UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FINAL DE GRADO

Seguridad en Redes definidas por software (SDN)

Javier Ruipérez Cuesta

Tutor: José Óscar Romero Martínez

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2020-2021

Valencia, 8 de marzo de 2021

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

VLC/ CAMPUS VALENCIA, INTERNATIONAL CAMPUS OF EXCELLENCE

EMAS Gestión ambiental verificada REG.NO. ES-CV- 000030

Resumen

En este proyecto se va a analizar la seguridad en las Redes Definidas por Software (SDN). Las redes definidas por software son la evolución de las redes convencionales, y la aparición de estas SDN redes ha aportado nuevas ventajas y funcionalidades en las diferentes áreas donde se pueden aplicar. Se realizará un estudio de su evolución, centrándose en OpenFlow, protocolo más utilizado en SDN, así como los principales controladores y, sobre todo, se hará un especial hincapié en la seguridad.

La seguridad es una cualidad fundamental y un aspecto muy importante a tener en cuenta en la actualidad, pues las redes SDN son prácticamente nuevas, y debido a la separación del plano de control y del plano de datos se introducen nuevas vulnerabilidades y tipos de ataques. En este trabajo se explicarán diversos ataques que se pueden producir y así como las soluciones que hay para poder mejorar la seguridad de una red SDN.

Existen muchas herramientas que sirven para probar y analizar la seguridad de una SDN. En este proyecto se explicarán los aspectos a tener en cuenta sobre la seguridad y su utilidad en este tipo de redes.

Palabras clave: SDN, OpenFlow, controlador SDN, ataques de red, vulnerabilidades, seguridad.

Resum

En aquest projecte s'analitzarà la seguretat en les Xarxes Definides per Programari (SDN). Les xarxes definides per programari són l'evolució de les xarxes convencionals, i l'aparició d'aquestes SDN xarxes ha aportat nous avantatges i funcionalitats en les diferents àrees on es poden aplicar. Es realitzarà un estudi de la seua evolució, centrant-se OpenFlow, protocol utilitzat en SDN, així com els principals controladors i, sobretot, es farà un especial recalcament en la seguretat.

La seguretat és una qualitat fonamental i un aspecte molt important a tindre en compte en l'actualitat, perquè les xarxes SDN són pràcticament noves, i a causa de la separació del pla de control i del pla de dades s'introdueixen noves vulnerabilitats i tipus d'atacs. En aquest treball s'explicaran diversos atacs que es poden produir i així com les solucions que hi ha per a poder millorar la seguretat d'una xarxa SDN.

Existeixen moltes eines que serveixen per a provar i analitzar la seguretat d'una SDN. En aquest projecte s’explicaran els aspects a tindre en compte de la seguretat i la seua utilitat aquest tipus de xarxes.

Paraules clau: SDN, OpenFlow, controlador SDN, atacs de xarxa, vulnerabilitats, seguretat.

Abstract

This proyect will analyze security in Software Defined Networking (SDN). Software Defined Networks are the evolution of conventional networks and the emergence of these SDN networks has brought new advantages and functionalities in the differents areas where they can be applied. A study will be made of their evolution, focusing on OpenFlow, the protocol used in SDN, as well as the main controllers and above all there will be a special emphasis on security.

Security is a fundamental quality and a very important aspect to take into account nowadays, since SDN are practically new an due to the separation of the control plane and the data plane new vulnerabilities and types of attacks are introduced. This paper will explain various attacks that can be produced and the solutions available to improve the security of an SDN network.

There are many tools that can be used to test and analyze the security of an SDN. In this proyect we will explain aspects to bear in mind about security and its usefulness of this type of networks.

Keywords: SDN, OpenFlow, controller SDN, network attacks, vulnerabilities, security.

Índice

Capítulo 1. Introducción....3
1.1 Redes Definidas por Software (SDN)....3
1.2 Objetivos Principales....3
1.3 Desafíos en la implementación de seguridad de red SDN....3
Capítulo 2. Arquitectura SDN y OpenFlow....5
2.1 Arquitectura SDN....5
2.1.1 Capa de Infraestructura o de datos....5
2.1.2 Capa de Control....6
2.1.3 Capa de Aplicación....6
2.2 OpenFlow....6
2.2.1 Funcionamiento e implementación de reenvío de paquetes (OpenFlow).....8
Capítulo 3. Controladores....9
3.1 Controladores SDN....9
3.2 Controladores OpenFlow....10
3.2.1 NOX/POX....10
3.2.2 OpenDaylight (ODL)....11
3.2.3 Floodlight....11
3.2.4 Ryu....11
3.3 Resumen de los principales controladores SDN....12
Capítulo 4. Seguridad SDN....13
4.1 La seguridad en las SDN....13
4.2 Amenazas y vulnerabilidades en SDN....14
4.2.1 Superficies de ataque y vectores de amenaza en SDN....14
4.2.2 Problemas de seguridad en la arquitectura SDN....15
4.2.3 Ataques a la arquitectura SDN....16
4.3 SDN para seguridad y seguridad SDN....19
4.3.1 Mejorar la seguridad a través de SDN....19
4.3.2 Mejorar la seguridad en SDN....21
Capítulo 5. Clasificación y visión general de las soluciones de seguridad en SDN....25
5.1 Detección de amenazas....26
5.2 Seguridad basada en NFV (Nube)....27
5.3 Mitigación de ataques....28
5.3.1 Ataques de inundación/denegación de servicios....28

5.3.2 Ataques de canal lateral 29
5.3.3 Infiltración del dispositivo de control de acceso 29
5.4 Gestión del acceso y de la identidad del usuario 29
5.5 Evaluación de la seguridad 31
5.6 Análisis forense 31
5.7 Marco de seguridad SDN integrado 31
Capítulo 6. Propuestas de seguridad SDN 33
6.1 Science DMZ: Banco de pruebas de nube segura basado en SDN 33
6.1.1 Detalles de la implementación 33
6.1.2 Arquitectura del sistema 34
6.1.3 Diagrama de flujo del banco de pruebas Science DMZ 35
6.1.4 Aprendizaje y conclusión 36
6.2 Detección y mitigación de ataques DDoS en el plano de datos de SDN 36
6.2.1 Introducción 36
6.2.2 Implementación y despliegue de la propuesta 37
6.2.3 Resultados 40
6.2.4 Análisis de los resultados 44
Capítulo 7. Conclusiones 45
7.1 Investigaciones futuras 46
Capítulo 8. Bibliografía 48

Capítulo 1. Introducción

1.1 Redes Definidas por Software (SDN)

Las redes SDN (Software defined network) son conjunto de técnicas relacionadas con el área de redes computacionales, cuyo objetivo es favorece la implantación e implementación de servicios de red de una forma determinista, dinámica y escalable, evitando así al administrador de una red coordinar dichos servicios a nivel bajo. Toda esta implementación se lleva a cabo mediante la separación del plano de datos que es el encargado de enviar los datos (en este caso, tramas) y del plano de control que se encarga de gestionar los dispositivos. Por lo tanto, toda la inteligencia de red y lógica de control ha migrado desde los dispositivos de red a una entidad basada en software lógicamente centralizada conocida como el "controlador" de red.

Además, en redes SDN aparece el concepto de programabilidad de red ya que todas las operaciones de red deben describirse como programas de software integrando algoritmos, estructuras de datos y conceptos de programación que pertenecen al entorno de desarrollo de software. La seguridad es un aspecto sensible en las redes de comunicación y de datos, así que estas redes pueden beneficiarse de las características que poseen las redes SDN incluyendo la programabilidad de la red. Varios problemas de seguridad que a menudo amenazan a las redes convencionales se pueden resolver en SDN de una manera oportuna y fiable haciendo que se cumplan las aplicaciones de software de seguridad de la red. [1]

1.2 Objetivos Principales

El principal objetivo de este trabajo es el análisis de la seguridad en las actuales redes definidas por software. Para ello se tendrán en cuenta los siguiente apartados o subobjetivos:

- Analizar la tecnología de redes definidas por software, así como sus componentes, protocolos (OpenFlow) y funcionamiento.
- Estudio general de los principales controladores de red SDN.
- Analizar la seguridad en esta tecnología, observar puntos críticos, vulnerabilidades, ataques que pueden ocurrir (Ataques de red, vectores de amenazas, etc.).
- Ejemplos prácticos

1.3 Desafíos en la implementación de seguridad de red SDN

El término plano de control y plano de control, que es utilizado para intercambiar información con los dispositivos de red, introducen nuevos desafíos de seguridad que necesitan ser determinados. Para el punto de vista de un atacante, el principal objetivo es el controlador de red puesto que su papel en la red SDN es fundamental.

La seguridad es un aspecto a tener en cuenta tanto en las redes convencionales como en las redes SDN, ya que nos garantiza disponibilidad, integridad y privacidad de la información. Además, la seguridad implementada debe ser simple de configurar y efectiva para hacer de la red un entorno escalable, eficiente y seguro. La arquitectura funcional SDN se puede dividir en tres capas, es decir, la capa de aplicación, la capa de control la capa de datos. Cada una puede tener múltiples vectores de ataque, por lo que los elementos que hay en cada capa deben ser protegidos. [2]

Principales acciones a llevar a cabo son:

- **Proteger y asegurar el controlador de red:** Puesto que el controlador es el centro de control de la red, este necesita ser vigilado muy de cerca. Si el controlador SDN se cae, por ejemplo, debido a un ataque de vectores, también lo hace la red, lo que significa que la disponibilidad del controlador debe mantenerse continua.
- **Establecer Confianza:** La red debe tener privacidad e integridad, por eso es imprescindible proteger las comunicaciones de toda la red. Esto significa garantizar que el controlador SDN, los dispositivos que gestiona y las aplicaciones que se cargan en él sean entidades de confianza que se están operando como deberían.
- **Elegir un "*framework*" o marco de política robusto:** Se necesita un sistema de controles y balances para asegurarse de que los controladores SDN están realizando sus tareas como realmente queremos que se hagan.
- **Realizar un profundo análisis forense de la red:** Básicamente este punto nos sirve para determinar, en caso de un ataque, quien lo ha podido realizar para así reportarlo y poder proteger la red de cara a nuevos ataques en el futuro.

Capítulo 2. Arquitectura SDN y OpenFlow

2.1 Arquitectura SDN

La idea fundamental de las Redes Definidas por Software es clara: trata de trasladar el plano de control del plano de datos en los dispositivos físicos como switches unificándolo en un solo elemento externo a la red física, llamado controlador.

En lugar de utilizar protocolos como OSPF o BGP, el controlador es el componente software común que se encarga de tomar la decisión del reenvío y gestionar las tablas de enrutamiento de los dispositivos de la red, así como controlar la red desde un punto, lo que da lugar a configurar todo tipo de switches, enrutadores o cortafuegos mediante un software de gestión modelo, haciendo de la red un sistema abierto. Por lo tanto, esto implica la centralización lógica del control y gestión de todos los dispositivos de reenvío de red que a su vez promueve la gestión de red como actividad de toda la red. [3]

La programabilidad del controlar SDN ofrece a los operadores de red y a los clientes unas interfaces de programación que pueden sacar todos los detalles bajo el nivel de infraestructura permitiendo la posibilidad de simplificar la aplicación de las políticas y el comportamiento de reenvío de red a través de lenguajes con políticas de alto nivel mas expresivos en lugar de usar comandos específicos del proveedor. La arquitectura SDN la componen tres partes principales: Capa de Aplicación, de control y de datos, cada una de estos componentes expone sus propias subcapas, funcionalidades e interfaces. [4]

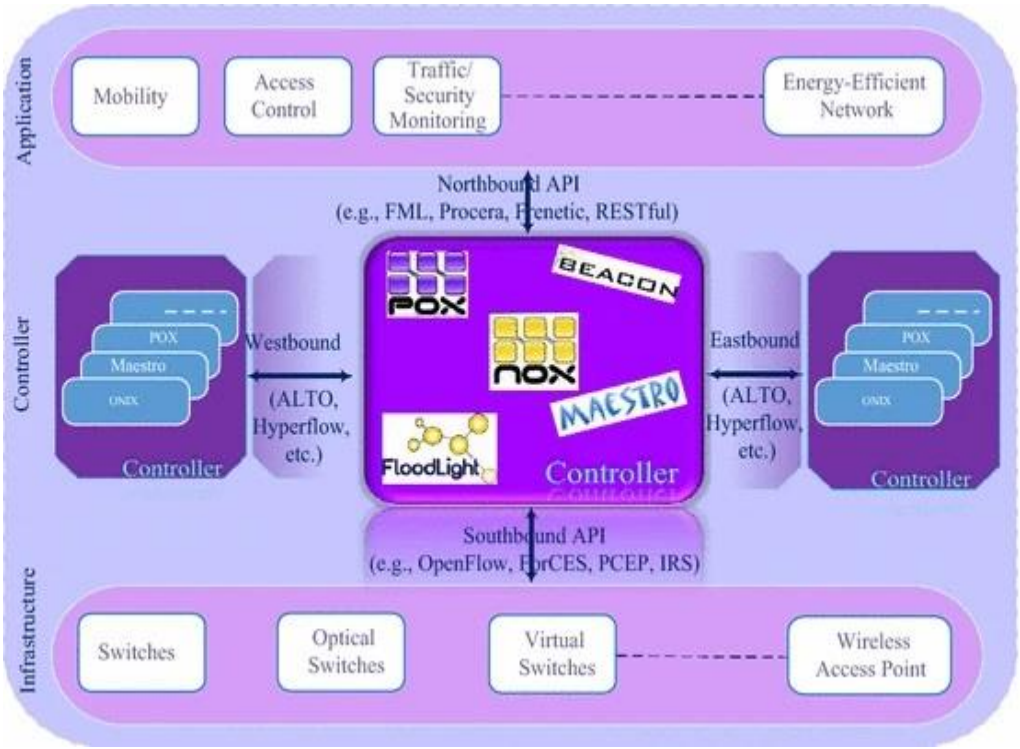

Figura 1. Arquitectura SDN y componentes. [5]

2.1.1 Capa de Infraestructura o de datos

Esta es la primera capa en la arquitectura SDN y se utiliza para el reenvío de un conjunto de paquetes basado en dispositivos de red que componen la infraestructura de red, cuyas funciones principales sirven para hacer cumplir las acciones de reenvío de paquetes de flujo de acuerdo con las instrucciones correspondientes proporcionadas por el controlador y para informar el estado de

la red cuando lo soliciten las aplicaciones de red. Hay varios protocolos que se pueden usar en la capa de datos, pero en este documento nos centraremos más en el protocolo OpenFlow ya que es de los más conocidos y con el que más trabajan las grandes empresas (Por ejemplo, Google). Los dispositivos de red más comunes que suelen estar presentes en esta capa son switches y routers. Un switch habitualmente funciona en la capa de enlace (L2). Puesto que la funcionalidad de un SDN basado en OpenFlow es definido por un controlador de red por software, nos referimos a todos los dispositivos de red como switches. [6]

2.1.2 Capa de Control

Como es de esperar en la capa de control, lógicamente centralizado, se sitúa el controlador de red que es el componente más importante de la arquitectura SDN pues tiene una vista centralizada de la red y gestiona tanto la capa de aplicación como la capa de datos o infraestructura. En definitiva, el plano de control de una configuración SDN consiste en uno o más controladores SDN que usan APIs abiertas para ejercer control sobre los switches de la red. Además de impulsar las reglas de reenvío a los switches, los controladores también vigilan el medio, de esta manera los controladores tienen la capacidad de tomar decisiones de reenvío integradas con la gestión del tráfico en tiempo real. [7]

Los controladores interactúan con el resto de la infraestructura de SND usando tres interfaces de comunicación, comúnmente llamadas sur (Southbound), norte (Northbound) e interfaces este y oeste. (Westbound/Eastbound) La división en sus funciones está definida de la siguiente manera:

- La interfaz hacia el Sur permite que le controlador se comunique, interactúe y maneje los elementos de envío. Aunque sí que es verdad que existen muchas otras soluciones patentadas como OnePK (Cisco) y Contrail (Jupiter Networks), OpenFlow es la implementación más común.
- La interfaz dirección Norte permite a las aplicaciones que se encuentra en la capa de aplicación programar los controladores haciendo modelos de datos abstractos y otras funcionalidades disponibles para ellos.
- Las interfaces Este/Oeste están pensadas para la comunicación entre grupos de controladores.

2.1.3 Capa de Aplicación

La capa que se encuentra en el nivel superior en el diagrama de bloques de la Figura.1 y que reside sobre la capa de control se llama capa de aplicación. En esta capa se encuentran las aplicaciones de negocio y el servicio que se encarga de dirigirlas. Estas aplicaciones SDN comunican sus requisitos a la red a través de una API (Northbound) que es la que conecta con la capa de control, y están diseñadas para satisfacer las necesidades de los usuarios. [8]

2.2 OpenFlow

OpenFlow fue propuesto originalmente por Nick McKeown en 2008 y además fue estandarizado por la ONF (Open Networking Foundation) en 2011. OpenFlow fue desarrollado para estandarizar la comunicación entre el conmutador OpenFlow y el controlador basado en software en la arquitectura SDN, permitiendo así la programación de las tablas de flujos por parte de las aplicaciones software. En la figura 2 podemos ver el avance de OpenFlow en orden cronológico. [9]

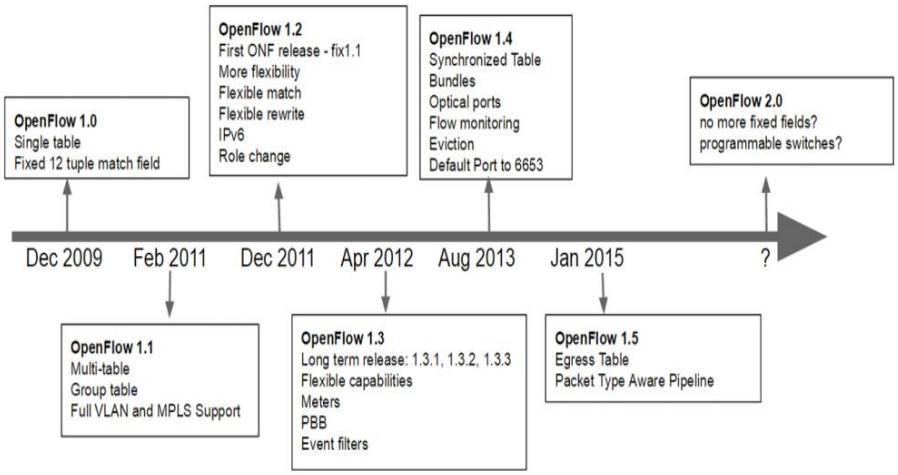

Figura 2. Orden cronológico de OpenFlow y sus distintas versiones. [10]

OpenFlow desacopla el plano de control del plano de datos y es el protocolo que más se usa habitualmente para la interfaz hacia el sur (Southbound). La arquitectura de OpenFlow comprende tres componentes principales, como se muestra en la figura 3; (1) los interruptores compatibles con OpenFlow constituyen el plano de datos; (2) el plano de control tiene uno o más controladores OpenFlow; (3) el plano de control está conectado con los interruptores a través de un canal de control seguro, es decir la interfaz OpenFlow.

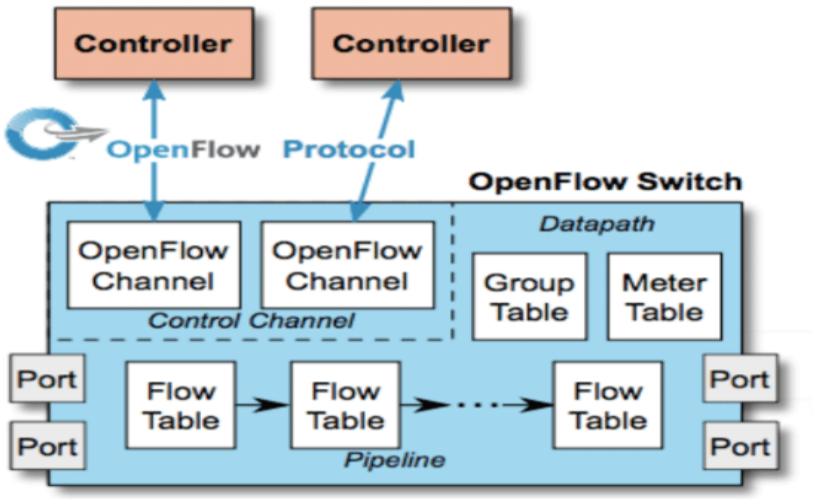

Figura 3. Arquitectura OpenFlow. [11]

Como bien hemos visto en la anterior figura, la arquitectura SDN se divide en, al menos, tres partes [12]:

- **Tabla de flujos**. Cada entrada de la tabla dispone de campos en los que debe buscar coincidencias con los paquetes entrantes, contadores e instrucciones sobre qué hacer con los paquetes que coinciden.
- **Canal seguro**. Conecta el dispositivo al controlador, permitiendo el envío de comandos y paquetes entre ambos mediante el protocolo OpenFlow.
- **Protocolo OpenFlow**. Proporciona una forma abierta y estandarizada en la comunicación entre el conmutador y el controlador, permitiendo al controlador añadir, eliminar, modificar y buscar en las entradas de la tabla de flujos a través del canal seguro.

2.2.1 Funcionamiento e implementación de reenvío de paquetes (OpenFlow).

Para entender el funcionamiento de cómo opera este protocolo, además de lo que ya sabemos del punto anterior de OpenFlow, también debemos entender que un conmutador conforme a OpenFlow en el plano de datos actúa simplemente como dispositivo de reenvío de paquetes según su tabla de flujo. Un cuadro de flujo comprende una lista de entradas de flujo. Cada entrada tiene campos de coincidencia, contadores e instrucciones. Cuando el paquete es recibido por el conmutador, este analiza el encabezamiento del paquete y la correspondencia se realiza con las entradas en el cuadro de flujo del conmutador. Si la entrada del diagrama de flujo se empareja con el encabezamiento del paquete, entonces se considera esa entrada en particular. Si se encuentran varias entradas de este tipo, pues en ese caso los paquetes se emparejarán dependiendo del orden de prioridad, de mayor a menor prioridad. Una vez terminado el proceso de emparejamiento y selección, se actualiza el contador de la entrada del diagrama de flujo. Finalmente, el conmutador ejecuta la acción sobre el paquete de acuerdo con la entrada del diagrama de flujo, por ejemplo, reenviar los paquetes al puerto, encapsular y reenviar al controlador, dejar ir al paquete y enviarlo a la red normalmente. En el caso de que el encabezamiento del paquete no coincida con la entrada del diagrama de flujo, el conmutador lo notifica al controlador y encapsula el paquete enviándolo al controlador con el mensaje "Packet_In". Cuando el conmutador recibe la notificación, el controlador encuentra la acción exacta para el paquete e instala una o más entradas adecuadas en la tabla de flujo del conmutador solicitante y luego los paquetes se envían de acuerdo con las reglas. [13]

Cuando todos los campos en la entrada del flujo coinciden con el contenido de todos los campos en el encabezado del paquete quiere decir que habrá una coincidencia entre un paquete entrante y una entrada de flujo. Después de una coincidencia exitosa, una de cada seis instrucciones incluida en la entrada de flujo podría ser ejecutada. Después, se ejecuta cierta acción sobre el paquete de acuerdo con las instrucciones. Cuando el proceso de emparejamiento de campos termina, un paquete de datos podría, o bien ser reenviado o bien ser descartado por el interruptor. Para entender un poco mejor la implementación de reenvió de paquetes (Packet Forwarding) nos apoyaremos en la siguiente figura 4. [14]

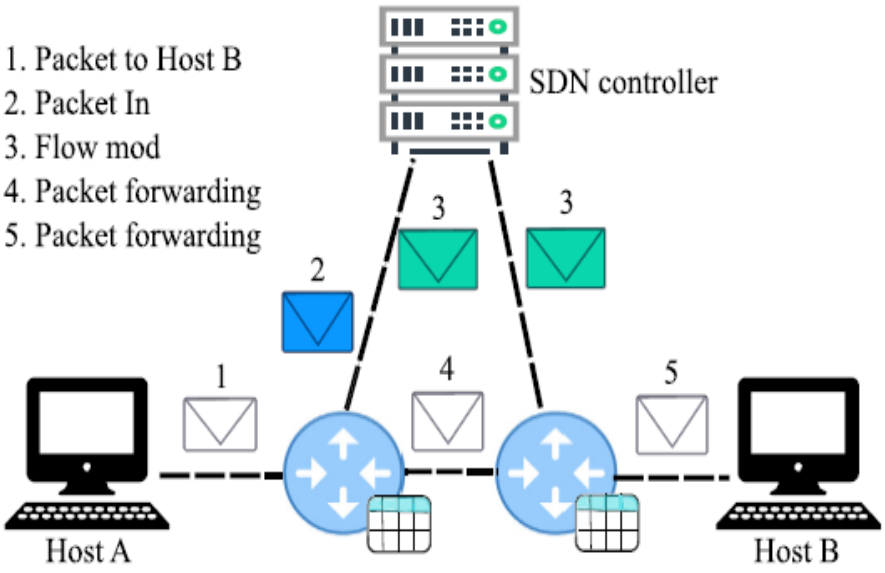

Figura 4. Reenvío de paquetes en OpenFlow. [15]

Los pasos del 1 al 5 ilustran el proceso seguido de acuerdo con la especificación OpenFlow para enviar un paquete del Host A al Host B. En el paso 3 el interruptor le pide al controlador instrucciones de reenvío y en el paso 4 el controlador instala nuevas reglas de reenvío en el interruptor.

Capítulo 3. Controladores

3.1 Controladores SDN

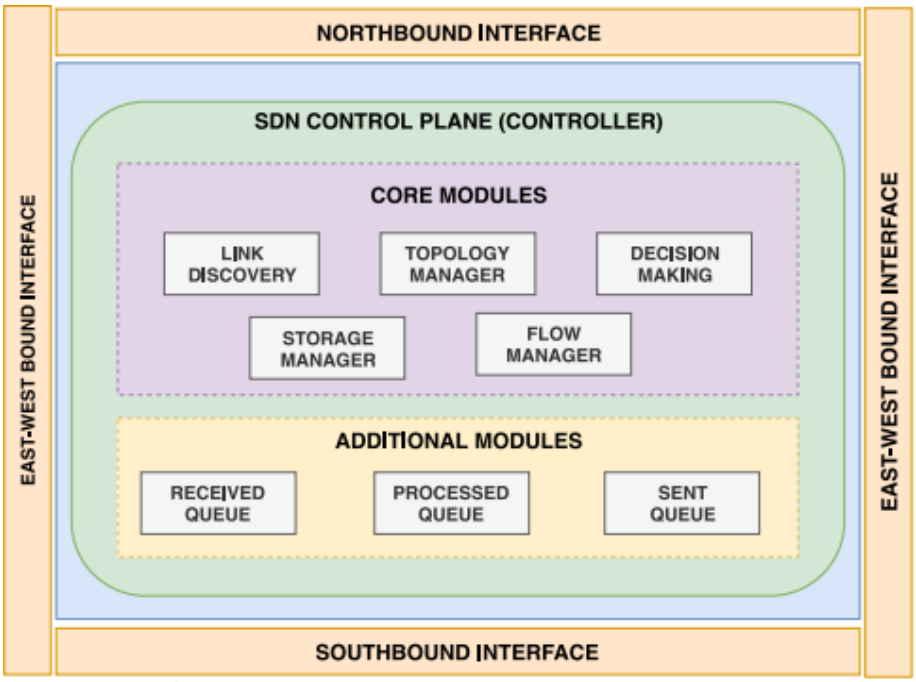

Figura 5. Arquitectura interna del plano de control SDN. [16]

El controlador es el cerebro de la operación de las redes SDN. Es él quien toma las decisiones, implementa las reglas de la red, ejecuta las instrucciones que le proporcionan las diferentes aplicaciones y las distribuye a los diferentes dispositivos de la capa física de la red. Se encuentra en la capa de control, concretamente, en un extremo entre los dispositivos del plano de datos y las aplicaciones de alto nivel de la capa de aplicación. El controlador es quien determina la manera de manejar los paquetes que no encajan en ninguna de las entradas de las tablas de flujo y quien se encarga de gestionar dichas entradas, añadiendo o eliminando a través del canal seguro a los dispositivos OpenFlow. Las entradas de flujo pueden ser añadidas a un dispositivo de plano de datos en un (1) modo proactivo, en el que las reglas de flujo se envían a los dispositivos de plano de datos tan pronto como el controlador se entera de ello; o (2) modo reactivo, en el que el controlador envía entradas de flujo a los dispositivos de plano de datos solo cuando es necesario. En el modo reactivo, cuando los dispositivos de plano de datos envían solicitudes de configuración de flujo al controlador, este primero comprobará el flujo con las políticas de la capa de aplicación y decidirá qué acciones van a llevarse a cabo. A continuación, el controlador determina una ruta para que los paquetes la superen e instala nuevas entradas de flujo en cada dispositivo a lo largo de la ruta. Las entradas de flujo que se agregan tienen valores de tiempo de espera específicos que indican a los dispositivos del plano de datos cuánto tiempo deben almacenarse en sus tablas de reenvío en caso de inactividad antes de eliminar la entrada. El equilibrio que hay entre el retardo de configuración y la memoria necesaria para mantener el reenvío en su entorno, determina la selección del administrador de la red. Además, en el modo reactivo, se ofrece a los administradores la posibilidad de estar de acuerdo con las condiciones presentes de la red. [17]

A parte del modo de funcionamiento, los administradores se enfrentan otro diseño, como es, la elección sobre la granularidad del flujo, donde el compromiso se encuentra entre la flexibilidad y la escalabilidad. Esto es similar a las rutas agregadas de los switches tradicionales. Mientras que las reglas de flujo precisas ofrecen flexibilidad y una capa adicional de seguridad. Por lo tanto, hay diferentes implementaciones de controladores. Desde un simple software que dinámicamente

añada y suprima flujos, donde el administrador controla toda la red de switches OpenFlow y es el responsable del proceso de todos los flujos, hasta una implementación con múltiples administradores, cada uno con diferentes cuentas y passwords, que les permite gestionar diferente conjunto de flujos. (Es lo que se podría asimilar a virtualizar una red con múltiples propietarios)

3.2 Controladores OpenFlow

Los principales controladores basados en OpenFlow que hay disponibles y que son los que habitualmente más se han utilizado son [18]:

3.2.1 NOX/POX

Figura 6. Controlador NOX. [19]

NOX es un controlador muy utilizado en la primera generación, puesto que su primera versión fue el primer controlador de OpenFlow. NOX es de código abierto, considerablemente usado y estable. En su primera versión este controlador, llamado NOX clásico se escribió en C++ y Python, pero este hoy en día ya no está en uso. Su nueva versión mucho más nueva solo usa C++, es mucho más rápido, todavía recibe actualizaciones y es soportado por varios equipos.

El funcionamiento de este controlador como el de la mayoría trabaja monitorizando eventos y proporciona una plataforma para programar una serie de tareas a realizar ante cada evento. NOX es recomendado para quienes programan en C++ y quieran usar instrucciones poco complejas, además por su simplicidad suele tener buenos resultados en cuanto a velocidad.

Figura 7. Controlador POX. [20]

Fundamentalmente POX es una versión de NOX, pero escrita en Python, con la desventaja de que solo soporta OpenFlow versión 1.0 y al ser Python no es tan rápida como el C++ de NOX. Actualmente POX recibe actualizaciones por lo que es muy utilizado y, además, es relativamente fácil de leer y escribir el código del mismo, por lo que, si se sabe programar en Python, este controlador es muy recomendable. También permite una rápida programación, por lo que se suele usar para hacer demostraciones, experimentos e investigaciones.

3.2.2 OpenDaylight (ODL)

Figura 8. Controlador OpenDaylight. [21]

OpenDaylight es un proyecto impulsado por la Linux Foundation y que, según su página oficial, es el controlador SDN de código abierto más implementado en la actualidad. Este software está escrito en Java, pero a diferencia del resto de controladores, este se trata de un controlador descentralizado que busca acelerar el proceso de aceptación de las redes SDN con un plataforma común y robusta a todas las redes. Cuenta con una interfaz Eastbound para hacer posible la comunicación entre los diferentes módulos del controlador, en unión con las otras dos interfaces Norte y Sur.

3.2.3 Floodlight

Figura 9. Controlador Floodlight. [22]

Floodlight se trata de un controlador de código abierto escrito en Java que es compatible con OpenFlow. El desarrollo de Floodlight fue llevado a cabo por la comunidad de ingenieros de Big Switch Networks y que en principio formaría parte del proyecto OpenDaylight, sin embargo, las diferencias que hubo entre Big Switch y Cisco hicieron que ambas empresas tomasen diferentes caminos y formaran proyectos independientes. Este controlador destaca entre los anteriores, aparte de estar escrito en Java, en que es el único que cuenta con una API REST [Aquí va lo que significa] para la interfaz Northbound haciendo posible la incorporación de aplicaciones externas.

3.2.4 Ryu

Figura 10. Controlador Ryu. [23]

Ryu es un controlador de código abierto completamente basado en Python que proporciona componentes de software con API bien definidas, las cuales facilitan a los desarrolladores la creación de nuevas aplicaciones de control y gestión de redes. Ryu soporta distintos protocolos para la gestión de dispositivos de red, entre ellos, OpenFlow. Unas de sus principales ventajas son que integra OpenStack con OpenFlow y que todo el código está disponible bajo la licencia de Apache 2.0.

3.3 Resumen de los principales controladores SDN

Nombre	NOX	POX	OpenDaylight	Floodlight	Ryu
Lenguaje	C++	Python	Java	Java	Python
Arquitectura	Centralizado	Centralizado	Descentralizado	Centralizado	Centralizado
Sistema Operativo	Linux	Linux, Windows, MacOS	Linux, Windows, MacOS	Linux, Windows, MacOS	Linux, MacOS
OpenFlow	Sí	Sí	Sí	Sí	Sí
OpenStack	No	No	Sí	Sí	Sí
Modularidad	Baja	Baja	Alta	Moderada	Moderada
Documentación	Limitada	Limitada	Muy buena	Buena	Muy buena

Tabla 1. Comparación entre controladores. [24]

Capítulo 4. Seguridad SDN

4.1 La seguridad en las SDN

Después de ver los elementos más importantes que forman una red definida por software, explicarlos y desarrollar sus características principales, así como las diferencias que hay con las redes tradicionales, se da comienzo a exponer y analizar los desafíos a nivel de seguridad que presentan el uso de estas redes SDN.

Como en cualquier otro ámbito o estructura de red, la seguridad es un aspecto importante a tener en cuenta ya que, con la mínima sospecha que se pueda tener se puede convertir en futuras debilidades que al fin y al cabo terminan poniendo en riesgo a nuestro sistema. Con el auge de infraestructuras de red modernas que sustentan aplicaciones cada vez más consistentes y dinámicas, como el Internet de las cosas, las redes sociales, los servicios en la nube, las aplicaciones móviles, etc. Por todo esto y mucho más, es necesario desarrollar tecnologías que se ajusten a la complejidad que requieren las aplicaciones ascendientes y a forma de atacar de sus atacantes.

Para poder entender la seguridad en SDN, antes debemos entender la seguridad asocia a un sistema informático, pero ¿Qué se entiende por seguridad asociada a un sistema informático? Pues bien, para poder decir que un sistema es seguro se deben dar tres atributos de seguridad esenciales: confidencialidad, integridad y disponibilidad. Además de las características que se ha mencionado anteriormente, se suelen incluir otras como son la autenticidad y no repudio. La confidencialidad, requiere que la información sea accesible únicamente por aquellos que estén autorizado, es decir, garantiza que la información privada o comprometida sobre diferente datos o personas no sea mostrada a usuarios no autorizados. La integridad hace posible que la información y el funcionamiento del sistema se mantenga inalterada ante intentos maliciosos. La disponibilidad asegura que el sistema informático pueda seguir trabajando sin sufrir ninguna degradación en cuanto a accesos y, además, ofrezca a los usuarios autorizados los recursos que requieran cuando éstos los necesiten. La autenticidad vela para que los usuarios puedan ser verificados como quienes dicen ser y que las entradas que llegan al sistema provengan de una fuente fiable. Por último, el no repudio, que garantiza al emisor que la información ha sido entregada y ofrece una prueba al receptor del origen de la información recibida. [25]

Las características de SDN como la visibilidad de toda la red, la inteligencia de la red centralizada y la programabilidad de la red reformaron la manera en que se realizan las tareas de control de paquetes y de la red básica en las redes programables. Sin embargo, estas características y la propia arquitectura SDN introducen nuevos retos de seguridad, a parte de los que ya existían en las redes tradicionales, y ataques de superficies que no están presentes en los despliegues de las redes convencionales. De acuerdo con esta afirmación y teniendo en cuenta los beneficios para el control de la red y el reenvío de paquetes que se pueden aprovechar de las características de SDN, se puede ver que la seguridad de SDN tiene una doble connotación: en primer lugar, la explotación de las características y mecanismos para proteger, reaccionar y proporcionar esquemas de mitigación contra riesgos de seguridad conocidos, ya sea mediante la introducción de nuevas propuestas de seguridad o mediante la ampliación de la funcionalidad de los sistemas y dispositivos de seguridad existentes. Y, en segundo lugar, el diseño de una arquitectura SDN segura que tenga como objetivo proporcionar un comportamiento proactivo contra los nuevos ataques de superficies y brechas de seguridad introducidas por la propia arquitectura SDN. [26].

Se considera oportuno que para poder entender los siguientes los siguientes apartados es necesario explicar brevemente dos definiciones clave como son [27]:

- **Vulnerabilidad:** es una debilidad en un sistema/entorno de información que pone en riesgo la seguridad de la información permitiendo que un atacante pueda poner en compromiso la integridad, disponibilidad, confidencialidad, autenticidad o el no repudio de ese mismo sistema.

- **Amenaza:** es toda acción que emplea una vulnerabilidad para atacar contra la seguridad de un sistema de información. Es decir, que podría tener un efecto potencialmente negativo sobre algún elemento del sistema atacado. Las amenazas pueden proceder de ataques como fraude, robo, virus; sucesos físicos como incendios, inundaciones; o negligencia y decisiones institucionales (mal manejo de contraseñas, no usar cifrado).

4.2 Amenazas y vulnerabilidades en SDN

Ahora que ya hemos definido brevemente algunos conceptos relacionados con la seguridad en SDN, podemos entender mejor lo que se va a explicar en los siguientes apartados.

SDN como cualquier nueva tecnología, tiene sus pros y sus contras. En lo que respecta a la seguridad, por ejemplo, la tecnología SDN puede ser aprovechada para mitigar totalmente algunos riesgos y vulnerabilidades que se explotan habitualmente en las redes convencionales. Por desgracia, la tecnología SDN introduce nuevas vulnerabilidades y vectores de amenaza que son inherentes a su novedosa arquitectura. De hecho, la separación de los planos de control y de datos y la centralización lógica de toda la inteligencia de la red exponen un único punto de fallo que puede ser explotado para comprometer toda la red SDN. Por lo tanto, la seguridad de SDN es un área importante de investigación. El diseño centralizado de SDN puede introducir de seguridad, como los ataques de denegación de servicio distribuidos (DDoS) contra el controlador de la SDN.

En un primer apartado se hablará de una visión generalizada de las superficies de ataque más visibles y los vectores de amenaza que se han identificado en los planos e interfaces de la arquitectura SDN. En la parte intermedia, se describen detalladamente los problemas, efectos y consecuencias derivados de la explotación de las superficies de ataque y vectores de amenaza presentes en dicha arquitectura SDN. Y, por último, una lista de los ataques y comportamientos maliciosos más comunes que tienen como objetivo las diferentes capas de la arquitectura SDN. [28]

4.2.1 Superficies de ataque y vectores de amenaza en SDN

Al igual que en las redes convencionales, cada instancia, protocolo, dispositivo o capa de red que participa en una red SDN puede ser objeto de un mal uso intencionado que en algunos casos se aprovecha para exponer los fallos del sistema. Este argumento es suficiente para poder afirmar que cada elemento o capa que forma parte de la arquitectura SDN es un vector de amenaza o superficie de ataque, es decir, cualquier mala configuración o despliegue inadecuado de cualquier elemento de una red SDN puede ser una fuente emergente de vulnerabilidades y riesgos de seguridad. A partir de este punto consideraremos el plano de control como una unión de la capa de aplicación y la capa de sistema operativo de red, siguiendo este enfoque, las vulnerabilidades y los ataques dirigidos a las aplicaciones de red o a las APIs del norte se considerarán vulnerabilidades o ataques al plano de control sin distinción alguna.

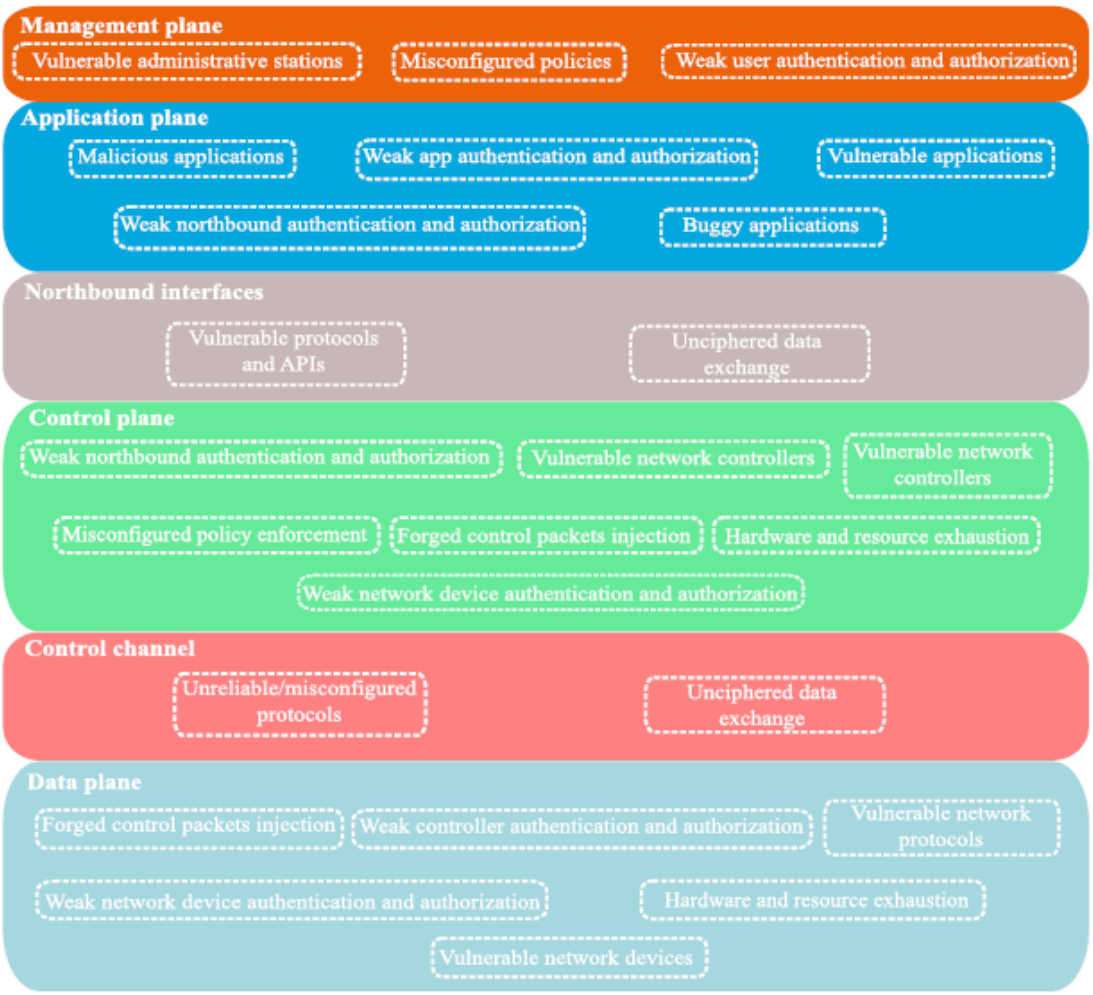

Figura 11. Amenazas y vulnerabilidades en la arquitectura SDN. [29]

En la Figura 11, cada plano/interfaz de la arquitectura SDN agrupa una lista con las superficies de amenaza más relevantes que podrían ser aprovechadas por usuarios maliciosos para comprometer una red bajo el paradigma SDN.

4.2.2 Problemas de seguridad en la arquitectura SDN

Todos los ataques a la seguridad de la red se pueden clasificar según el objetivo principal del ataque en cuestión, por ejemplo, la escucha de una interfaz de control se puede etiquetar como un ataque cuyo objetivo principal es manipular los datos privados y sensibles que se intercambian entre los aparatos de la red, lo que representa una divulgación no autorizada de información. En la siguiente lista se describen los problemas y amenazas a las que se puede ver expuesto la seguridad de un sistema basado en redes definidas por software, explicando las superficies de ataque, los comportamientos y los fallos de seguridad [30].

- **Acceso no autorizado:** Los atacantes se conceden a sí mismos acceso no supervisado a elementos de la red SDN, pudiendo manipular mecanismos de control débiles, lanzando ataques de fuerza bruta contra terminales administrativas y API REST que están expuestas a sesiones de registro, o explotando vínculos de vulnerabilidad en los componentes de la red y luego instalando dispositivos falsos.
- **Fuga de datos:** Hay varias maneras en las que un atacante pueda determinar una superficie de ataque para sustraer información sensible sobre la red, por ejemplo, un atacante puede inferir sobre el comportamiento de reenvío de la red mediante la difusión de paquetes a elementos de la red. Un atacante que logre comprometer una aplicación de

red vulnerable, puede obtener acceso a las bases de datos de políticas de red y a otros datos internos almacenados en la red. Los canales inseguros pueden ser aprovechados para el espionaje y el rastreo de paquetes. Por último, los ataques de suplantación de identidad de dispositivos permiten a un atacante recibir información que originalmente estaba destinada a un elemento de la red comprometido por lo que el atacante puede generar entonces un conjunto de solicitudes de flujo falsas que desembocan en un ataque de denegación de servicios (DoS).

- **Modificación de datos no autorizado:** Si los atacantes son capaces de apropiarse el controlador y de aprovechar los protocolos vulnerables, entonces estos tendrían el control de todo el sistema. Partiendo desde esta posición, los atacantes pueden modificar o anular las reglas de flujo existentes a través de aplicaciones maliciosas lo que permitiría dirigir los paquetes a través de la red en su beneficio. Además, un acceso no autorizado al almacenamiento interno ofrece a un atacante la posibilidad de introducir políticas de red contradictorias. Por otra parte, la falta de protocolos de intercambio seguro de paquetes, como TLS (Trasnport Layer Security), hace posible que los cambios en la topología de la red puedan ser incluidos aprovechando las desconfiguraciones del protocolo junto con la suplantación de dispositivos y los ataques de inyección de paquetes.
- **Aplicaciones comprometidas/maliciosas:** Como el controlador actúa como una contemplación del plano de datos para las aplicaciones y SDN permite que las aplicaciones de terceros se integren en la arquitectura; una aplicación maliciosa puede tener un efecto perjudicar en la red. De la misma manera que si una aplicación está mal diseñada o con fallos podría involucrar involuntariamente vulnerabilidades en el sistema.
- **Interrupción del servicio:** Una de las principales debilidades de seguridad de SDN viene dada por su propia arquitectura. Debido a la necesidad de comunicación entre controlador y el dispositivo de red, un atacante podría provocar tres fuentes principales de interrupción de servicio, ataques de inundación, ataque de inyección de paquetes y envenenamiento de la topología.

4.2.3 Ataques a la arquitectura SDN

Todas las capas e interfaces son susceptibles a ciertos ataques específicos que pueden comprometer los componentes de la red que residen en la propia capa o dirigirse a elementos de otra capa. Los problemas de seguridad más comunes de la SDN incluyen ataques en las diferentes capas de la arquitectura SDN. Veamos en los siguientes párrafos el conjunto de ataques más habituales que podrían producirse en cada una de estas capas. A continuación, se muestra una imagen para ilustrar una arquitectura SDN típica y desde donde pueden venir los atacantes.

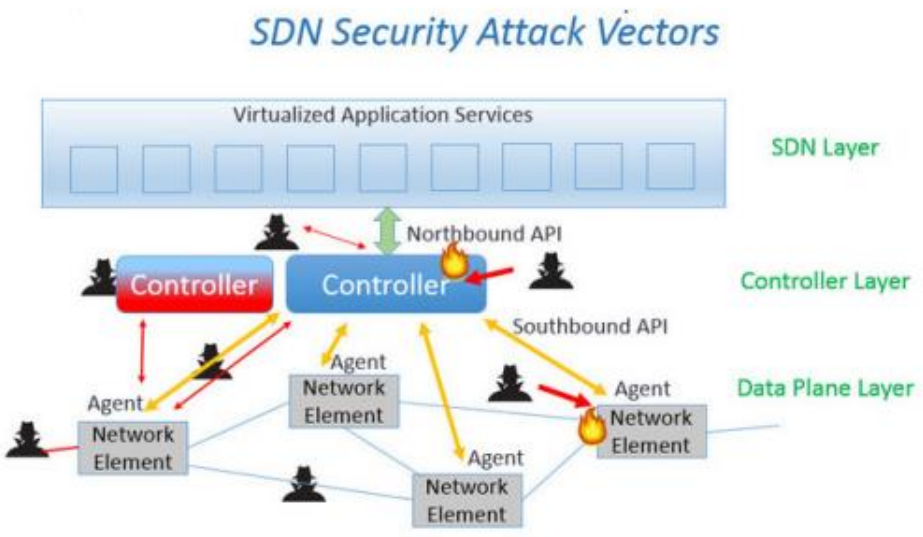

Figura 12. Arquitectura SDN y atacantes. [31]

1) Capa de aplicación [32]

Terminación de aplicaciones por abuso de privilegios fijados y autoridad: Las aplicaciones de terceros y de control con autoridad limitada en el sistema de red pueden verse comprometidas para englobar la ejecución de comandos del sistema que se utilizan principalmente para desconectar/apagar ciertas APIs o aplicaciones de red sensibles.

Neutralización de servicios: Las aplicaciones maliciosas que se instalan con éxito sobre el controlador, pueden utilizarse para manipular y manejar los paquetes de control, permitiendo así ejecutar una interrupción del servicio, por ejemplo, mediante el descarte de los paquetes de control para evitar que lleguen a dichas aplicaciones.

Ataques a las APIs vulnerables del Northbound (Norte): Hay muchas APIs Northbound que son utilizadas por los controladores SDN. Las APIs Northbound podrían utilizar lenguaje como Python, Java, C, XML, JSON, entre otros, si hay algún tipo de error de configuración o de programación, el atacante podría aprovechar la API Northbound vulnerable, entonces el atacante tendría todo el control de la red SDN a través del controlador. Si el controlador carece de cualquier forma de seguridad para la API de entrada norte, entonces el atacante podría ser capaz de crear sus propias políticas de SDN y así obtener el control del entorno SDN.

A menudo, hay una contraseña que viene definida por defecto y se utiliza para una API REST que es trivial de determinar. Si un despliegue de SDN no realiza el cambio de esta contraseña por defecto, el atacante crearía paquetes hacia la interfaz de gestión del controlador permitiéndole a este consultar la configuración del entorno SDN y poner su propia configuración.

2) Capa de control [33]

En la capa de control se encuentra el controlador SDN y es evidente que es el objetivo principal de ataque. Un atacante trataría de apuntar al controlador SDN con varios propósitos, algunos de ellos son;

Túnel dinámico de reglas de flujo: Los atacantes podrían instanciar nuevos flujos suplantado los mensajes API de dirección norte o suplantando los mensajes de dirección sur hacia los dispositivos de red. Si un atacante es capaz de falsificar con éxito los flujos del controlador, entonces el atacante tendría la capacidad de permitir que el tráfico fluya a través de la SDN a su voluntad y posiblemente pueda eludir las políticas en las que se puede confiar para la seguridad.

Envenenamiento del controlador: Los protocolos de red vulnerables y las aplicaciones con errores o maliciosas pueden utilizarse para envenenar la información del controlador. Un atacante podría intentar realizar un DDoS o DoS del controlador o utilizar otro tipo de método, como el protocolo LLDP (Link Layer Discovery Protocol), para hacer que el controlador falle. El atacante también podría intentar algún tipo de ataque de consumo de recursos en el controlador para atascarlo y hacer que responda con mayor lentitud a los eventos Packet_In y hacer que envíe muy lentamente los mensajes Packet_Out.

Uso indebido del sistema operativo: La mayoría de las veces, los controladores SDN se ejecutan en algún tipo de sistema operativo. Si el controlador SDN se ejecuta en un sistema operativo de propósito general, entonces las vulnerabilidades de ese sistema operativo se convierten en vulnerabilidades para el controlador. También las configuraciones erróneas y los dispositivos de plano de datos falsos pueden explotar las vulnerabilidades del controlador para lograr diferentes objetivos.

Inundación de paquetes: Por medio de hosts o/y switches comprometidos, un adversario puede difundir paquetes de red maliciosos que los receptores de la infraestructura de red traducen en mensajes de paquetes de entrada al controlador debido a la ocurrencia de un considerable porcentaje de fallos en la tabla del switches. El controlador podría desperdiciar todos sus recursos computacionales respondiendo a la gran cantidad de paquetes de entrada que llegan a través de su interfaz hacia el sur.

Creación de otro controlador: Sería perjudicial que un atacante creara su propio controlador e hiciese creer a los que elementos de la red que los flujos proceden del controlador "falso". Entonces el atacante podría crear entradas en las tablas de flujo de los elementos de red y los ingenieros SDN no tendrían visibilidad de esos flujos desde la perspectiva del controlador. En este caso, el atacante tendría el control total de la red.

3) Capa de datos [34]

Negación de servicios aprovechando el ataque de envenenamiento ARP: Un atacante puede lograr el aislamiento del switch mediante la suplantación del controlador. Usando el ataque de envenenamiento ARP (Address Resolution Protocol), el atacante se apropia la identidad del controlador y obliga a un switch de destino a abandonar la conexión que tiene con el controlador SDN y conectarse en su lugar al controlador "falso". Lo que el atacante logra con este ataque es la desconexión el switch a la red.

Modificación de las reglas de flujo: Los atacantes pueden manipular la información alojada en las tablas de flujo de los switches, sobrescribiendo o borrando las reglas de flujo existentes. Los atacantes pueden planificar este ataque desde una aplicación comprometida o desde un controlador de red comprometido.

Inundación de reglas de flujo: Para llevar a cabo este ataque se utilizan técnicas de ataque de canal lateral, en el que se envían un tipo de paquetes que generan una falta en la tabla, forzando al switch a realizar una solicitud al controlador para la instalación de una nueva regla de flujo. Dada la situación en el que el atacante ha deducido dicha información, entonces es capaz de lanzar un ataque de inundación de tabla de flujo obligando al switch a pedir constantemente nuevas reglas y a llenar su tabla de flujos; este comportamiento tiene efectos negativos en el rendimiento y en la estabilidad del conmutador.

Inyección de paquetes de control malformados: Consiste en que el switch en cuestión recibe paquetes de control manipulados que contengan cabeceras malformadas que son falsificadas para expones las vulnerabilidades existentes.

Ataques de canal lateral: También conocidos como ataques de reconocimiento, en el que el atacante querría espiar los flujos para ver que flujos están en uso y que tráfico se está permitiendo a través de la red. El atacante puede vigilar la comunicación hacia el sur entre el elemento de red y el controlador SDN. Esta información podría ser útil para un ataque de repetición o simplemente para fines de reconocimiento. Por ejemplo, un atacante podría registrar el RTT (tiempo de ida y vuelta) experimentado por un paquete concreto cuando se ha enviado a un switch determinado, dicha información podría ser aprovechada más tarde para lanzar un ataque de inundación de la tabla de flujo en ese switch.

Ataques	**Puntos Vulnerables**	**Objetivos**				
(A: Aplicación, NB: Northbound, C: Control, SB: Southbound, D: Datos)		**A**	**NB**	**C**	**SB**	**D**
Abuso de privilegios y autoridad	Aplicaciones de terceros	X	X	X		
Interrupción de servicio	Malware	X		X		
Túnel de reglas de flujo dinámico	Malware y switches	X		X		
Vista envenenada de la red	Protocolos y servicios de red	X	X	X	X	X
Inundación de paquetes	Controlador y switches			X	X	X
Inundación de la tabla de switches	Controlador y switches			X	X	X
Man in the Middle	Canal control e interfaz sur			X	X	X
Inundación de tablas de flujo	Desconexión de switches			X	X	X
Uso indebido del Sistema Operativo	Controlador	X		X	X	X

Tabla 2. Ataques a la arquitectura SDN. [35]

La Tabla 2 hace un pequeño resumen de algunos ataques que se pueden lanzar con las redes SDN, haciendo hincapié en los puntos vulnerables y los objetivos que tiene cada ataque con el fin de estables una relación causa-efecto entre una fuente puntual de un ataque y todos los componentes de la arquitectura SDN que pueden ser objetivo. Dicha relación podría ser útil para determinar los niveles de gravedad de daños que pueden atribuirse a determinados ataques.

4.3 SDN para seguridad y seguridad SDN

4.3.1 Mejorar la seguridad a través de SDN

Podemos sugerir tres características relevantes de una red SDN que pueden ser usadas para implementar una gran variedad de soluciones de seguridad en el entorno de una red SDN. En este apartado se describirá brevemente estas características y como estas podrían ser aprovechadas para mejorar la seguridad en SDN. [36]

La primera característica es el Control de flujo dinámico, la cual beneficial a la seguridad de dos formas distintas: 1) garantizando y reforzando la funcionalidad de los middleboxes (Dispositivo de red informática que trasforma, inspecciona, filtra y manipula el tráfico con fines distintos al reenvío de paquetes) de seguridad como una composición de diferentes agrupaciones de reglas de control de flujo distribuidas en toda la infraestructura de la red; y 2) en forma de aplicaciones de red instaladas en la capa de control vinculadas al controlador a través de una interfaz con dirección norte, por lo que no se necesitan dispositivos de hardware adicionales que pueden sustituirse incorporando reglas de seguridad en dispositivos de red básicos destinados principalmente al reenvío de paquetes. Por lo tanto, el control de flujo dinámico de SDN puede aprovecharse para el despliegue de, por ejemplo, firewalls internes y perimetrales, listas de control de acceso (ACL) completas y activas, y esquemas básicos de redireccionamiento de tráfico. Las reglas de flujo dinámicas pueden ser dirigidas para forzar a los dispositivos de red y así redirigir ciertos tipos de tráfico a sistemas de seguridad más especializados o al propio controlador para que analice el tráfico a través de una aplicación de seguridad especifica.

La segunda característica es una composición de Visibilidad en toda la red con control de flujo centralizado. Esta característica es una distinción que refuerza la distinción entre redes SDN y las arquitecturas de red convencionales (Véase en la Figura 13). La visibilidad en toda la red significa que el control de la red puede conocer el estado de cualquier elemento de red desplegado en cualquier lugar y en cualquier momento. Mientras que el control de flujo centralizado se refiere a la forma en que las decisiones de reenvío de flujo pueden ser tomadas desde una instancia única de red y lógicamente centralizada; este comportamiento libera a cada dispositivo de la red de hacer cualquier calculo o aplicar cualquier algoritmo de reenvío cuando llega un paquete de datos. Esta agrupación de características puede beneficiar a varias tareas de seguridad, ya que el controlador puede solicitar muestras de flujo y estadísticas de flujo. La información que se recopila se puede enviar a las aplicaciones de seguridad especializada, por ejemplo, un IDS, IPS o DPI, o incluso puede ser utilizada directamente por el propio controlador para instruir cualquier tipo de reacción según la situación de la red. [37]

Mediante la supervisión del tráfico de la red, el controlador puede mantener registros actualizados sobre el comportamiento del tráfico, tanto de flujos de tráfico inesperados procedentes de elementos de red no supervisados como de flujos de tráfico sospechosos o malformados que atraviesen la infraestructura. La detección y prevención de ataques, las características de la red y la visibilidad de la red permiten identificar el origen de los ataques y, a continuación, el plano de control puede decidir qué acciones son las apropiadas para mitigar cualquier situación anómala; por ejemplo, las estadísticas de flujo y los patrones de tráfico pueden avisar de un posible ataque de denegación de servicio (DoS), entonces el controlador o la aplicación de seguridad tiene que tomar las medidas necesarias para detenerlo, bien sea instalando nuevas reglas de flujo o denegando por completo cualquier tráfico desde el origen del ataque.

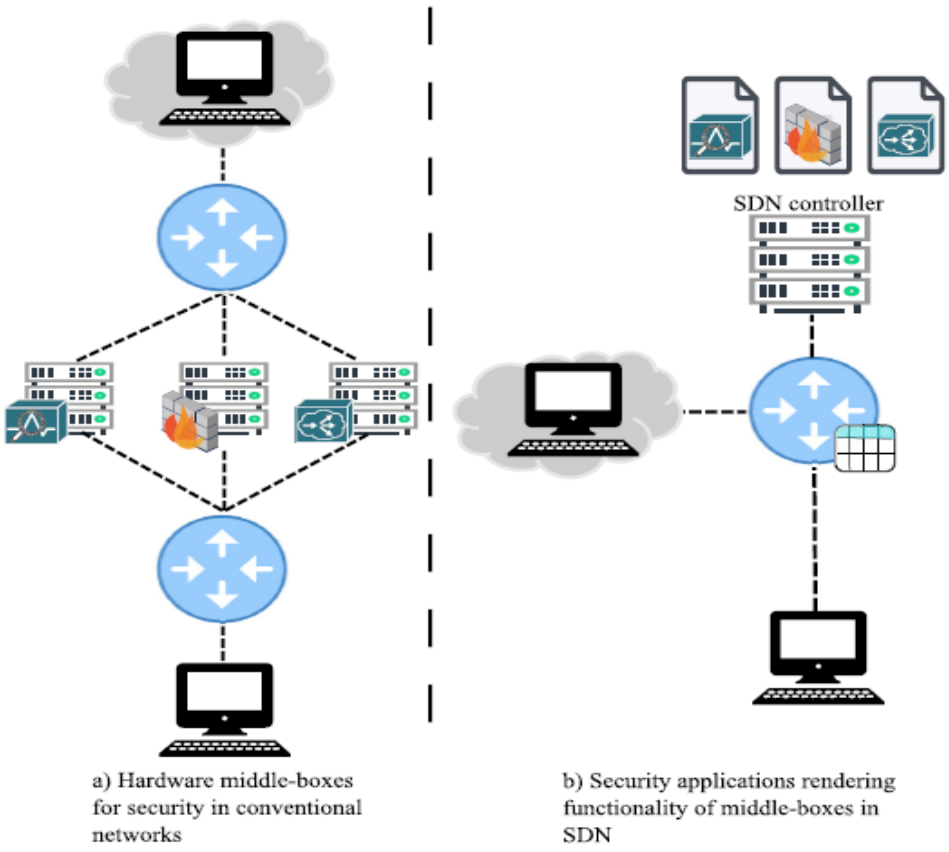

Figura 13. Despliegue de Middleboxes en redes convencionales (a) y en SDN (b). Esta figura nos muestra como los dispositivos de hardware conectados a la red convencional pueden ser sustituidos por aplicaciones software en las SDN. [38]

La tercera característica es la programabilidad de la red. En las redes convencionales, los usuarios/administradores no pueden modificar los dispositivos o aplicaciones de la red a su antojo. Estos dispositivos vienen equipados con conjuntos de comandos/instrucciones que, en SDN, gracias a la programabilidad de la red, la infraestructura puede estar formada por un conjunto de dispositivos en los que el cliente puede instalar diferentes funcionalidades que se expresan a través de conjuntos específicos de reglas de flujo. Por ejemplo, un caso muy práctico son las listas de control de acceso que pueden configurarse en los switches de la red para que apliquen reglas de flujo que expresen acciones de rechazo o abandono contra diferentes acciones que se puedan producir. La programabilidad de la red mejora la flexibilidad de la misma y también beneficia el despliegue de servicios y sistemas de seguridad como aplicaciones de red, ya sea instaladas en el controlador o en una capa de aplicaciones de seguridad separada. Esta característica aporta rutinas de software flexibles seguras que son diseñadas para procesar una gran cantidad de operaciones. Gracias a estas nuevas tecnologías software se pueden implementar soluciones de seguridad más robustas capaces de predecir ataques. Así mismo estas propiedades que ofrece la programabilidad hace posible que se pueda ejecutar contraataques contra los adversarios de la red, por ejemplo, si una aplicación de red de seguridad detecta un host maligno, esta aplicación puede liberar un ataque contra ese host con el objetivo de detener el ataque.

La programabilidad de la red introdujo la capacidad de reconfigurar dispositivos, este comportamiento puede aprovecharse para implementar funciones y servicios de seguridad o políticas de seguridad. Está claro que las propiedades de la SDN pueden ser explotadas para beneficiar la seguridad de la red en diferentes contextos, ya sea adaptando esquemas de seguridad bien conocidos o incluso introduciendo nuevos enfoques novedosos para hacer frente a los riesgos de seguridad existentes. Sin embargo, como ya hemos dicho antes, la arquitectura SDN introduce nuevas superficies de ataque y brechas de seguridad. Por eso es importante señalar que se

necesitan nuevos desarrollos de seguridad para poder garantizar un entorno SDN seguro, a esto se le ha denominado seguridad a nivel de sistema o lo que es lo mismo seguridad SDN. [39]

4.3.2 Mejorar la seguridad en SDN

Como se ha mencionado anteriormente, la arquitectura SDN introduce nuevas vulnerabilidades y riesgo de seguridad debido a que incorpora nuevos componentes de red. Además, la introducción de nuevas interfaces y protocolos facilita a la aparición de nuevas superficies de ataque y objetivos explotables. Ante estos hechos, surge una conclusión importante: a pesar de los esfuerzos por mejorar la seguridad de la red aprovechando las novedades e innovaciones de SDN, una arquitectura de red podría seguir siendo insegura si el propio marco de SDN no es completamente seguro contra amenazas de seguridad introducidas por las características de SDN. Los canales e interfaces utilizados para el intercambio de información en SDN puede exponer las comunicaciones de red si estas no están debidamente protegidas, pues las comunicaciones en texto plano pueden ser intervenidas y los datos se pueden utilizar para comprometer distintas entidades de la red. Por lo tanto, todos los canales de comunicación e interfaces de datos de la red SDN deben ir cifrados, de modo que, si un atacante consigue infiltrarse en los intercambios de datos, este no pueda extraer la información real que se transmite ya que los datos están cifrados.

Tanto los esquemas de cifrado como la seguridad de la red SDN debería reforzarse más usando los mecanismos de autenticación y confianza, especialmente en el plano de control. El controlador debe ser capaz de reconocer y autorizar dispositivos de confianza (otros controladores, conmutadores y dispositivos adicionales), aplicaciones de red mediante el intercambio de certificados firmados, claves simétricas y asimétricas o códigos de autenticación de mensajes. Asegurar que solo los dispositivos y aplicaciones de confianza tengan acceso necesario a los recursos de la red garantiza que los dispositivos y aplicaciones maliciosas permanezcan aisladas de la red (Véase en la figura 14). La selección de los mecanismos de protección del canal de control o de alguna otra interfaz debe hacerse con cuidado y teniendo en cuenta los servicios y protocolos que utilizara el canal. Lograr un alto nivel de protección y seguridad en los límites de los planos de control y aplicación requiere desarrollo de diferentes mecanismos para poder identificar y mitigar los riesgos introducidos por aplicaciones de red poco fiables. Por ejemplo, esquemas de autorización y autenticación, evaluación de seguridad, y acceso controlado a los servicios y aplicaciones de gestión. [40]

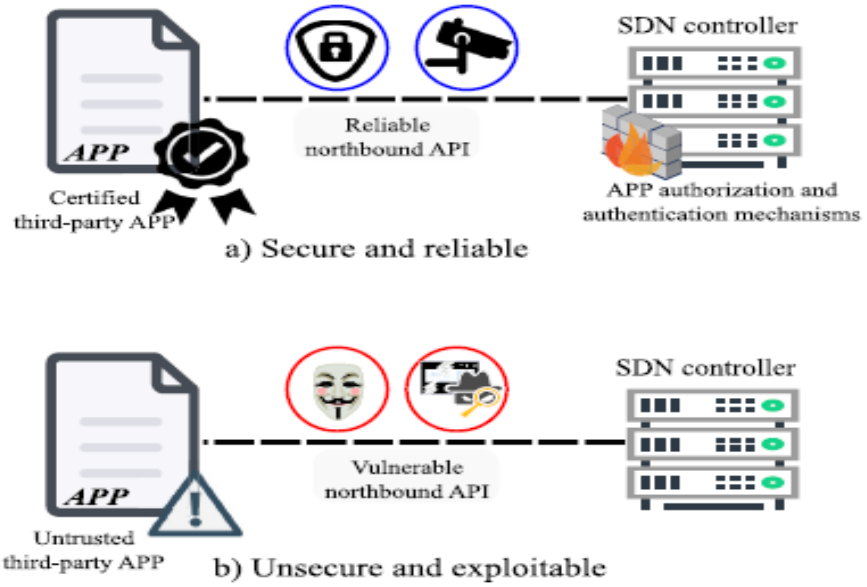

Figura 14. Canales de control, a) Canal seguro y fiable con mecanismos de seguridad necesarios y b) Canal inseguro y fácil de atacar. [41]

Un fallo importante en SDN es la imposibilidad que tiene el controlador para detectar aplicaciones de red que puedan estar emitiendo reglas de flujo que entren en conflicto con las reglas instaladas anteriormente por otra aplicación o incluso reglas de flujo que contradigan a las políticas de seguridad preestablecidas. Como medidas para mitigar estas situaciones, se debería de incluir en el plano de control de la red SDN un esquema mediador para las reglas conflictivas y también unos esquemas de verificación de políticas. El mediador de reglas conflictivas se utilizaría para identificar las reglas que sobrescriben el comportamiento de aplicación de la política indicado por un conjunto previo de reglas, mientras que el verificador de políticas sería capaz de garantizar que las nuevas reglas de flujo listas para ser emitidas no entren en conflicto con las políticas de seguridad preestablecidas, evitando así infringir el esquema de seguridad y exponiendo las brechas de seguridad.

Como ya se ha mencionado anteriormente en otro capítulo, hay diferentes ataques que están dirigidos al estado de la red y tienen como objetivo tumbar esta misma como, por ejemplo, envenenamiento de la topología, denegación de servicios (DoS), inyección de reglas de flujo y la infiltración de dispositivos fraudulentos. Estos ataques pueden ser mitigados gracias a los sólidos mecanismos de supervisión del estado de la red que ofrecen la posibilidad de mantener un registro actualizado del comportamiento y de estado actual de la red. Por lo tanto, la información de la red recopilada en un determinado marco temporal puede equipararse con un archivo de configuración de red o un listado (que contiene tablas de flujo, información de topología, lista de dispositivos de red autorizados, estadísticas de reenvío de paquetes, estado de las interfaces de datos y de control, etc.) para detectar cualquier incoherencia en el estado de la red. Además, la supervisión del estado de la red se puede aprovechar para clasificar los diferentes estados de la red, riesgos de seguridad, estados estables, si es propenso a fallos; de modo que la red puede disponer de un mecanismo de adaptación que, a partir de la información histórica recopilada sobre el estado de la red, se pueda predecir nuevos estados de red y evitar que caiga en estados propensos a fallos, es decir, que la red se mantenga funcionando en estados libres de fallos y segura. [42]

Un método utilizado hoy en día que contribuye a la mejora de la seguridad de una red SDN es la adopción de entorno de nube y virtualización de funciones de red. [43] Pues este método permite la integración de una variedad de soluciones de seguridad ofrecidas por múltiples proveedores y desarrolladores de seguridad a costa de introducir componentes y capas adicionales en la arquitectura SDN (Véase en la figura 15). La seguridad virtualizada contribuye a la escalabilidad de la red SDN ya que los dispositivos de seguridad desplegados en la nube pueden ser invocados y liberados en cualquier lugar de la topología, y tanto las funciones de seguridad como las aplicaciones pueden compartirse y migrar entre diferentes nubes de seguridad. De hecho, los elementos de la red pueden ser liberados del procesamiento adicional que se necesita para ejecutar algunas funciones de seguridad complejas que consumen muchos recursos.

En los entornos de seguridad basados en la nube, las funciones de seguridad virtualizadas son instanciadas y eliminadas bajo demanda, de esta manera se puede construir un esquema de seguridad más robusto usando una composición de aplicaciones de seguridad básicas e intermedias de monitorización y detección que pueden identificar ataques nuevos. Si se detectan anomalías, la red se pone en estado de alerta y este esquema construye un conjunto de propiedades que son capaces de alimentar una función de seguridad. Esta función de seguridad puede ser solicitada bajo demanda por lo que una vez se ha aplicado puede ser desvinculada (Véase en la figura 16).

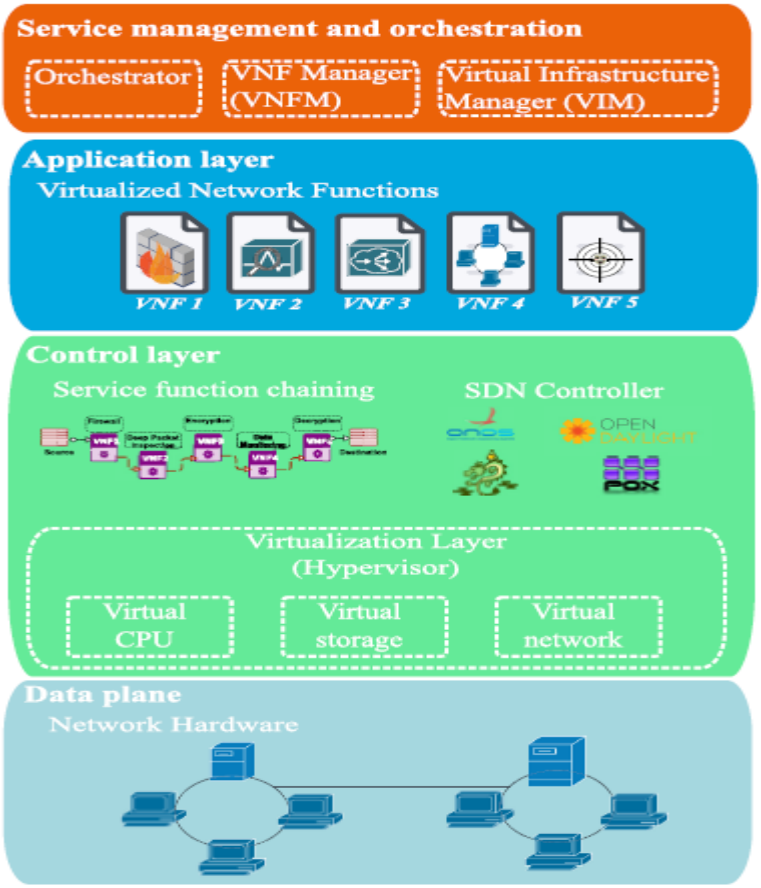

Figura 15. Introducción de NFV a la arquitectura de SDN. [44]

En la figura 15 podemos ver como se ha introducido la virtualización de funciones de red a la arquitectura SDN. Gestión y orquestación de instancias e interfaces virtualizadas son creadas sobre la capa de aplicación como una capa adicional. Las funciones de red virtualizadas residen, en general, en las capas de aplicación, aunque algunas pueden residir en los dispositivos del plano de datos. La asignación de recursos y el control de los componentes virtualizados son tareas de la capa de control.

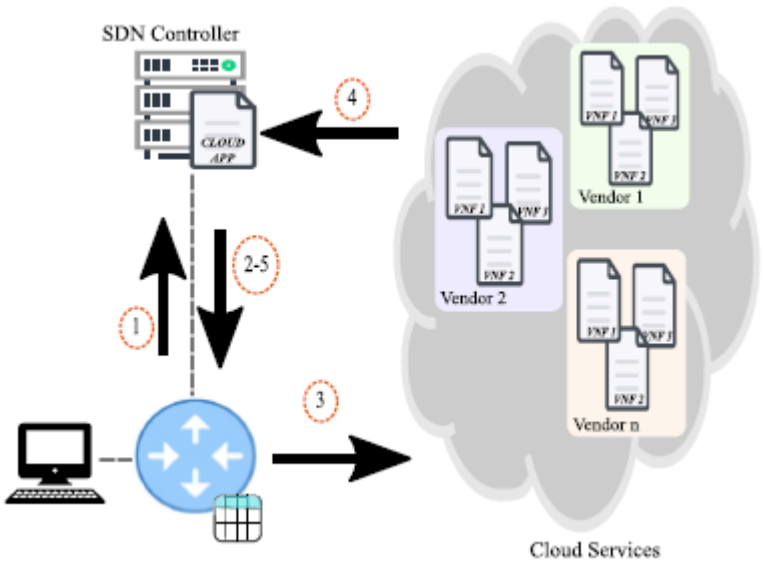

Figura 16. Solicitud de funciones de seguridad basadas en la nube bajo demanda. [45]

En la figura 16 podemos ver las solicitudes que se han de seguir en un sistemas basado en la nube bajo demanda: 1) Un dispositivo conectado a la red solicita una función de seguridad; 2) El controlador (o la aplicación de red) ordena al dispositivo que solicite la función de seguridad a un servicio en la nube; 3) El dispositivo reenvía la solicitud al servicio en la nube; 4) La función de seguridad virtualizada manda acciones de seguridad al controlador y 5) El controlador da las instrucciones necesarias a las reglas de flujo y estas son las que realizan las acciones de seguridad.

Emulando el concepto de los servicios unificados en las redes convencionales, en SDN se debe desarrollar un marco de seguridad unificado para poder garantizar la seguridad en todos los niveles de la misma red SDN, incluyendo todos las capas y dispositivos implicados en la arquitectura SDN. Esto significa que la seguridad debe de comenzar desde las primeras etapas de la construcción de la red y el desarrollo de las aplicaciones, pasando por el despliegue de la red y la instalación de servicios, la supervisión, la comprobación del estado de la red y la aplicación de políticas y finalmente llegando hasta la aplicación de esquemas forenses y de resiliencia de la red. El marco de desarrollo de la seguridad SDN incluye lenguajes de programación, depuradores y compiladores, todo esto proporciona a los desarrolladores de SDN herramientas para el avance de aplicaciones seguras y políticas de seguridad. También les proporciona mecanismo para localizar vulnerabilidades, puntos críticos de la red y fallos de software antes del despliegue de las aplicaciones.

Por un lado, los esquemas de monitorización del estado de la red, las políticas de seguridad y los sistemas de detección de ataques garantizan tanto la protección de la red como la posibilidad de reaccionar ante comportamientos sospechosos de la red y cambios inesperados. Mientras que, por otro lado, los esquemas de encriptación y cifrado se encargan de proteger los datos que se intercambian en la red a través de todas las interfaces y canales de comunicación, y, además, los mecanismos de confianza permiten que solo los elementos que este autorizados y autenticados puedan acceder a las distintas capas de la red.

Los esquemas forenses y de resiliencia de la red garantizan, en primer lugar, que, tras una situación de ataque, la red pueda seguir funcionando de forma continua sin interrupción y que se realicen tanto las reconfiguraciones adecuadas de la red como los procesos de mitigación de los ataques para poder restablecer el funcionamiento de la red. Y, en segundo lugar, rastrear las fuentes del ataque en concreto, reunir pruebas del mismo y aislar los dispositivos y aplicaciones involucradas en el ataque para luego generar informes y descripciones que posteriormente se utilizarán para llevar a cabo un análisis exhaustivo.

Capítulo 5. Clasificación y visión general de las soluciones de seguridad en SDN

En este capítulo se hará un repaso absoluto de las nuevas soluciones de seguridad para los vectores de ataque y las superficies de amenaza cuyo objetivo principal es la arquitectura SDN. Las propuestas están clasificadas en varias categorías que reflejan la principal contribución para mejorar la seguridad de una red definida por software, estas son algunas de ellas: Detección de ataques, Seguridad virtualizada/basada en la nube, Mitigación de amenazas y ataques, Garantizar la seguridad de la sesión del usuario, evaluación de la seguridad, análisis forense y, por último, marco de seguridad integrado. A continuación, se detallarán algunos de ellos.

.

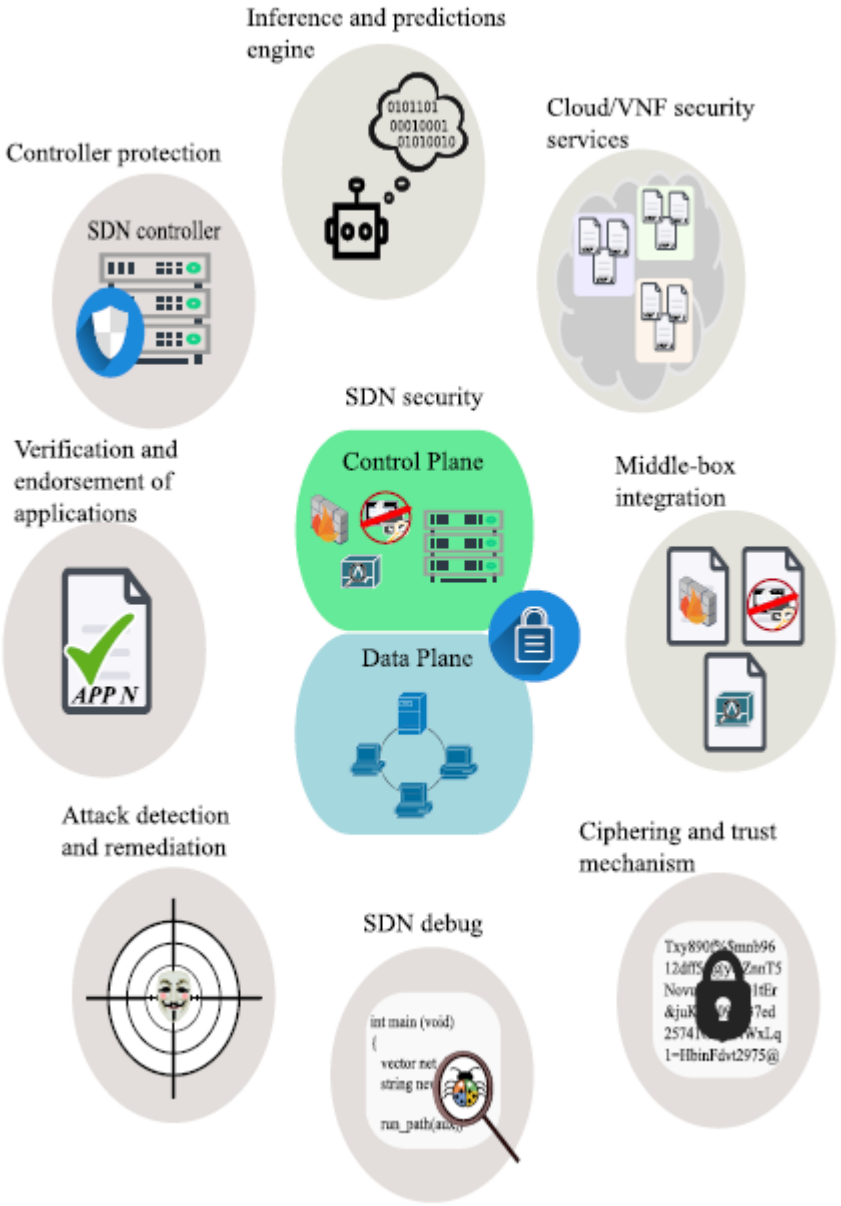

Figura 17. Framework de seguridad SDN unificado. [46]

5.1 Detección de amenazas

El desarrollo de aplicaciones de seguridad para la detección de ataques puede lograrse aprovechando la capacidad que tiene el plano de control de SDN para solicitar características de flujo a partir de los dispositivos de la infraestructura y luego inferir sobre el estado de la red, los patrones de tráfico y otras propiedades con la información recopilada. Las rutinas de software especializadas en aplicaciones de seguridad están diseñadas para detectar comportamientos anómalos y analizar la información de la red requerida por el plano de control.

La programabilidad del software y la flexibilidad de la red ayudan a imponer la sustitución de aparatos fijos en las redes SDN, lo que permite a los usuarios asegurar su propia infraestructura lógica de una manera perceptiblemente privada. Cualquier implementación de seguridad por parte del usuario, como los sistemas de detección de intrusiones (IDS), los sistemas de prevención de intrusiones (IPS), la inspección profunda de paquetes (DPI), las redes privadas virtuales (VPN), la defensa de objetos móviles (MTD), etc., se llevaría a cabo mediante la instalación de nuevas reglas de flujo en el entorno de nube basado en SDN [47]. Las soluciones basadas en software y los enfoques de aprendizaje automáticos cada vez son más comunes ya que estos pueden gestionarse para ofrecer esquemas de seguridad eficaces. Por ejemplo, aplicaciones que incorporen plataformas de software de código abierto y con capacidades de aprendizaje automático para ofrecer detección y mitigación de ataques.

También hay estudios como en [48], el cual se basa en realizar operaciones de aprendizaje automático (conocido como machine-learning) en la inspección de paquetes de flujo para proporcionar una solución inteligente llamada SDN-IDPS. El diseño es básicamente una integración de un conjunto completo de características de aprendizaje automático que pueden utilizarse para construir un “framework” eficaz para la inspección de paquetes y la detección de ataques. Las pruebas que se llevan a cabo por los autores de dicho estudio, enseñan que la solución IDPS muestra un rendimiento aceptable en condiciones de ataque de gran carga de trabajo.

En la práctica, existe el despliegue de Network-IDS o NIDS en SDN que se enfoca en la detención de amenazas monitorizando el tráfico de la red a la que están conectados los hosts. En primer lugar, se consolida un esquema de defensa perimetral eficiente contra los ataques externos para reducir el conjunto de switches que reportan características de la red al IDS. Una segunda medida consiste en dejar que los switches reenvíen las estadísticas de la red en intervalos de tiempo fijos en lugar de reflejar todos los flujos de tráfico cada vez. Además, las funciones NIDS podrían no ser procesadas por el controlador, sino que deberían ejecutarse en paralelo en el mismo dispositivo de hardware del controlador. La arquitectura NIDS se beneficia de la combinación de dos partes: por un lado, aprovecha un enfoque basado en firmas que se apoya en el uso de sistemas de mitigación de ataques de seguridad bien conocidos y, por otro lado, un enfoque de aprendizaje automático, construido mediante un algoritmo de retro propagación, que actúa como un sistema de reconocimiento de patrones con el fin de garantizar la defesan contra amenazas de seguridad desconocidas y difíciles de detectar. El enfoque basado en firmas se implementa como un dispositivo físico de detección física instala en un lugar fijo de la topología. Por su parte el sistema de reconocimiento de patrones se implementa sobre el controlador SDN y aprovecha las estadísticas de flujo recogidas por el controlador para detectar patrones de tráfico anómalos y posteriormente instruir al controlador SDN para que instale nuevas entradas de flujo en los dispositivos del plano de datos.

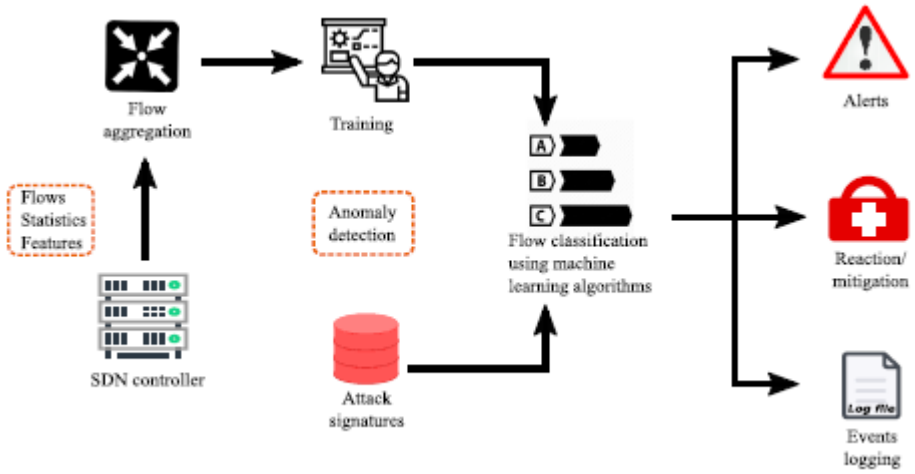

Figura 18. Técnicas basadas en software y enfoques de aprendizaje automáticos. [49]

5.2 Seguridad basada en NFV (Nube)

La virtualización de las funciones de red y los servicios de nube virtualizados comparten una característica en común, ambas tecnologías utilizan sistemas para permitir que diferentes soluciones de software coexistan en un marco de hardware compartido. Un conjunto de servicios virtualizados está disponible localmente en el sistema y puede ser instanciado cuando sea necesario, mientras que las funciones virtuales están distribuidas en sistemas externos y pueden ser aplicadas en cualquier momento que la red pueda tener acceso a ellas. En SDN, un entorno de seguridad virtualizado permite construir un sistema de seguridad más sofisticado, pues permite la integración de múltiples soluciones proporcionadas por diferentes proveedores y, por tanto, permite una construcción en línea de los sistemas de seguridad que pueden ser solicitados bajo demanda. La aplicación de instancias de seguridad virtualizada contribuye a mejorar el rendimiento de la red SDN, ya que se observa una reducción de la carga de procesamiento del controlador cuando se transfieren las funciones de seguridad a entidades externas (como pueden ser nubes y grupos de servicios virtuales en los switches). Así mismo, el uso de servicios virtuales fomenta la innovación ya que los desarrolladores de SDN se centran en el diseño de interfaces de red para entidades virtuales, mientras que los proveedores de servicios de seguridad se centran en desarrollar soluciones de seguridad.

Las instancias actuales de los Firewalls o cortafuegos en las redes SDN se basan en uno de estos dos métodos, por un lado, un método centrado en el controlador, en el que las funciones del firewall están diseñadas como una aplicación que se ejecutar en el controlador y por lo tanto las funciones de reenvío, denegación y caída de datos pueden ser instruidas a los switches del plano de datos en forma de reglas de flujo. Por otro lado, el método centrado en la NFV, en el que las aplicaciones del firewall se despliegan como funciones de red virtualizadas en entornos de nube distribuidos. Estos dos métodos presentaban algunos inconvenientes y por ello se propuso una solución para superar sus deficiencias con una fusión hibrida en la que se aprovecharía las mejores características de ambos enfoques. La solución propuesta ofrece un esquema similar a un enfoque centrado en la NFV con la aplicación de firewall instanciada como una función virtualizada que se ejecuta en la nube. De esta manera la aplicación NFV-firewall beneficia la escalabilidad de la red y ayuda en la reducción de la latencia de los paquetes.

Otro campo a tener en cuenta son los Sistemas de Redes Industriales (INS), los cuales presentan un enfoque proactivo del plano de datos seguro mediante la introducción de switches SDN activos, llamados A-switches. Estos conmutadores activos están diseñados de tal manera que poseen una capa de virtualización adicional que permite la instalación de las funciones de seguridad virtualizadas en forma de aplicaciones de seguridad más robustas. El controlador SDN manda la activación de las funciones de seguridad virtualizada a los switches a través de un módulo de gestión y estos eliminan la necesidad de usar dispositivos fijos en el middlebox

consiguiendo así que las funciones de seguridad se desplieguen directamente en los dispositivos de reenvío y se distribuyan con precisión.

Las funciones de seguridad de la red desplegadas en entornos de nube pueden integrarse junto con funcionalidades de filtrado básico en SDN para permitir la seguridad total en la arquitectura. Esta propuesta se basa en proporcionar la integración entre las funciones de seguridad de la red y el filtrado de paquetes en el plano de datos con el objetivo de desplegar una inspección de paquetes dinámicos y una infraestructura de detección de ataque donde la seguridad de la red pueda decidir si los paquetes que provienen de un flujo especifico necesitan un tratamiento mayor por parte de las funciones de seguridad especializadas en la nube o, por el contrario, el controlador SDN esta instruido para emitir las reglas de flujo pertinentes en el plano de datos. Esto puede beneficiar a la seguridad de la red permitiendo que múltiples funciones de seguridad (Firewalls), actúen sobre un solo paquete de flujo, también puede beneficiar al rendimiento de la red, ya que confía las funciones de filtrado a los dispositivos del plano de datos, evitando así que todos los paquetes sean reenviados a la nube. [50]

5.3 Mitigación de ataques

Como la mayoría de las amenazas de seguridad que se observan normalmente en las redes convencionales pueden reproducirse en escenarios SDN, debería de existir la posibilidad de realizar estrategias de mitigación de amenazas aplicadas a las características de la arquitectura de la SDN, por ejemplo, la programabilidad de la red y la gestión centralizada de flujos. Se pueden llevar a cabo diversas soluciones de control, ya que toda la información relativa al estado de la red está a disposición total de los controladores. Entonces en respuesta a cualquier comportamiento desviado, la aplicación puede instruir al controlador para que emita las entradas pertinentes de flujo que deben mitigar dicha situación.

Por ejemplo, una aplicación eficaz de mitigación de ataques SDN podría estar compuesta por un mecanismo de detección que sea capaz de distinguir las características específicas de un comportamiento irregular de la red, de acuerdo con ciertas propiedades únicas descritas en un conjunto de políticas de seguridad, y un módulo de reacción que aplica directamente un conjunto de acciones de mitigación en los elemento de red comprometidos (flujos, switches, canales, interfaces de red), de acuerdo con las propiedades descritas en la política. A continuación, se detallarán brevemente algunos enfoques de mitigación basados en SDN para diferentes ataques de seguridad.

5.3.1 Ataques de inundación/denegación de servicios

Para mitigar ataques de congestión de enlace (ruta, inundación, DoS, DDoS, etc.) en redes SDN se necesita ofrecer un marco de gestión y aplicación de políticas intuitivo en el que las políticas de red de alto nivel que estén almacenadas en un repositorio puedan traducirse en rutas para redistribuir los flujos de ataque y aliviar la congestión de los enlaces. Tras la detección de una anomalía, los clientes pueden dar una alerta a un componente de supervisión de la red conectado al controlador del IPS (Proveedor de servicios de Internet), y luego el componente de vigilancia puede extraer la información de los diferentes caminos que en ese momento se encuentren cogestionados. Un dispositivo de decisión de políticas reúne las características recogidas por el componente de supervisión y solicita a una base de datos de políticas que acciones deben tomarse contra el ataque. A continuación, un módulo de orquestación e implementación de políticas utiliza la información de los flujos y las acciones de decisión descritas para elaborar el cálculo de las nuevas rutas de reenvío y emitir las reglas de flujo a los switches de la red para redistribuir los flujos congestionados en las rutas de flujo no congestionadas.

Los ataques DDoS lanzados mediante el uso de flujo de bajo tráfico son difíciles de detectar. La mayoría de los sistemas de detección tradicionales no detectan este tipo de ataques y a menudo reportan tasas significativas de falsos positivos y falsos negativos. Una propuesta para solucionar este tipo de ataques es la prueba de relación de probabilidad secuencial (SPRT) que proporciona

un esquema que puede detectar ataques de inundación de paquetes distribuidos de gran tamaño, pero de bajo tráfico en muy pocas rondad de análisis de flujo. [51]

5.3.2 Ataques de canal lateral

La propuesta que se da [52] para este tipo de ataque es una contramedida de seguridad contra el ataque Know-Your-Enemy (KYE), el cual es un ataque de canal lateral que explota el comportamiento de instalación de reglas de flujo bajo demanda que presentan las SDN basadas en OpenFlow. El general, el ataque KYE se divide en dos fases, la primera corresponde a la fase de sondeo en la que el atacante envía paquetes de prueba al switch de destino y observa las reglas de flujo instaladas. En la segunda fase, según la información recopilada en la fase anterior el atacante es capaz de descubrir información sobre la configuración de la red, la política de red aplicada para cada flujo de tráfico especifico e incluso las entradas de flujo. La ofuscación de flujos es la contramedida de seguridad sugerida contra el ataque KYE.

Esta contramedida aprovecha la capacidad de los switches OpenFlow para modificar los paquetes en circulación. Cuando el atacante envía un flujo de prueba a un switch comprometido, el controlador emite una única regla de flujo que ordena al switch que modifique algunos campos de la cabecera del flujo y que reenvíe los paquetes a otro switch, el proceso se repite para un numero preestablecido de switches. Cuando el flujo llega al último switch, el controlador emite la regla de flujo correspondiente que aplicara una política de seguridad para el flujo original de la prueba de ataque. Se dice que el conjunto de switches que participan en el proceso de ofuscación del flujo pertenecen a los que se denomina ruta de ofuscación.

5.3.3 Infiltración del dispositivo de control de acceso

Network Flow Guard (NFG) es una solución diseñada para defender las redes SDN contra los ataques infringidos por intrusos ilegales que se infiltran en las redes privadas aprovechando los puntos de acceso inseguros desplegados intencionadamente o no en la topología de la red. NFG detecta los puntos de acceso fraudulentos basándose en un esquema de inspección pasiva de paquetes combinado con un sistema de detección activa.

El esquema de inspección pasiva de NFG incluye tres etapas de filtrado, en la primera, un dispositivo sospechoso contrasta en una “lista blanca” predefinida los dispositivos que están autorizados a conectarse a la red. En la segunda etapa, se utiliza un algoritmo para extraer los valores TTL (Tiempo de vida) de las cabeceras de los paquetes y luego se buscan los valores decrecientes para rastrear la existencia de puntos de acceso no autorizados en la red. En una tercera etapa, la supervisión sobre los puertos de los switches permite detectar la cantidad de direcciones MAC e IP asociadas a un mismo puerto, y en caso de que el recuento sea superior a uno, entonces el puerto se marca y el switch se aísla de la red. En la última etapa del esquema pasivo lo que se hace es registrar los valores promedio de TTA (Tiempo de reconocimiento) en los mensajes iniciales de saludo TCP SYN en cada puerto del switch de la red, se calcula un promedio global y si se observa algún valor que este fuera de lo normal con respecto al promedio, entonces ese puerto se marca y su tráfico se dirige al sistema de detección activo para realizar pruebas más profundas y aislar los dispositivos sospechosos de la red. [53]

5.4 Gestión del acceso y de la identidad del usuario

En SDN, la información especialmente sensible se transmite a través del canal de control, por lo que muchos ataques se centran en espiar la capa de control para obtener la información suficiente para poder comprometer la red. La aplicación de cifrado en los canales de comunicación, mejora la seguridad ya que, aunque los atacantes puedan filtrar los datos cifrados, no podrían obtener los datos reales en texto plano (Véase en la Figura 19). La separación de los planos de control y de datos expone un único punto de fallo para toda la SDN. Entonces, la implementación de medidas de seguridad especializadas en la interfaz de control es obligatoria en cualquier despliegue de red SDN, ya que un canal de control mal protegido podría

comprometer el controlador SDN, los dispositivos de la infraestructura y podría espiar los datos de flujo sensibles sobre el estado y la configuración de la red.

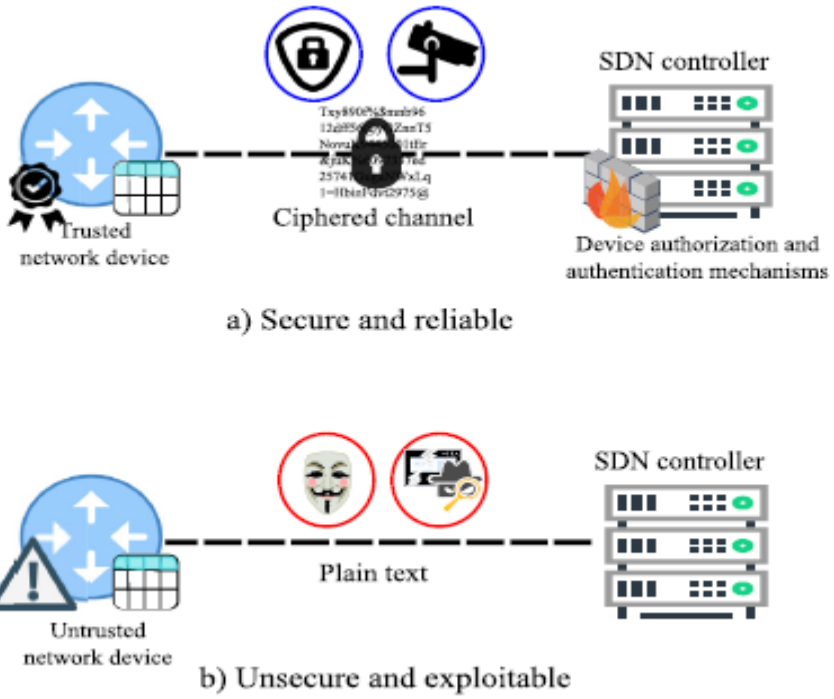

Figura 19. Protección del canal de control: a) Seguro y fiable, b) inseguro y explotable. [54]

Sin embargo, los atacantes muy especializados podrían tener la capacidad de infiltrarse en la red utilizando un elemento de la red comprometido o falso, en cuyo caso la protección de los datos exige medidas de seguridad adicionales para garantizar que los agentes de red ajenos a la red no puedan participar en el intercambio de datos. Para evitar que cualquier persona ajena a la red se infiltre en ella, se debe establecer un mecanismo de confianza para la autenticación y la autorización de los privilegios de la red. De este modo no solo se protegerán los datos, mediante esquemas de cifrado, sino que también se evita el acceso no autorizado a la red.

Como los mecanismos de encriptación que ofrece el pack de OpenFlow para la protección del canal de control son opcionales y difíciles de desplegar, [55] se proponen nueves mecanismo de autenticación como la huella digital de dispositivos de red que proporciona una estrategia de autenticación precisa. Puesto que los switches de red suelen presentar diferentes características de implementación y exponen alguna que otra discrepancia en cuanto a tiempos de procesamiento, estas características constituyen una huella digital del dispositivo. Todos los dispositivos de la red deben de ser registrados en el módulo de huella digital antes de que se pongan en funcionamiento. Entonces, los operadores de red pueden activar la construcción de la huella digital y cada vez que un dispositivo se conecta al controlador, el módulo de huella digital bloquea cualquier aplicación de red que intente establecer un enlace con el nuevo dispositivo. Este estado continúa hasta que se finaliza el proceso de autorización y si el proceso de autenticación tiene éxito, el bloqueo se elimina, en caso contrario el dispositivo de red se mantiene aislado del funcionamiento de la red.

Otro mecanismo propuesto para la seguridad y protección del canal de control SDN puede ser un comprobador de credenciales cuando se intercambie información en la fase de inicio de una sesión segura. La seguridad del canal de control se basa en el uso de algoritmos de seguridad, así como en las fechas de caducidad y los emisores de los certificados de seguridad utilizados en el intercambio de información. Las políticas de seguridad mencionadas anteriormente incluyen un conjunto amplio de restricciones de seguridad definidas por el usuario, por ejemplo, la versión del protocolo TLS permitida para las sesiones de canal seguro en la red. Lo que permite este protocolo es que se pueda establecer una ruta fiable entre los dispositivos que actúan como extremos en el intercambio de datos [56].

Existe también, un mecanismo [57] cuyo diseño incluye una aplicación para la autenticación de dispositivos que se despliega para instruir al controlador sobre las reglas de reenvío de acuerdo con los resultados de un proceso de autenticación. Este mecanismo hace posible un control de acceso seguro en SDN que impide el acceso no autorizado a recursos de la red aplicando el estándar IEEE802.1x existente y el EAP (Protocolo de Autenticación Extensible) junto con la implantación del servidor RADIUS, estudiados en algunas de las asignaturas de telemática de la carrera.

5.5 Evaluación de la seguridad

Una gran cantidad de riesgos y problemas de seguridad de la red pueden atribuirse a las siguientes situaciones: malas configuraciones, implementaciones erróneas, reacciones no previstas, situaciones de ejecución inesperadas, especificaciones y diseños de protocolo erróneos, etc. Es evidente que antes de la puesta en marcha de una red, hay que realizar pruebas y evaluaciones exigentes para poder garantizar tanto su funcionalidad cono su seguridad. Por desgracia, muchas de las pruebas de seguridad no son estrictas y solo comprueban algunas características comunes, dejándose problemas de seguridad ocultos. Por esta razón anterior, se necesita desarrollar soluciones de seguridad que proporcionen una prueba exhaustiva para descubrir todas las vulnerabilidades posibles. Un despliegue completo de red seguridad debería garantizar al menos que la red no es vulnerable y que está debidamente protegida a los vectores de ataque o derivados de cualquiera de las situaciones mencionadas anteriormente. En las redes SDN este tipo de solución es posible mediante aplicaciones de seguridad que pueden realizar una evaluación completa de las vulnerabilidades, ya sea bajo demanda o periódicamente.

5.6 Análisis forense

El análisis forense desempeña un papel importante en la reconstrucción de escenas de ciberdelincuencia y en el conjunto de registro de datos válidos que podrían utilizarse para rastrear y descubrir el origen de ciertas amenazas a la seguridad. Una de las pocas propuestas forenses que hay es SDN Forensics [58] y consiste en un enfoque multicapa que permite funciones en cada capa de la estructura SDN para:

- El tratamiento de los datos de las pruebas.
- La detección de anomalías aprovechando la inteligencia artificial.
- Activación de alarmas.
- Formateo de instancias de datos de las pruebas recogidas en la escena del ataque.

Una instancia de adquisición y extracción de datos es capaz de construir conjuntos de datos de pruebas utilizando la información recopilada de los registros de control y ejecución, imágenes de memoria y almacenamiento de datos físicos (Discos). A continuación, una instancia de fusión de datos integra en forma de clusters (grupos o conglomerados) toda la información extraída de las fuentes físicas y de software. Esos clusters se convierten en la fuente de entrada que se alimenta para el análisis de datos, mediante técnicas de aprendizaje de máquinas especializadas en la detección de anomalías, activación de alarmas, reconstrucción de la escena y la documentación del ataque reportado.

5.7 Marco de seguridad SDN integrado

En el capítulo 4 se ha destacado la necesidad de una solución de seguridad SDN más extensa concebida para cooperar con la materialización de una arquitectura de seguridad unificada para la SDN. Para la arquitectura se debería plantear una solución de seguridad integrada capaz de ofrecer un control alto de todos los aspectos de seguridad en la red, mientras que al mismo tiempo se desempeña el papel de una composición de componentes que atienden individualmente a cuestiones de seguridad específicas. Los componentes de la arquitectura deben ofrecer mecanismos y esquemas centrados en la mejora y protección de todos los componentes e interfaces del sistema, en lugar de limitarse a ofrecer soluciones a vectores de ataque o situaciones anómalas.

Además, la arquitectura de seguridad debería centrarse en atender a las características de alto nivel que se encuentran en los problemas de seguridad en lugar de ocuparse de los detalles de bajo nivel. Esto podría garantizar que la solución de seguridad actúe en una gama más amplia de escenarios y situaciones, al tiempo que proporciona una respuesta a los incidentes de seguridad eficaces y específicos. Por ejemplo, una solución orientada a la arquitectura para la protección del canal de control debe concebirse como un mecanismo que sea capaz de garantizar condiciones de seguridad como la confidencialidad, la integridad y la disponibilidad de los datos sin preocuparse por cubrir detalles relacionados con ataques específicos que ponen en peligro estas condiciones de seguridad.

Una propuesta que construye un marco de seguridad SDN bastante completo es NOSArmor [59], la cual incluye varios enfoques individuales de seguridad en SDN. Los distintos enfoques de seguridad se integran en el marco de seguridad como bloques de construcción de seguridad, llamados SBB, que por separado abordan vectores de ataque específicos y riesgos de seguridad que amenazan el controlador de la red. Son ocho SBB los que conforman el marco de seguridad, cinco de ellos permiten adoptar medidas de seguridad contra vectores de ataque originados en la capa de aplicación. Mientras que otros tres SBB son los que se encargan de mitigar los riesgos originados en el plano de datos. La selección de los SBBs se hizo con el fin de abordar los siguientes problemas de seguridad: 1) las aplicaciones no autorizadas que acceden al almacenamiento interno; 2) la verificación de hosts de confianza; 3) los mensajes de protocolo manipulados por hosts maliciosos que pueden envenenar la información de la red; 4) las aplicaciones que instalan reglas de política conflictivas en el controlador; 5) aplicaciones maliciosas que hacen uso indebido de las funcionalidades del sistema central del controlador, 6) dispositivos fraudulentos que emiten mensajes OpenFlow malformados; 7) aplicaciones que abusan de los recursos de la red.

Según se ha demostrado en la siguiente citación [59], NOSArmor muestra un rendimiento competitivo en comparación con otros controladores existentes en cuanto a la seguridad de los contenidos de la red. Pues este enfoque aprovecha un concepto ampliamente difundido que es la integración de diferentes soluciones específicas para materializar nuevas soluciones que sirvan para muchos propósitos de seguridad. La única preocupación con respecto a esta propuesta es la introducción de una carga adicional de procesamiento que se podría ver reflejada en el rendimiento del controlador, aunque los autores afirman que el rendimiento no se ve afectado.

Capítulo 6. Propuestas de seguridad SDN

Después de haber hecho un repaso de sus inicios, de la arquitectura SDN, de describir varios de sus controladores, de analizar y explicar sus vulnerabilidades o ataques que pueden afectar al funcionamiento de una red definida por software y dar algunas soluciones teóricas para poder mejorar su seguridad, en esta sección del proyecto se van a aportar varias propuestas importantes que se pueden llevar a cabo en la práctica para ver verdaderamente la utilidad de una buena base de seguridad en un SDN.

La forma de trabajar en Internet, tanto a nivel de usuarios como de empresas, ha evolucionado mucho en los últimos años. Desde que se introdujo el modelo de una SDN se ha podido solucionar problemas de flexibilidad entre otras que existían antiguamente. Las SDN han sido un gran factor que ha hecho posible que exista Internet tal y como lo conocemos hoy en día. También cabe destacar que, con el crecimiento de estas redes, paralelamente, han ido en aumento diferentes ataques o malware que pueden dañar una estructura SDN. A continuación, se muestra una gráfica con la comparativa de todo el malware detectado en los últimos 10 años.

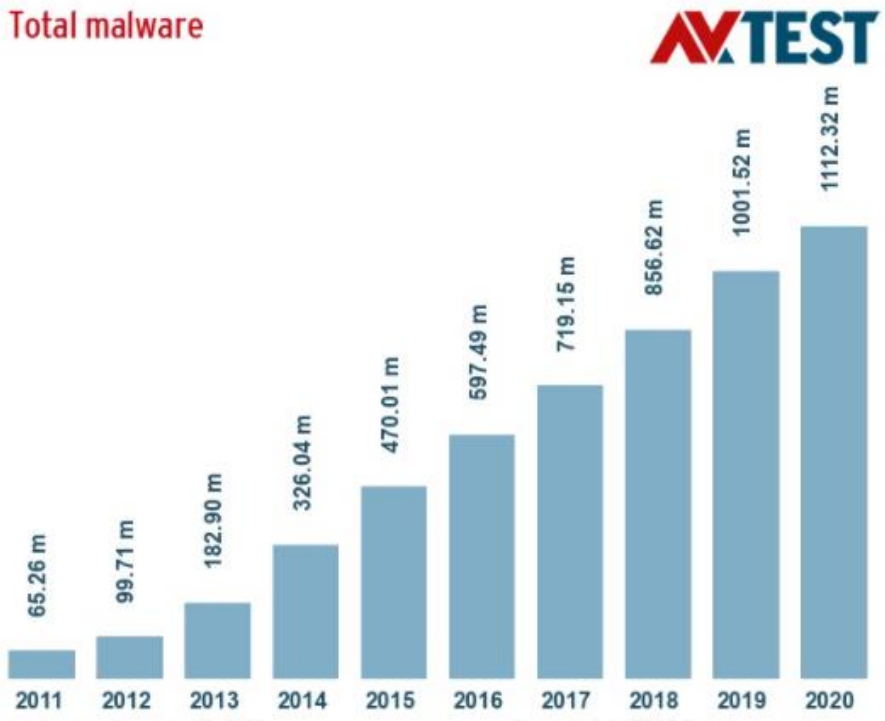

Figura 20. El malware total detectado que ha ido aumentado en los últimos 10 años. [60]

Debido a esto es muy importante tener en cuenta, a la hora de crear una red SDN, que esta posea una seguridad robusta y con dispositivos fiables ya que tarde o temprano puede ser atacada con el fin de extraer información de todo tipo. Por eso, a continuación, se proponen varias aplicaciones o diseños que se han probado y desarrollado en la actualidad con el fin de garantizar una red definida por software segura.

6.1 Science DMZ: Banco de pruebas de nube segura basado en SDN

6.1.1 Detalles de la implementación

En primer lugar, una DMZ o Zona desmilitarizada es una red perimetral que protege la red de área local (LAN) interna de una organización del tráfico malicioso o no fiable. En definitiva, una DMZ es una subred que se ubica entre Internet público y las redes privadas.

En segundo lugar, en esta propuesta los autores de [61], utilizan un honeypot basado en SDN, es decir, hacen un HoneyMix para identificar la actividad de los atacantes usando un honeypot de baja interacción. Utilizan la vulnerabilidad del sistema de la nube y la información de accesibilidad para generar gráficos de ataques que posteriormente servirán para la evaluación de la seguridad y para la selección de contramedidas basadas en SDN para las redes en la nube. También se dispone de un cortafuegos SDN que ayudara a comprobar las reglas de flujo para los conflictos de seguridad en el entorno DMZ. Utilizan el controlador SDN de OpenDaylight para la detección, resolución y visualización de conflictos de políticas de seguridad. La GUI (Interfaz gráfica de usuario) de Science DMZ está basada en un framework PHP laravel con Bootstrap [62] y utiliza API Rest para gestionar el controlador SDN, el back-end de OpenStack y otros segmentos de la red.

Se demostrará un banco de pruebas seguro, Science DMZ, para la gestión del tráfico de red, la identificación de vectores de ataques sospechosos y la aceptación de las contramedidas necesarias para evitar las brechas de seguridad. Science DMZ permitirá aprovechar la SDN para la gestión de incidentes de seguridad, eventos, políticas de trafico de red y reconocimiento de patrones de ataque. El marco de seguridad de esta DMZ detecta y resuelve automáticamente cualquier violación de las reglas de flujo de SDN.

6.1.2 Arquitectura del sistema

La arquitectura del sistema se basa en una red definida por software basada en comando y control (C&C). El servidor físico de la pasarela en el borde de la red consta de un sistema de detección de intrusos, conocido habitualmente como IDS y un proxy honeypot. El IDS comprueba las firmas de ataque para el tráfico normal y malicioso. El tráfico normal puede interactuar con la infraestructura de nube que está basada en OpenStack.

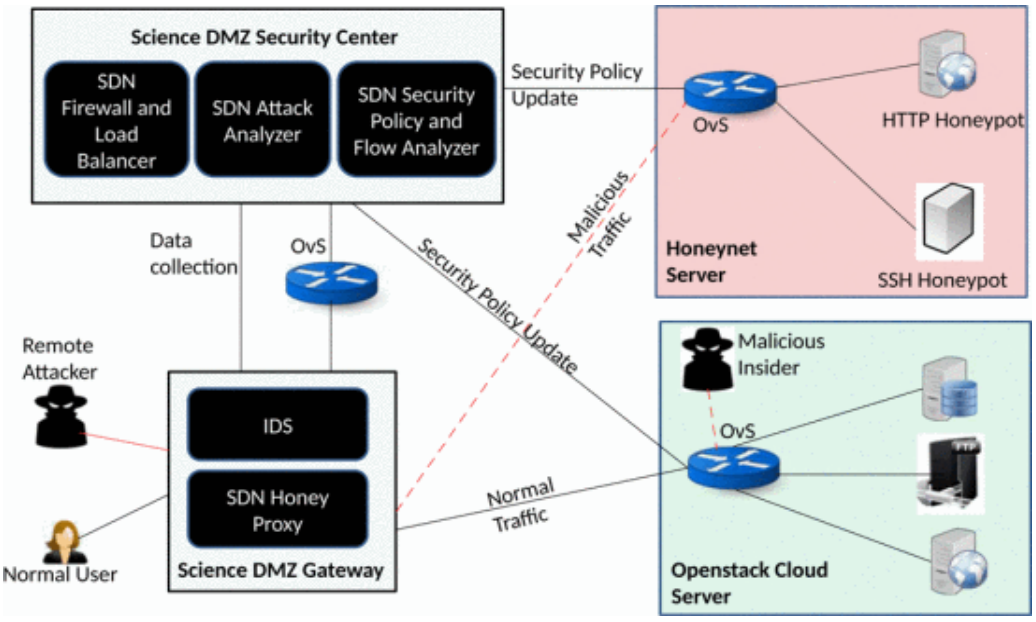

Figura 21. Arquitectura completa del sistema.

El tráfico malicioso es enviado al servidor Honeypot, donde se registran todas las actividades de ataque para realizar un análisis profundo de seguridad. A continuación, se describirán varios componentes del sistema:

- **Sistema de detección de intrusos en la red (NIDS).** El sistema de detección de intrusos (IDS), realiza la detección de patrones de ataques basados en firmas en la puerta de enlace de la red. Esta información se transmite al controlador SDN mediante las APIs de dirección norte.

- **Proxy SDN Honey.** El proxy honey y el NIDS forman la puerta de enlace o pasarela de la Science DMZ. Este componente es utilizado por el controlador SDN para detectar ataques basados en huellas digitales. El controlador SDN utilizara esta información a su beneficio para redirigir el tráfico a un honeypot de baja o alta interacción en fusión del tipo de trafico de ataque.
- **Firewall SDN y balanceador de carga.** Como se puede observar en la figura 21, el controlador SDN en el centro de seguridad Science DMZ utilizara la información de tráfico de streaming registrada por el IDS para proporcionar la funcionalidad de equilibrio de carga. El firewall/cortafuegos SDN redirigirá el tráfico normal a la nube OpenStack y el tráfico inusual al servidor Honeynet a través del proxy SDN honey.
- **Analizador de ataques SDN.** El analizador de ataques ubicado en el centro de seguridad Science DMZ, utilizará la información del sistema de vulnerabilidades para varias máquinas virtuales que se ejecutan en el sistema, registros del servidor honeynet, registros del servidor de la nube OpenStack, la información de la política del firewall para generar gráficos de ataque escalables y realizar análisis de seguridad basados en varias rutas de ataque identificadas a partir del grafico de ataque.
- **Política de seguridad de SDN y analizador de flujo.** Se encargará de detectar y resolver automáticamente cualquier infracción de la política de flujos mediante una función de seguridad adaptativa para el controlador SDN. Para encontrar cualquier incoherencia, las reglas de flujo presentes en el plano de datos se contrastan con el espacio de autorización que deniega el firewall. Al mismo tiempo, también se le comunicará al módulo de análisis unos registros para clasificar los ataques y generar nuevas restricciones de seguridad.

6.1.3 Diagrama de flujo del banco de pruebas Science DMZ

El diagrama de flujo del centro de seguridad de Science DMZ anterior describe como se gestionarán las solicitudes de conexión dirigidas a los servicios alojados en la nube de Science DMZ y los servidores honeypot.

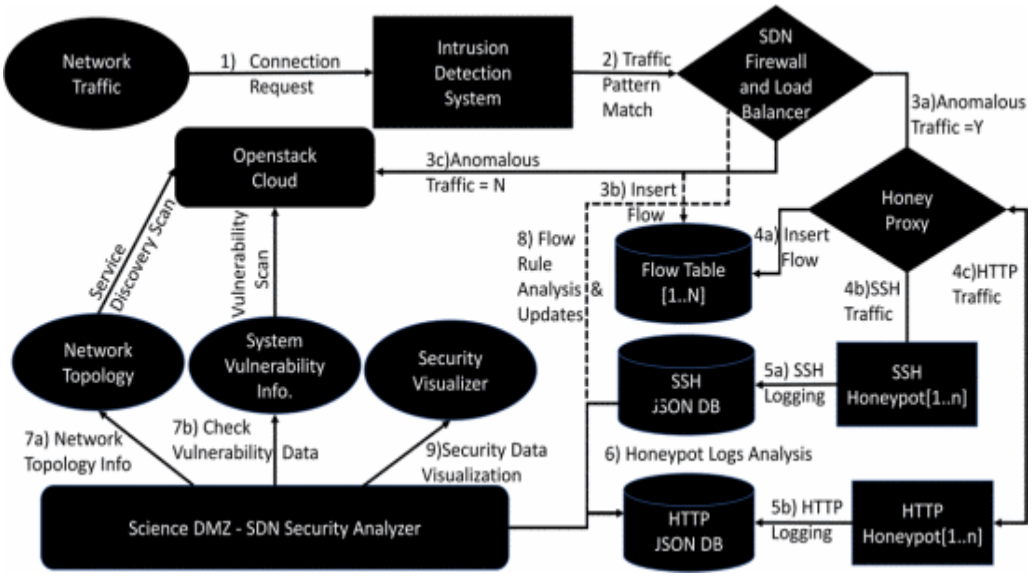

Figura 22. Información del diagrama de flujo de la Science DMZ.

- Paso 1) y 2) → Se inspecciona el tráfico en busca de patrones maliciosos y se redirige al honeypot correspondiente.
- Paso 3) → Si hay una coincidencia de firma para el tráfico inusual, reenviamos el tráfico malicioso a HoneyProxy. Cuando el volumen de tráfico es alto, el controlador SDN realiza un equilibrio de carga. El trafico normal se reenvía a la nube OpenStack.
- Paso 4) → El tráfico inusual es dirigido por HoneyProxy al honeypot apropiado y las reglas de tráfico se insertan en la tabla de flujo correspondiente.

- Paso 5) → Los registros de los honeypots SSH y HTTP se almacenan en la base de datos como recopilaciones en formato JSON.
- Paso 6) → El analizador de seguridad SDN comprueba los registros de los honeypots SSH y HTTP para reconocer el comportamiento de los usuarios.
- Paso 7) → Se inspecciona la topología de la red y la información de vulnerabilidades para crear un gráfico de ataque escalabre para el sistema y comprobar el impacto potencial del ataque desde una fuente proporcionada.
- Paso 8) → El analizador de seguridad utiliza un observador en el almacén de datos del controlador SDN OpenDaylight para saber de los conflictos de las políticas de seguridad y las actualizaciones de las reglas de flujo. La información recopilada en los pasos 6 y 7 se usa para actualizar las políticas del firewall en el controlador SDN.
- Paso 9) → La información de los pasos 7 y 8 se usa para la visualización, regeneración de los gráficos de ataques y la actualización de los conflictos de reglas de flujo de la SDN.

6.1.4 Aprendizaje y conclusión

Este banco de pruebas de Science DMZ llevado a cabo en la práctica, muestra la gestión y la seguridad de una red en la nube utilizando redes definidas por software. Con la realización de la investigación de esta propuesta se ha conocido los componentes principales como el firewall habilitado para SDN, el comprobador de conflictos de reglas de flujo, el honeypot SDN, el equilibrador de carga y el analizador de seguridad Science DMZ. La interfaz gráfica de usuario que se ha propuesto con este diseño permite a los usuarios del sistema organizar y controlar los problemas de seguridad en un entorno de nube. Por lo tanto, esta Science DMZ puede ser implantada y aprovechada por grandes empresas para garantizar la seguridad y el buen funcionamiento de las redes que estén bajo su uso.

6.2 Detección y mitigación de ataques DDoS en el plano de datos de SDN

6.2.1 Introducción

En el plano de datos de SDN, los dispositivos de red individuales y los switches son bastantes vulnerables a distintos tipos de ataques, como por ejemplo el ataque de denegación de servicio (DoS), el ataque de denegación de servicio (DDoS), la modificación de datos, el repudio o el ataque de canal lateral, los cuales ya se han visto e introducido en el capítulo 4/5 de este proyecto.

El ataque DDoS es el más popular en el plano de datos, por lo que los dispositivos de la red no pueden ser utilizados o consultados por los usuarios legítimos. Los ataques se concentran en función del tipo de víctima, clasificándolos como ataques de protocolo, ataques de ancho de banda o ataques lógicos. Muchos expertos/analistas han declarado que actualmente no existen defensas totalmente exitosas contra un ataque de denegación de servicio distribuido. Hay muchas medidas de seguridad que puede implementar un host o una red para hacer que la red y las redes subyacentes sean más seguras.

Los autores de esta propuesta [63], han decidido desarrollar una aplicación de red definida por software aprovechando el protocolo OpenFlow para estudiar una defensa de la red contra un ataque DDoS. El banco de pruebas que utilizaron es el conocido emulador de redes Minimet, donde se construyen switches OVS (Open vSwitch) y se controlan mediante un controlador SDN, llamado OpenDaylight (ODL). Desarrollaron un script en Python para detectar y mitigar un ataque de inundación distribuido, ICMP (Protocolo de mensajes de control de Internet) y UDP (Protocolo de datagramas del usuario), desde todos los hosts posibles hacia una víctima en este caso, un servidor online. Los resultados obtenidos de la evaluación muestran los rendimientos de TCP, el tiempo de ida y vuelta, la perdida de paquetes antes y después del ataque, además del tiempo de detección y mitigación medible por los usuarios de la red emulada para demostrar la aplicación de SDN en la resolución de los efectos adversos del ataque.

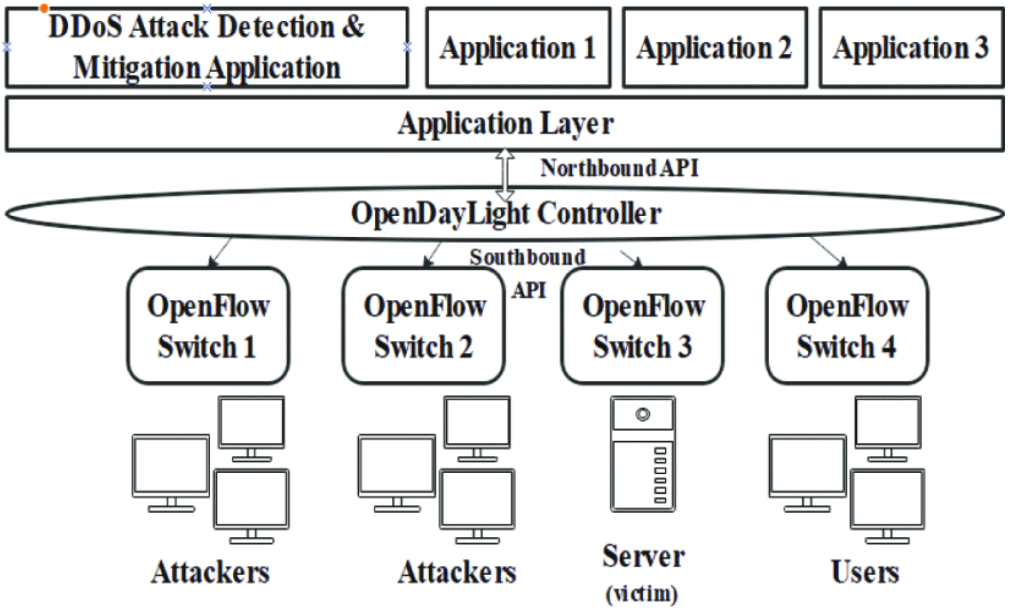

Figura 23. Arquitectura SDN con el potencial ataque DDoS.

6.2.2 Implementación y despliegue de la propuesta

En este apartado se describirá las características y el diseño de la propuesta que los autores plantearon en el artículo [63]. Esta propuesta mostrará como los ataques DDoS pueden consumir los recursos de los servidores en línea y proporcionará una solución para detectarlos y mitigarlos. El sistema que se propuso aprovecha la programación con naturaleza dinámica de la SDN e implementa un mecanismo de protección DDoS adaptativos. Como se ha descrito antes, se utiliza el controlador OpenDaylight y el banco de pruebas Minimet ya que soporta OpenFlow para un enrutamiento personalizado altamente flexible en redes definidas por software. La topología que se utilizó para llevar a cabo este diseño es una topología modificada, la cual podemos observar en la figura 25.

6.2.2.1 Arquitectura del sistema

El sistema funciona mediante un control de bule entre tres componentes básicos de la arquitectura. Como podemos observar en la figura 24: La parte A contiene el controlador OpenDaylight, la parte B incluye la topología de la red y la parte C contiene la aplicación SDN.

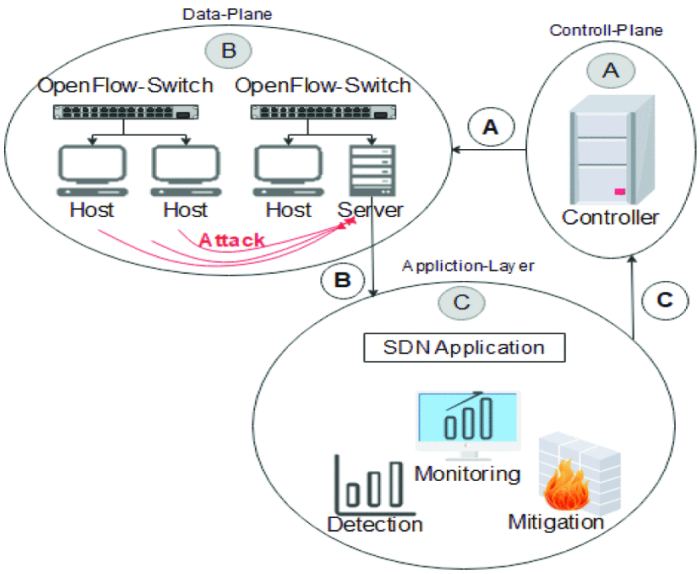

Figura 24. Modelo conceptual del sistema.

Como ya se ha explicado en el capítulo 3, el controlador OpenDaylight es un controlador SDN de código abierto, gestionado por la Fundación Linux. Es uno de los controladores más populares del momento ya que expone APIs abiertas de dirección sur, en este caso será utilizada por el enlace A para comunicarse con la parte B. La parte B es la que se encargará de emular los dispositivos de red donde se puede iniciar un ciberataque de manera potencial. El emulador de red se utilizará para desplegar un dominio de red con algunos hosts y switches OVS, el switch basado en software ejecutará la parte del cliente del protocolo OpenFlow y los hosts ejecutaran software de red Linux estándar.

La topología de la red utilizada por los autores consiste en sesenta y cuatro hosts, denominados desde Hosts-1 a Hosts-64 (Con la dirección de red de 10.0.1.0/24 a 10.0.8.0/24 cada red relacionada con un switch). Los atacantes son algunos de estos hosts que quieren interrumpir la conectividad entre usuarios y el servidor afectado. La víctima es un servidor conectado al switch número dos con la dirección IP 10.0.2.60.

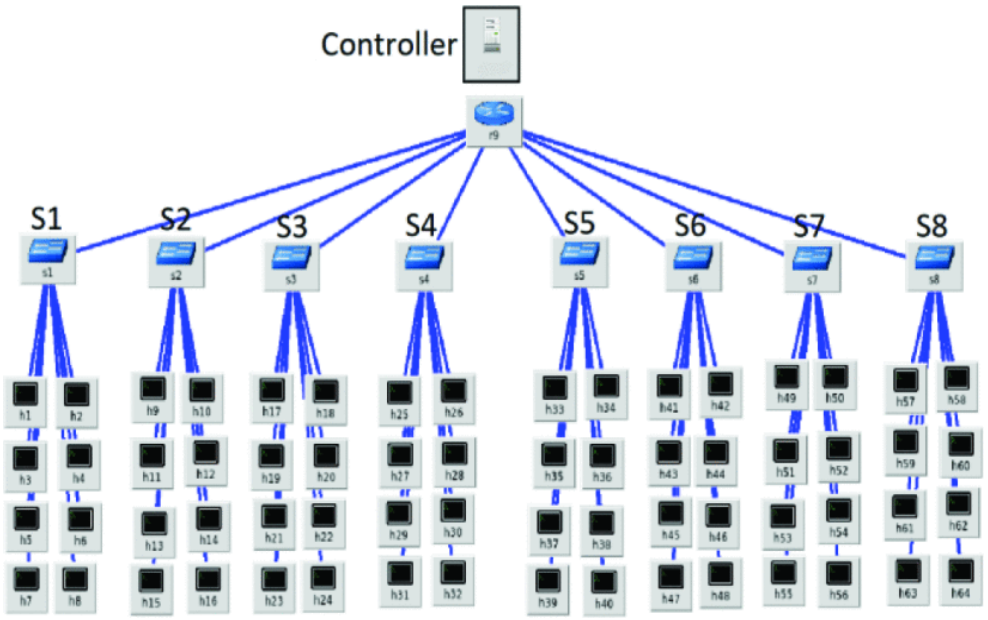

Figura 25. Topología de la red creada en Minimet para llevar a cabo esta propuesta.

6.2.2.2 Planteamiento del escenario de ataque

El escenario planteado asume que el ataque DDoS debe ser preferentemente mitigado, para evitar que el servidor sufra los efectos negativos de rendimiento introducidos por un ataque DDoS. Los autores seleccionan aleatoriamente el host 1, el host 42 y el host 64 para atacar al servidor, y el host 12 será el que mide la accesibilidad del servidor y el rendimiento del servicio antes y después del ataque a la red.

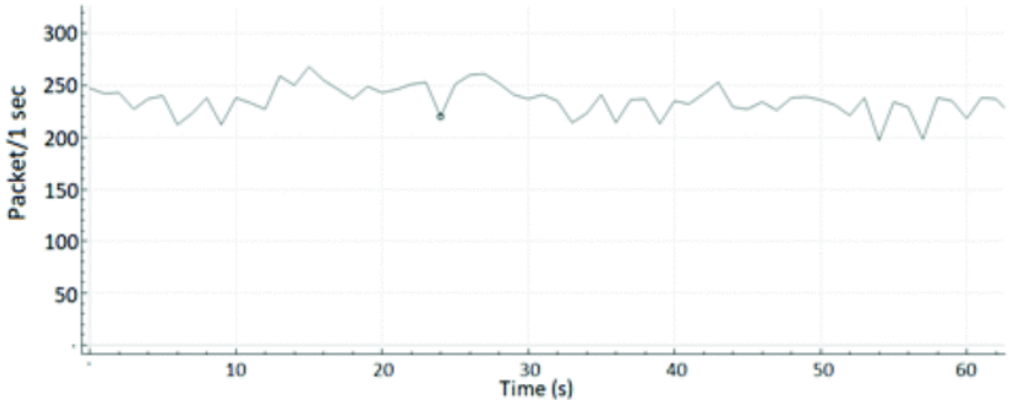

Figura 26. Gráfico que muestra el tráfico normal del servidor cuando no hay ningún ataque.

Los hosts atacaran el servidor utilizando la herramienta Hping3. Hping3 es la herramienta de red elegida para llevar a cabo este ataque ya que es capaz de enviar paquetes TCP/IP personalizados. Soporta los protocolos TCP, UDP, ICMP y RAW-IP, también permite enviar paquetes manipulados y permite controlar el tamaño, la cantidad y fragmentación de los paquetes para sobrecargar el objetivo y eludir o ataque los firewalls. Se realiza un ataque de ancho de banda (inundación ICMP y UDP) contra la víctima, estos ataques DDoS de ancho de banda o volumétricos están diseñados para abrumar la capacidad de la red interna con volúmenes de tráfico malicioso significativamente altos. En este caso se aplican los siguientes comandos con el fin de consumir el ancho de banda del servidor 10.0.2.60:

```
Icmp-flooding.....
root@mininet-vm:~# hping3 -V -1 -d 1400 --faster
10.0.2.60
UDP-Flooding.....
root@mininet-vm:~# hping3 --udp  10.0.2.60 -p 445
-flood
```

Figura 27. Comandos ICMP y UDP aplicados para llevar a cabo el ataque de inundación.

6.2.2.3 Solución propuesta

Las practicas actuales en las redes tradicionales consisten en confiar en el firewall situado en la frontera del dominio de red o en la puerta de enlace para que deje caer los paquetes dañinos. Con la SDN todos los conmutadores OVS pueden reprogramarse para dejar caer el tráfico de los atacantes lo antes posible. La solución que se propone es utilizar una aplicación SDN (la cual se escribió en Python por parte de los autores) para capturar y analizar el tráfico hacia el servidor en cuestión. Cuando se detecta un comportamiento inusual en el tráfico, la aplicación comienza a analizar los paquetes para extraer la IP del atacante de acuerdo con los mayore remitentes de tráfico, que reenvían un tráfico enorme. Para que los usuarios normales puedan llegar fácilmente al servidor, aplican una regla de caída al switch al que se conectó el servidor. En la figura 28 se muestra la regla que se generó desde la aplicación a la API del controlador OpenDaylight.

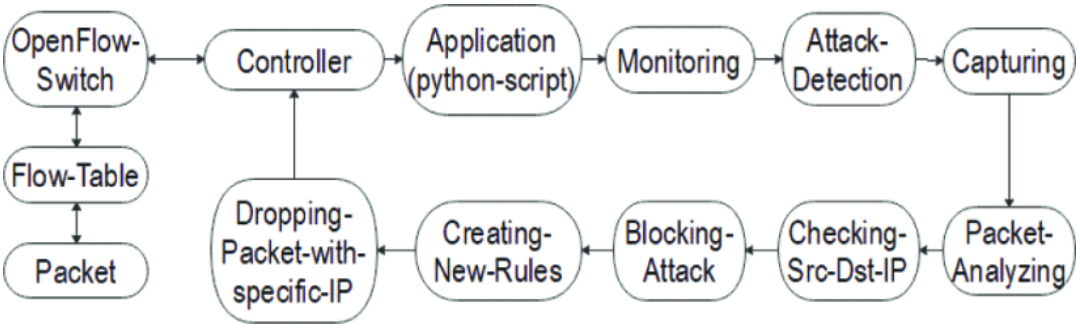

Figura 28. Flujo de trabajo del sistema para cada paquete recibido.

6.2.3 Resultados

En este apartado se describirán las pruebas de evaluación de la propuesta y, así pues, se mostrará los resultados obtenidos por los autores de [63]. En la siguiente imagen podemos ver una mejor idea del impacto que tiene el ataque con respecto a la QoS en los hosts mientras que la red se encuentra bajo el ataque. Los resultados del tiempo medio de respuesta ICMP se obtuvieron del comando ping del host12 hacia el servidor.

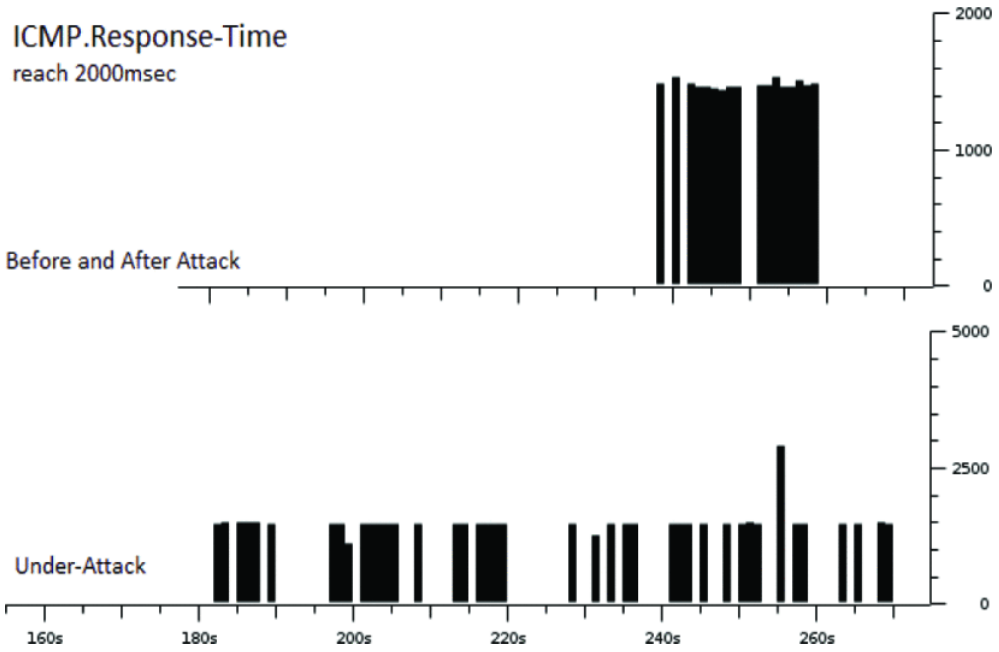

Figura 29. Tiempo de respuesta ICMP del host12 al servidor (Antes y después del ataque, y durante el ataque).

En la figura 30 se muestra una caída significativa en el rendimiento de TCP debido al comportamiento malicioso que hay en la red mientras la duración del ataque DDoS. En primer lugar, podemos ver cómo se comporta el ataque DDoS, en segundo lugar, la gráfica tenemos el tráfico TCP antes y después del ataque y por último nos encontramos con la superposición de los dos anteriores.

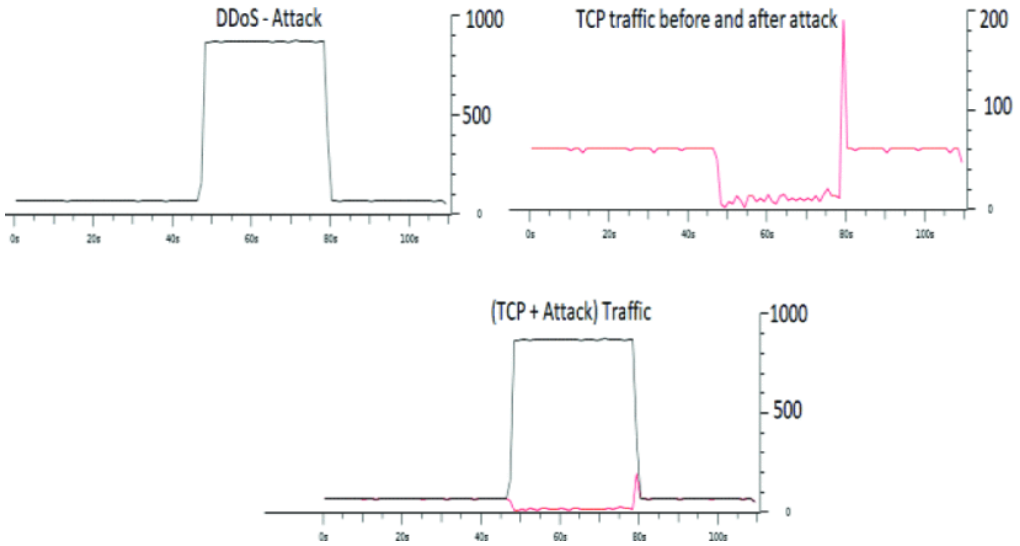

Figura 30. Las peticiones TCP del host 2 al servidor a través del ataque.

La aplicación está diseñada para añadir las IPs de los atacantes en el archivo JSON para aplicarlo al OVSswitch. A continuación, se muestra una imagen que contiene la tabla de flujo abierto en el switch 2 después de la mitigación del ataque DDoS. Se puede ver como las tres primeras reglas fueron establecidas por la aplicación con alta prioridad ya que las IPs coinciden con cualquier paquete con las direcciones IPs de los atacantes, por lo tanto, la aplicación ejecuta la acción de **drop** y descarta dichos paquetes.

```
    root@mininet-vm:~# sudo ovs-ofctl --o dump-
flows s2
    NXST_FLOW reply (xid=0x4):
     cookie=0x0,     duration=2340.97s,     table=0,
n_packets=1864,  n_bytes=1099760,  idle_age=2340,
priority=1000,ip,nw_src=10.0.6.0/24 actions=drop
     cookie=0x0,     duration=2340.97s,     table=0,
n_packets=146971, n_bytes=211922774, idle_age=43,
priority=1000,ip,nw_src=10.0.1.0/24 actions=drop
    cookie=0x0,duration=2340.947s,table=0,n_packe
ts=81435285,n_bytes=63764255230,   idle_age=2340,
priority=1000,ip,nw_src=10.0.8.0/24 actions=drop
     cookie=0x0,     duration=1032.778s,     table=0,
n_packets=0,      n_bytes=7378,      idle_age=1032,
priority=10,ip,nw_dst=10.0.2.30 actions=output:4
     cookie=0x0,     duration=1032.767s,     table=0,
n_packets=75,       n_bytes=0,       idle_age=1032,
priority=10,ip,nw_dst=10.0.2.60 actions=output:7
     cookie=0x0,     duration=1032.775s,     table=0,
n_packets=0, n_bytes=0, idle_age=1032,
priority=10,ip,nw_dst=10.0.2.80 actions=output:9
     cookie=0x0,     duration=1032.781s,     table=0,
n_packets=74,      n_bytes=7280,      idle_age=1032,
priority=10,ip,nw_dst=10.0.2.20 actions=output:3
     cookie=0x0,     duration=1032.786s,     table=0,
n_packets=0, n_bytes=0, idle_age=1032,
priority=10,ip,nw_dst=10.0.2.50 actions=output:6
     cookie=0x0,     duration=1032.762s,     table=0,
n_packets=0,       n_bytes=0,       idle_age=1032,
priority=10,ip,nw_dst=10.0.2.70 actions=output:8
     cookie=0x0,      duration=1032.79s,     table=0,
n_packets=0,       n_bytes=0,       idle_age=1032,
priority=65535,ip,dl_dst=00:00:00:00:01:02
actions=output:1
     cookie=0x0,     duration=1032.793s,     table=0,
n_packets=10,       n_bytes=420,       idle_age=61,
priority=1,arp actions=FLOOD=1,arp actions=FLOOD
```

Figura 31. Tabla de reglas de flujo de OpenvSwitch tras la mitigación del ataque DDoS.

El ataque DDoS fue detectado y mitigado cerca de 100 y 150 segundos, a partir de entonces, la red empieza a estabilizarse en torno a su estado de funcionamiento "normal" después de utilizar los dos ataques de flujo: 1) Inundación ICMP y 2) Inundación UDP.

En las figuras 32 y 33 podemos ver como se realizó la inundación de paquetes ICMP.

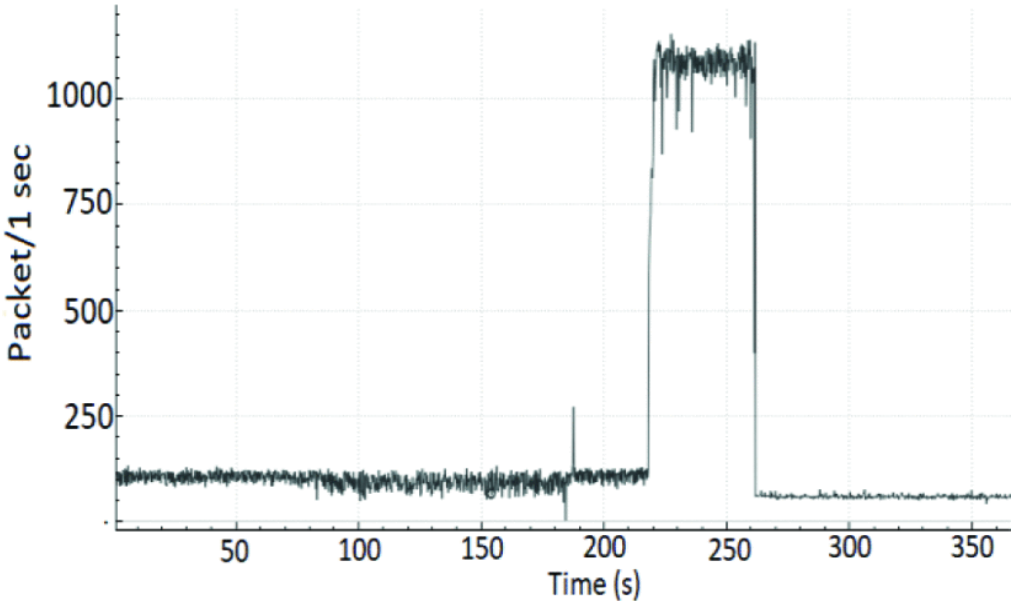

Figura 32. Tasa de paquetes del servidor antes y después de la mitigación del ataque DDoS (Inundación ICMP).

Time	Source	Destination	Protoco	Lengtl	Info
46.006276006	10.0.1.20	10.0.2.60	ICMP	1442	Echo (ping) request
46.006364006	10.0.2.40	10.0.2.60	ICMP	1442	Echo (ping) request
46.006404006	10.0.1.10	10.0.2.60	ICMP	1442	Echo (ping) request
46.006448006	10.0.8.80	10.0.2.60	ICMP	1442	Echo (ping) request
46.006517006	10.0.7.20	10.0.2.60	ICMP	1442	Echo (ping) request
46.006801006	10.0.2.40	10.0.2.60	ICMP	1442	Echo (ping) request
46.006849006	10.0.1.10	10.0.2.60	ICMP	1442	Echo (ping) request
46.006875006	10.0.7.20	10.0.2.60	ICMP	1442	Echo (ping) request
46.006898006	10.0.8.80	10.0.2.60	ICMP	1442	Echo (ping) request
46.006920006	10.0.1.20	10.0.2.60	ICMP	1442	Echo (ping) request
46.006992006	10.0.1.10	10.0.2.60	ICMP	1442	Echo (ping) request
46.007039006	10.0.2.40	10.0.2.60	ICMP	1442	Echo (ping) request
46.007062006	10.0.7.20	10.0.2.60	ICMP	1442	Echo (ping) request
46.007085006	10.0.8.80	10.0.2.60	ICMP	1442	Echo (ping) request
46.007109006	10.0.1.20	10.0.2.60	ICMP	1442	Echo (ping) request
46.007200006	10.0.2.40	10.0.2.60	ICMP	1442	Echo (ping) request
46.007253006	10.0.1.10	10.0.2.60	ICMP	1442	Echo (ping) request
46.007280006	10.0.7.20	10.0.2.60	ICMP	1442	Echo (ping) request
46.007304006	10.0.8.80	10.0.2.60	ICMP	1442	Echo (ping) request
46.008792006	10.0.1.10	10.0.2.60	ICMP	1442	Echo (ping) request
46.008925006	10.0.2.40	10.0.2.60	ICMP	1442	Echo (ping) request

Figura 33. Ataque de inundación ICMP captura a través de Wireshark.

Mientras que, en las siguientes figuras, 34 y 35, se muestra la inundación de paquetes UDP.

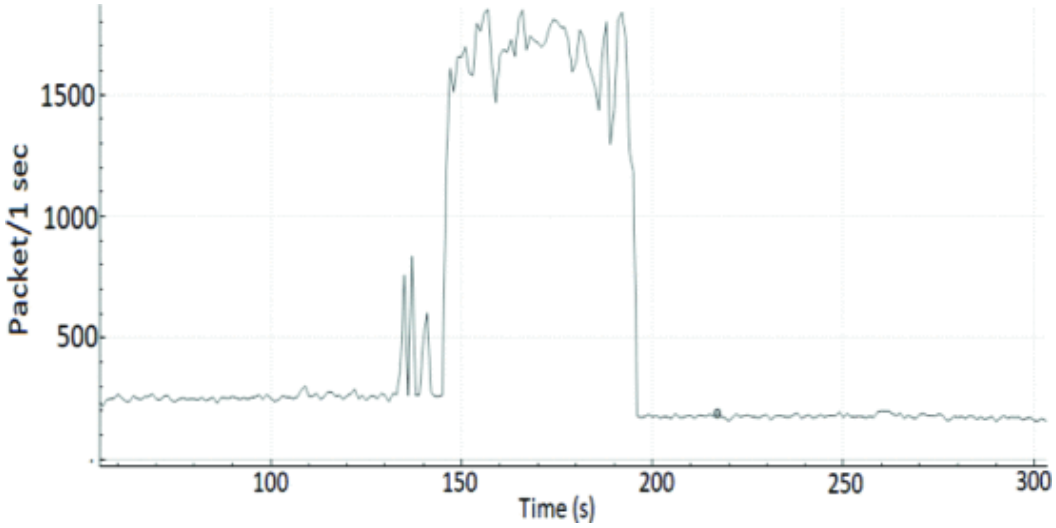

Figura 34. Tasa de paquetes del servidor antes y después de la mitigación del ataque DDoS (Inundación UDP).

Time	Source	Destination	Protoco	Lengtl	Info
82.31134700(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41863
82.31141300(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41865
82.31148300(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41866
82.31155300(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41868
82.31162500(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41870
82.31169500(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41873
82.31174700(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41876
82.31181400(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41879
82.31188000(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41882
82.31194600(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41885
82.31201200(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41888
82.31207800(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41891
82.31215800(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41895
82.31222400(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41898
82.31228800(	10.0.6.20	10.0.2.60	UDP	42	Source port: 41901
82.31235400(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25495
82.31242100(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25496
82.31248700(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25497
82.31261000(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25498
82.31262400(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25499
82.31268300(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25500
82.31275100(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25501
82.31282000(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25502
82.31288600(	10.0.1.10	10.0.2.60	UDP	42	Source port: 25503

Figura 35. Ataque de inundación UDP capturado a través de Wireshark.

6.2.4 Análisis de los resultados

Esta propuesta que se ha explicado anteriormente detecta y mitiga los ataques DDoS y los limita en su origen, a través de la escritura, desarrollada por los autores en Python, de una aplicación basada en SDN que trabaja con el controlador OpenDaylight con el fin de captura y analizar el tráfico hacia la víctima en un rango de tiempo aproximado de entre 100 y 150 segundos. Como hemos visto el diseño de la aplicación se centra en una solución que funciona particularmente bien para SDN, basándose en sus especificaciones, puntos de fuerza, limitaciones y utilizando el hecho de que la especificación SDN dicta el reenvió de paquetes al controlador. Puesto que en los últimos años ha incrementado el uso de ataques DDoS, esta aplicación nos sirve de gran utilidad para detectar y mitigar estos ataques, pero también se puede utilizar para limitar la velocidad de salida de los paquetes de un puerto de conmutación y para la implementación de la QoS, en lugar de la regla de acción de caída.

Capítulo 7. Conclusiones

La arquitectura SDN es una revolución en la gestión y en el control de la red, añadiendo características especiales que mejoran las distintas funciones de la red y al mismo tiempo proporcionan soluciones a problemas engorrosos que están presentes en las redes convencionales. El control centralizado y la programabilidad de la red contribuyen en la agilización de la creación de prototipos y desarrollo de funciones de red, en general, la mayoría de las funciones de red que se encuentren en las arquitecturas convencionales pueden ser plasmadas en la SDN en forma de sencillas implementaciones de software. La seguridad completa de un sistema o de la red también está en el ámbito de la innovación de la red a través de la aplicación de las características de SDN; los trabajos y las propuestas en el estado del arte detallan el aumento de las características SDN en el desarrollo y prestación de diferentes funciones de seguridad de la red.

A pesar de la introducción de nuevos esquemas de seguridad y la mejora de los que ya existen, que a su vez proporcionan nuevas herramientas y mecanismos para reforzar la seguridad y la protección de la red, la seguridad fiable en la SDN no puede garantizarse por completo y la obsesión por conseguir la seguridad completa de la red o de un sistema puede desembocar en sistemas inestables e ineficientes. Además, las capas e interfaces adicionales en la SDN propician sin problemas la aparición de nuevas vulnerabilidades y amenazas de seguridad. Esta última afirmación define dos hojas de ruta de investigación y desarrollo en la disciplina de seguridad de las SDN, por un lado, se encuentra la rama de la investigación destinada a aprovechar las características constitutivas de la SDN para mejorar la seguridad de la red, mientras que, por otro lado, está la rama que se centra en promover una arquitectura SDN segura y fiable con sus capas e interfaces asociadas.

Las vulnerabilidades y los ataques de la red en SDN son cada vez más complejos y sofisticados. Por lo tanto, permiten la aparición de nuevos retos que obligan a la investigación continua de la seguridad y así dar forma a la arquitectura SDN para: estar en constante interacción con diferentes tecnologías; e intentar integra elementos y atributos que pueden ser aplicados en la construcción de marcos robustos de seguridad innovadores y competentes. Algoritmos de aprendizaje automático (machine learning), servicios en la nube, funciones de red virtualizadas, son buenos ejemplos de tecnológicas que pueden unirse para la construcción del entorno de seguridad SDN de esta generación y de generaciones próximas.

El despliegue de SDN en las redes de producción es todavía una visión y aún queda mucho trabajo por hacer en cuanto a nivel de seguridad de SDN para convertir esta visión en una realidad. Hay problemas de seguridad esenciales que requieren atención y cuestiones latentes que aún no se han descubierto, por lo que el campo de investigación sigue abierto y cada nueva aplicación o nuevo diseño contribuye en la ayuda para cerrar la brecha entre la teoría y la práctica. El objetivo de la seguridad de la SDN debe apuntar a lograr un marco de seguridad automatizado, auto-construible, verificable y con la capacidad de dar un diagnostico propio, es decir, un marco de seguridad lo suficientemente amplio como para atender diferentes situaciones y niveles de criticidad, lo suficientemente reconfigurable como para configurar eficazmente su estructura según la gravedad y el impacto de la situación correspondiente, y que sea capaz de distinguir y resolver las incompatibilidades que residen dentro de la propia estructura del sistema, ya que esas inconsistencias podrían desencadenar un rendimiento anómalo imprevisible.

Aunque ha pasado mucho tiempo desde que se publicaron las primeras investigaciones y propuestas en seguridad SDN, hay varios retos y temas abiertos que no han madurado lo suficiente y se necesitan grandes esfuerzos en esos campos específicos, por ejemplo, en el análisis forense de SDN, mediadores de políticas, depuradores SDN y el descubrimiento de vulnerabilidades. De alguna manera los despliegues y los ensayos de las investigaciones sobre la seguridad de las redes SDN ha llevado a la comunidad científica a centrarse en algunos temas de moda, como los motores de detección basados en el aprendizaje automático, los planos de datos programables y las funciones de seguridad virtualizadas, dejando un poco de lado aspectos importantes de

seguridad y abriendo una brecha en el desarrollo de nuevas estrategias de seguridad que no solo atienden a estos temas sobrevalorados si no que son necesarias para la construcción de marcos de seguridad unificados, robustos y completos.

Finalmente se ha optado por escoger y analizar dos aplicaciones o propuestas realizadas a nivel practico, con las cuales hemos podido observar que realmente existen diseños que proporcionan un nivel de seguridad robusto basado en las redes definidas por software. Por lo tanto, se puede afirmar que es conveniente el uso de la implementación de esta tecnología SDN, tanto para mejorar la seguridad como para agilizar otros procesos de red, ya que tiene ventajas sobre las redes convencionales y además pueden convivir con ellas.

7.1 Investigaciones futuras

Podemos afirmar que SDN es y será una de las tecnologías más prometedores e innovadoras de esta década pues aprovecha la separación entre el plano de control y el plano de datos para compensar las deficiencias de escalabilidad, flexibilidad y eficacia de las redes tradicionales. Las SDN han recibido una gran atención por parte del mundo académico y de parte del sector de la industria, y se ha utilizado ampliamente en centro de datos, computación en la nube, LAN inalámbrica, redes inteligentes, casas y hogares inteligentes, y otros escenarios de aplicación. En cuanto a seguridad se refiere, la SDN mejora la seguridad de la red gracias a la visibilidad global del estado de la red, la inteligencia centralizada y la programabilidad de la red. Así, una capa de distribución común recoge información sobre los requisitos de seguridad de los distintos servicios, recursos y hosts para aplicar la seguridad a los elementos de la red y poder aplicar las políticas de seguridad con el objetivo de formar una aplicación de seguridad robusta y escalable. Sin embargo, los mismos atributos principales de la SDN, es decir, esa inteligencia centralizada y los elementos de red programables, ha traído consigo una serie de nuevos problemas que hacen que la seguridad en SDN sea una tarea difícil de realizar. Por ello, se han propuesto diversas soluciones y plataformas de seguridad SDN que se describen en las secciones anteriores. Las soluciones que existen no son suficientes para aliviar por completo los problemas de seguridad de las redes SDN, por lo tanto, a continuación, se indican algunos retos de investigación y direcciones futuras [64].

A. Escalabilidad del controlador y comunicación entre dominios

La escalabilidad es uno de los principales retos a los que se enfrenta la arquitectura SDN lógicamente centralizada. En las SDN, a mediad que el tamaño y el diámetro de la red, la cantidad de trafico de control que va dirigido al controlador aumento y por lo tanto el tiempo de establecimiento del flujo crece. Además, se sabe que la capacidad y el conjunto de operaciones de un controlador OpenFlow es probablemente limitado. Por lo tanto, la falta de escalabilidad en SDN puede permitir ataques de inundación dirigidos a la comunicación entre el controlador y el switch para causar la saturación del plano de control. Entonces, ¿Cómo se puede garantizar la comunicación segura y en tiempo real entre múltiples controladores?, esta será una cuestión importante a resolver en el futuro de la seguridad de la SDN [64].

B. Problemas de autentificación de la aplicación

Los problemas de las aplicaciones maliciosas son uno de los mayores problemas de seguridad en SDN. Para evitar desplegar aplicaciones maliciosas o dañadas, se debe de establecer una conexión de confianza entre la capa de aplicación y la capa de control. Además, se necesita autenticar la identidad de los dispositivos que van a estar conectados a la red antes de intercambiar mensajes de control.

C. Estandarización de la interfaz norte

Con el continuo desarrollo de la tecnología SDN en los últimos años, OpenFlow se ha convertido en el protocolo estándar de la interfaz de dirección sur. Sin embargo, la estandarización del protocolo de interfaz norte todavía se enfrenta a diferentes problemas. La complejidad de los diversos tipos de controladores, la diversificación de los sistemas

operativos y la finalización de la composición funcional limitan la portabilidad y la escalabilidad de la interfaz norte. Por lo tanto, la estandarización de la interfaz norte será un factor importante del trabajo de seguridad de la SDN en el futuro.

D. Modelos de desarrollo y programación

La SDN permite a los desarrolladores establecer nuevas arquitecturas de red, protocolos y aplicaciones, y probarlas o utilizarlas en redes operativas. Esta capacidad aportara innovación en las redes, pero por otro lado también puede introducir problemas de seguridad debido a la gran cantidad de nuevas aplicaciones que puede ejecutarse en la red.

Por ejemplo, si una aplicación se desarrolla en un entorno que usa el plano de control y este se encuentra en un entorno diferente, la funcionalidad del plano de datos podría verse afectada por vulnerabilidad de seguridad además de otros retos, como la colisión de políticas de seguridad. Por lo tanto, los modelos y paradigmas de programación adecuados y los entornos de desarrollo deben ser estandarizados para minimizar las posibilidades de módulos conflictivos que creen vulnerabilidades de seguridad. Los módulos conflictivos en los sistemas distribuidos pueden crear vulnerabilidades de seguridad, como la exposición de información sensible de la red o APIs, y crear reglas de flujo contradictorias.

E. Automatización de la seguridad de la red

Otro de los retos futuros que se encuentran en materia de seguridad SDN y que es uno de los más cautivador y con mucha proyección es, el uso de aprendizaje automático. La creciente complejidad, el dinamismo, los requisitos de fiabilidad y la escalabilidad están haciendo que la gestión y la supervisión de las redes de comunicación sean mucho más difíciles. A pesar de esto, los desarrolladores están trabajando cada vez más en las técnicas de automatización ya que estas darían cierta estabilidad a la red si consiguiesen por sí solas, actualizar acciones futuras y aprender a resolver los desafíos a los que se expone la red.

La ONF afirma que las SDN ofrecen un marco flexible de automatización y gestión de la red, que permite desarrollar herramientas para automatizar las tareas de gestión para reducir la sobrecarga operativa, disminuir la inestabilidad de la red y apoyar los nuevos modelos. Existen marcos de automatización para la QoS, implementación de políticas a través de Procera, plataforma de control automático de respuesta OMNI o el proyecto Poseidon combinado con Faucet como controlador SDN para dar lugar a un sistema inteligente y reactivo. Sin embargo, aún no se han demostrado mecanismos claros y viables de automatización de la seguridad de la SDN.

Capítulo 8. Bibliografía

[1] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, "Security in SDN: A comprehensive survey"

[2] SDxCentral Studios, 2013. SDN Security - Challenges Implementing SDN Network Security in SDN Environments. https://www.sdxcentral.com/networking/sdn/definitions/security-challenges-sdn-software-defined-networks/

[3] Meru Networks, "Demystifying Software-Defined Networking for Enterprise Networks" (2013) http://www.merunetworks.com/collateral/solution-briefs/anintroduction-to-sdn-sb.pdf

[4] Lara, A., Kolasani, A., Ramamurthy, B., 2014a. Network innovation using openflow: a survey. IEEE Commun. Surv. Tutor. 16 (1), Paginas 493–512, https://doi.org/10.1109/SURV.2013.081313.00105

[5] Journal of Network and Systems Management, Sanjeev Singh & Rakesh Kumar Jha, "A Survey on Software Defined Networking: Architecture for Next Generation Network" Recuperado de https://link.springer.com/article/10.1007/s10922-016-9393-9#Sec36

[6] Kreutz, D., Ramos, F.M.V., Verssimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S., 2015. Software-defined networking: a comprehensive survey. Proc. IEEE 103 (1), 14–76, https://doi.org/10.1109/JPROC.2014.2371999.

[7] Dijiang Huang, Ankur Chowdhary, Sandeep Pisharody, (2018), "Software-Defined Networking and Security: From Theory to Practice", Apartado 3.3.3

[8] Latif, Z., Sharif, K., Li, F., Monjorul Karim, M., Biswas, S., & Wang, Y. (2020, 15 abril). A comprehensive survey of interface protocols for software defined networks. ScienceDirect, 156(2020). Recuperado de https://www.sciencedirect.com

[9] Wang, R., Butnariu, D., Rexford, J.: OpenFlow-based server load balancing gone wild. In: Proceedings of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE '11), USENIX Association Berkeley, CA, USA, Pagina 12 (2011)

[10] Blog de Kspviswa, "OpenFlow Version RoadMap", (2016), Imagen recuperada de https://kspviswa.github.io/OpenFlow_Version_Roadmap.html

[11] Mathias Duarte, Eduardo Grampin, Martin Giachino, "Herramientas de simulación/emulación SDN", (Tesis de licenciatura en computación), Imagen recuperada de https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/19028/1/2526.pdf

[12] McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. (2008) "OpenFlow: Enabling Innovation in Campus Networks" http://archive.openflow.org/documents/openflow-wplatest.pdf

[13] Braun, W., Menth, M.: Software-defined networking using OpenFlow: protocols, applications and architectural design choices. Future Internet 6, 302–336

[14] Stallings, W., (2013). Software-defined networks and openflow. "Inter. Protocol" J. 16 (1), Páginas 2–14. Recuperado de https://wxcafe.net/pub/IPJ/ipj16-1.pdf

[15] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey"

[16] Zhu, Liehuang & Karim, Md Monjurul & Sharif, Kashif & Li, Fan & Du, Xiaojiang & Guizani, Mohsen. (2019). SDN Controllers: Benchmarking & Performance Evaluation. Recuperado de https://arxiv.org/pdf/1902.04491.pdf

[17] Dijiang Huang, Ankur Chowdhary, Sandeep Pisharody, (2018), "Software-Defined Networking and Security: From Theory to Practice", Apartado 3.3.6. https://doi.org/10.1201/9781351210768

[18] Óscar Roncero Hervás, Software Defined Networking, (Master en ingeniería Telemática) Recuperado de https://upcommons.upc.edu/bitstream/handle/2099.1/21633/Memoria.pdf

[19] NOX. Recuperado de http://www.noxrepo.org/

[20] POX Manual Current documentation. (2015). POX Manual. Recuperado de https://noxrepo.github.io/pox-doc/html/

[21] OpenDaylight Project, Recuperado de https://en.wikipedia.org/wiki/OpenDaylight_Project Linux Foundation, OpenDaylight. Recuperado de www.opendaylight.org

[22] Foodlight, Recuperado de https://docs.huihoo.com/open-networking-summit/2012/floodlight-openflow-controler.pdf

[23Ryu documentation. (s. f.). Ryu Docs.

Recuperado de https://ryu.readthedocs.io/en/latest/getting_started.html

[24] Alejandro García Centeno, Carlos Manuel Rodríguez Vergel, Caridad Anías Calderon, Frank Camilo, "Controladores SDN, elementos para su selección y evaluación", 2014.

[25] Antonio Castillo Jiménez, "SEGURIDAD LOGICA. Gestión de la información: Confidencialidad, integridad, disponibilidad y estanqueidad", Recuperado de https://cronicaseguridad.com/2017/03/31/seguridad-logica-gestion-la-informacion-confidencialidad-integridad-disponibilidad

[26] Zhen Yan, Peng Zhang, Athanasios V. Vasilakos, "A security trust framework for virtualized networks and software defined networking", 2015. Recuperado de https://doi.org/10.1002/sec.1243

[27] Blog INCIBE, "Amenaza vs Vulnerabilidad, ¿sabes en qué se diferencian?", 2017, Recuperado de https://www.incibe.es/protege-tu-empresa/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian

[28] Hogg, S., 2014. Sdn Security Attack Vectors and Sdn Hardening: Securing Sdn Deployments Right from the Start.

Kreutz, D., Ramos, F., Verissimo, P., 2013. Towards secure and dependable software-defined networks. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. ACM, pp. 55–60.

[29] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey" https://doi.org/10.1016/J.JNCA.2020.102595

[30] Andrés Felipe Murillo, Sandra Rueda, Laura Victoria Morales, Álvaro Cardenas, "SDN and NFV Security: Challenges for Integrated Solutions", 2017. Recuperado de https://www.researchgate.net/publication/321008006_SDN_and_NFV_Security_Challenges_for_Integrated_Solutions

[31] Scott Hogg, "SDN Security Attack Vectors and SDN Hardening", 2014, Recuperado de https://www.networkworld.com/article/2840273/sdn-security-attack-vectors-and-sdn-hardening.html

[32] Röpke, C., 2016. Sdn Malware: Problems of Current Protection Systems and Potential Countermeasures. Sicherheit, Sicherheit, Schutz und Zuverlssigkeit. Recuperado de https://dl.gi.de/handle/20.500.12116/884

Scott Hogg, "SDN Security Attack Vectors and SDN Hardening", 2014, Recuperado de https://www.networkworld.com/article/2840273/sdn-security-attack-vectors-and-sdn-hardening.html

[33] Scott Hogg, "SDN Security Attack Vectors and SDN Hardening", 2014, Recuperado de https://www.networkworld.com/article/2840273/sdn-security-attack-vectors-and-sdn-hardening.html

Benton, K., Camp, L.J., Small, C., 2013. Openflow vulnerability assessment. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. Páginas 151–152. https://doi.org/10.1145/2491185.2491222

[34] Yoon, C., Lee, S., Kang, H., Park, T., Shin, S., Yegneswaran, V., Porras, P., Gu, G., 2017a. Flow wars: systemizing the attack surface and defenses in software-defined networks. IEEE/ACM Trans. Netw. 25 (6), 3514–3530, https://doi.org/10.1109/TNET.2017.2748159.

[35] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey". https://doi.org/10.1016/J.JNCA.2020.102595 (Página 7)

[36] Shin, S., Xu, L., Hong, S., Gu, G., 2016. Enhancing network security through software defined networking (sdn). In: 2016 25th International Conference on Computer Communication and Networks. ICCCN, pp. 1–9, https://doi.org/10.1109/ICCCN.2016.7568520.

[37] Dacier, M.C., Knig, H., Cwalinski, R., Kargl, F., Dietrich, S., 2017. Security challenges and opportunities of software-defined networking. IEEE Secur. Priv. 15 (2), Páginas 96–100, https://doi.org/10.1109/MSP.2017.46.

[38] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey". https://doi.org/10.1016/J.JNCA.2020.102595 (Imagen sacada de la página 7)

[39] Matias, J., Garay, J., Toledo, N., Unzilla, J., Jacob, E., 2015. Toward an sdn-enabled nfv architecture. IEEE Commun. Mag. 53 (4), 187–193, https://doi.org/10.1109/MCOM.2015.7081093.

[40] Schehlmann, L., Abt, S., Baier, H., 2014. Blessing or curse? revisiting security aspects of software-defined networking. In: 10th International Conference on Network and Service Management (CNSM) and Workshop, Páginas 382–387, https://doi.org/10.1109/CNSM.2014.7014199.

[41] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey". https://doi.org/10.1016/J.JNCA.2020.102595 (Imagen sacada de la página 8)

[42] Dacier, M.C., Knig, H., Cwalinski, R., Kargl, F., Dietrich, S., 2017. Security challenges and opportunities of software-defined networking. IEEE Secur. Páginas 96–100, https://doi.org/10.1109/MSP.2017.46.

[43] Bernardo, D.V., Chua, B.B., 2015. Introduction and analysis of sdn and nfv security architecture (sn-seca). In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Páginas 796–801, https://doi.org/10.1109/AINA.2015.270.

Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J.J., Lorca, J., Folgueira, J., 2017. Network slicing for 5g with sdn/nfv: concepts, architectures, and challenges. IEEE Commun. Mag. 55 (5), Páginas 80–86, https://doi.org/10.1109/MCOM.2017.1600935.

[44] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey". https://doi.org/10.1016/J.JNCA.2020.102595 (Imagen sacada de la página 9)

[45] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey". https://doi.org/10.1016/J.JNCA.2020.102595 (Imagen sacada de la página 10)

[46] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey". https://doi.org/10.1016/J.JNCA.2020.102595 (Imagen sacada de la página 11)

[47] Sandeep, P., Janakarajan, N., Ankur Chowdhary, Abdullah, A., Dijiang H., 2019. Brew: A securiry policy análisis framework for distributed SDN-based cloud environments. In: 2019 IEEE Latin-American Conference on Communications (LATINCOM), página 1, https://doi.org/10.1109/TDSC.2017.2726066

[48] Le, A., Dinh, P., Le, H., Tran, N.C., 2015. Flexible network-based intrusion detection and prevention system on software-defined networks. In: 2015 International Conference on Advanced Computing and Applications. ACOMP, Páginas 106–111, https://doi.org/10.1109/ACOMP.2015.19

[49] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey". https://doi.org/10.1016/J.JNCA.2020.102595 (Imagen sacada de la página 12)

[50] Ivan Farris, Tarik Taleb, Yacine Khettab, Jaeseung Song, 2019. A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems. https://doi.org/10.1109/COMST.2018.2862350

Manuel C., Luca D., Lucia Seno, Fluvio V., Adriano V., Claudio Z., 2017. Leveraging SDN to improve security in industrial networks. Páginas 1-7, https://doi.org/10.1109/WFCS.2017.7991960

[51] Dong, P., Du, X., Zhang, H., Xu, T., 2016. A detection method for a novel ddos attack against sdn controllers by vast new low-traffic flows. In: 2016 IEEE International Conference on Communications. ICC, Paginas 1–6, https://doi.org/10.1109/ICC.2016.7510992.

[52] Conti, M., Gaspari, F.D., Mancini, L.V., 2018. A novel stealthy attack to gather sdn configuration-information. IEEE Trans. Emerg. Top. Comput. Páginas 1–12, https://doi.org/10.1109/TETC.2018.2806977.

[53] Cox, J.H., Clark, R., Owen, H., 2017. Leveraging sdn and webrtc for rogue access point security. IEEE Trans. Netw. Serv. Manag., Páginas 756–770, https://doi.org/10.1109/TNSM.2017.2710623

[54] Journal of Network and Computer Applications. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, 2018, "Security in SDN: A comprehensive survey". https://doi.org/10.1016/J.JNCA.2020.102595 (Imagen sacada de la página 14)

[55] Gray, N., Zinner, T., Tran-Gia, P., 2017. Enhancing sdn security by device fingerprinting. In: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management. IM, Páginas 879–880, https://doi.org/10.23919/INM.2017.7987393

[56] Alireza, R., Miika Komu, Patrick Salmela, Tuomas Aura, 2016. An SDN-based approach to enhance the end-to-end security: SSL/TLS case study. IEEE/IFIP Network Operations and Management Symposium, Páginas 281–288, https://doi.org/10.1109/NOMS.2016.7502823

[57] Mattos, D.M.F., Duarte, O.C.M.B., 2016. Authflow: authentication and access control mechanism for software defined networking. Páginas 1-16, Recuperado de https://www.gta.ufrj.br/ftp/gta/TechReports/MaDu16.pdf

[58] Zhang, S.-h., Feng, Y.-J., Wen, X.-L., Zhou, S.-B., Lu, D.-L., 2017. Sdnforensics: a comprehensive Forensics framework for software defined network. Recuperado de https://www.scopus.com/record/display.uri?eid=2-s2.0-85040237962&origin=inward

[59] Hyeonseong Jo, Jaehyun Nam, Seungwon Shin, 2018. NOSArmor: Building a Secure Network Operatin System. Páginas 1-15, https://doi.org/10.1155/2018/9178425

[60] Jerry Low, 2020. 24 estadísticas alarmantes de ciberseguridad que necesita saber. https://www.webhostingsecretrevealed.net/es/blog/security/cybersecurity-statistics/

[61] Chowdhary A., Hermant Dixit V., Tiwari N., Kyung S., Huang D., Ahn G., 2017. Science DMZ: SDN based Secured Cloud Testbed. Arizona State University. 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). https://doi.org/10.1109/NFV-SDN.2017.8169868

[62] https://laravel.com/docs/6.x/frontend

[63] Huda Saleh Abdulkarem, Ammar Dawod, 2020. DDoS Attack Detection and Mitigation at SDN Data Plane Layer, Published in: 2020 2nd Global Power, Energy and Communication Conference (GPECOM). https://doi.org/10.1109/GPECOM49333.2020.9247850

[64] Ijaz Ahmad, Suneth Namal, Mika Ylianttila, and Andrei Gurtov, 2015. Security in Software Defined Networks: A survey. IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 17, NO. 4. https://doi.org/10.1109/COMST.2015.2474118

Yifan Liu, Bo Zhao, Pengyuan Zhao, Peiru Fan, Hui Liu. 2019. A survey: Typical security issues of software-defined networking. China Communications (Volume:16, Issue:7) https://doi.org/10.23919/JCC.2019.07.002