en la red de datos de la Universidad de Auckland y en un ISP pequeño, en el cual se concluyó que alrededor del 90% de los usuarios frecuentes enviaron al menos cinco paquetes a cada destino, mientras que los usuarios "anormales" enviaron menos de cinco paquetes por conexión. En la Figura 6 se muestra el flujograma de este método. El número mínimo de paquetes por conexión de un usuario frecuente se denota por *n*, mientras que el número promedio de conexiones que los usuarios frecuentes corresponde a *k*.

Figura 6

Flujograma del Método Basado en el Análisis del Tráfico del Usuario



Nota: Tomado de Dao (2015)

Inicialmente se define una tabla *T* en el controlador para almacenar las direcciones IP de origen de los paquetes reenviados desde el conmutador. Cada dirección IP única tiene un contador *ci* para rastrear la cantidad de paquetes recibidos. Durante el ataque, cada vez que un nuevo paquete reenviado por el conmutador llega al controlador, el controlador supone que podría ser de la dirección de ataque DDoS. El controlador crea una nueva entrada específica con *hard_timeout* e *idle_timeout* con valores menores que los de las entradas normales, esto se hace para limitar su vida útil. Luego, la dirección IP de origen se actualiza en la tabla *T* para su seguimiento, y su contador *ci* se incrementa en 1 (Dao et al., 2015).

Cuando el valor de *ci* alcanza *k*, se analiza el número promedio de contadores de paquetes *s*. Si *s* es mayor que *n*, quiere decir que la dirección de origen transmitió conexiones de datos legítimas, por lo que se considera un usuario frecuente. En consecuencia, el controlador restablece el *hard_timeout* e *idle_timeout* al valor normal. Por otra parte, si *s* es menor que *n*, esto es tráfico malicioso (Dao et al., 2015).

Debido a que se establecen valores cortos para el *hard_timeout* e *idle_timeout* de los nuevos paquetes reenviados desde el conmutador al controlador, se reduce el impacto del ataque DDoS. Después de que el controlador analiza el comportamiento del tráfico de datos, las direcciones de origen maliciosas se cancelan mediante entradas de bloque. Este método puede contener el ataque DDoS, sin embargo, no es efectivo cuando la cantidad de tráfico de ataque es muy grande.

- **Método basado en tiempo**

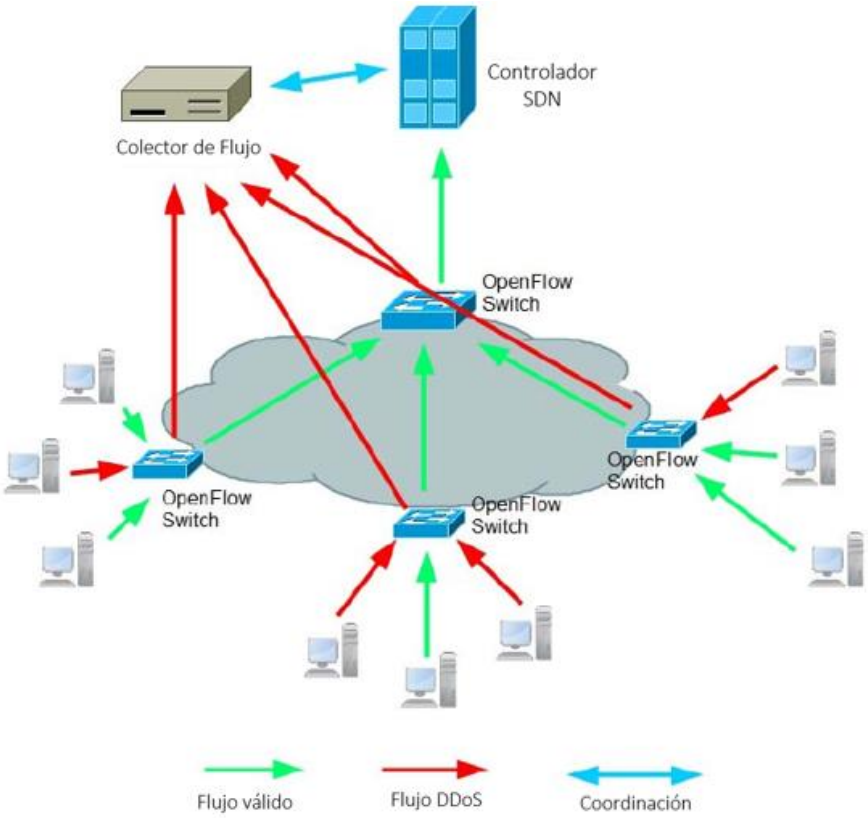
En el trabajo realizado por Dharma (2015), se considera la dirección de destino, el tiempo necesario para lograr un tráfico de alta velocidad y un patrón de ataque de tiempo de ataque DDoS. La duración de tiempo se usa para detectar el ataque y el patrón de ataque de tiempo se usa para prevenir futuros ataques DDoS.

Con este método, ilustrado en la Figura 7, se asume que cada paquete que llega al controlador es un paquete nuevo, por lo que se verificará una dirección de destino válida. Si esta dirección no es válida o es desconocida en la red, el controlador reenviará el paquete al colector de flujo. Un colector de flujo es un módulo que se implementa en el controlador para almacenar un paquete no válido y analizarlo mediante un método estadístico.

Si la acumulación de paquetes no válidos aumenta significativamente en un determinado periodo de tiempo, el colector de flujo notifica al controlador.

Figura 7

Escenario de la Detección de DDoS Basada en Tiempo



Nota. Tomado y traducido de Dharma (2015)

El controlador aplica una nueva regla para que cada dispositivo de red reenvíe el paquete no válido directamente al colector de flujo. El colector de flujo se aplica luego al procesamiento adicional del patrón de tiempo de agrupación del ataque DDoS. Este patrón se utilizará para evitar el próximo ataque DDoS.

En la Tabla 3 se muestra la comparación entre los tres métodos antes mencionados, se destacan las principales características de cada uno.

Tabla 3

Comparación Entre los Tres Métodos Mencionados

Criterio \ Método de Detección	Método Basado en Entropía	Método Basado en el Análisis de Tráfico del Usuario	Método Basado en Tiempo
Dispositivo donde se implementa la solución	Controlador de red	Controlador de red	Controlador de red
Parámetro de detección	Dirección IP de destino.	Dirección IP de origen.	Dirección de destino y tiempo característico del ataque.
Escenario de Simulación	Mininet con nueve Switches OpenFlow.	Opnet Modeler con un Switch OpenFlow.	Mininet con cuatro Switches OpenFow.
Módulos del controlador adicionales	Ninguno	Ninguno	Colector de flujo
Flexibilidad en el cambio de umbrales de detección.	Si	Si	Si
Tasa de detección	96%	No especifica	No especifica
Efectividad en un ataque dirigido a toda la red	No se comprueba	No se comprueba	No se comprueba
Efectividad en un ambiente multicontrolador	No se comprueba	No se comprueba	No se comprueba
Documentación	Alta	Baja	Baja

Nota. Elaboración propia basada en (Mousavi & St-hilaire, 2016), (Dao et al., 2015) y (Dharma et al., 2015).

El método basado en la entropía es el que ha tenido más desarrollo durante los últimos años, pues son varias las investigaciones que se han realizado usando esta técnica y muchas de sus implementaciones están documentadas en la plataforma GitHub.

2.4.Norma ISO/IEC 27001

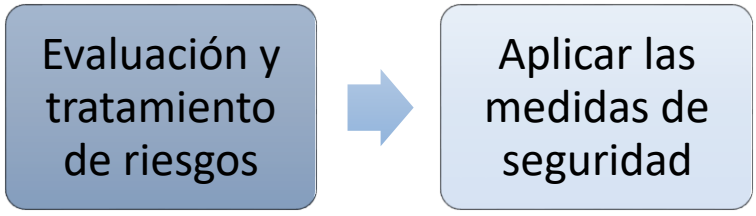
ISO/IEC 27001:2013 es el estándar internacional para la gestión de la seguridad de la información. Su objetivo es el de "especificar los requisitos para establecer, implementar, mantener y mejorar continuamente un sistema de gestión de seguridad de la información" que se integra en el sistema de gestión global de la organización, con el fin de ayudar a asegurar los recursos de información. Su aplicación permite a la organización determinar y evaluar los riesgos de seguridad de la información e implementar procedimientos y mecanismos que preserven su integridad, confidencialidad y disponibilidad de la información. La estructura organizativa de la norma ISO / IEC 27001: 2013 indica que su funcionamiento sigue el ciclo de mejora PDCA: Plan - Do - Check – Act, que busca hacer que los procesos de gestión sean más ágiles, claros y objetivos (Carvalho & Marques, 2019).

i. Funcionamiento

El objetivo de la norma ISO 27001 es proteger la integridad, disponibilidad y confidencialidad de la información en la empresa. Para cumplir este objetivo es necesario determinar los riesgos a los que está expuesta la información que maneja la organización y luego determinar lo que debe hacerse para impedir dichos riesgos. Por lo tanto, la ISO 27001 se estructura como se muestra en la Figura 8.

Figura 8

Estructura de ISO 27001



Nota. Basado en Robalino (2018)

Los procedimientos, políticas, software y equipos están dentro de los controles a aplicar.

Esta norma se basa en un enfoque del ciclo de mejora continua. Este ciclo está formado por cuatro etapas: Planificar, Hacer, Verificar y Actuar. Por ello se le denomina también como

ciclo PDCA (por sus siglas en inglés Plan-Do-Check-Act) o ciclo de Deming, en honor a su creador.

ii. Estructura

La norma ISO 27001 se dividida en 11 secciones y un Anexo. Las tres primeras secciones son introductorias y no obligatorias. Desde la cuarta sección su cumplimiento es obligatorio para la norma.

La estructura es la siguiente:

0. **Introducción:** explica el objetivo de ISO 27001 y su compatibilidad con otras normas.
1. **Objeto y campo de aplicación:** explica que esta norma se puede aplicar en cualquier tipo de organización.
2. **Referencias normativas:** hace referencia al estándar ISO/IEC 27000, en el que se proporcionan términos y definiciones.
3. **Términos y definiciones:** hace referencia a ISO/IEC 27000.
4. **Contexto de la organización:** Define los requerimientos para comprender cuestiones externas e internas, también define las partes interesadas, sus requisitos y el alcance del SGSI.
5. **Liderazgo:** Define las responsabilidades de la dirección, el establecimiento de roles y responsabilidades y el contenido de la política de alto nivel sobre seguridad de la información.
6. **Planificación:** Define los requerimientos para la evaluación de riesgos, el tratamiento de riesgos, la declaración de aplicabilidad, el plan de tratamiento de riesgos y la determinación de los objetivos de seguridad de la información.
7. **Soporte:** Define los requerimientos sobre disponibilidad de recursos, competencias, concienciación, comunicación y control de documentos y registros.
8. **Funcionamiento:** Define la implementación de la evaluación y el tratamiento de riesgos, como también los controles y demás procesos necesarios para cumplir los objetivos de seguridad de la información.
9. **Evaluación de desempeño:** Define los requerimientos para monitoreo, medición, análisis, evaluación, auditoría interna y revisión por parte de la dirección.

10. **Mejora:** Define los requerimientos para el tratamiento de no conformidades, correcciones, medidas correctivas y por último la mejora continua.

Anexo A: Proporciona 114 controles agrupados en 14 secciones.

iii. Controles

En el Anexo A de la norma ISO 27001 se establecen los controles a aplicar con el fin de evitar o dar solución a situaciones de riesgo. Los controles se deben utilizar en el contexto 6.1.3 (Tratamiento de riesgos de seguridad de la información) y se enumeran de la siguiente manera:

- A.5 Política de seguridad de la información
- A.6 Organización de seguridad de la información
- A.7 Seguridad en recursos humanos
- A.8 Gestión de activos
- A.9 Control de acceso
- A.10 Criptografía
- A.11 Seguridad física y del entorno
- A.12 Seguridad de las operaciones
- A.13 Seguridad de las comunicaciones
- A.14 Adquisición, desarrollo y mantenimiento del sistema
- A.15 Relaciones con proveedores
- A.16 Gestión de incidentes de seguridad de la información
- A.17 Aspectos de seguridad de la información para la gestión de la continuidad del negocio
- A.18 Cumplimiento

iv. Norma ISO 27001 y su Relación con la Protección contra DDoS

La norma ISO 27001 busca proteger la confidencialidad, integridad y disponibilidad de la información en una organización. Debido a que la disponibilidad es el punto crítico que se debe preservar durante un ataque DDoS, la norma ISO 27001 puede ayudar a las organizaciones de las maneras que se muestran en la Tabla 4.

Tabla 4

Relación de los Controles de la ISO 27001 con la Protección Contra Ataques DDoS

Dominio	Objetivo	Nombre del Control	Control	Relación con la Protección de DDoS
A.12 Operaciones de seguridad	A.12.6 Gestión de vulnerabilidad técnica	A.12.6.1 Gestión de vulnerabilidades técnicas	Obtener información sobre las vulnerabilidades técnicas de los sistemas de información, evaluar la exposición de la organización e implementar medidas para afrontar el riesgo.	Las encuestas periódicas aseguran que las vulnerabilidades recientemente descubiertas se manejen rápidamente.
A.13 Seguridad de las comunicaciones	A.13.1 Gestión de seguridad de red	A.13.1.1 Controles de red	Gestionar y controlar la red para proteger la información en los sistemas y aplicaciones.	El uso de mecanismos de seguridad ayuda a reducir el impacto inicial de los ataques DDoS.
		A.13.1.2 Seguridad de los servicios de red	Identificar los mecanismos de seguridad, niveles de servicio, y requisitos de gestión de los servicios de red.	
A.14 Adquisición, desarrollo y mantenimiento del sistema	A.14.1 Requisitos de seguridad de los sistemas de información	A.14.1.1 Análisis de requisitos y especificaciones de seguridad de la información	Los requisitos relacionados con la seguridad de la información deben tomarse en cuenta para los nuevos sistemas de información.	Los sistemas adecuadamente implementados reducen las posibilidades de que una vulnerabilidad sea explotada para permitir ataques DDoS
A.16 Gestión de incidentes de seguridad de la información	A.16.1 Gestión de incidentes de seguridad de la información y mejoras	A.16.1.5 Respuesta a incidentes de seguridad de la información	Responder a los incidentes de seguridad de la información conforme a los procedimientos documentados.	Al definir responsabilidades y procedimientos sobre cómo manejar los ataques, la organización puede reaccionar con rapidez.
		A.16.1.6 Aprendizaje de los incidentes de seguridad de la información	Analizar la resolución de incidentes de seguridad de la información para reducir en el futuro su probabilidad o impacto.	

Nota. Tomado de la norma ISO/IEC 27001

Aunque muchos elementos de un ataque DDoS están fuera del control de una organización, al adoptar las prácticas y controles de la norma ISO 27001, una organización puede implementar las medidas de seguridad necesarias para detectar y responder adecuadamente ante estos ataques.

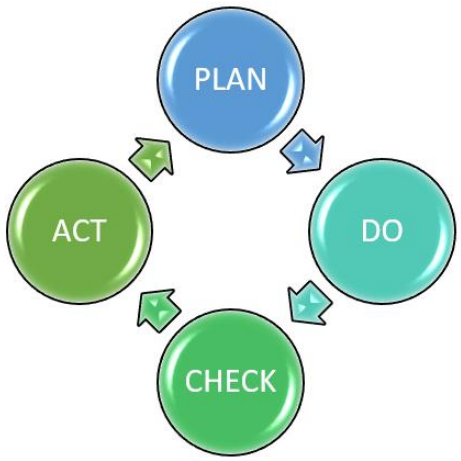
2.5.Ciclo de Deming

Actualmente es una herramienta gerencial muy utilizada en la gestión de los sistemas de calidad. En muchas organizaciones del mundo se ha podido demostrar la mejora continua obtenida tanto de eficacia como eficiencia de sus procesos.

La norma ISO 27001 adopta el ciclo de Deming como metodología, para aplicarse en procesos de sistemas de gestión de seguridad. El ciclo de Deming también, representado en la Figura 9, se conoce como ciclo de mejora continua o por sus siglas en inglés PDCA: Plan-Do-Check-Act. A continuación, se detalla cada una de las etapas del ciclo.

Figura 9

Ciclo de Deming



Nota. Basado en Pedraza (2017)

- **Plan (Planear):** En esta etapa se establecen las políticas, objetivos, procedimientos y procesos para la gestión de riesgos y mejora de la seguridad de la información para alcanzar los resultados esperados por las partes interesadas.

- **Do (Hacer):** Consiste en la implementación de acuerdo con el plan, implementar las medidas prácticas y controlar el proceso, de manera que las actividades avancen como se esperaba y alcancen el objetivo establecido.
- **Check (Verificar):** Se comparan los resultados de la etapa "Do" con los resultados esperados de la etapa "Plan". Se comprueba que las medidas adoptadas han surtido efecto, para lo cual se debe recopilar datos y monitorizar el comportamiento del sistema.
- **Act (Actuar):** En esta etapa se realizan acciones correctivas para alcanzar los resultados esperados en caso de haber alguna diferencia con los resultados obtenidos.

La norma ISO 27001 en su versión 2005 armoniza con el ciclo de mejora continua PDCA. ISO/IEC 27001 se divide en 11 secciones más el anexo A; las secciones 0 a 3 son introductorias (no obligatorias), mientras que las secciones 4 a 10 son obligatorias (Pedraza, 2017). En la Tabla 5 se muestra la manera en que se alinean las secciones de la norma ISO 27001 con las fases del PDCA.

Tabla 5

Relación Entre las Secciones de la ISO 27001 y el PDCA.

Sección de la ISO 27001	Nombre	Fase de PDCA
Sección 4	Contexto de la organización	Planear
Sección 5	Liderazgo	
Sección 6	Planificación	
Sección 7	Apoyo	
Sección 8	Funcionamiento	Hacer
Sección 9	Evaluación del desempeño.	Verificar
Sección 10	Mejora	Actuar

Nota. Tomado de Pedraza (2017)

2.6.Metodología MAGERIT

La norma ISO 27001, recomienda utilizar una metodología de análisis de riesgos para determinar los activos de información, identificar las amenazas, vulnerabilidades e impactos, con la finalidad de establecer los controles idóneos para el tratamiento de los riesgos. Para este trabajo se ha optado por utilizar MAGERIT (Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información) debido a que proporciona una metodología concreta y detallada que suele usarse en proyectos de seguridad.

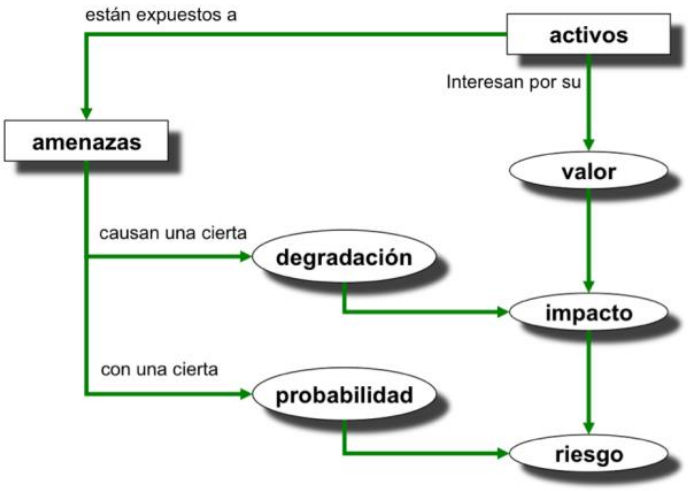
El análisis de riesgos de MAGERIT sigue los pasos:

1. Determinar los activos relevantes para la organización.
2. Determinar a qué amenazas se exponen los activos relevantes.
3. Determinar qué salvaguardas hay dispuestas y su eficacia frente al riesgo.
4. Estimar el impacto, definido como el daño sobre el activo derivado de la materialización de la amenaza.
5. Estimar el riesgo, definido como el impacto ponderado con la tasa de ocurrencia (o expectativa de materialización) de la amenaza.

La Figura 10 recoge los elementos que forman parte de este recorrido.

Figura 10

Elementos del Análisis de Riesgos



Nota. Tomado de MAGERIT (2012)

Se define como activos a los componentes susceptibles a ser atacados deliberada o accidentalmente con consecuencias para la organización. Incluyen: información, datos, servicios, aplicaciones (software), equipos (hardware), comunicaciones, recursos administrativos, recursos físicos y recursos humanos (MAGERIT, 2012). De un activo puede interesar calibrar diferentes dimensiones: confidencialidad, integridad y disponibilidad.

i. Amenazas

Una amenaza es todo lo que le puede ocurrir a los activos y causar daño. Las amenazas típicas pueden ser de origen natural, de origen industrial, defectos de las aplicaciones, causadas por personas de forma accidental y causadas por las personas de forma deliberada (MAGERIT, 2012).

Una vez que se ha identificado una amenaza, hay que valorar su influencia en el valor del activo, en los siguientes sentidos:

- **Degradación:** Describe cuán perjudicado resultaría el valor del activo. Se puede modelar cualitativamente por medio de una escala nominal como se muestra en la Tabla 6, en donde se proponen cinco criterios, que pueden ser valorados en base a cualquiera de las tres categorías.
- **Probabilidad:** Describe cuán probable o improbable es que se materialice la amenaza. Se puede modelar numéricamente como una frecuencia de ocurrencia, los valores más comunes se presentan en la Tabla 7. Los distintos criterios se pueden valorar en base a una de las tres categorías definidas en la metodología MAGERIT.

Tabla 6

Degradación de Valor

Criterio	Categoría 1	Categoría 2	Categoría 3
MA	Muy alta	Casi seguro	Fácil
A	Alta	Muy alto	Medio
M	Media	Posible	Difícil
B	Baja	Poco probable	Muy difícil
MB	Muy baja	Muy raro	Extremadamente difícil

Nota. Adaptado de MAGERIT (2012)

Tabla 7

Probabilidad de Ocurrencia

Criterio	Categoría 1	Categoría 2	Categoría 3
MA	100	Muy frecuente	A diario
A	10	Frecuente	Mensualmente
M	1	Normal	Una vez al año
B	1/10	Poco frecuente	Cada varios años
MB	1/100	Muy poco frecuente	Siglos

Nota. Tomado de MAGERIT (2012)

ii. Determinación del Impacto Potencial

El impacto es la medida del daño sobre el activo derivado de la materialización de una amenaza. Conociendo el valor de los activos y la degradación provocada por las amenazas, es necesario determinar el impacto que estas tendrían sobre el sistema. Se puede calcular el impacto en base a la Tabla 8:

Tabla 8

Estimación de Impacto

Impacto		Degradación		
		1%	10%	100%
Valor	MA	M	A	MA
	A	B	M	A
	M	MB	B	M
	B	MB	MB	B
	MB	MB	MB	MB

Nota. Tomado de MAGERIT (2012)

Aquellos activos que reciban una calificación de impacto muy alto (MA) deberían ser atendidos de manera inmediata.

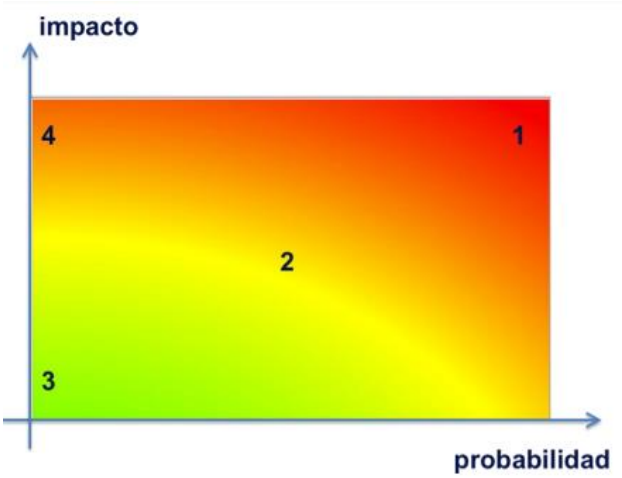
iii. Determinación del Riesgo Potencial

El riesgo se define como la medida del daño probable sobre un sistema. El riesgo se incrementa con el impacto y la probabilidad, se puede dividir en cuatro zonas que se muestran en la Figura 11 y se detallan a continuación:

- **Zona 1**: El riesgo es muy probables y de muy alto impacto.
- **Zona 2:** Cubre un amplio rango desde situaciones improbables y de impacto medio, hasta situaciones muy probables, pero de impacto bajo o muy bajo.
- **Zona 3:** El riesgo es improbable y de bajo impacto.
- **Zona 4:** El riesgo es improbable y de muy alto impacto.

Figura 11

El Riesgo en Función del Impacto y la Probabilidad



Nota. Tomado de MAGERIT (2012)

El cálculo del riesgo en base al impacto y la frecuencia se define en la Tabla 9.

Tabla 9

Riesgo en Función del Impacto y Frecuencia

Riesgo		Probabilidad				
		MB	B	M	A	MA
Impacto	MA	A	MA	MA	MA	MA

	A	M	A	A	MA	MA
	M	B	M	M	A	A
	B	MB	B	B	M	M
	MB	MB	MB	MB	B	B

Nota. Tomado de MAGERIT (2012)

Por otra parte, se califica el riesgo por medio de las escalas cualitativas mostradas en la Tabla 10.

Tabla 10

Calificación y Valoración del Riesgo

Riesgo	Calificación	Descripción
MA	Crítico	Requiere atención urgente.
A	Grave	Requiere atención.
M	Apreciable	Puede ser objeto de estudio para su tratamiento.
B	Asumible	No se van a tomar acciones para atajarlo.

Nota. Adaptado de MAGERIT (2012)

iv. Salvaguardas

Las salvaguardas o contra medidas se pueden definir como los procedimientos o mecanismos tecnológicos que se implementan para reducir el riesgo. Hay amenazas que requieren simplemente una organización adecuada, otras requieren elementos técnicos (programas o equipos), otras necesitan seguridad física y, finalmente, está la política del personal (MAGERIT, 2012). En la Tabla 11 se resumen los tipos de salvaguardas.

Tabla 11

Tipos de Salvaguardas

Efecto	Tipo
Reducen la probabilidad	Preventivas, disuasorias y eliminatorias
Acotan la degradación	Minimizadoras, correctivas y recuperativas
Consolidan el efecto de las demás	De monitorización, de detección, de concienciación y administrativas

Nota. Tomado de MAGERIT (2012)

2.7.Simuladores y Emuladores SDN

Debido a que en SDN la operación e inteligencia de la red están centralizadas en un controlador, la exactitud y la eficiencia de las funciones implementadas por el controlador deben ser probadas antes de su uso en un ambiente real (Valencia et al., 2015), para ello existe una variedad de herramientas. Actualmente los simuladores más utilizados para la evaluación de SDN son: Mininet, EstiNet y NS-3 (Torres & Zuñiga, 2020).

- **Mininet**

Mininet es un emulador de red en el que se puede crear redes virtuales con hosts, conmutadores, controladores y enlaces. Los hosts Mininet ejecutan software de red Linux estándar, y sus conmutadores soportan el protocolo OpenFlow para un enrutamiento basado en SDN. Las redes Mininet ejecutan código real, incluidas las aplicaciones de red estándar de Unix / Linux, así como el núcleo real de Linux y la pila de red. Gracias a esto, el código desarrollado en Mininet de un controlador OpenFlow, conmutador o host puede pasar a un sistema real con cambios mínimos, para pruebas, evaluación de rendimiento e implementación (Mininet Team, 2018).

- **EstiNet**

EstiNet es un simulador y emulador que utiliza el enfoque “kernel re-entering” para probar las funciones y el rendimiento de los controladores OpenFlow. En una red simulada con esta herramienta, cada host es capaz de ejecutar el sistema operativo Linux real, y

cualquier programa de aplicación real basado en UNIX. Otra característica destacable es que en EstiNet no es necesario que la simulación se ejecute en tiempo real, sino que puede ser más rápida o más lenta. Esta capacidad le permite simular correctamente el rendimiento de una red compuesta por una gran cantidad de conmutadores OpenFlow y hosts(Torres & Zuñiga, 2020).

- **NS-3**

Es un simulador basado en eventos discretos, destinado principalmente para la investigación y educación. Implementa una amplia gama de protocolos de redes cableadas y redes inalámbricas. La versión actual está diseñada para soportar todo el flujo de trabajo de la simulación desde la configuración hasta la recolección y análisis de tramas. Permite simular el funcionamiento de un controlador OpenFlow real implementado en C++ (Torres & Zuñiga, 2020).

En la Tabla 12 se muestra una tabla comparativa con las principales características de las herramientas de simulación y emulación mencionadas.

Tabla 12

Características de Simuladores de SDN

Características	Mininet	EstiNet	NS-3
Versión	2.2.1	9.0	3.25
Lenguaje de programación	Python	Sin información	C++ y Python
Sistema Operativo	Ubuntu	Fedora	Linux, FreeBSD y MAC OS
Desarrollador	Universidad de Standford	EstiNet Technologies Inc.	Ns-3 Project
Versión de OpenFlow	1.0, 1.1, 1.2, 1.3	1.0, 1.3	0.8.9, 1.3

Características	Mininet	EstiNet	NS-3
Compatibilidad con controladores reales	Si	No	Si
Soporte GUI	Solo para observación	Solo para observación	Para configuración y observación
Facilidad de uso	Alta	Alta	Media
Escalabilidad	Media	Alta	Alta
Licencia	Código abierto	Propietaria	Código abierto
Documentación	Buena	Media	Buena
Rendimiento	Medio	Alto	Alto

Nota. Basado en Valencia (2015) y Torres (2020)

CAPITULO III

3. MARCO METODOLÓGICO

En esta sección se describe el procedimiento para responder a la problemática planteada en la presente investigación.

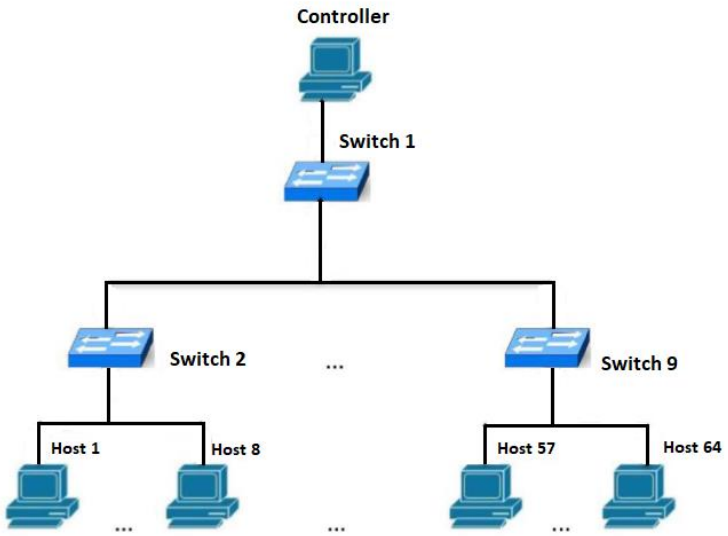
3.1.Descripción del Área de Estudio

El área de estudio de investigación corresponde a un entorno de simulación controlado, en donde se implementará una SDN. Para ejecutar las pruebas se usará la herramienta Mininet, ya que es la herramienta de emulación de red estándar para SDN (Mousavi & St-hilaire, 2016).

Para la definición del escenario de prueba se ha tomado como referencia la topología de red utilizada en los trabajos de Kia (2016) y Sahoo (2017), que consiste en una estructura tipo árbol de profundidad de dos niveles y ocho de fanout que crea sesenta y cuatro hosts, como se muestra en la Figura 12. Se elige este tipo de topología en función del hecho de que es una estructura de red ampliamente utilizada en los centros de datos (Kia, 2015).

Figura 12

Topología de Red



3.2.Diseño y Tipo de Investigación

El enfoque asumido para abordar el problema de investigación es cuantitativo. Se realizará una interpretación de datos cuantitativos obtenidos de la simulación que se llevará a cabo, así como la validación de la metodología propuesta.

Se utilizarán los siguientes tipos de investigación:

- **Investigación Documental.** Porque se investiga todo lo relacionado al entorno de investigación, para de esta manera conocer más sobre el tema y a la vez tener una visión más amplia de todos los factores que involucran las variables que permitirá realizar la investigación y el análisis para lograr el objetivo planteado.

- **Investigación Aplicada.** Porque busca la aplicación o utilización de conocimientos, desde una o varias áreas especializadas, con el objetivo de implementarlos de forma práctica para satisfacer necesidades concretas, proporcionando una solución a problemas del sector social o productivo.

3.3. Procedimiento de Investigación

Se utilizarán las siguientes técnicas:

- Matriz de revisión bibliográfica: Para realizar la investigación documental, es decir, recopilar información ya existente relacionada a la presente propuesta.

- Ficha de observación: instrumentos de investigación, evaluación y recolección de datos que se van a obtener de la simulación.

3.4.Consideraciones Bioéticas

No se tienen consideraciones bioéticas debido a que esta investigación no realizará ninguna modificación o experimentación con elementos naturales y/o su información genética.

3.5.Diseño de la Metodología

El desarrollo de la metodología se realizó tomando como referencia las secciones de la norma ISO 27001, en concordancia con las fases del ciclo de Deming para tener un conjunto lógico de pasos secuenciales que permitan alcanzar el objetivo de la metodología.

Para la fase de diseño se toma como referencia el trabajo de Robalino (2018), en cuyo diseño metodológico se toman en cuenta los siguientes puntos:

- **Definición del alcance:** Especifica los objetivos para el proceso e identifica a quienes se benefician de él.
- **Criterios para la elaboración de subprocesos:** Establece criterios y determina los resultados que se utilizarán para evaluar la calidad del proceso.
- **Inventario de información y recursos:** Reúne experiencia en el uso del proceso objetivo, incluida la calidad, cantidad, puntualidad y el costo de la información relevante.
- **Orden lógico de subprocesos:** Organiza el proceso en pasos, incluyendo la retroalimentación cuando aplica.

i. Definición del Alcance

En este trabajo se pretende desarrollar una metodología para la aplicación de un mecanismo de detección y mitigación de ataques DDoS dirigidos al plano de control de SDN. Al ser una tecnología prácticamente nueva, las SDN y sus respectivos problemas de seguridad no tienen las propuestas metodológicas necesarias para su implementación.

La estructura de la norma se basará en las secciones obligatorias definidas en la norma ISO 27001, mismas que concuerdan también con el ciclo de mejora continua o ciclo de Deming. El guiarse en estos estándares avalados internacionalmente asegura una metodología completa y ordenada que pueda ser aplicada en cualquier tipo de organización.

El desarrollo de la metodología iniciará identificando los riesgos y vulnerabilidades técnicas del controlador SDN, para luego establecer una solución adecuada de acuerdo con los requerimientos que demande la solución. La solución que se considere se implementará en

un entorno de simulación controlado para recopilar datos, determinar y corregir falencias en la metodología para finalmente validarla.

Como resultado de la implementación de esta metodología se obtendrá una guía para detectar y mitigar este tipo de incidentes de seguridad. Específicamente será una guía que al cumplirse puntualmente ayudará a los profesionales de las áreas tecnológicas y especialmente de la seguridad de la información, a mejorar la seguridad del plano de control de SDN, que es el punto más crítico en este tipo de arquitecturas.

ii. Criterios para la Elaboración de Subprocesos

Debido a que la implementación de un método para solventar la problemática planteada es un punto crucial en la metodología, es necesario definir los subprocesos necesarios para realizar la implementación de la manera óptima.

Los subprocesos que se detallan a continuación se basan en las secciones obligatorias de la norma ISO 27001, se toman en cuenta todas las actividades necesarias para llevar a cabo el desarrollo de la metodología propuesta y se omiten actividades que no se consideran necesarias para alcanzar los objetivos de esta investigación y que quedan fuera del alcance.

- **Identificación de Riesgos**

Este subproceso corresponde a revisar la información acerca del controlador SDN de la red de datos de la organización para la identificación de amenazas, vulnerabilidades y riesgos que se encuentren respecto a posibles ataques DDoS.

- **Planificación**

En el subproceso *Planificación s*e establecen los objetivos de seguridad del controlador SDN respecto a la protección contra ataques DDoS, para conforme a ello realizar una planificación que dicte la manera en que se llegará a estos objetivos. El documento resultante de este subproceso será fundamental para el cumplimiento de todos los subprocesos siguientes.

- **Selección del mecanismo**

Al existir distintos métodos para la detección y mitigación de ataques DDoS, los principales deben ser analizados. Este subproceso tiene el objetivo de determinar el mecanismo más idóneo para ser implementado como solución.

- **Pruebas**

Se implementa el mecanismo seleccionado en un escenario de prueba lo más apegado a la realidad de la red de la organización, para verificar que el funcionamiento de la solución corresponda a lo planeado. Esta fase de prueba y verificación ayuda a los ingenieros a que evidencien posibles falencias que se deben corregir antes de realizar la implementación en el ambiente de producción.

- **Implementación**

Se implementa en el ambiente de producción la solución elegida y probada, el proceso se realiza de acuerdo con la planificación, para cumplir con los objetivos de seguridad del controlador SDN.

(Al no contar con un escenario real este subproceso se llevará a cabo en un ambiente de simulación controlado)

- **Monitoreo**

En este subproceso se pone a prueba el correcto funcionamiento de la solución implementada. Se recopila toda la información necesaria para evaluar el sistema mediante el monitoreo de distintos factores que se consideren relevantes. Pasado el tiempo de prueba se evalúa si los resultados del monitoreo cumplen satisfactoriamente con los objetivos planteados.

- **Mejora**

Si la solución resulta ser satisfactoria se implementa de manera definitiva. Por otra parte, en caso de encontrar falencias o inconformidades respecto a los objetivos planteados, es necesario realizar las correcciones pertinentes o descartar la solución. Las correcciones realizadas en este subproceso también deben ser validadas.

Una vez terminado el proceso, se debe volver al primer paso de manera periódica para estudiar nuevas mejoras a implementar.

iii. Inventario de Información y Recursos

En este apartado se determinan las entradas (recursos o insumos) necesarias para ejecutar cada uno de los subprocesos y las salidas, que corresponden al resultado que se desea obtener tras la realización de los subprocesos. Los recursos y resultados definidos para cada

subproceso se determinan en base a la norma ISO 27001 y se muestran de manera resumida en la Tabla 13.

Tabla 13

Subprocesos, Recursos y Resultados de la Metodología Propuesta

SUBPROCESO	RECURSOS	RESULTADO
Identificación de riesgos	Información del Controlador SDN respecto a la seguridad	Documento de análisis de riesgos
Planificación	Documento de análisis de riesgos	Documento de Planificación
Selección del mecanismo	Documento de análisis de riesgos Documento de Planificación	Mecanismo a implementar
Pruebas	Mecanismo a implementar Documento de planificación	Resultado de pruebas
Implementación	Mecanismo a implementar Documento de planificación Resultado de pruebas	Implementación del mecanismo de detección y mitigación de DDoS
Monitoreo	Implementación del mecanismo de detección y mitigación de DDoS Documento de planificación	Reporte de desempeño
Mejora	Documento de planificación Reporte de desempeño Implementación del mecanismo de detección y mitigación de DDoS	Aprobación de la solución

En la Tabla 13 se presenta cada subproceso de la metodología a desarrollar con sus respectivas entradas y salidas. Debido a que se trata de una guía secuencial el producto de cada subproceso se convierte en un recurso del siguiente subproceso. Los insumos mencionados se describen a mejor detalle en el apartado 3.6.

- **Identificación de Riesgos:**

  Se debe obtener la información sobre las vulnerabilidades técnicas de los sistemas de información y evaluar que tan expuestos están. Como esta metodología busca proteger al controlador SDN, este es el activo a evaluar. De este subproceso se obtiene el "Análisis de riesgos del controlador" a partir de la Información del Controlador SDN respecto a la seguridad.

- **Planificación:**

  La planificación parte del resultado del subproceso anterior "Análisis de riesgos del controlador", adicionalmente se requiere de fijar objetivos respecto a la protección del controlador SDN frente a ataques DDoS. Con esta información se obtiene el "Documento de Planificación".

- **Selección del Mecanismo:**

  El "Documento de planificación" proporciona los lineamientos para elegir el mecanismo más idóneo para abordar la problemática de la mejor manera posible. El resultado de este subproceso es el "Mecanismo a implementar".

- **Pruebas:**

  Corresponde a la verificación del mecanismo en un ambiente de simulación. Los insumos para este subproceso son: "Documento de planificación" y "Mecanismo a implementar". Con la ejecución de todas las pruebas que se consideren necesarias, se obtiene el "Resultado de pruebas".

- **Implementación:**

  Con el "Documento de planificación", "Mecanismo a implementar" y "Resultado de pruebas" de subprocesos anteriores se obtiene como resultado la "Implementación del mecanismo de detección y mitigación de DDoS".

- **Monitoreo:**

  Requiere como entrada los insumos: “Implementación del mecanismo de detección y mitigación de DDoS” y “Documento de planificación” de subprocesos anteriores. El resultado de este subproceso es el “Reporte de desempeño”

- **Mejora:**

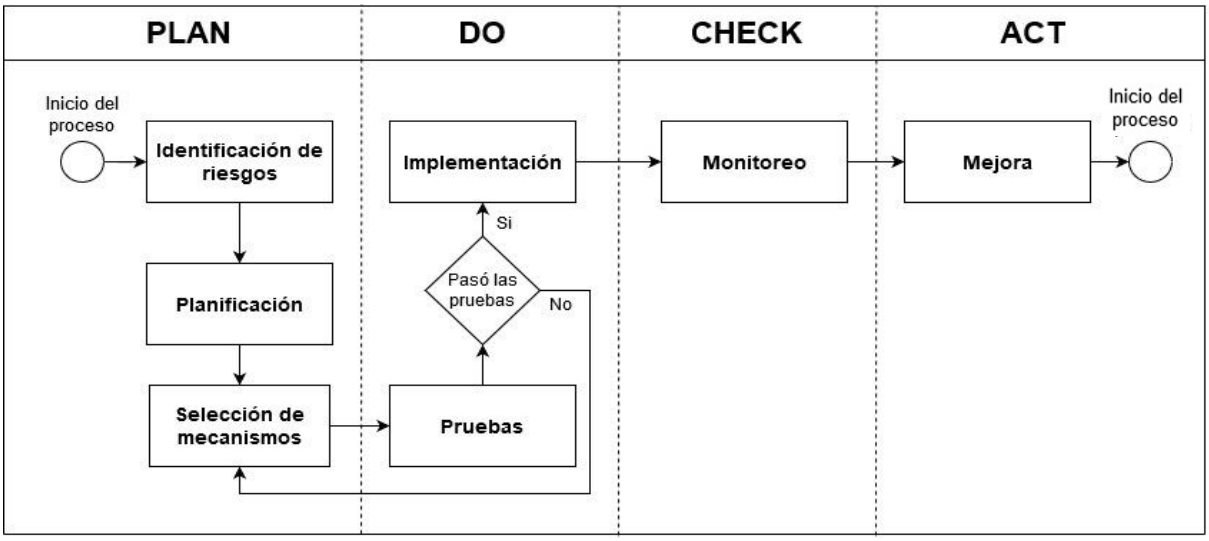
  Requiere de los insumos: “Implementación del mecanismo de detección y mitigación de DDoS”, “Reporte de funcionamiento” y “Documento de planificación” de subprocesos previos. Si el resultado de la mejora es satisfactorio se tiene como salida del subproceso la “Aprobación de la solución”. Una vez finalizado el proceso se vuelve al primer paso conforme lo dicta el proceso de mejora continua del ciclo de Deming.

iv. Orden Lógico de Subprocesos

Los subprocesos definidos en el apartado ii siguen un orden lógico y secuencial, dado que el resultado de un subproceso se convierte en un recurso del proceso siguiente. En la Figura 13 se presenta el flujograma correspondiente a la metodología propuesta, señalando también a que fase del PDCA corresponden.

Figura 13

Orden Lógico de Subprocesos



3.6. Metodología de Detección y Mitigación de Ataques DDoS Dirigidos al Plano de Control de SDN

A continuación, se desarrollan cada uno de los subprocesos de la metodología, así como las actividades a realizar. La Figura 14 muestra un diagrama modelo para representar gráficamente el subproceso, actividades, elementos de entrada y elementos de salida.

Entradas: Insumos o recursos que se requieren para cada subproceso.

Actividades: Acciones que se deben cumplir en cada subproceso.

Salidas: Objetivo(s) a alcanzar por cada subproceso.

Figura 14

Plantilla Para el Diagrama de Subprocesos

NOMBRE DEL SUBPROCESO		
ENTRADAS	ACTIVIDADES	SALIDAS
Recursos/insumos para el subproceso	Actividades del subproceso	Resultados del subproceso

A continuación, se establece la metodología partiendo del primer subproceso y sus respectivos elementos:

3.6.1. Identificación de Riesgos

Este subproceso corresponde al primer paso de la metodología, la Figura 15 muestra el flujograma del subproceso con sus respectivas entradas, actividades y salidas.

Figura 15

Flujograma del Subproceso "Identificación de Riesgos"

Entradas:

- Información del Controlador SDN respecto a la seguridad.

Actividades:

- **Definir los activos a evaluar**

Se definen los activos de información asociados al plano de control SDN utilizando el modelo presentado en la Tabla 14. La dimensión en las que se deben evaluar los activos es la disponibilidad, pues es donde afectan los ataques de DDoS.

Tabla 14

Tabla Modelo "Activos a Evaluar"

Activo	Descripción	Tipo	Persona responsable	Ubicación

La valoración de Degradación, Probabilidad, Impacto y Riesgo se realiza conforme las métricas señaladas en el apartado 2.6

- **Determinar las vulnerabilidades y amenazas por cada activo**

Determinar las vulnerabilidades del plano de control frente a ataques de DDoS. Valorar la degradación del valor de los activos usando los criterios de la Tabla 6, en base a la categoría 1.

Valorar la probabilidad de ocurrencia de la amenaza usando los criterios definidos en la Tabla 7, en base a la categoría 2.

Utilizar el modelo de la Tabla 15 para resumir esta información.

Tabla 15

Tabla Modelo "Vulnerabilidades Por Cada Activo"

Activo	Vulnerabilidad	Amenaza	Degradación	Probabilidad

- **Estimar el impacto**

Conociendo el valor de los activos y la degradación provocada por las amenazas se determina el impacto en base a la Tabla 8. Utilizar el modelo presentado en la Tabla 16 para resumir esta información.

Tabla 16

Tabla Modelo "Estimación del Impacto"

Activo	Valor	Amenaza	Degradación	Impacto

- **Estimar el riesgo**

Estimar el riesgo en base al Impacto y la Probabilidad como se realiza en la Tabla 9. Utilizar el modelo presentado en la Tabla 17 para resumir esta información.

Tabla 17

Tabla Modelo "Estimación del Riesgo"

Activo	Amenaza	Impacto	Probadilidad	Riesgo

- **Calificación del riesgo**

Calificación del riesgo de acuerdo a la Tabla 10 para determinar si :

- Es **crítico** en el sentido de que requiere atención urgente.
- Es **grave** en el sentido de que requiere atención.
- Es **apreciable** en el sentido de que pueda ser objeto de estudio para su tratamiento.
- Es **asumible** en el sentido de que no se van a tomar acciones para atajarlo.

Estas actividades se enmarcan en el control A.12.6.1 (Gestión de vulnerabilidades técnicas) de la norma ISO 27001.

Salidas:

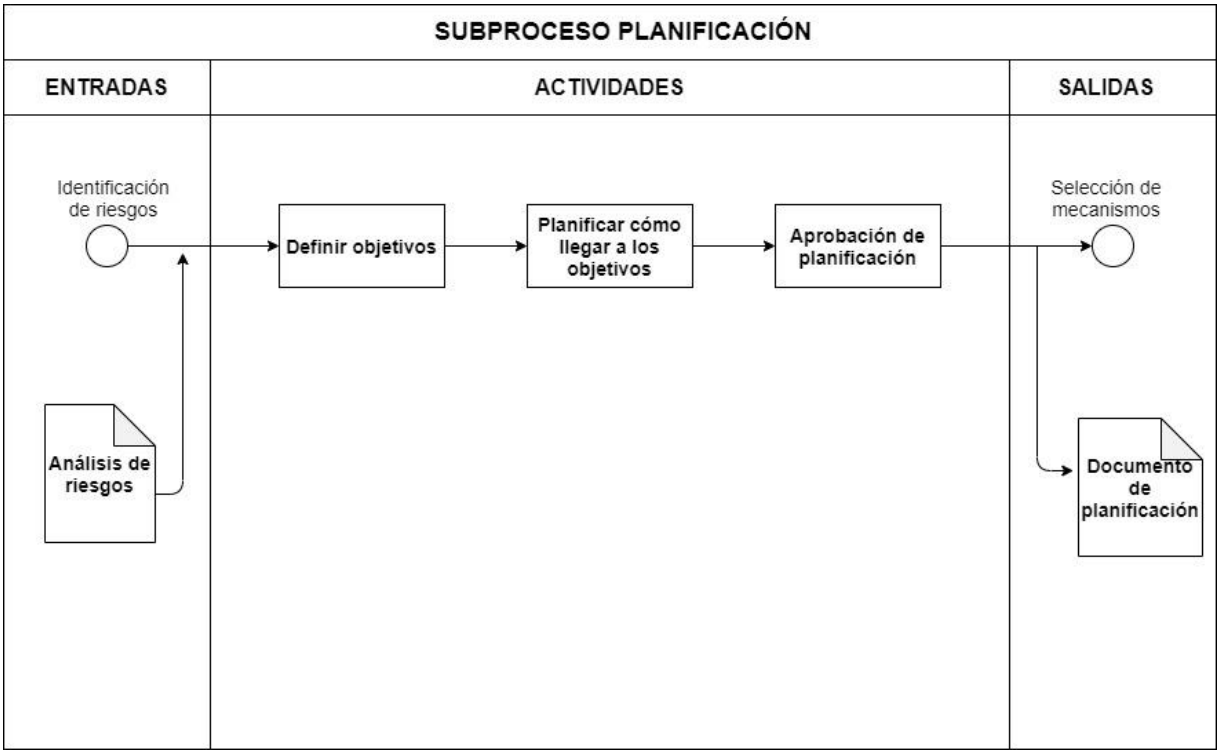
- Documento de análisis de riesgos del controlador.

3.6.2. Planificación

El resultado del subproceso anterior es uno de los recursos necesarios para el subproceso *Planificación.* La Figura 16 muestra este subproceso gráficamente.

Figura 16

Flujograma del Subproceso "Planificación"



Entradas:

- Documento de análisis de riesgos.

Actividades:

- **Definir qué se hará respecto a los riesgos**

  De acuerdo a la calificación de los riesgos, determinar si es necesario tomar salvaguardas o contra medidas.

  Si se decide que no es necesario tomar medidas respecto a los riesgos, el proceso termina en este paso con la aceptación del riesgo y su respectiva justificación.

Si los riesgos requieren acciones, se debe elegir la o las salvaguardas que se consideren necesarias. La Tabla 11 resume los tipos de salvaguardas que se pueden tomar según la metodología MAGERIT.

- **Definir objetivos de seguridad:**
  Plantear objetivos de seguridad de la información para tratar los riesgos encontrados. Estos objetivos deben ser medibles en lo posible.

- **Planificar como cumplir los objetivos:**
  Realizar una planificación que indique cómo se llegará a los objetivos, respondiendo principalmente a las siguientes interrogantes:
  - ¿Qué recursos son necesarios?
  - ¿Quién será el responsable?
  - ¿Cuándo se finalizará?
  - ¿Cómo se evaluará el cumplimiento de los objetivos?

- **Aprobación de la planificación:**
  Conseguir que la dirección conozca y apruebe la planificación.

Las actividades de este subproceso se fundamentan en el control A.13.1.1 (Controles de red) de la norma ISO 27001.

Salidas:

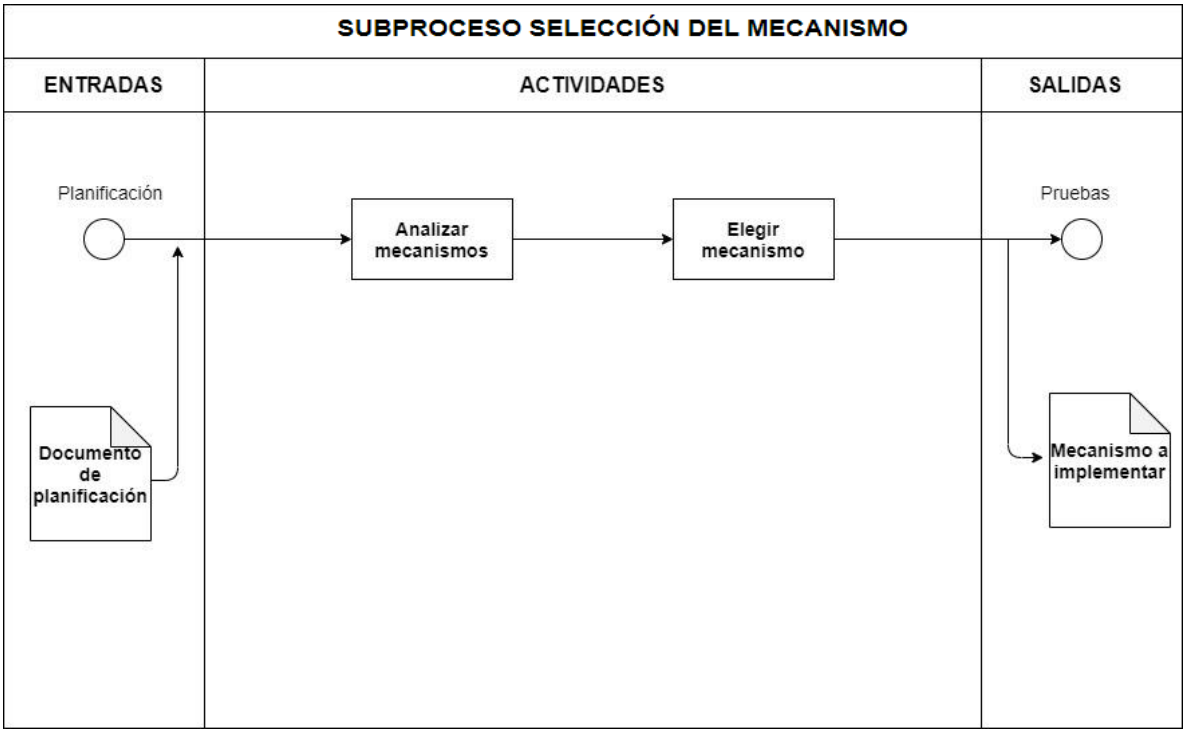
- Documento de Planificación.

3.6.3. Selección del Mecanismo

Este subproceso, representado en la Figura 17, requiere del resultado del anterior subproceso *Planificación* y otros insumos que se detallan a continuación:

Figura 17

Flujograma de Subproceso "Selección del Mecanismo"



Entradas:

- Documento de Planificación.

Actividades:

- **Análisis de mecanismos de seguridad:**
  Analizar los mecanismos más utilizados para la detección y mitigación de ataques DDoS dirigidos al controlador, determinar sus características principales.

- **Elección del mecanismo:**
  Elegir de entre las opciones, la solución que ayude a cumplir los objetivos de seguridad del controlador SDN.

Las actividades de este subproceso se fundamentan en el control A.13.1.2 (Seguridad de los servicios de red) de la norma ISO 27001.

Salidas:

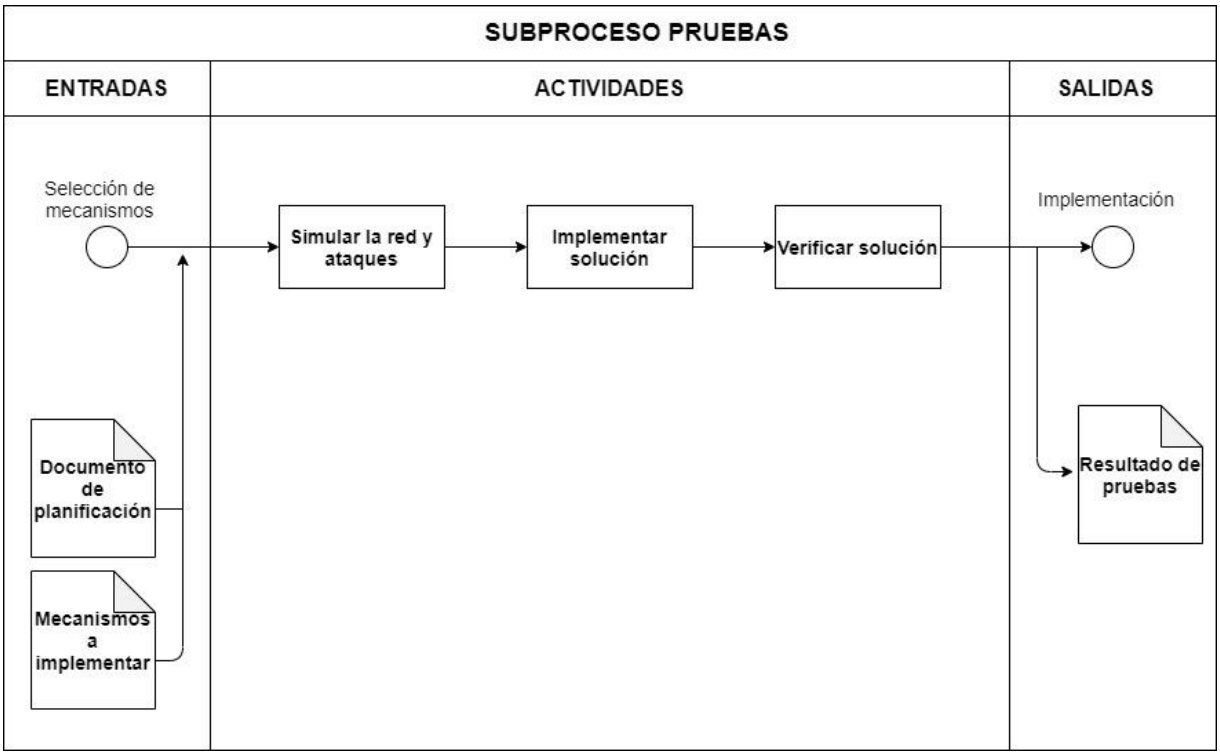
- Mecanismo a implementar.

3.6.4. Pruebas

En este subproceso se busca probar el mecanismo seleccionados en un ambiente de simulación. La Figura 18 muestra gráficamente el subproceso “Pruebas”.

Figura 18

Flujograma de Subproceso "Pruebas"



Entradas:

- Documento de planificación.
- Mecanismo a implementar.

Actividades:

- **Simular la red de datos junto con los ataques:**

Utilizar un software de simulación para representar los elementos de la red de datos de la organización. Utilizar las herramientas necesarias para simular ataques DDoS dirigidos hacia el controlador SDN en el ambiente de simulación.

- **Implementar la solución:**
  Implementar el mecanismo de detección y mitigación elegidos en el subproceso anterior con todas las configuraciones que sean necesarias.

- **Verificar la solución:**
  Verificar que el mecanismo detecte y mitigue los ataques en el escenario de simulación. En caso de que esto no suceda, se debe volver al subproceso *Selección del mecanismo.*

Las pruebas se enmarcan en el control A.14.2.8 (Pruebas de seguridad del sistema) y A.14.2.9 (Pruebas de aprobación de sistemas) de la norma ISO 27001.

Salida:

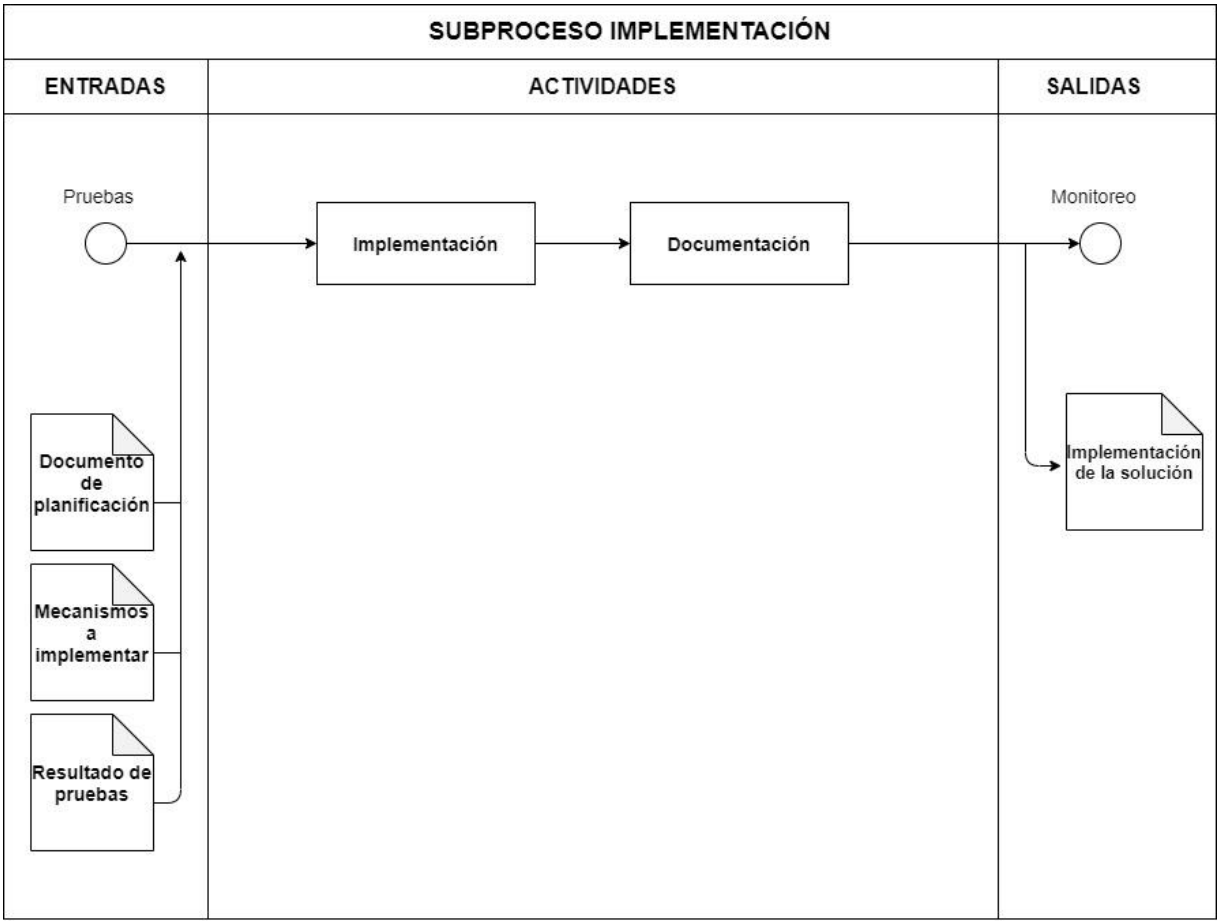
- Resultado de pruebas

3.6.5. Implementación

El objetivo de este subproceso es realizar la implementación de la solución elegida dentro de la infraestructura de la organización. La Figura 19 muestra gráficamente el subproceso "Implementación".

Figura 19

Flujograma de Subproceso "Implementación"



Entradas:

- Documento de planificación.
- Mecanismo a implementar.

Actividades:

- **Implementación:**

  Implementar el mecanismo de seguridad en el escenario real teniendo en cuenta el procedimiento realizado en las pruebas de simulación.

- **Documentación:**

  Documentar la implementación y configuración del mecanismo elegido. Es necesario que cuente también con la siguiente información:

- Nombres del personal a cargo de la implementación.
- Firmas de responsabilidad.
- Fecha de implementación.

La implementación se enmarca en el control A.13.1.1 (Controles de red) de la norma ISO 27001.

Salidas:

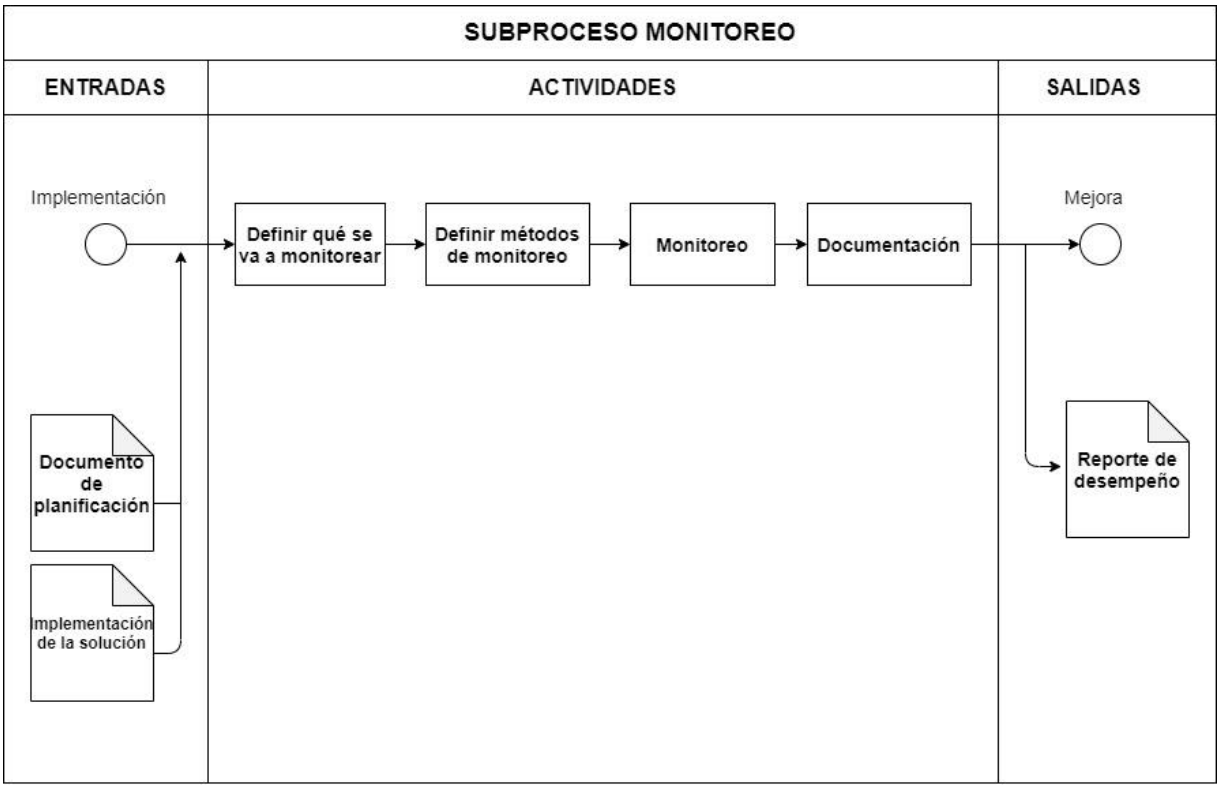
- Implementación del mecanismo de detección y mitigación de DDoS.

3.6.6. Monitoreo

Este subproceso busca poner a prueba la solución. La Figura 20 muestra los elementos necesarios de este subproceso.

Figura 20

Flujograma del subproceso "Monitoreo"



Entradas:

- Documento de planificación.
- Implementación del mecanismo de detección y mitigación de DDoS.

Actividades:

- **Definir qué es necesario monitorear:**

  Seleccionar los factores que determinen el nivel de efectividad de la solución implementada.
- **Definir los métodos/detalles de monitoreo:**

  Definir cómo se llevará a cabo el monitoreo dentro de los siguientes aspectos:
  - ¿Cuándo se llevan a cabo los monitoreos?
  - ¿Quién debe monitorear?
  - ¿Cuándo analizar y evaluar los resultados?
  - ¿Quién debe evaluar y analizar los resultados?
- **Documentación**

  Documentar los resultados del monitoreo, no conformidades, y el nivel de cumplimiento de objetivos.

La respuesta a incidentes de seguridad se enmarca en el control A.16.1.5 de la norma ISO 27001.

Salidas:

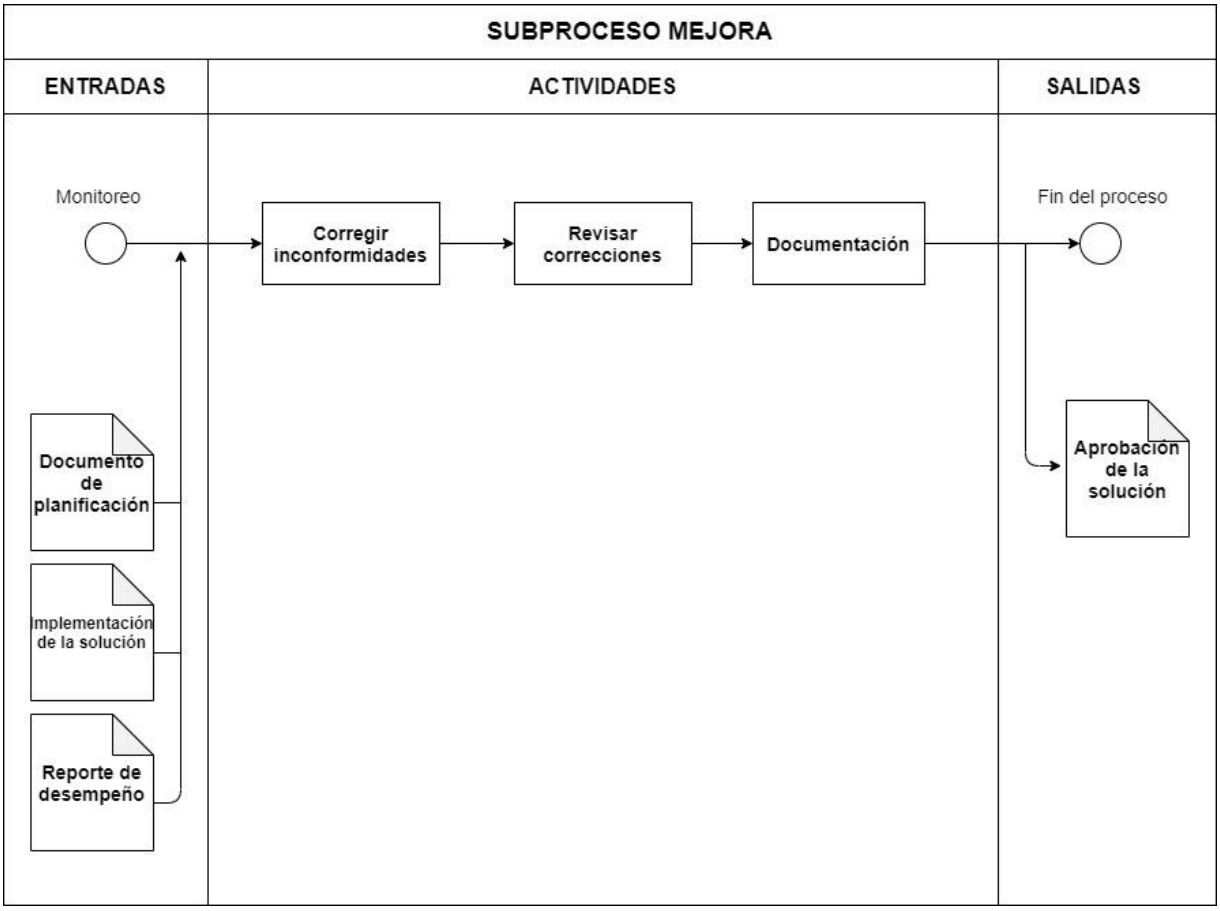
- Reporte de desempeño.

3.6.7. Mejora

Este subproceso tiene el objetivo de corregir las no conformidades que pueden identificarse tras el *Monitoreo.* Como subproceso final, se busca también la aprobación de la solución. La Figura 21 muestra gráficamente el subproceso “Mejora”.

Figura 21

Flujograma de Subproceso "Mejora"



Entradas:

- Implementación del mecanismo de detección y mitigación de DDoS.
- Reporte de desempeño.
- Objetivos de seguridad.

Actividades:

- **Corregir inconformidades:**

  Realizar las configuraciones necesarias para controlar y corregir las inconformidades encontradas.

- **Revisar las correcciones:**

  Revisar la eficacia de las acciones correctivas realizadas respecto a las no conformidades.
- **Documentar:**

  Documentar la naturaleza de las no conformidades, las acciones llevadas a cabo y sus resultados.

Las mejoras se enmarcan en el control A.16.1.6 (Aprendizaje de los incidentes de seguridad de la información) de la norma ISO 27001.

Salidas:

- Documento de aprobación de la solución.

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

Para validar la metodología propuesta, esta se implementa en un ambiente de simulación. El software a utilizar es Mininet, una herramienta ampliamente usada en la simulación de arquitecturas SDN. El programa se ejecuta en una máquina virtual de VirtualBox con el sistema operativo Ubuntu.

Para la implementación del escenario de simulación se utilizó un computador con las características que se detallan en la Tabla 18:

Tabla 18

Características de Hardware

Característica	Detalle
Marca	Dell
Procesador	Intel(R) Core(TM) i7-3632QM CPU 2.20GHz
RAM	12,0 GB
Disco duro	1 TB mecánico y 256 GB estado sólido
Sistema Operativo	Windows 10
Red inalámbrica	Intel(R) Centrino(R) Wireless-N 2230

Por otra parte, dentro de las herramientas de software utilizadas en este trabajo se tiene:

- **VirtualBox**

En este trabajo se utiliza la versión más reciente de VirtualBox correspondiente a la 6.1.6 r137129 para alojar una máquina virtual con la última versión de Ubuntu hasta el momento, ya que es lo que se recomienda para la ejecución del software de simulación Mininet. Los detalles de la máquina virtual se observan en la Tabla 19.

Tabla 19

Características de la Máquina Virtual

Característica	Detalle
Nombre de la máquina virtual	Mininet
Tipo	Ubuntu
Versión	Ubuntu (64-bit) 20.04
Cantidad de memoria (RAM)	2048 MB
Configuración de Red	Adaptador puente
Tamaño de disco duro virtual	30 GB
Procesador(es)	1 CPU

- **Mininet**

Se utiliza la última versión de Mininet ya que es la herramienta de emulación de red estándar para SDN. Para este programa es recomendable usar lanzamientos de Ubuntu más recientes, ya que admiten versiones más nuevas de Open vSwitch (OVS).

- **PuTTY**

Se utiliza esta herramienta para acceder a la línea de comandos de la máquina virtual a través del puerto 22 desde la máquina física, su uso no es indispensable pero facilita la copia de códigos y scripts entre las dos máquinas.

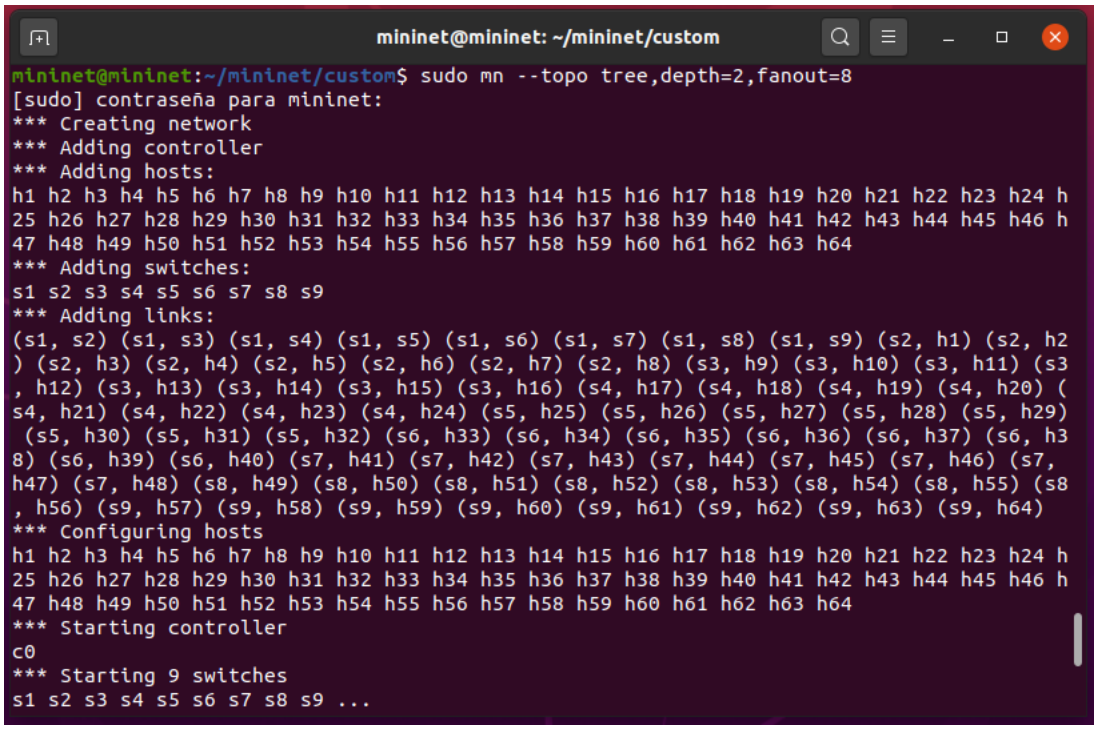
4.1.Simulación de la Topología de Red

Se implementa la topología de red mencionada en el apartado 3.1 que consiste en una estructura tipo árbol. Se elige este tipo de topología debido a que es una estructura de red ampliamente utilizada en los centros de datos y se ha utilizado para realizar investigaciones relacionadas al presente trabajo.

La topología planteada se crea con el comando *sudo mn –topo tree,depth=2,fanout=8*, mostrado en la Figura 22, en donde se genera una topología tipo árbol con los siguientes elementos de red: Un controlador (*c0*), nueve switches OVS (*s1, s2, s3…*) y sesenta y cuatro hosts (*h1, h2, h3…*). Automáticamente se generan los enlaces de red respectivos.

Figura 22

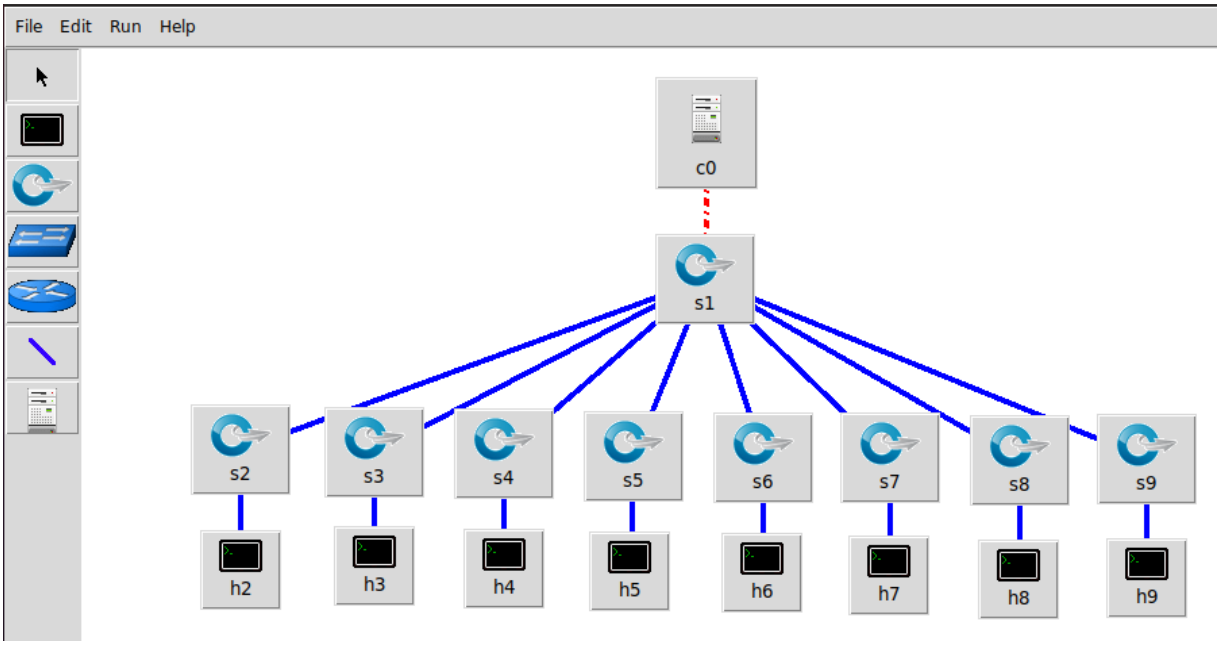
Creación de la Topología de Red



La topología también se puede generar a través de Mininet seleccionando cada uno de los elementos con sus respectivos enlaces de red. En la Figura 23 se muestra la topología de red en Mininet, en este caso, para visualizar de mejor manera se ha representado un solo host por switch de nivel 2.

Figura 23

Topología de Red en MiniEdit



Los elementos de red utilizados para representar la topología de red en MiniEdit se detallan en la Tabla 20.

Tabla 20

Elementos de Red en MiniEdit

Nombre	Gráfico	Descripción
Host		Host o terminal
Switch		Open vSwitch, que permite habilitar de forma programable las funciones de forwarding o transmisión de la información.
Controlador		Controlador de OpenFlow. También se pueden configurar otro tipo de controladores modificando las propiedades o configuración de los mismos.

Nombre	Gráfico	Descripción
NetLink		Enlaces entre los elementos de la red.

Entre los controladores disponibles, se ha elegido utilizar el controlador POX para este trabajo, debido a que es un controlador rápido y ligero, además la mayoría de los autores lo han utilizado en sus investigaciones.

El controlador POX se instala automáticamente con la herramienta Mininet, en este proyecto se utiliza la misma máquina virtual como controlador OpenFlow, aunque también es posible ejecutarlo desde una máquina virtual diferente.

El controlador por defecto está configurado con la dirección IP 10.1.1.1 y puerto 6634, por lo que es necesario configurarlo con la dirección IP de la máquina que se desea que actúe como controlador. Esto se configura mediante el comando:

./pox.py openflow.of_01 --address=192.168.10.108 --port=6633

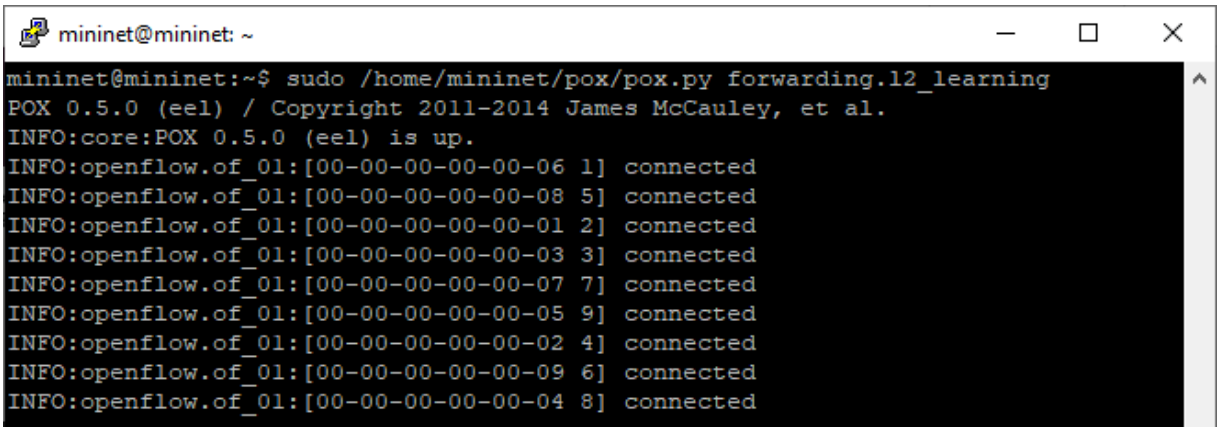
Luego es necesario activar mediante la línea de comandos al controlador POX con la ayuda del comando:

sudo /home/mininet/pox/pox.py forwarding.l2_learning

La Figura 24 muestra la activación exitosa del controlador con su respectiva versión y el resumen de los switches conectados a él. Sin este paso no puede existir convergencia en la red.

Figura 24

Activación del controlador POX



La Figura 25 es el resultado del comando *pingall* ejecutado en la línea de comandos de Mininet, se observa que todos los paquetes se han recibido exitosamente denotando la convergencia de la red. Este comando es una forma sencilla de comprobar la conectividad entre los hosts de la red, aunque también es posible usar el comando *ping* desde cada terminal.

Figura 25

Prueba de Conectividad Entre Hosts

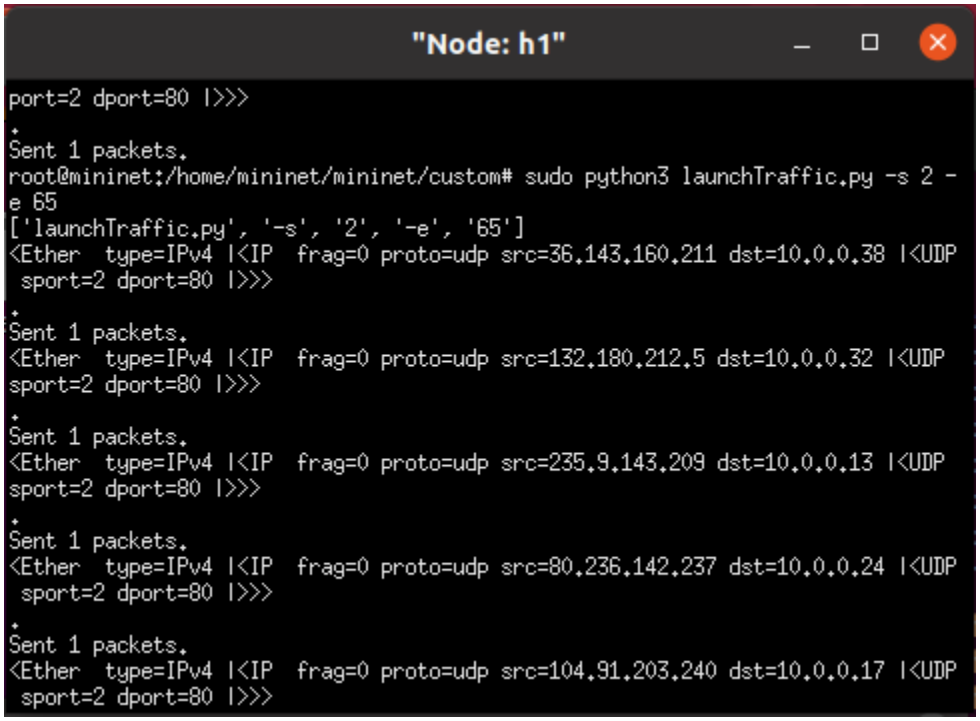
4.2.Generación de Tráfico

La herramienta utilizada para generar tanto el tráfico legítimo como de ataque es Scapy. Este programa se puede ejecutar desde una ventana del terminal y mediante ejecución de scripts. Para este proyecto se han programado dos scripts, uno para los flujos de tráfico normales o legítimos y otro para el flujo de ataque. El tipo de paquete elegido para el tráfico normal y de ataque es UDP.

Mininet asigna las direcciones IP de los hosts a partir de 10.0.0.1 de manera secuencial. Para el tráfico normal, la dirección IP de destino se genera en función del rango que se definió en el script (10.0.0.1 - 10.0.0.64) y las direcciones IP de origen se generan aleatoriamente utilizando la función aleatoria *randrange*. En la Figura 26 se muestra la generación del tráfico normal a través de la ejecución de un script desde un host.

Figura 26

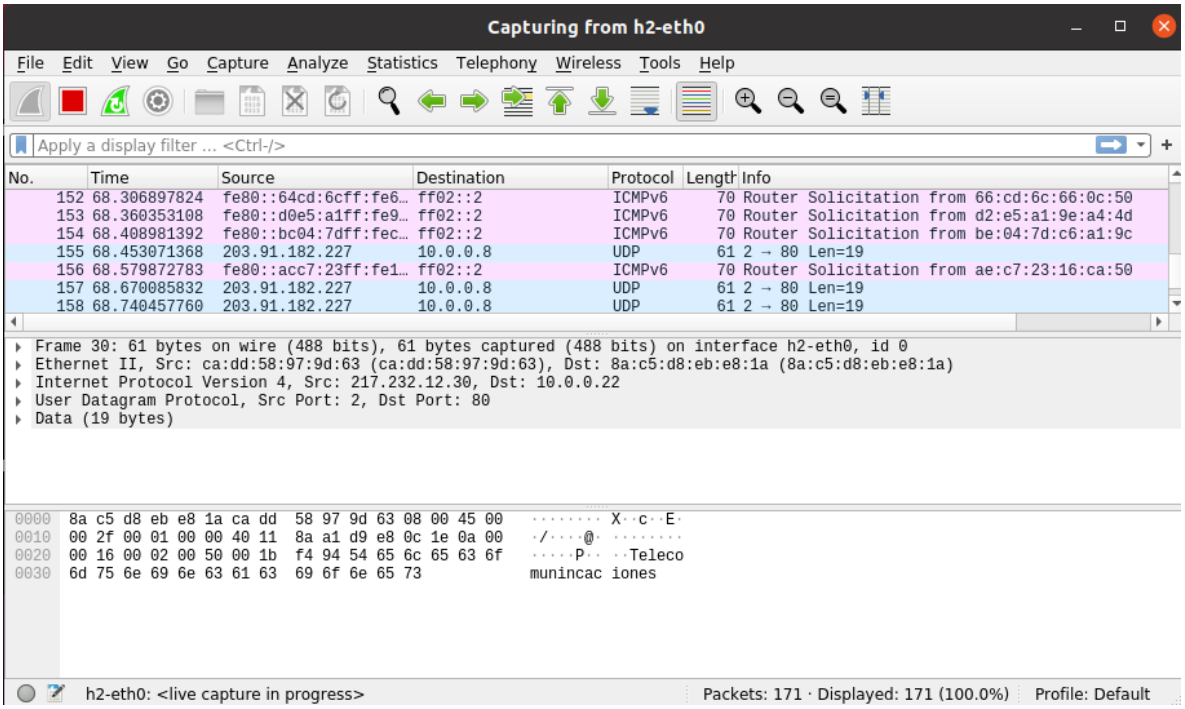
Generación de Tráfico Legítimo



En la Figura 27 se muestra la captura del tráfico legítimo desde otro host, se observa con la ayuda de Wireshark.

Figura 27

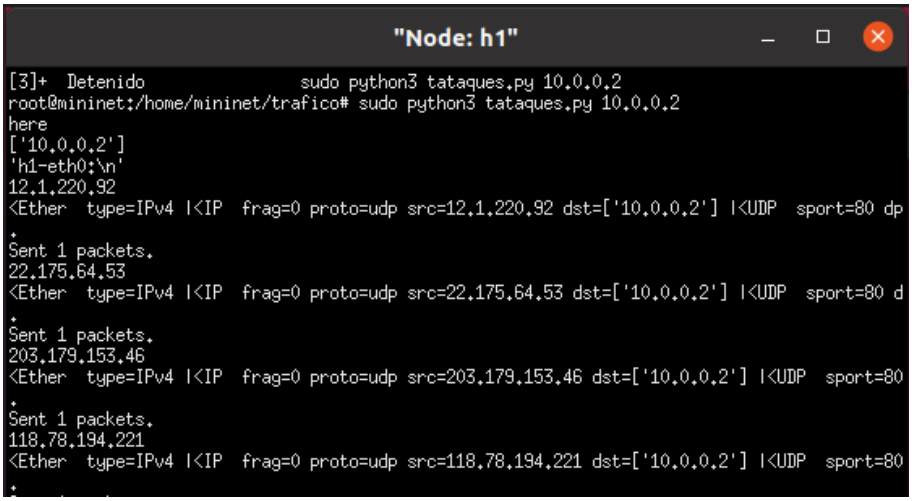
Captura del Tráfico Legítimo en Wireshark



Para ejecutar el ataque hacia un host en específico es necesario ejecutar el script de generación de paquetes con direcciones IP de origen aleatorias. La dirección IP de la víctima se especifica al ejecutar el comando como se muestra en la Figura 28.

Figura 28

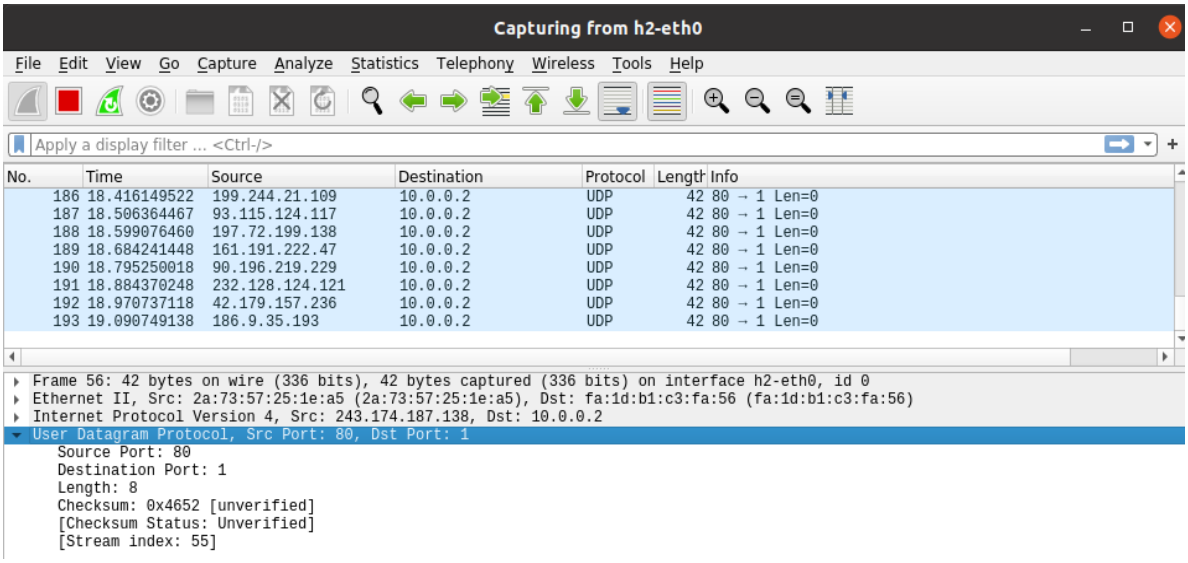
Generación del Tráfico de Ataque



La Figura 29 muestra la captura en Wireshark de los paquetes UDP de ataque dirigidos hacia el host con dirección 10.0.0.2. Las direcciones de origen son aleatorias y corresponden a los rangos fijados en el script.

Figura 29

Captura del Tráfico de Ataque en Wireshark



4.3.Aplicación de la Metodología

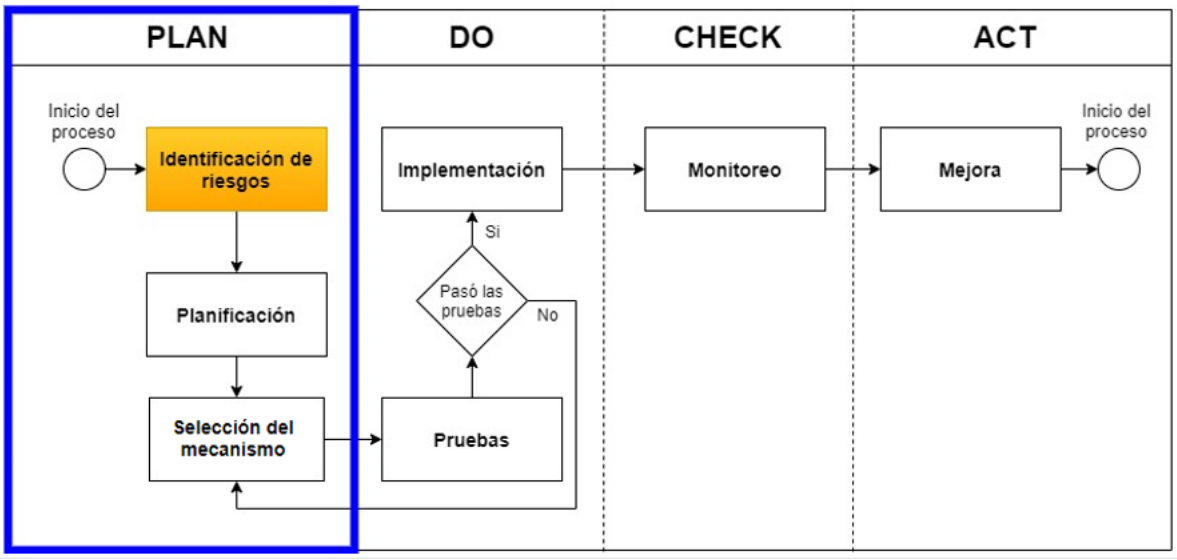
Con el escenario de simulación completamente funcional se procede a aplicar la metodología propuesta. A continuación, se aplica la metodología partiendo del primer subproceso y sus respectivos elementos:

4.3.1. Identificación de Riesgos

Este subproceso corresponde al primer paso dentro de la fase "PLAN" como se muestra en la Figura 30. El objetivo es realizar un análisis de riesgos del plano de control de SDN.

Figura 30

Avance de la Aplicación: Identificación de Riesgos



Entradas:

- Información del Controlador SDN respecto a la seguridad.

Actividades:

- Se definieron los activos a evaluar.
- Se determinaron las vulnerabilidades y amenazas por cada activo.
- Se estimó el impacto.
- Se estimó y calificó el riesgo.

Tras finalizar el análisis de riesgo al plano de control SDN se obtuvo el resultado mostrado en la Tabla 21.

Tabla 21

Resumen del Análisis de Riesgos

Activo	Amenaza	Riesgo	Calificación del riesgo
Servidor del controlador	Ataque de DDoS por inundación de paquetes	MA[1]	Crítico
Controlador	Ataque de DDoS por inundación de paquetes	MA	Crítico

Salida:

- Documento de análisis de riesgos del plano de control (ver ANEXO C).

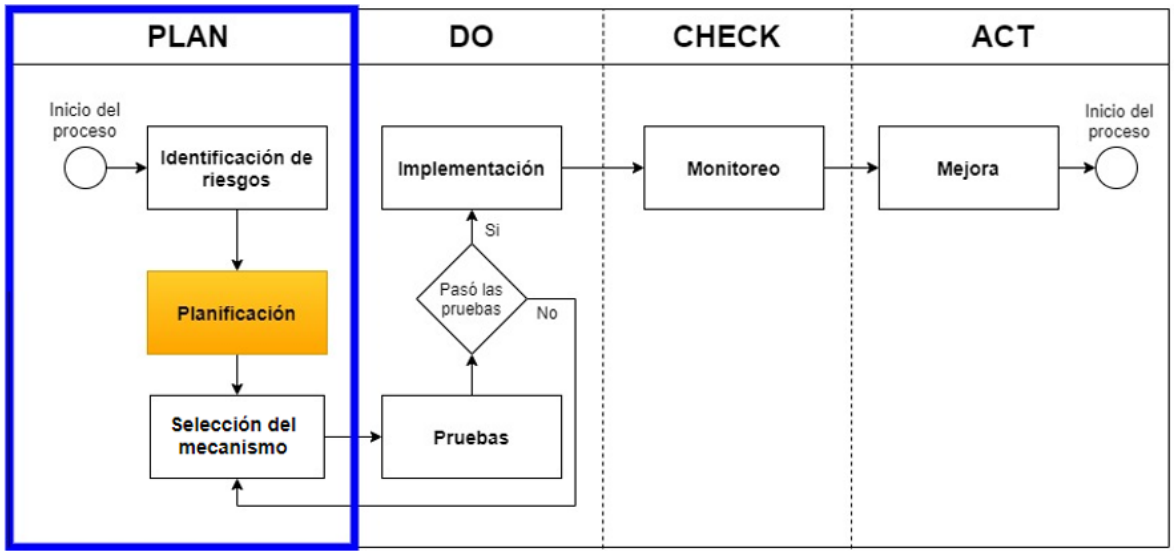
4.3.2. Planificación

La planificación es el segundo paso dentro de la fase "PLAN" (Ver Figura 31). El objetivo de este subproceso es definir cómo se tratarán los riesgos.

[1] Muy Alto

Figura 31

Avance de la Aplicación: Planificación



Entradas:

- Documento de análisis de riesgos.

Actividades:

- **Definir qué se hará respecto a los riesgos**

  Debido a que la calificación de los riesgos fue crítica se determinó que no era posible aceptar el riesgo, por lo que se debían tomar salvaguardas.

  El tipo de salvaguardas adecuadas para este caso, tomando en cuenta la metodología MAGERIT, son las de monitorización y detección.

- **Definir el objetivo de seguridad:**

  Se definió el siguiente objetivo:

  - Implementar un mecanismo de detección temprana y mitigación de ataques DDoS en el controlador SDN.

- **Planificar como cumplir el objetivo:**

  Para cumplir el objetivo se definieron los siguientes recursos:

- Documentación sobre mecanismos de detección y mitigación de ataques DDoS.
- Topología de red.
- Escenario de simulación controlado (con todos elementos que se presentan en la topología de red).
- Administrador de red y personal de TI.

Se designó al administrador de la red como responsable de velar por el cumplimiento del objetivo de seguridad.

Se definió como fecha máxima de implementación de la solución el día 24/06/2021

Salidas:

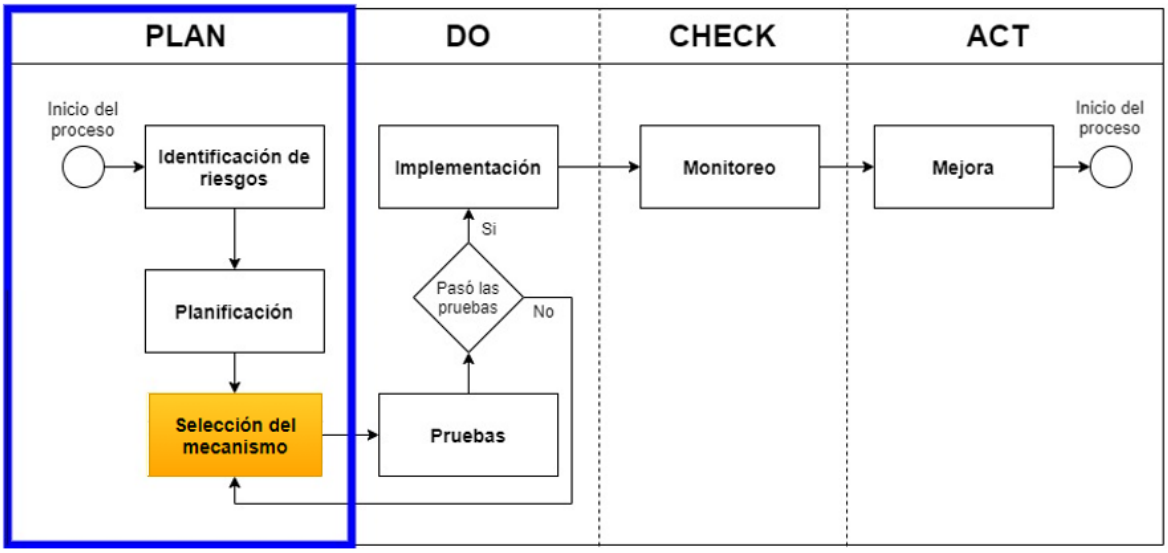
- Documento de Planificación (Ver ANEXO D).

4.3.3. Selección del Mecanismo

Como se indica en la Figura 32 la selección del mecanismo es el último paso dentro del a fase "Plan". El objetivo de ese subproceso es seleccionar la solución más adecuada para tratar el o los riesgos encontrados.

Figura 32

Avance de la Aplicación: Selección del Mecanismo



Entradas:

- Documento de Planificación.

Actividades:

- **Análisis de mecanismos de seguridad:**

  El análisis de mecanismos de detección y mitigación de ataques DDoS en el controlador SDN se realizó en la sección 2.3 y sus características principales se resumen en la Tabla 3.

- **Elección del mecanismo:**

  Se eligió el mecanismo basado en el cálculo de la entropía de la red, debido a que según las investigaciones realizadas tiene una alta efectividad de detección, además se ha implementado en topologías similares a la que se propone en esta investigación. Otro factor que se tomó en cuenta fue la documentación del mecanismo, pues son varias las investigaciones que se han realizado usando esta técnica y muchas de sus implementaciones están documentadas en la plataforma GitHub, a diferencia de los otros mecanismos cuya implementación no se describe de manera detallada, por lo que es muy difícil ponerlas en práctica.

Salidas:

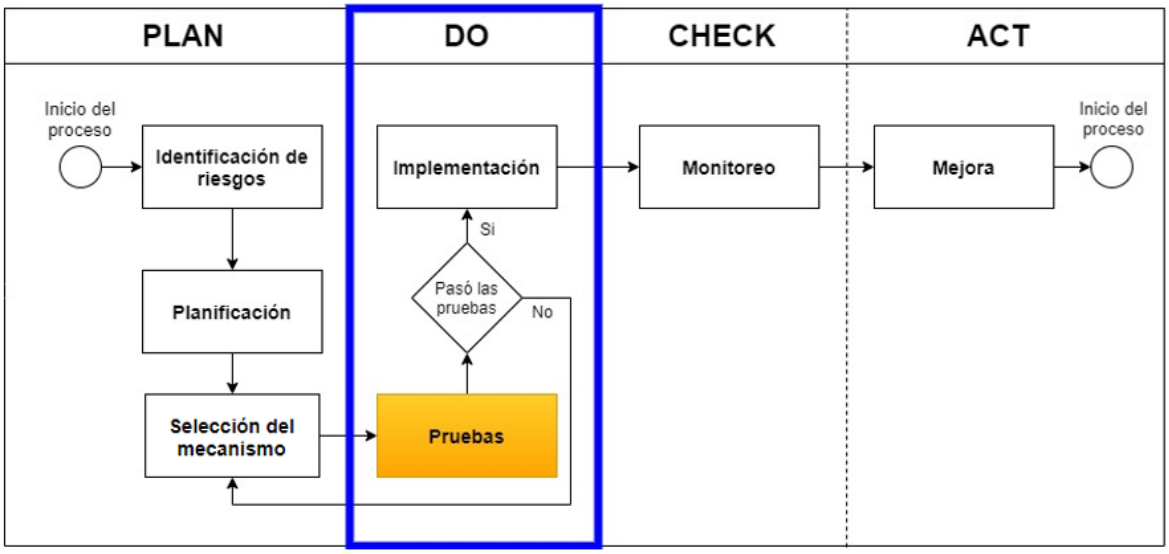
- Mecanismo a implementar: Mecanismo basado en el cálculo de la entropía de la red.

4.3.4. Pruebas

Las pruebas marcan el inicio de la fase "DO", como se muestra en la Figura 33. El objetivo de este subproceso es probar el funcionamiento del mecanismo seleccionado.

Figura 33

Avance de la Aplicación: Pruebas



- **Simular la red de datos junto con los ataques:**
  La simulación de la red y de los ataques DDoS se detalló en las secciones 4.1 y 4.2.

- **Implementar la solución en el escenario de prueba:**
  La solución elegida requiere únicamente del controlador pues la detección y mitigación de ataques DDoS se realiza mediante la programación del mismo. El módulo de detección se encarga de calcular la entropía de la red, la cual es mayor a uno al existir sólo tráfico normal (ver Figura 34) y toma valores menores a 0,5 cuando hay un ataque en curso (ver Figura 35).

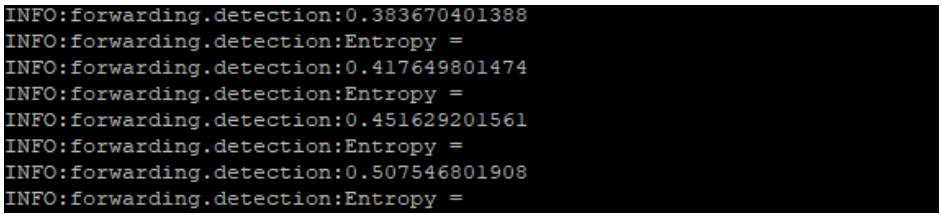
Figura 34

Entropía con Tráfico Normal

```
INFO:forwarding.detection:Entropy =
INFO:forwarding.detection:1.00385965316
INFO:forwarding.detection:Entropy =
INFO:forwarding.detection:1.05977725351
INFO:forwarding.detection:Entropy =
INFO:forwarding.detection:1.11569485386
INFO:forwarding.detection:Entropy =
```

Figura 35

Entropía con un Ataque DDoS en Curso



- **Verificar la solución:**

La Figura 36 muestra la detección y mitigación del ataque en el controlador SDN, este escenario corresponde a un ataque hacia el host 64 desde los hosts 2 y 3, mientras el host 1 genera tráfico legítimo. Cuando el valor de entropía disminuye del límite fijado (0,5), el controlador muestra un mensaje indicando que hay un ataque en curso y procede a bloquear el puerto del switch desde donde proviene el ataque, en este caso el puerto 2 del s2.

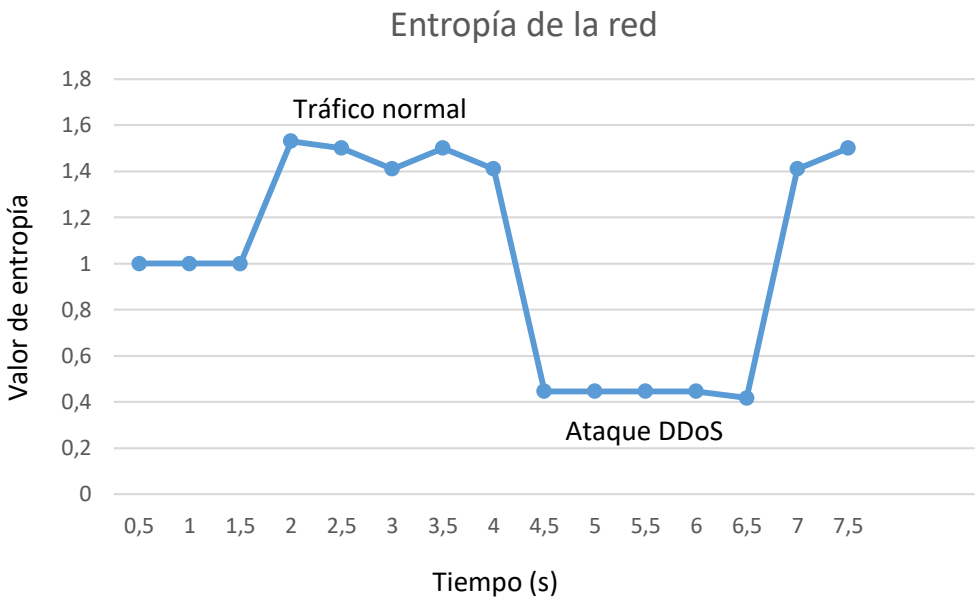
Figura 36

Detección y Mitigación del Ataque de DDoS

La Figura 37 muestra la medición de la entropía en el escenario planteado durante la prueba. Inicialmente sin tráfico cursante en la red la entropía tuvo el valor de 1, al generar tráfico legítimo la entropía ascendió a 1,5. Al momento de generar el ataque DDoS hacia el host 64 el valor de la entropía descendió a 0,4. La gráfica muestra que desde que se genera el ataque (4,5s) hasta que se detecta (6,5s) transcurren dos segundos aproximadamente, siendo esta una detección temprana.

Figura 37

Medición de la Entropía



Salida:

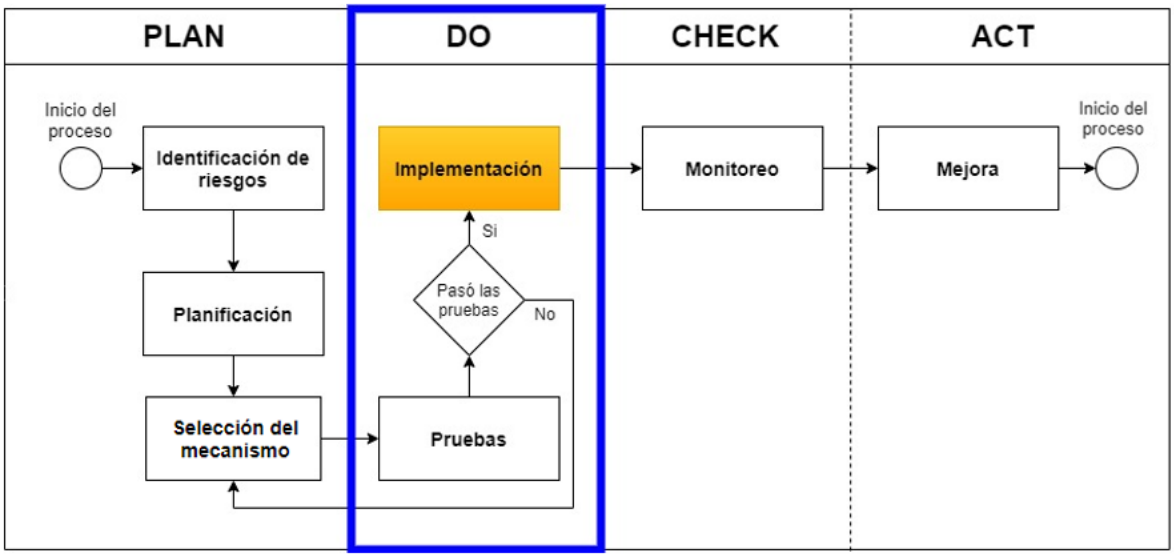
- Resultado de pruebas

4.3.5. Implementación

El objetivo de este subproceso es realizar la implementación de la solución elegida dentro de la infraestructura de la organización. La implementación es parte de la fase “DO” (ver Figura 38).

Figura 38

Avance de la Aplicación: Implementación



Entradas:

- Documento de planificación.
- Mecanismo a implementar.

Actividades:

- **Implementación:**

  Este subproceso se omite en la fase de ejecución de la metodología de este trabajo, ya que en el alcance se definió que se usaría únicamente un escenario de simulación controlado.

  Si se contara con un escenario real la implementación se realizaría en el controlador SDN mediante la creación de un módulo de detección y la modificación del archivo l3_learning.py del controlador POX, de la misma manera como se realizó en el escenario de prueba.

- **Documentación:**

  La documentación se realizó sobre lo realizado en el escenario de simulación, detallando las configuraciones realizadas sobre el controlador SDN, las cuales se podrían aplicar en un controlador real.

Salidas:

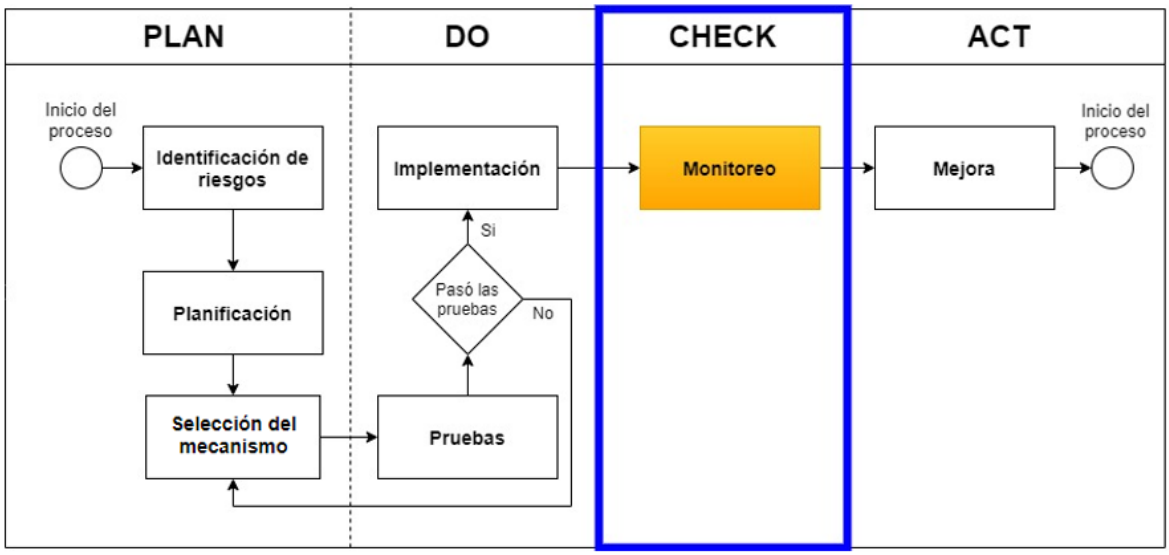
- Implementación del mecanismo de detección y mitigación de DDoS (Ver ANEXO E).

4.3.6. Monitoreo

El monitoreo corresponde a la fase "CHECK", como se señala en la Figura 39. El objetivo de este subproceso es obtener un reporte de desempeño de la solución implementada.

Figura 39

Avance de la Aplicación: Monitoreo



Entradas:

- Documento de planificación.
- Implementación del mecanismo de detección y mitigación de DDoS.

Actividades:

- **Definir qué es necesario monitorear:**

  Se definió monitorear el tiempo de detección, pues es el factor que determina si la detección y mitigación se realiza en una fase temprana del ataque.

- **Definir los métodos/detalles de monitoreo:**

  Si se tratara de un escenario real el monitoreo se realizaría en el lapso de tiempo especificado en la planificación, observando el desempeño de la solución implementada frente a ataques reales hacia el controlador.

  Al tratarse de un escenario de prueba controlado el monitoreo se realizó mediante la generación de distintos tipos de tráfico de ataque TCP, UDP e ICMP.

  El tiempo de detección se obtuvo mediante la revisión de los logs generados en el terminal del controlador.

  Se definió al administrador de la red como el encargado de realizar el monitoreo, y posteriormente analizar los resultados obtenidos.

- **Documentación**

  Se documentaron los resultados del monitoreo frente a varios ataques de DDoS de inundación TCP, UDP e ICMP.

  Para el ataque TCP Flood, también llamado SYN Flood, se generaron paquetes de solicitud de sincronización (SYN) del protocolo TCP con direcciones IP de origen aleatorias y una dirección IP de destino específica (víctima). La Figura 40 muestra una captura de Wireshark de los paquetes generados para este ataque.

Figura 40

Ataque TCP Flood

No.	Time	Source	Destination	Protocol	Length	Info
798	38.148825063	72.167.139.21	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
799	38.180567964	13.117.90.128	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
800	38.215500797	114.85.113.143	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
801	38.251180729	109.138.220.113	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
802	38.283002108	99.233.85.32	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
803	38.312817793	247.102.172.141	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
804	38.343965763	67.135.204.24	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
805	38.372318285	150.213.114.199	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
806	38.401875790	34.224.66.135	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
807	38.432022720	84.21.62.188	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
808	38.460469362	182.6.177.166	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
809	38.494801176	160.164.105.193	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
810	38.527416739	178.131.191.3	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
811	38.574467185	162.169.155.18	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
812	38.605612586	51.235.149.81	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
813	38.636257175	80.231.131.157	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
814	38.667898869	176.216.187.193	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
815	38.696431697	217.13.101.180	10.0.0.64	TCP	54	2 → 80 [SYN] Seq
816	38.726104888	119.229.236.39	10.0.0.64	TCP	54	2 → 80 [SYN] Seq

El ataque UDP flood se generó de manera similar al TCP Flood, en este caso se usó el protocolo UDP y se probó con diferentes puertos de destino que

corresponden a servicios comunes que manejan este protocolo, por ejemplo, el puerto 53/UDP (DNS). La captura de Wireshark de los paquetes enviados en este ataque se muestra en la Figura 41.

Figura 41

Ataque UDP Flood

No.	Time	Source	Destination	Protocol	Length	Info
2293	162.941497807	183.180.130.218	10.0.0.64	UDP	42	53 → 53 L
2294	162.971582798	82.207.134.79	10.0.0.64	UDP	42	53 → 53 L
2295	163.002777052	141.92.82.36	10.0.0.64	UDP	42	53 → 53 L
2296	163.033778409	84.3.42.86	10.0.0.64	UDP	42	53 → 53 L
2297	163.064299358	153.243.108.129	10.0.0.64	UDP	42	53 → 53 L
2298	163.098929539	180.194.11.36	10.0.0.64	UDP	42	53 → 53 L
2299	163.128455258	92.67.159.113	10.0.0.64	UDP	42	53 → 53 L
2300	163.159855541	233.14.254.64	10.0.0.64	UDP	42	53 → 53 L
2301	163.191294437	129.17.84.221	10.0.0.64	UDP	42	53 → 53 L
2302	163.222926918	128.151.220.153	10.0.0.64	UDP	42	53 → 53 L
2303	163.251258935	120.112.150.175	10.0.0.64	UDP	42	53 → 53 L
2304	163.282579771	6.144.168.173	10.0.0.64	UDP	42	53 → 53 L
2305	163.314115539	101.144.61.132	10.0.0.64	UDP	42	53 → 53 L
2306	163.344757167	20.212.10.121	10.0.0.64	UDP	42	53 → 53 L
2307	163.383168217	130.213.63.61	10.0.0.64	UDP	42	53 → 53 L
2308	163.411537072	20.155.46.190	10.0.0.64	UDP	42	53 → 53 L
2309	163.442778110	250.75.174.246	10.0.0.64	UDP	42	53 → 53 L
2310	163.471178632	221.227.95.54	10.0.0.64	UDP	42	53 → 53 L
2311	163.500677685	163.95.194.252	10.0.0.64	UDP	42	53 → 53 L

Para el ataque ICMP Flood se generaron paquetes de Echo Request, los cuales utilizan el protocolo ICMP. La captura de Wireshark de los paquetes generados en este ataque se muestra en la Figura 42.

Figura 42

Ataque ICMP Flood

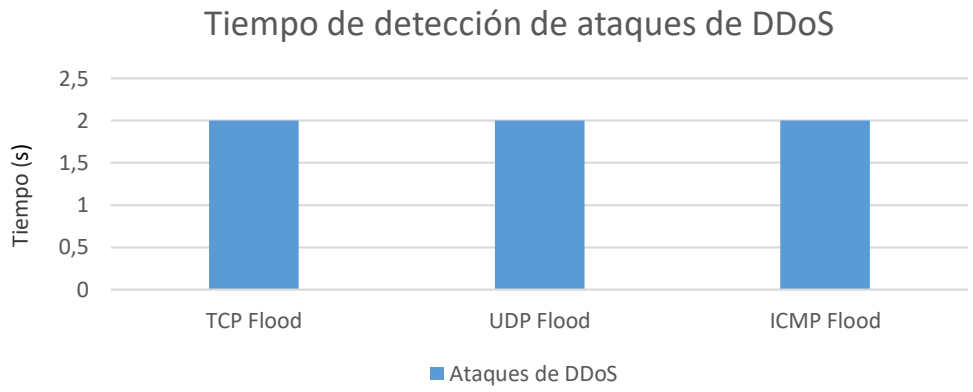
No.	Time	Source	Destination	Protocol	Length	Info
3158	213.347800863	131.184.98.155	10.0.0.64	ICMP	42	Echo (ping) request id=(
3159	213.376523621	156.25.62.214	10.0.0.64	ICMP	42	Echo (ping) request id=(
3160	213.427846024	245.139.1.70	10.0.0.64	ICMP	42	Echo (ping) request id=(
3161	213.456659048	191.87.115.160	10.0.0.64	ICMP	42	Echo (ping) request id=(
3162	213.495504959	31.19.227.255	10.0.0.64	ICMP	42	Echo (ping) request id=(
3163	213.527504432	123.107.242.238	10.0.0.64	ICMP	42	Echo (ping) request id=(
3164	213.557278202	27.70.199.91	10.0.0.64	ICMP	42	Echo (ping) request id=(
3165	213.589692285	167.203.56.243	10.0.0.64	ICMP	42	Echo (ping) request id=(
3166	213.622753353	126.28.81.23	10.0.0.64	ICMP	42	Echo (ping) request id=(
3167	213.652716765	15.65.117.44	10.0.0.64	ICMP	42	Echo (ping) request id=(
3168	213.683801724	128.12.232.86	10.0.0.64	ICMP	42	Echo (ping) request id=(
3169	213.723062401	218.247.143.181	10.0.0.64	ICMP	42	Echo (ping) request id=(
3170	213.757689331	52.25.73.62	10.0.0.64	ICMP	42	Echo (ping) request id=(
3171	213.790915129	40.70.81.28	10.0.0.64	ICMP	42	Echo (ping) request id=(
3172	213.868447264	81.189.176.231	10.0.0.64	ICMP	42	Echo (ping) request id=(
3173	213.905246615	201.193.212.83	10.0.0.64	ICMP	42	Echo (ping) request id=(
3174	213.936934550	155.210.236.67	10.0.0.64	ICMP	42	Echo (ping) request id=(
3175	213.973817691	87.37.64.194	10.0.0.64	ICMP	42	Echo (ping) request id=(
3176	214.004911812	235.66.119.90	10.0.0.64	ICMP	42	Echo (ping) request id=(

La Figura 43 indica que la detección de ataques de DDoS utilizando el mecanismo implementado es independiente del tipo de paquete, ya que se obtiene el mismo

tiempo de detección (2s) para los tres tipos de ataques simulados. Esto se debe a que el mecanismo utilizado se basa en la dirección IP de destino para medir la entropía de la red.

Figura 43

Tiempo de detección de ataques DDoS (TCP, UDP e ICMP Flood)



Se concluyó que la solución cumple con los objetivos planteados en la planificación, sin embargo, se encontró una inconformidad respecto a la excesiva cantidad de mensajes en el terminal del controlador que impiden al administrador de la red monitorear los eventos notificados por controlador.

Salidas:

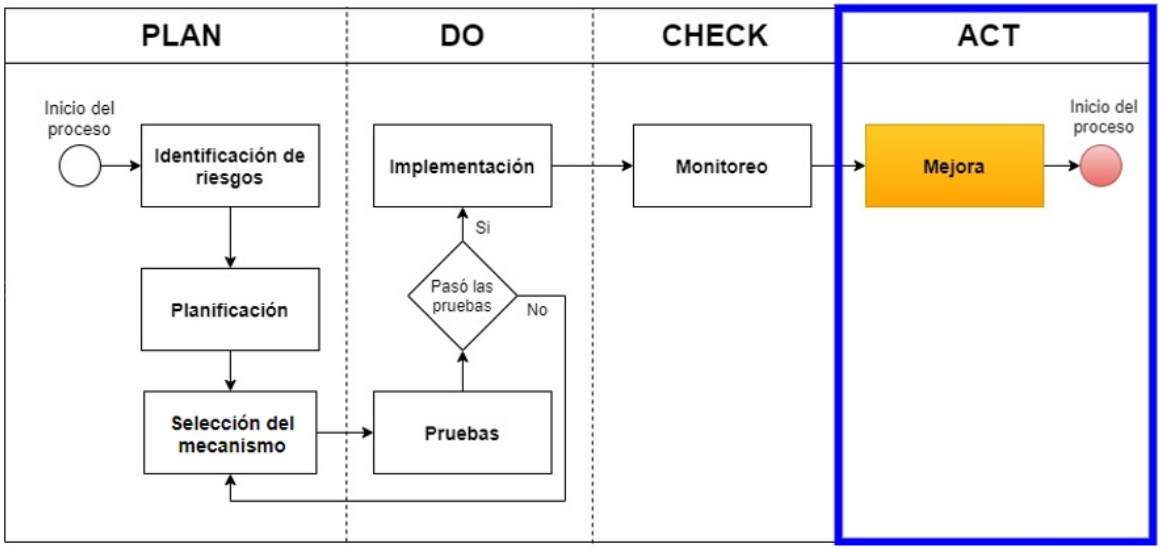
- Reporte de desempeño (Ver ANEXO F).

4.3.7. Mejora

El subproceso mejora está dentro de la fase "ACT" y corresponde al último paso del ciclo de mejora continua. El objetivo de este subproceso es aprobar la solución siempre y cuando se hayan corregido las inconformidades encontradas en el paso anterior.

Figura 44

Avance de la Aplicación: Mejora



Entradas:

- Implementación del mecanismo de detección y mitigación de DDoS.
- Reporte de desempeño.
- Objetivos de seguridad.

Actividades:

- **Corregir inconformidades:**

  Se corrigió la inconformidad encontrada en el subproceso *Monitoreo*, mediante la eliminación de las líneas de código que imprimen los logs de la información relacionada al valor de la entropía en la red, dejando solamente la alerta de detección y mitigación del ataque.

  Las acciones correctivas se realizaron en el módulo de detección y el módulo del controlador POX.

  En el módulo de detección se utilizó el símbolo '#' para comentar las líneas que generaban logs relacionados con la entropía de la red:

  31 #log.info(self.entDic)

  55 #log.info('Entropy = ')

55 #log.info(sum(elist))

En el módulo del controlador POX se también se utilizó el símbolo '#', en este caso para inhabilitar todas las líneas con el comando *print* que imprimían salidas en el terminal con información sobre la entropía de la red. Las líneas 202 y 271 imprimen información mientras se confirma que hay un ataque de DDoS en curso. La línea de comando 223 imprime el valor de la entropía de la red todo el tiempo, incluso cuando no hay ningún tipo de tráfico.

202 #print("Enpty diction ",str(event.connection.dpid), str(event.port))

*271 #print "\n***** Entropy Value = ",str(ent_obj.value),"*****\n"*

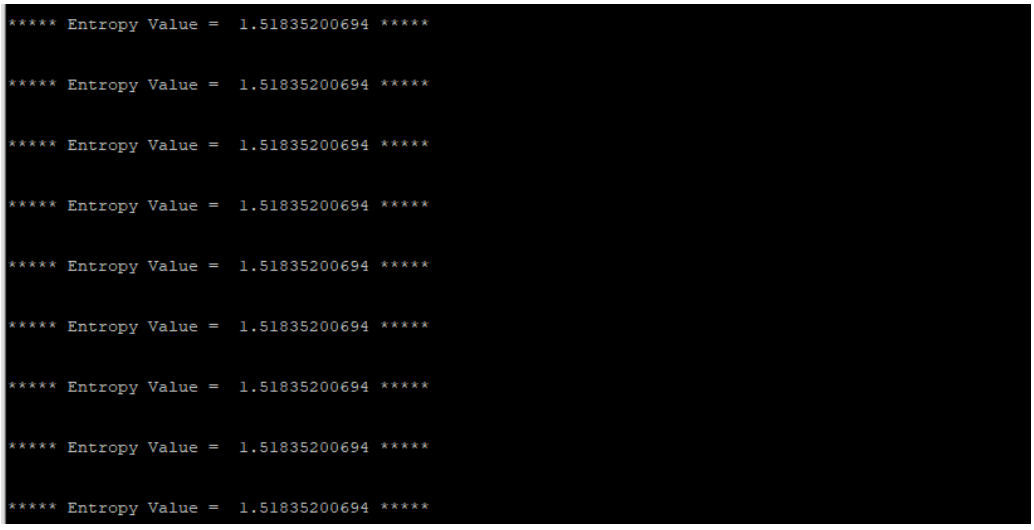
223 #print "\n",datetime.datetime.now(), ": printing diction ",str(diction),"\n"

- **Revisar las correcciones:**

  La Figura 45 muestra al terminal del controlador POX inundado de mensajes relacionados con la medición de la entropía de la red, los cuales no permiten visualizar en qué momento se detectó y mitigó un ataque de DDoS ni tampoco las alertas generadas por el controlador a causa de otro tipo de eventos.

Figura 45

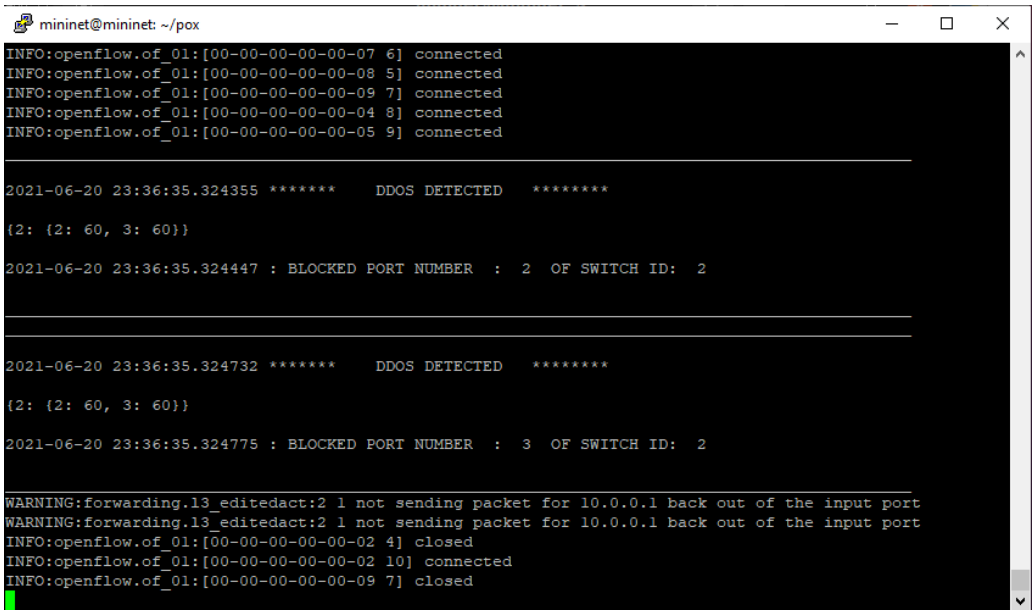
Terminal del Controlador POX Sin Correcciones



Con las correcciones realizadas se logró una salida del controlador más limpia, en el sentido de que ya no se muestran mensajes innecesarios que dificultan el monitoreo de eventos en el controlador. La Figura 46 muestra la salida del terminal con las correcciones, en donde ya no hay logs relacionados con la entropía de la red, sino que solo se imprime la alerta del ataque de DDoS detectado y mitigado. También quedan a la vista otros eventos cuya notificación es propia del controlador POX y no tienen relación con la solución de seguridad implementada.

Figura 46

Terminal del Controlador POX con Correcciones



El rendimiento de la solución implementada frente a los ataques de DDoS en el controlador no se vio afectada por los cambios descritos en esta sección, ya que se deshabilitaron únicamente las líneas necesarias para corregir la inconformidad.

Salida:

- Documento de aprobación de la solución.

4.4.Resultados

En esta sección se evalúan los resultados obtenidos en los subprocesos: Identificación de riesgos, Planificación, Selección del mecanismo, Pruebas, Implementación, Monitoreo y Mejora. De manera que se evidencia el cumplimiento de la metodología de detección y mitigación de ataques DDoS en entornos SDN basado en la norma ISO/IEC 27001 para mejorar la seguridad en el plano de control. La Tabla 22 resume los resultados obtenidos en cada subproceso y sus respectivas actividades

Tabla 22

Resultados por Cada Subproceso

Subproceso	Actividades	Resultado	Logro
Identificación de riesgos	Definir los activos a evaluar	Actividad cumplida	Activos definidos ANEXO C
	Determinar las vulnerabilidades y amenazas por cada activo	Actividad cumplida	Vulnerabilidades y amenazas definidas ANEXO C
	Estimar el impacto	Actividad cumplida	Impacto estimado ANEXO C
	Estimar el riesgo	Actividad cumplida	Riesgo estimado ANEXO C
Planificación	Definir objetivos	Actividad cumplida	Objetivos definidos ANEXO D
	Planificar cómo llegar a los objetivos	Actividad cumplida	Planificación realizada ANEXO D
	Aprobación de la planificación	Actividad cumplida	Formato de documento de planificación ANEXO D
Selección del mecanismo	Analizar mecanismos	Actividad cumplida	Mecanismos analizados Sección 2.3

Subproceso	Actividades	Resultado	Logro
	Elegir mecanismo	Actividad cumplida	Mecanismo elegido Sección 4.3.3
Pruebas	Simular la red y ataques	Actividad cumplida	Red y ataques simulados Sección 4.3.4
	Implementar la solución	Actividad cumplida	Solución implementada Sección 4.3.4
	Verificar la solución	Actividad cumplida	Solución verificada Sección 4.3.4
Implementación	Implementación	Actividad cumplida	Implementación realizada ANEXO E
	Documentación de la implementación	Actividad cumplida	Documentación realizada ANEXO E
Monitoreo	Definir qué se va a monitorear	Actividad cumplida	Factor de monitoreo definido ANEXO F
	Definir métodos de monitoreo	Actividad cumplida	Método de monitoreo definido ANEXO F
	Monitoreo	Actividad cumplida	Monitoreo realizado ANEXO F
	Documentación del monitoreo	Actividad cumplida	Documentación realizada ANEXO F
Mejora	Corregir inconformidades	Actividad cumplida	Corrección realizada Sección 4.3.7
	Revisar correcciones	Actividad cumplida	Revisión realizada Sección 4.3.7

Subproceso	Actividades	Resultado	Logro
	Documentación de las correcciones	Actividad cumplida	Documentación realizada Sección 4.3.7
	Aprobación de la solución	Actividad cumplida	Formato de documento de aprobación ANEXO G

Tras cumplir todas las actividades propuestas en la metodología se evidenciaron las siguientes observaciones que podrían mejorar los resultados de la implementación:

- El documento de planificación resultante del subproceso *Planificación* corresponde a una propuesta, por lo que al aplicarse en una organización real debería ser revisada y aprobada por un profesional que tenga un cargo superior, como por ejemplo el director o jefe del departamento.
- La aprobación de la solución, al igual que el documento de planificación, debería ser realizada por alguna autoridad para que tenga mayor validez al aplicarse en una organización.

4.5. Tiempo de Aplicación de la Metodología

No existe un tiempo definido para la aplicación de cada uno de los subprocesos ni de la metodología en sí, pues esto puede variar dependiendo de la organización. Por lo tanto, se toma como referencia el tiempo de implantación del sistema de gestión de seguridad de la información con la norma ISO 27001, que puede ser de mínimo cuatro meses y máximo un año para evitar que la implementación quede obsoleta debido a distintos factores como los continuos cambios en los riesgos, cambio en las prioridades de la organización con respecto a la protección de activos, aparición de nuevas amenazas, entre otros (INTECO, 2016).

La organización debe estimar el tiempo de implementación principalmente en función de la disponibilidad de los recursos, por ejemplo, considerar si el personal responsable de la aplicación estará dedicado exclusivamente a esa tarea o tiene asignadas otras

responsabilidades de mayor o menor prioridad que no le permitirían culminar con la aplicación en un periodo de tiempo reducido.

También es importante que se tome en cuenta el nivel de conocimiento del personal encargado acerca del tema en cuestión, ya que se podría dar el caso que necesite un periodo adicional de formación y capacitación.

4.6.Costos Asociados a la Aplicación de la Metodología

La aplicación de la metodología requiere que el personal encargado tenga conocimientos básicos sobre gestión de procesos y sobre la tecnología SDN, por ello en esta sección se incluyen costos de capacitación en cada una de estas áreas. Para la capacitación en gestión de procesos se toma como referencia el curso "Gestión por procesos 1" disponible en la plataforma de aprendizaje Udemy, en donde se abordan los conocimientos básicos sobre la implementación de metodologías (Udemy, 2021). En cuanto a la capacitación técnica se toma como referencia el costo del curso "Redes Definidas por Software" dentro de la categoría de la seguridad informática que imparte la Universidad de Chicago (coursera, 2021).

La Tabla 23 muestra los costos asociados a la capacitación requerida para la aplicación de la metodología, cabe recalcar que los costos mostrados corresponden a la capacitación de una sola persona, la cual podría replicar los conocimientos adquiridos a su grupo de trabajo en caso de ser necesario.

Tabla 23

Costo de Capacitación

Descripción	Costo (dólares)
Capacitación en gestión de procesos	49,99
Capacitación en SDN	49,00
Total	98,99

En cuanto a los costos de hardware, la solución no requiere de la adquisición de ningún dispositivo adicional pues se realiza sobre el mismo controlador SDN y bajo este contexto

tampoco se generan costos de software, ya que POX es un controlador de licencia libre, en donde se pueden realizar las configuraciones que se crean convenientes.

La Tabla 24 muestra de manera resumida los costos asociados a la implementación de la metodología. El costo de capacitación se toma del valor total mostrado en la Tabla 23.

Tabla 24

Costos de la aplicación de la metodología

Descripción	Costo (dólares)
Hardware	0,00
Software	0,00
Controlador	0,00
Capacitación	98,99
Total	98,99

La solución se establece para una red SDN en producción, en caso de no tener esta infraestructura se debe considerar los costos asociados al hardware para desplegar la red SDN.

4.7.Discusión

- La solución implementada cumple con el objetivo de detectar y mitigar los ataques DDoS en la red, sin embargo, la acción de bloquear el puerto del switch conectado al host que realiza el ataque tiene la desventaja de que no le permite a dicho host enviar o recibir tráfico legítimo. Como mejora para este inconveniente se podrían implementar técnicas de Machine Learning para filtrar los paquetes, de manera que no se bloquee al host en su totalidad, solamente al tráfico malicioso que genera.
- El mecanismo de seguridad implementado ha sido validado en investigaciones previas mediante la simulación en Mininet y la generación de distintos ataques de inundación de paquetes, lo que ha contribuido a que se tenga un gran nivel de efectividad mediante la selección del umbral de entropía adecuado para detectar un ataque DDoS. Tras aplicar este mecanismo en el escenario de simulación se coincide

con los resultados de investigaciones previas permitiendo una detección temprana para ataques DDoS y es efectivo con cualquier tipo de paquete (TCP, UDP e ICMP).

- La solución fue exitosa para el escenario planteado con el controlador POX, sin embargo, no se encontró mucha información para la aplicación del mecanismo en otros controladores, en el futuro debería investigarse esta técnica en los controladores Ryu, ODL y FloodLight, ya que son muy reconocidos.
- En el presente trabajo se realizaron ataques controlados para el subproceso de monitoreo, debido a que se trata de un entorno simulado y no es posible tener ataques reales. La aplicación de este subproceso en un ambiente real no requiere de la generación de ataques por parte de la organización, sino solo el monitoreo de su rendimiento ante los eventos comunes que ocurren en la red en un periodo de tiempo previsto en la planificación. Esto proporcionaría resultados más precisos para el reporte de desempeño de la solución.

CONCLUSIONES

La arquitectura SDN desacopla el plano de control del plano de datos proporcionando ventajas en la reprogramación y gestión centralizada de la red. Sin embargo, la característica centralizada de SDN hace que el controlador se convierta en un punto de falla que puede ser aprovechado por los ataques de DDoS, comprometiendo a toda la red.

La ejecución de un ataque de DDoS dirigido hacia el controlador SDN es relativamente sencilla desde el punto de vista de que no se necesita generar un tipo de paquete especial para el ataque, pues se puede utilizar cualquiera de los protocolos UDP, TCP o ICMP para inundar al controlador y consumir sus recursos.

La metodología diseñada en este trabajo se basa en la norma ISO/IEC 27001 y el ciclo PDCA. Consta de siete subprocesos (Identificación de riesgos, Planificación, Selección del mecanismo, Pruebas, Monitoreo y Mejora), en donde las salidas de cada subproceso se convierten en entradas para el siguiente de manera que se tiene una secuencia de actividades que deben ser ejecutadas y documentadas.

La aplicación exitosa de cada uno de los subprocesos de la metodología dio como resultado la implementación de una solución basada en el cálculo de la entropía de red, capaz de detectar ataques UDP flood, TCP flood e ICMP flood en una etapa temprana y mitigarlo mediante el bloqueo del puerto del switch desde donde proviene el tráfico malicioso.

Implementar la solución dentro del módulo del controlador es ventajoso a nivel de costos, pues la organización no necesita adquirir software o hardware adicional, en vista que la solución se basa en software libre.

La implementación del mecanismo elegido no es compleja siempre y cuando se tengan conocimientos básicos sobre cómo usar el controlador POX, para realizar modificaciones más avanzadas también se requiere de conocimientos en el lenguaje de programación Python.

La entropía de la red es un valor que se puede utilizar para detectar anomalías en el tráfico de la red, en este caso detectar que un host está recibiendo una cantidad excesiva de paquetes respecto al resto de hosts de la red.

Utilizar el controlador POX tiene la ventaja de que al ser de distribución libre ya ha sido ampliamente investigado por lo que es posible encontrar mucha información respecto a las implementaciones que se pueden hacer con él, en contraste con otros controladores cuya información es escaza.

RECOMENDACIONES

Se debe volver a aplicar la metodología desde el primer paso para cumplir con el ciclo de mejora continua, considerando que los resultados del último subproceso “Mejora” constituyen una entrada para el subproceso inicial “Análisis de riesgos” al momento de volver a aplicar la metodología.

Seleccionar un mecanismo que cuente con amplia documentación teórica y sobre todo práctica, para tener toda la información que se necesita al momento de realizar la implementación.

No realizar configuraciones sobre el módulo del controlador POX l3_learning, ya que si este tiene un error no se iniciará el controlador, es mejor hacer una copia del archivo y sobre esta realizar las configuraciones.

Se recomienda planificar un tiempo prudencial para completar cada uno de los subprocesos, dependiendo de la disponibilidad del personal y de los recursos que se requieren para ejecutar la metodología.

REFERENCIAS BIBLIOGRÁFICAS

Ahmad, I., Namal, S., Ylianttila, M., & Gurtov, A. (2015). *Security in Software Defined Networks : A Survey*. *c*. https://doi.org/10.1109/COMST.2015.2474118

Bannour, F., Souihi, S., & Mellouk, A. (2017). *Distributed SDN Control : Survey , Taxonomy and Challenges*. *c*, 1–25. https://doi.org/10.1109/COMST.2017.2782482

Bawany, N. Z., & Shamsi, J. A. (2016). *Application Layer DDoS Attack Defense Framework for Smart City using SDN*. *May*.

Carvalho, C., & Marques, E. (2019). *Adapting ISO 27001 to a Public Institution*. *June*, 19–22.

Centeno, A. G., Manuel, C., Vergel, R., & Calderón, C. A. (2016). Controladores SDN, elementos para su selección y evaluación. *Revista Telem@tica*, *13*(3), 10–20.

coursera. (2021). *Software Defined Networking*. https://www.coursera.org/learn/sdn

Dao, N. N., Park, J., Park, M., & Cho, S. (2015). A feasible method to combat against DDoS attack in SDN network. *International Conference on Information Networking*, *2015-Janua*, 309–311. https://doi.org/10.1109/ICOIN.2015.7057902

Deepa, V., Sudar, K. M., & Deepalakshmi, P. (2018). Detection of DDoS Attack on SDN Control plane using Hybrid Machine Learning Techniques. *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, *Icssit*, 299–303. https://doi.org/10.1109/ICSSIT.2018.8748836

Dharma, N. I. G., Muthohar, M. F., Prayuda, J. D. A., Priagung, K., & Choi, D. (2015). Time-based DDoS detection and mitigation for SDN controller. *17th Asia-Pacific Network Operations and Management Symposium: Managing a Very Connected World, APNOMS 2015*, 550–553. https://doi.org/10.1109/APNOMS.2015.7275389

Dong, S., Abbas, K., & Jain, R. (2019). A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments. *IEEE Access*, *7*, 80813–80828. https://doi.org/10.1109/ACCESS.2019.2922196

Haque, M. R., Ali, S., Tan, S. C., Yusoff, Z., Kwang, L. C., Kaspin, I. R., & Ziri, S. R. (2017). Motivation of DDoS Attack-Aware in Software Defined Networking Controller Placement. *2017 International Conference on Computer and Applications, ICCA 2017*, 36–42. https://doi.org/10.1109/COMAPP.2017.8079751

INTECO. (2016). *Implantación de un SGSI en la empresa*.

https://www.incibe.es/extfrontinteco/img/File/intecocert/sgsi/img/Guia_apoyo_SGSI.pdf

Kandoi, R., & Antikainen, M. (2015). Denial-of-service attacks in OpenFlow SDN networks. *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, 1322–1326. https://doi.org/10.1109/INM.2015.7140489

Kia, M. U. (2015). *Early Detection and Mitigation of DDoS Attacks In Software Defined Networks*.

Latif, Z., Sharif, K., Li, F., Karim, M. M., Biswas, S., & Wang, Y. (2020). A comprehensive survey of interface protocols for software defined networks. *Journal of Network and Computer Applications*, *156*(July 2019), 102563. https://doi.org/10.1016/j.jnca.2020.102563

Lawal, B. H., & At, N. (2018). Real-Time Detection and Mitigation of Distributed Denial of Service (DDoS) Attacks in Software Defined Networking (SDN). *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 1–4.

MAGERIT. (2012). MAGERIT – versión 3.0. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Libro I - Método. *Ministerio de Hacienda y Administraciones Públicas*, *2006*(630-12-171–8), 127. https://www.ccn-cert.cni.es/documentos-publicos/1789-magerit-libro-i-metodo/file.html

Mininet Team. (2018). *Mininet: An Instant Virtual Network on your Laptop (or other PC)*. Mininet.Org. http://mininet.org

Mladenov, B. (2019). Studying the DDoS Attack Effect over SDN Controller Southbound Channel. *2019 X National Conference with International Participation (ELECTRONICA)*, 1–4.

Mousavi, S. M., & St-hilaire, M. (2016). *Early Detection of DDoS Attacks against SDN Controllers*. 77–81.

Mousavi, S. M., & St-Hilaire, M. (2018). Early Detection of DDoS Attacks Against Software Defined Network Controllers. *Journal of Network and Systems Management*, *26*(3), 573–591. https://doi.org/10.1007/s10922-017-9432-1

MS-ISAC. (2017). Guide to DDoS Attacks November 2017. *Guide to DDoS Attacks, November*.

Pedraza, G. (2017). Plan de implementación de un sistema de gestión de seguridad de la información de una entidad del sector público basado en la NTC ISO 27001:2013. *Universidad de América*, 1–14. http://hdl.handle.net/20.500.11839/7008

Polat, H., Polat, O., & Cetin, A. (2020). Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability (Switzerland)*, *12*(3). https://doi.org/10.3390/su12031035

Robalino Diaz, J. W. (2018). *Propuestas Metodológica Y Simulación De La Implementación De Un Siem Basado En La Norma Iso 27001 Y/O 27002*.

Saharan, S., & Gupta, V. (2019). Prevention and Mitigation of DNS based DDoS attacks in SDN Environment. *2019 11th International Conference on Communication Systems and Networks, COMSNETS 2019*, *2061*, 571–573. https://doi.org/10.1109/COMSNETS.2019.8711258

Sahoo, K. S. (2017). Detection of Control Layer DDoS Attack using Entropy metrics in SDN : An Empirical Investigation. *2017 Ninth International Conference on Advanced Computing (ICoAC)*, 281–286.

Plan Nacional de Desarrollo 2017-2021-Toda una Vida, 82 (2017).

Singh, M. P., & Bhandari, A. (2020). New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges. *Computer Communications*, *154*(March), 509–527. https://doi.org/10.1016/j.comcom.2020.02.085

Thomas, R. M., & James, D. (2017). DDOS Detection a nd Denial u sing Third Party Application i n SDN. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 3892–3897.

Torres, Á., & Zuñiga, A. (2020). Análisis de herramientas que permitan el modelado de tráfico en redes SDN. *Centro Sur. Social Science Journal*, *4*. http://centrosureditorial.com/index.php/revista

Udemy. (2021). *Curso virtual gestión por procesos 1*. https://www.udemy.com/course/curso-virtual-vestion-por-procesos-1/?utm_source=adwords&utm_medium=udemyads&utm_campaign=LongTail_la.ES_cc.LATAM&utm_term=_._ag_118028630821_._ad_515853463327_._kw__._de_c_._dm__._pl__._ti_dsa-1190286610239_._li_9076633_._pd__._&matchtype=b&gclid=CjwKCAjw87SHB

hBiEiwAukSeUZ0VeIoGWopYgd7a1znZhYYPeuXmUCncFTtxhQ8jEo1gsieYQmW6KBoCW60QAvD_BwE

Valencia, B., Santacruz, S., & Padilla, L. Y. B. J. J. (2015). *Mininet : una herramienta versátil para emulación y prototipado de Redes Definidas por Software 1 Mininet : a versatile tool for emulation and prototyping of Software Defined Networking*. *17*, 62–70.

Zerkane, S., Espes, D., Le Parc, P., & Cuppens, F. (2017). Vulnerability Analysis of Software Defined Networking. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10128 LNCS*, VI. https://doi.org/10.1007/978-3-319-51966-1

ANEXO A

Mesa de trabajo para determinar las vulnerabilidades del plano de control de SDN

Objetivo: Determinar las vulnerabilidades que presenta el plano de control de SDN frente a ataques DDOS, así como su impacto y probabilidad de ocurrencia.

Fecha: 26 de mayo de 2021

Participantes: Jessica Báez, Fabián Cuzme y Mauricio Domínguez

Vulnerabilidades del plano de control de SDN

El análisis de vulnerabilidades es un proceso de gran importancia en la gestión de riesgos de seguridad porque permite descubrir las debilidades de un sistema y sus impactos en la seguridad. Debido a que no existe una clasificación universal estandarizada en el análisis de vulnerabilidades, esto se convierte en un proceso complejo y subjetivo. Este hecho se intensifica para las SDN, pues se trata de un entorno dinámico y emergente donde no hay vulnerabilidades SDN históricas y bien conocidas (Zerkane et al., 2017).

Cada organización implementa la arquitectura SDN de diferentes formas de acuerdo con sus necesidades y objetivos específicos. Por lo tanto, es necesario realizar un análisis de vulnerabilidad genérico para SDN para adoptar contramedidas adecuadas contra los ataques de seguridad (Zerkane et al., 2017).

La metodología MAGERIT define en su catálogo de amenazas a la denegación de servicio como un ataque intencionado que actúa sobre la dimensión de seguridad “Disponibilidad” y afecta tanto a servicios como a equipos informáticos (hardware). La carencia de recursos suficientes provoca la caída del sistema cuando la carga de trabajo es desmesurada.

Debido a que en SDN el controlador es el cerebro principal de la red, el rendimiento de la red depende de él. Una de las principales amenazas hacia el controlador SDN son los ataques DDoS, que al ejecutarse con éxito pueden hacer que la red colapse al inhabilitar al controlador de atender peticiones de usuarios legítimos (Dong et al., 2019). Las

vulnerabilidades del plano de control SDN que pueden ser aprovechadas para lanzar ataques DDoS se han estudiado en algunas investigaciones y son las siguientes:

- **Mecanismo de nuevo paquete**

Con respecto al funcionamiento básico de SDN, un atacante puede aprovechar el mecanismo de nuevo paquete para hacer que el controlador sea inaccesible (Dharma et al., 2015).

La operación básica de SDN consiste en transferir tráfico para un paquete en la red. Hay una búsqueda en la tabla de flujo para cada paquete nuevo al conmutador. Para una coincidencia exitosa, se llevará a cabo la acción de flujo. De lo contrario, el paquete se enviará al controlador para obtener más instrucciones mediante mensajes packet_in. A su vez, el controlador agregará una regla de flujo o eliminará los flujos de la tabla de flujo. Si la tasa de llegada de packet_in es muy alta en caso de ataque DDoS, los recursos del controlador comenzarán a agotarse (Sahoo, 2017).

- **Tamaño de la tabla de flujo limitada**

Debido al tamaño de la memoria, el tamaño de la tabla de flujo en el controlador y los dispositivos de red es limitado. Un ataque DDoS puede aprovechar esta limitación y hacer que el controlador tenga todos sus recursos destinados al procesamiento de los paquetes maliciosos dejando de atender a los usuarios legítimos (Dharma et al., 2015).

El controlador SDN se aloja en un servidor, ya sea físico o virtual. Por ello se considera también las vulnerabilidades del servidor frente a los ataques DDoS.

- **Recursos de memoria y procesamiento limitados.**

Dado que cada paquete nuevo tiene que llegar al controlador, un ataque DDoS puede, en última instancia, desperdiciar los recursos del sistema del controlador, como sus capacidades de procesamiento (CPU) y memoria física (RAM), etc., en el manejo de estos paquetes maliciosos (Singh & Bhandari, 2020). En un ataque DDoS exitoso, los recursos del sistema pueden agotarse, provocando el colapso de toda la red.

De acuerdo a Zerkane (2017), el impacto de la explotación de la vulnerabilidad en la accesibilidad a los recursos SDN se estima de la siguiente manera:

- **Bajo:** La disponibilidad del controlador se ve parcialmente afectada (no está disponible durante un tiempo determinado o si está disponible todo el tiempo con algunas interrupciones).
- **Alto:** el controlador es completamente inaccesible.

La complejidad del ataque es baja porque el atacante hace un mal uso del comportamiento predeterminado del controlador SDN y puede repetir el ataque. No necesita privilegios especiales ni interacciones de usuario.

Probabilidad de ocurrencia de la amenaza:

Los ataques DDoS han aumentado exponencialmente en tamaño en los últimos años. Estos ataques son capaces de causar más daños significativos a las empresas más grandes, los sistemas ciberfísicos, los centros de datos, y los proveedores de servicios donde la tecnología SDN acaba de comenzar a brotar y tomar forma. De acuerdo al último Informe de seguridad de infraestructura mundial anual (WISR) de Arbor Networks, el 61 % de los operadores de data centers aseguraron haber recibido ataques que saturaron por completo el ancho de banda de su data center (Singh & Bhandari, 2020).

Un estudio realizado por la empresa Black Lotus Communications anotó que la frecuencia de ataques DDoS continúa en crecimiento y que algunos Data Centers son el objetivo de estos ataques docenas de veces al mes. Otro estudio más específico afirma que el 35% de Data Centers se enfrentan a uno o más ataques DDoS por semana (Saharan & Gupta, 2019). Con esta frecuencia se estima según la metodología MAGERIT que la probabilidad de ocurrencia de la amenaza es MUY ALTA.

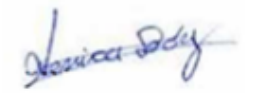
El resumen de la información obtenida en esta mesa de trabajo se presenta en la Tabla 25.

Tabla 25

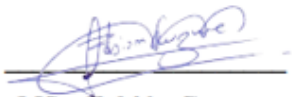
Vulnerabilidades del Plano de Control

Vulnerabilidades del Plano de Control	Amenaza	Impacto	Probabilidad de Ocurrencia
Mecanismo de nuevo paquete	Ataque DDoS	Alto	Muy Alta
Tamaño de la tabla de flujo limitada	Ataque DDoS	Alto	Muy Alta
Recursos de memoria y procesamiento limitados	Ataque DDoS	Alto	Muy Alta

Firmas de responsabilidad de los participantes:



Ing. Jessica Báez
Maestrante



MSc. Fabián Cuzme
Tutor

MSc. Mauricio Domínguez
Asesor

ANEXO B

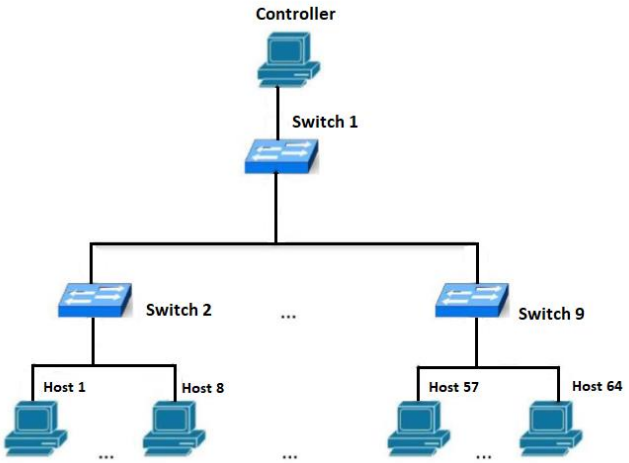
b.1 Formato Acta de entrevista

Nº:	1
Fecha:	
Tiempo:	
Objetivo de la entrevista:	Validar el análisis de riesgos a presentar en el trabajo de titulación: "METODOLOGÍA DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DDOS EN ENTORNOS SDN BASADO EN LA NORMA ISO/IEC 27001 PARA MEJORAR LA SEGURIDAD EN EL PLANO DE CONTROL".
Nombre de la entrevistadora:	Ing. Jessica Báez
Nombre del entrevistado:	Ing. Fernando Obando
Referencias/Experiencia profesional del entrevistado:	Encargado de la seguridad de la información de la empresa SAITEL.

Topología de red:

Consiste en una estructura tipo árbol de profundidad de dos niveles y ocho de fanout

- 1 controlador SDN
- 9 switches OpenFlow
- 64 hosts



Cuestionario:

¿Cuáles son los activos de información que deben ser evaluados en el escenario planteado para el análisis de riesgos? ¿Qué valor tendrían estos activos?

¿Está de acuerdo con las vulnerabilidades presentadas en el ANEXO A? ¿Consideraría otra vulnerabilidad?

Según su experiencia ¿Cuál es la degradación del valor que tienen los activos al ser víctimas de un ataque DDoS?

Según su experiencia, ¿Cuál es la probabilidad de ocurrencia de los ataques DDoS en este tipo de arquitecturas?

Ing. Jessica Báez	Ing. Fernando Obando
Entrevistadora	Entrevistado

Versión:	Elaborado por:	Aprobado por:
1.0	 Ing. Jessica Báez	 MSc. Fabián Cuzme
Fecha: 01/06/2021	Fecha: 01/06/2021	Fecha: 02/06/2021

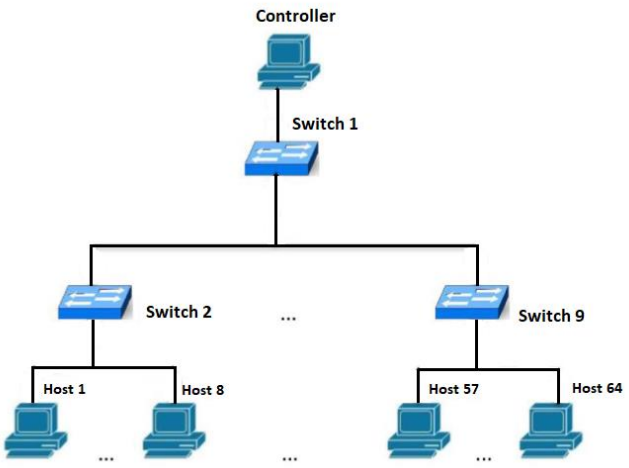
b.2 Acta de entrevista

Nº:	1
Fecha:	3 de junio de 2021
Tiempo:	2 horas
Objetivo de la entrevista:	Validar el análisis de riesgos a presentar en el trabajo de titulación: "METODOLOGÍA DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DDOS EN ENTORNOS SDN BASADO EN LA NORMA ISO/IEC 27001 PARA MEJORAR LA SEGURIDAD EN EL PLANO DE CONTROL".
Nombre de la entrevistadora:	Ing. Jessica Báez
Nombre del entrevistado:	Ing. Fernando Obando
Referencias/Experiencia profesional del entrevistado:	Encargado de la seguridad de la información de la empresa SAITEL.

Topología de red:

Consiste en una estructura tipo árbol de profundidad de dos niveles y ocho de fanout

- 1 controlador SDN
- 9 switches OpenFlow
- 64 hosts



Cuestionario:

¿Cuáles son los activos de información que deben ser evaluados en el escenario planteado para el análisis de riesgos? ¿Qué valor tendrían estos activos?

El controlador (MA) ----------

¿Está de acuerdo con las vulnerabilidades presentadas en el ANEXO A? ¿Consideraría otra vulnerabilidad?

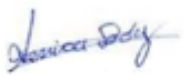
Si de acuerdo, son los escenarios de vulnerabilidades con mayor probabilidad de ocurrencia considerando en DDoS ----------

Según su experiencia ¿Cuál es la degradación del valor que tienen los activos al ser víctimas de un ataque DDoS?

Depende del valor que el activo representa para la continuidad del negocio, si nos ubicamos en la topología planteada la degradación del controlador SDN es MUY ALTA, debido a que es un punto crítico en la red mostrada, el cual, en caso de verse afectado afectaría de gran manera la disponibilidad de la información a los host o dispositivos que necesiten acceso.

Según su experiencia, ¿Cuál es la probabilidad de ocurrencia de los ataques DDoS en este tipo de arquitecturas?

Debido a que es uno de los ataques que no requieren mayor experticia es uno de los más frecuentes, pero de la misma manera es uno de los ataques que se pueden mitigar con mayor facilidad, teniendo en cuenta que la empresa tiene a un experto dedicado a la seguridad de la información, caso contrario puede afectar de gran manera la continuidad del negocio y generar intermitencias en la disponibilidad de los datos o servicios que ofrece una determinada organización.



Ing. Jessica Báez
Entrevistadora

Ing. Fernando Obando
Entrevistado

ANEXO C

Mesa de trabajo para realizar el análisis de riesgos del plano de control de SDN

Objetivo: Realizar el análisis de riesgos de plano de control SDN referente a los ataques SDN mediante la metodología MAGERIT.

Fecha: 3 de junio de 2021

Participantes: Jessica Báez y Fernando Obando

Se realiza el análisis de riesgos del plano de control, evaluando las amenazas por cada activo.

- **Definir los activos a evaluar**

  Se definen los activos a evaluar con sus respectivos detalles en la Tabla 26.

Tabla 26

Activos a Evaluar

Activo	Descripción	Tipo	Persona responsable	Ubicación
Servidor del controlador	Servidor físico/virtual que aloja al controlador SDN	Hardware	Administrador del controlador	Data Center
Controlador	Controlador SDN de la red	Software	Administrador del controlador	Servidor del controlador

La valoración de Degradación, Probabilidad, Impacto y Riesgo se realiza conforme las métricas señaladas en el apartado 2.6.

- **Determinar las amenazas por cada activo**

  Se valoran las vulnerabilidades y sus respectivas amenazas ante los ataques DDoS dirigidos al controlador SDN (ver Tabla 27).

Tabla 27

Vulnerabilidades por Cada Activo

Activo	Vulnerabilidad	Amenaza	Degradación	Probabilidad
Servidor del controlador	• Recursos de memoria y procesamiento limitados.	Ataque de DDoS	MA	MA
Controlador	• Mecanismo de nuevo paquete. • Tamaño de la tabla de flujo limitada.	Ataque de DDoS	MA	MA

Nota. Basado en Sahoo (2017), Dharma (2015) y Singh (2015) (ver ANEXO A).

- **Estimar el impacto**

Conociendo el valor de los activos y la degradación provocada por las amenazas se determina el impacto en la Tabla 28. Se define el valor de los activos señalados como MA (Muy alto), debido a que son un punto fundamental de la arquitectura SDN, pues su funcionamiento compromete a toda la red (Sahoo, 2017).

Tabla 28

Estimación del Impacto

Activo	Valor	Amenaza	Degradación	Impacto
Servidor del controlador	MA	Ataque de DDoS por inundación de paquetes	MA	MA
Controlador	MA	Ataque de DDoS por inundación de paquetes	MA	MA

- **Estimar el riesgo**

Se estima el riesgo en base al Impacto y la Probabilidad. Ver Tabla 29.

Tabla 29

Estimación del Riesgo

Activo	Amenaza	Impacto	Probabilidad	Riesgo
Servidor del controlador	Ataque de DDoS por inundación de paquetes	MA	MA	MA
Controlador	Ataque de DDoS por inundación de paquetes	MA	MA	MA

- **Calificación del riesgo**

Se califica el riesgo de acuerdo a la Tabla 10.

Se determina que el riesgo para el activo "Servidor del controlador" es crítico y requiere de atención urgente, al igual que el activo "Controlador".

Ing. Jessica Báez
Entrevistadora

Ing. Fernando Obando
Entrevistado

ANEXO D

Documento de Planificación

Debido a que los riesgos encontrados en el proceso "Identificación de riesgos" han sido calificados como CRÍTICOS, se concluye que se deben tomar salvaguardas para mitigarlo.

Como contramedida para mitigar el riesgo se propone implementar un mecanismo de detección y mitigación de ataques DDoS.

Objetivo de seguridad:

Implementar un mecanismo de detección temprana y mitigación de ataques DDoS en el controlador SDN.

Recursos a utilizar:

- Documentación sobre mecanismos de detección y mitigación de ataques DDoS.
- Topología de red.
- Escenario de simulación controlado (con todos elementos que se presentan en la topología de red).
- Administrador de red y personal de TI.

Personal responsable:

Nombre	Cargo	Firma

Fecha de inicio:

3 de junio de 2021

Fecha de finalización:

24 de junio de 2021

Cronograma:

Se define un cronograma para la ejecución de los siguientes subprocesos indicados en la metodología.

Tabla 30

Cronograma de Ejecución

Subproceso	Fecha de Inicio	Fecha de Finalización
Selección del mecanismo	4 de junio de 2021	7 de junio de 2021
Pruebas/ Implementación	8 de junio de 2021	16 de junio de 2021
Monitoreo	17 de junio de 2021	19 de junio de 2021
Mejora	20 de junio de 2021	24 de junio de 2021

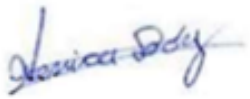
Métrica de evaluación:

- Tiempo de detección del ataque DDoS

Firma de aprobación:

_____

Director del Departamento de TI

Versión:	Elaborado por:	Aprobado por:
1.0	 Ing. Jessica Báez	 MSc. Fabián Cuzme
Fecha: 02/06/2021	Fecha: 02/06/2021	Fecha: 03/06/2021

ANEXO E

Documento de Implementación del mecanismo de detección y mitigación de ataques DDoS en el controlador

En este documento se detalla la implementación del mecanismo de seguridad contra DDoS.

1. Crear el módulo de detección

El módulo de detección es un archivo escrito en Python que tiene la función de calcular la entropía en la red, este se ubicó en la carpeta:

pox/pox/forwarding

El archivo se creó mediante el siguiente comando:

vim detection.py

Dentro del módulo detection.py se pegó el código que se muestra a continuación.

```
import math
from pox.core import core
log = core.getLogger()
class Entropy(object):
        count = 0
        entDic = {}
        ipList = []
        dstEnt = []
        value = 1
        def statcolect(self, element):
                #print "Self values"
                #print "count is " + str(self.count)
                #print "Length of IP list is"
                #print len(self.ipList)
                #print "*********"
                l = 0
                self.count +=1
                self.ipList.append(element)
                if self.count == 50:
                        for i in self.ipList:
                                l +=1
                                if i not in self.entDic:
                                        self.entDic[i] =0
                                self.entDic[i] +=1
                        self.entropy(self.entDic)
                        log.info(self.entDic)
                        self.entDic = {}
                        self.ipList = []
                        l = 0
                        self.count = 0

        def entropy (self, lists):
                #print "Entropy called"
                l = 50
```

```
            elist = []
            for k,p in lists.items():
                    '''
                    log.info("p is")
                    log.info(p)
                    log.info("P is obtained from")
                    log.info(k)
                    log.info("l is")
                    log.info(l)
                    '''
                    c = p/float(l)
                    #log.info("Value of c is ")
                    #log.info(c)
                    c = abs(c)
                    elist.append(-c * math.log(c, 10))
                    log.info('Entropy = ')
                    log.info(sum(elist))
                    #log.info("****")
                    self.dstEnt.append(sum(elist))
            if(len(self.dstEnt)) == 80:
                    print self.dstEnt
                    self.dstEnt = []
             self.value = sum(elist)

    def __init__(self):
            pass
```

2. Crear el módulo l3_edited:

El módulo l3_edited.py es una copia del módulo original del controlador l3_forwarding, con la diferencia de que tiene la función de llamar al módulo de detección para obtener el valor de la entropía de la red y además contiene las funciones de notificación y mitigación del ataque. El archivo l3_edited se ubicó en la misma carpeta del módulo de detección:

pox/pox/forwarding

El archivo se creó mediante el siguiente comando:

vim l3_edited.py

Dentro del módulo l3_edited.py se pegó el código que se muestra a continuación.

```
import datetime
from pox.core import core
import pox

from pox.lib.packet.ethernet import ethernet, ETHER_BROADCAST
from pox.lib.packet.ipv4 import ipv4
from pox.lib.packet.arp import arp
from pox.lib.addresses import IPAddr, EthAddr
from pox.lib.util import str_to_bool, dpid_to_str
from pox.lib.recoco import Timer

import pox.openflow.libopenflow_01 as of
from pox.lib.revent import *
import itertools
```

```
import time
#editing
from .detection import Entropy
diction = {}
ent_obj = Entropy()
set_Timer = False
defendDDOS=False
log = core.getLogger()
FLOW_IDLE_TIMEOUT = 10
ARP_TIMEOUT = 60 * 2
MAX_BUFFERED_PER_IP = 5
MAX_BUFFER_TIME = 5
class Entry (object):
  def __init__ (self, port, mac):
    self.timeout = time.time() + ARP_TIMEOUT
    self.port = port
    self.mac = mac
  def __eq__ (self, other):
    if type(other) == tuple:
      return (self.port,self.mac)==other
    else:
      return (self.port,self.mac)==(other.port,other.mac)
  def __ne__ (self, other):
    return not self.__eq__(other)
  def isExpired (self):
    if self.port == of.OFPP_NONE: return False
    return time.time() > self.timeout
def dpid_to_mac (dpid):
  return EthAddr("%012x" % (dpid & 0xffFFffFFffFF,))

class l3_switch (EventMixin):
  def __init__ (self, fakeways = [], arp_for_unknowns = False, wide = False):

    self.fakeways = set(fakeways)
    self.wide = wid
    self.arp_for_unknowns = arp_for_unknowns
    self.outstanding_arps = {}
    self.lost_buffers = {}
    self.arpTable = {}
    self._expire_timer = Timer(5, self._handle_expiration, recurring=True)

    core.listen_to_dependencies(self)

  def _handle_expiration (self):
    empty = []
    for k,v in self.lost_buffers.iteritems():
      dpid,ip = k
      for item in list(v):
        expires_at,buffer_id,in_port = item
        if expires_at < time.time():
          # This packet is old.  Tell this switch to drop it.
          v.remove(item)
          po = of.ofp_packet_out(buffer_id = buffer_id, in_port = in_port)
          core.openflow.sendToDPID(dpid, po)
      if len(v) == 0: empty.append(k)

    for k in empty:
      del self.lost_buffers[k]

  def _send_lost_buffers (self, dpid, ipaddr, macaddr, port):
    if (dpid,ipaddr) in self.lost_buffers:
      bucket = self.lost_buffers[(dpid,ipaddr)]
      del self.lost_buffers[(dpid,ipaddr)]
      log.debug("Sending %i buffered packets to %s from %s"
                % (len(bucket),ipaddr,dpid_to_str(dpid)))
      for _,buffer_id,in_port in bucket:
```

```
        po = of.ofp_packet_out(buffer_id=buffer_id,in_port=in_port)
        po.actions.append(of.ofp_action_dl_addr.set_dst(macaddr))
        po.actions.append(of.ofp_action_output(port = port))
        core.openflow.sendToDPID(dpid, po)
  def _handle_openflow_PacketIn (self, event):
    dpid = event.connection.dpid
    inport = event.port
    packet = event.parsed
    global set_Timer
    global defendDDOS
    global blockPort
    timerSet =False
    global diction
    def preventing():
      global diction
      global set_Timer
      if not set_Timer:
        set_Timer =True
      if len(diction) == 0:
        print("Enpty diction ",str(event.connection.dpid), str(event.port))
        diction[event.connection.dpid] = {}
        diction[event.connection.dpid][event.port] = 1
      elif event.connection.dpid not in diction:
        diction[event.connection.dpid] = {}
        diction[event.connection.dpid][event.port] = 1
        #print "ERROR"
      else:
        if event.connection.dpid in diction:
          if event.port in diction[event.connection.dpid]:
            temp_count=0
            temp_count =diction[event.connection.dpid][event.port]
            temp_count = temp_count+1
            diction[event.connection.dpid][event.port]=temp_count
            #print   "printting   dpid   port   number   and   its   packet   count:   ",
str(event.connection.dpid),                              str(diction[event.connection.dpid]),
str(diction[event.connection.dpid][event.port])
          else:
            diction[event.connection.dpid][event.port] = 1

      print "\n",datetime.datetime.now(), ": printing diction ",str(diction),"\n"

    def _timer_func ():
      global diction
      global set_Timer
      if set_Timer==True:
        #print datetime.datetime.now(),": calling timer fucntion now!!!!!"
        for k,v in diction.iteritems():
          for i,j in v.iteritems():
            if j >=50:
              print
"_____________________________________________________________________________________"
              print "\n",datetime.datetime.now(),"*******    DDOS DETECTED   ********"
              print "\n",str(diction)
              print "\n",datetime.datetime.now(),": BLOCKED PORT NUMBER  : ", str(i), " OF
SWITCH ID: ", str(k)
              print
"\n_____________________________________________________________________________________"

              #self.dropDDOS ()
              dpid = k
              msg = of.ofp_packet_out(in_port=i)
              #msg.priority=42
              #msg.in_port = event.port
              #po = of.ofp_packet_out(buffer_id = buffer_id, in_port = in_port)
              core.openflow.sendToDPID(dpid,msg)
```

```
      diction={}
    if not packet.parsed:
      log.warning("%i %i ignoring unparsed packet", dpid, inport)
      return
    if dpid not in self.arpTable:
      # New switch -- create an empty table
      self.arpTable[dpid] = {}
      for fake in self.fakeways:
        self.arpTable[dpid][IPAddr(fake)] = Entry(of.OFPP_NONE,
         dpid_to_mac(dpid))
    if packet.type == ethernet.LLDP_TYPE:
      # Ignore LLDP packets
      return
    if isinstance(packet.next, ipv4):
      log.debug("%i %i IP %s => %s", dpid,inport,
                packet.next.srcip,packet.next.dstip)
      ent_obj.statcolect(event.parsed.next.dstip)#editing
      print "\n***** Entropy Value = ",str(ent_obj.value),"*****\n"
      if ent_obj.value <0.5:
        preventing()
        if timerSet is not True:
          Timer(2, _timer_func, recurring=True)
          timerSet=False
      else:
        timerSet=False
      # Send any waiting packets...
      self._send_lost_buffers(dpid, packet.next.srcip, packet.src, inport)

      # Learn or update port/MAC info
      if packet.next.srcip in self.arpTable[dpid]:
        if self.arpTable[dpid][packet.next.srcip] != (inport, packet.src):
          log.info("%i %i RE-learned %s", dpid,inport,packet.next.srcip)
          if self.wide:
            # Make sure we don't have any entries with the old info...
            msg = of.ofp_flow_mod(command=of.OFPFC_DELETE)
            msg.match.nw_dst = packet.next.srcip
            msg.match.dl_type = ethernet.IP_TYPE
            event.connection.send(msg)
      else:
        log.debug("%i %i learned %s", dpid,inport,packet.next.srcip)
      self.arpTable[dpid][packet.next.srcip] = Entry(inport, packet.src)

      #print    "switcID:    "+str(dpid)+"    ,Port:    "+str(event.port)+"    ,MAC    address:
"+str(myPacketInSrcEth)+"      ,SrcIP:      "+      str(myPacketInSrcIP)+",      Dst      Mac:
"+str(myPacketInDstEth)+", Dst IP: "+str(myPacketInDstEth)
      # Try to forward
      dstaddr = packet.next.dstip
      if dstaddr in self.arpTable[dpid]:
        # We have info about what port to send it out on...

        prt = self.arpTable[dpid][dstaddr].port
        mac = self.arpTable[dpid][dstaddr].mac
        if prt == inport:
          log.warning("%i %i not sending packet for %s back out of the "
                      "input port" % (dpid, inport, dstaddr))
        else:
          log.debug("%i %i installing flow for %s => %s out port %i"
                    % (dpid, inport, packet.next.srcip, dstaddr, prt))
          actions = []
          actions.append(of.ofp_action_dl_addr.set_dst(mac))
          actions.append(of.ofp_action_output(port = prt))
          if self.wide:
            match = of.ofp_match(dl_type = packet.type, nw_dst = dstaddr)
          else:
            match = of.ofp_match.from_packet(packet, inport)
```

```
      msg = of.ofp_flow_mod(command=of.OFPFC_ADD,
                            idle_timeout=FLOW_IDLE_TIMEOUT,
                            hard_timeout=of.OFP_FLOW_PERMANENT,
                            buffer_id=event.ofp.buffer_id,
                            actions=actions,
                            match=match)
      event.connection.send(msg.pack())
  elif self.arp_for_unknowns:
    if (dpid,dstaddr) not in self.lost_buffers:
      self.lost_buffers[(dpid,dstaddr)] = []
    bucket = self.lost_buffers[(dpid,dstaddr)]
    entry = (time.time() + MAX_BUFFER_TIME,event.ofp.buffer_id,inport)
    bucket.append(entry)
    while len(bucket) > MAX_BUFFERED_PER_IP: del bucket[0]
    self.outstanding_arps = {k:v for k,v in
     self.outstanding_arps.iteritems() if v > time.time()}

    if (dpid,dstaddr) in self.outstanding_arps:
      return
    self.outstanding_arps[(dpid,dstaddr)] = time.time() + 4
    r = arp()
    r.hwtype = r.HW_TYPE_ETHERNET
    r.hwlen = 6
    r.protolen = r.protolen
    r.opcode = r.REQUEST
    r.hwdst = ETHER_BROADCAST
    r.protodst = dstaddr
    r.hwsrc = packet.src
    r.protosrc = packet.next.srcip
    e = ethernet(type=ethernet.ARP_TYPE, src=packet.src,
                 dst=ETHER_BROADCAST)
    e.set_payload(r)
    log.debug("%i %i ARPing for %s on behalf of %s" % (dpid, inport,
     r.protodst, r.protosrc))
    msg = of.ofp_packet_out()
    msg.data = e.pack()
    msg.actions.append(of.ofp_action_output(port = of.OFPP_FLOOD))
    msg.in_port = inport
    event.connection.send(msg)

elif isinstance(packet.next, arp):
  a = packet.next
  log.debug("%i %i ARP %s %s => %s", dpid, inport,
   {arp.REQUEST:"request",arp.REPLY:"reply"}.get(a.opcode,
   'op:%i' % (a.opcode,)), a.protosrc, a.protodst)

  if a.prototype == arp.PROTO_TYPE_IP:
    if a.hwtype == arp.HW_TYPE_ETHERNET:
      if a.protosrc != 0:
        if a.protosrc in self.arpTable[dpid]:
          if self.arpTable[dpid][a.protosrc] != (inport, packet.src):
            log.info("%i %i RE-learned %s", dpid,inport,a.protosrc)
            if self.wide:
              msg = of.ofp_flow_mod(command=of.OFPFC_DELETE)
              msg.match.dl_type = ethernet.IP_TYPE
              msg.match.nw_dst = a.protosrc
              event.connection.send(msg)
        else:
          log.debug("%i %i learned %s", dpid,inport,a.protosrc)
        self.arpTable[dpid][a.protosrc] = Entry(inport, packet.src)

        self._send_lost_buffers(dpid, a.protosrc, packet.src, inport)

        if a.opcode == arp.REQUEST:

          if a.protodst in self.arpTable[dpid]:
```

```
                if not self.arpTable[dpid][a.protodst].isExpired():

                  r = arp()
                  r.hwtype = a.hwtype
                  r.prototype = a.prototype
                  r.hwlen = a.hwlen
                  r.protolen = a.protolen
                  r.opcode = arp.REPLY
                  r.hwdst = a.hwsrc
                  r.protodst = a.protosrc
                  r.protosrc = a.protodst
                  r.hwsrc = self.arpTable[dpid][a.protodst].mac
                  e = ethernet(type=packet.type, src=dpid_to_mac(dpid),
                               dst=a.hwsrc)
                  e.set_payload(r)
                  log.debug("%i %i answering ARP for %s" % (dpid, inport,
                   r.protosrc))
                  msg = of.ofp_packet_out()
                  msg.data = e.pack()
                  msg.actions.append(of.ofp_action_output(port =
                                                          of.OFPP_IN_PORT))
                  msg.in_port = inport
                  event.connection.send(msg)
                  return
      log.debug("%i %i flooding ARP %s %s => %s" % (dpid, inport,
       {arp.REQUEST:"request",arp.REPLY:"reply"}.get(a.opcode,
       'op:%i' % (a.opcode,)), a.protosrc, a.protodst))

      msg = of.ofp_packet_out(in_port = inport, data = event.ofp,
          action = of.ofp_action_output(port = of.OFPP_FLOOD))
      event.connection.send(msg)

def launch (fakeways="", arp_for_unknowns=None, wide=False):
  fakeways = fakeways.replace(","," ").split()
  fakeways = [IPAddr(x) for x in fakeways]
  if arp_for_unknowns is None:
    arp_for_unknowns = len(fakeways) > 0
  else:
    arp_for_unknowns = str_to_bool(arp_for_unknowns)
  core.registerNew(l3_switch, fakeways, arp_for_unknowns, wide)
```

3. Detener el módulo del controlador POX en ejecución

Se utilizó la combinación de teclas Ctrl+Z en el terminal del controlador para detenerlo.

4. Detener el proceso que escucha al puerto del controlador

El controlador no permite iniciar otro proceso en el mismo puerto, por lo que fue necesario matar el proceso anterior con el siguiente comando:

sudo fuser -k 6633/tcp

De esta manera se liberó el puerto 6633/TCP para ser usado con el nuevo módulo que tiene la solución los ataques de DDOS.

5. Ejecutar el nuevo módulo del controlador POX con la solución.

Se utilizó el siguiente comando en el terminal de controlador para ejecutar el nuevo módulo con la solución:

python ./pox.py forwarding.l3_edited

Los códigos del módulo de detección y módulo del controlador se encontraron en la plataforma GitHub en el siguiente enlace:
https://github.com/jagan103/DDos-SDN/tree/master/code

Participantes en la implementación y firmas de responsabilidad:

Nombre	Cargo	Fecha	Firma

ANEXO F

REPORTE DE DESEMPEÑO

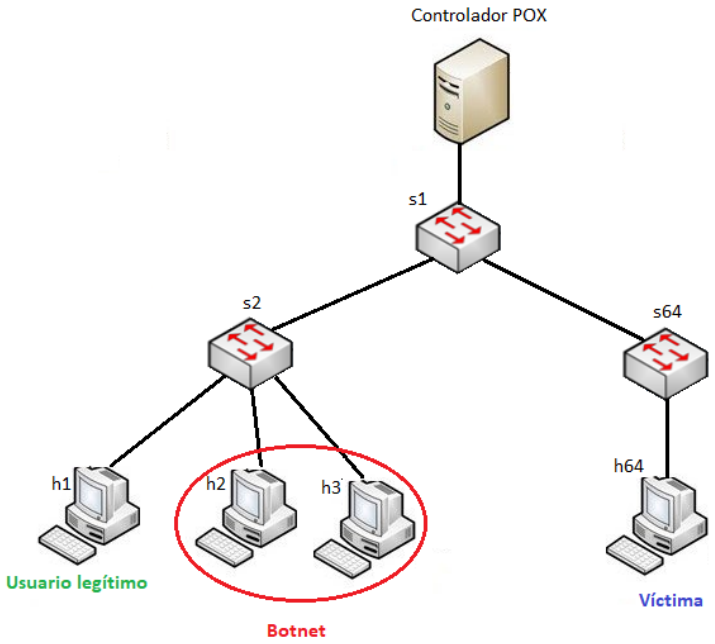
Fecha de inicio del monitoreo: 17/6/2021

Fecha de finalización del monitoreo: 17/6/2021

Para realizar el monitoreo se decidió generar diferentes tipos de ataques de DDoS de inundación en el escenario de prueba, siguiendo el esquema mostrado en la Figura 47.

Figura 47

Escenario de Prueba



El Multi-State Information Sharing and Analysis Center (MS-ISAC) contempla tres tipos de ataques de DDoS de inundación: SYN flood, UDP flood e ICMP flood (MS-ISAC, 2017).

El SYN Flood es uno de los ataques de DDoS más comunes observados por MS-ISAC. Ocurre cuando un atacante envía una sucesión de solicitudes de sincronización (SYN) del Protocolo de control de transmisión (TCP) al objetivo. Un ataque de inundación UDP es

similar a un SYN Flood en el sentido de que se utiliza una botnet para enviar una gran cantidad de tráfico al servidor de destino. Algunos puertos UDP comunes son 53 (DNS), 88 (Kerberos), 137/138/139/445 (Windows) y 161 (SNMP).(MS-ISAC, 2017)

En base a lo mencionado anteriormente se realizaron los ataques descritos en Tabla 31.

Tabla 31

Ataques Realizados en el Subproceso "Monitoreo"

Tipo de Ataque	Puerto Destino	Inicio del Ataque	Detección y Mitigación	Tiempo de Detección y Mitigación
UDP flood	53/UDP	21:54:11	21:54:13	00:00:02
	88/UDP	21:56:43	21:56:45	00:00:02
	137/UDP	21:59:23	21:59:25	00:00:02
	138/UDP	22:02:27	22:02:29	00:00:02
	139/UDP	22:03:54	22:03:56	00:00:02
	445/UDP	22:05:12	22:05:14	00:00:02
	161/UDP	22:07:16	22:07:18	00:00:02
SYN Flood	22/TCP	22:10:46	22:10:48	00:00:02
	80/TCP	22:11:13	22:11:15	00:00:02
ICMP flood	-	22:14:37	22:14:39	00:00:02

Resultados del monitoreo:

De acuerdo al monitoreo realizado se puede concluir que el mecanismo de detección y mitigación de ataques de DDoS implementado en el controlador SDN es efectivo para los

tipos de inundación de paquetes TCP flood, UDP flood e ICMP flood, es decir, que la detección no depende del tipo de paquete de ataque.

Nivel de cumplimiento de los objetivos:

El mecanismo implementado es efectivo para los ataques de DDoS de inundación TCP, UDP e ICMP.

Se cumple con el objetivo de detección temprana, ya que en los ataques simulados se tiene un tiempo de detección y mitigación de 2 segundos.

No conformidades:

El mecanismo implementado cumple con los objetivos planteados, sin embargo, dificulta al administrador de la red visualizar el momento en que se produjo y mitigó el ataque, pues dicha alerta se pierde en el terminal del controlador debido a la excesiva cantidad de mensajes que indican la entropía actual de la red. Esto también impide que se identifiquen otras eventualidades que suceden en el controlador, por lo que se dificulta el monitoreo de dicho equipo y por lo tanto de la red.

ANEXO G

Documento de Aprobación de la solución

Ibarra, 24/06/2021

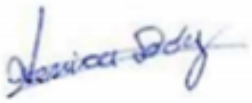
Referente al proyecto relacionado con la implementación de un mecanismo de detección y mitigación de ataques de DDoS hacia el controlador SDN, ejecutado por el administrador de la red, cumplo con informar lo siguiente:

Una vez que dicho proyecto ha pasado por las cuatro fases del ciclo PDCA, la solución implementada, técnicamente, se encuentra en condiciones de ser aprobada.

Cabe destacar que el proceso deberá ser realizado nuevamente de manera periódica cada seis meses o cuando haya cambios significativos en la red, esto con el fin de cumplir con el ciclo de mejora continua.

_____

Director del Departamento de TI

Versión:	Elaborado por:	Aprobado por:
1.0	 Ing. Jessica Báez	 MSc. Fabián Cuzme
Fecha: 24/06/2021	Fecha: 24/06/2021	Fecha: 04/07/2021