PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

ESTUDIO DEL RENDIMIENTO DE SISTEMAS DE DETECCIÓN DE INTRUSOS (IDS) EN REDES SDN

Trabajo de investigación para obtener el grado académico de BACHILLER EN CIENCIAS CON MENCIÓN EN INGENERÍA DE LAS TELECOMUNICACIONES

AUTOR:

Renzo Edú Manrique Huamaní

ASESOR:

César Augusto Santiváñez Guarniz

Lima, diciembre de 2020

Resumen

El presente trabajo tiene como enfoque realizar un estudio sobre el rendimiento de diversas soluciones de sistemas de detección de intrusos (IDS) basados en redes definidas por software (SDN). Debido a que SDN ha cobrado relevancia en los últimos años, es importante abordar el tema de la ciberseguridad en esta y realizar un análisis; ya que, si bien SDN puede agregar mejoras respecto a este ámbito, también puede generar nuevas vulnerabilidades que ponen en riesgo los datos de los usuarios y empresas. Por lo tanto, este estudio tiene como objetivos: analizar el rendimiento de diversas soluciones IDS aplicadas a entornos SDN, hacer una comparación entre ellas basándonos en los objetivos y resultados de las evaluaciones de cada solución propuesta y; finalmente, determinar que soluciones son las más prometedoras.

El estudio evidencia que existen una gran cantidad de soluciones relacionas con el tema de ciberseguridad y SDN; estas tienen como fin el abordar diversos tipos de ataques como DoS, escaneo de puertos, redes *botnet*; así como también, proponer nuevas funcionalidades como es el caso de mejorar el rendimiento de la red mediante el bypass de algunos firewalls o disminuir la carga de tráfico reflejado al IDS. Para el desarrollo del presente trabajo se realiza una investigación bibliográfica en los temas referentes a aplicaciones de IDS enfocados a entornos SDN.

Tabla de contenidos

Índice de Figuras

Índice de Tablas

Introducción

En la actualidad, la ciberseguridad es un tema que afecta a todos, tanto a empresas como usuarios, esto debido a que los atacantes cada vez tienen modos de operación más complejos y cuentan con herramientas y métodos más sofisticados. Por ello, es indispensable contar con herramientas que brinden seguridad a las redes como es el caso de firewalls, IDS, entre otros. Por otro lado, las redes definidas por software (SDN) han cobrado importancia en los últimos años debido a sus diversos beneficios que ofrecen; sin embargo, el tema de la ciberseguridad en éstas aún es ambiguo, ya que puede añadir nuevas vulnerabilidades; así como también, proporcionar nuevas facilidades para mejorar la seguridad en las redes.

Como resultado de esto, diversos estudios se han realizado centrándose en la ciberseguridad de las redes definidas por software. Muchas de estas investigaciones se han apoyado en la aplicación de sistemas de detección de intrusos (IDS) para combatir esta problemática en las redes SDN. Por lo tanto, este trabajo, mediante un estudio bibliográfico, tiene como objetivo realizar un análisis de diversas soluciones IDS aplicados a entornos SDN, de esta forma se podrá determinar el rendimiento de estos y realizar una comparación basándonos en objetivos y resultados.

El primer capítulo tratará la problemática, la justificación y los objetivos detrás de este trabajo. En el segundo capítulo se realizará una recopilación bibliográfica de los diversos tipos de IDS y de algunas soluciones IDS basadas en SDN. En el tercer capítulo se hará un análisis y una comparación de las diversas soluciones IDS que se han estudiado en el presente trabajo. En el cuarto capítulo se dará una reflexión acerca del trabajo realizado y de la aplicabilidad de la solución IntelliFlow. Finalmente, se presentan las conclusiones del trabajo.

1. Problemática de la Ciberseguridad en Redes SDN

El presente capítulo contiene una introducción al concepto de *Software Defined Network* (SDN); además, se realiza una breve descripción sobre el estado actual de la ciberseguridad. También, se presenta la problemática existente en las redes SDN en lo que se refiere al tema de ciberseguridad. Finalmente, se exponen los objetivos y la metodología del presente trabajo.

1.1 Software Defined Network

En los últimos años, se ha visto un incremento en el desarrollo de las tecnologías de la información, a causa de esto las redes actuales son cada vez más complejas debido a los requerimientos que necesitan los servicios de las empresas en la actualidad. Por lo tanto,

dada esta complejidad de las redes tradicionales resulta muy complicado el mantenimiento de estas, ya que se necesita la configuración específica de cada uno de los dispositivos siendo una labor tediosa, propensa a errores y demandante de tiempo para los operadores de la red [1].

A modo de solución, *Software Defined Network* (SDN) emerge como un nuevo paradigma en el mundo de las redes. La propuesta de SDN consiste en separar el plano de control de los equipos de la red, centralizando la inteligencia de la red de manera lógica en un dispositivo llamado controlador. De esta forma, se consigue tener una vista centralizada de la red para un fácil manejo y mejor uso de recursos; además, provee a la red de una mayor agilidad y flexibilidad debido a que la vuelve a está totalmente programable, permitiéndole también volverla altamente escalable en el plano de Datos. Cabe mencionar, que las redes SDN disminuye los gastos de OPEX al ser una red programable; y también reduce el CAPEX al poder reutilizar los dispositivos y no necesitar del uso de dispositivos propietarios, ya que SDN opera sobre código abierto (*open source*) [2].

A continuación, se muestra una imagen con los componentes básicos para una arquitectura SDN:

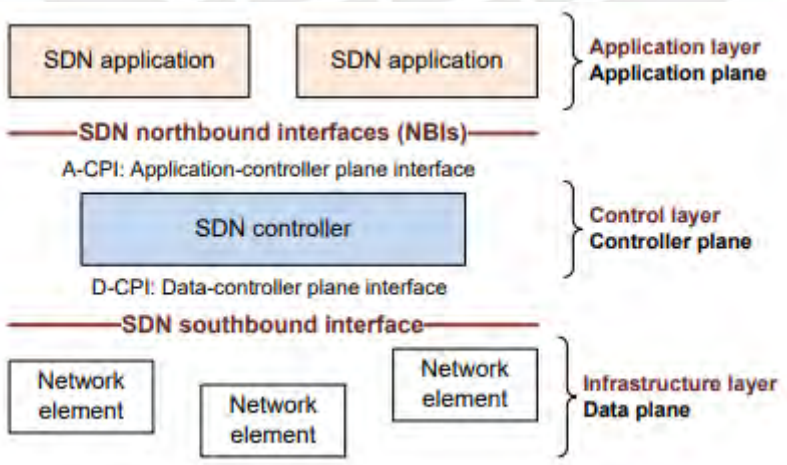

Figura 1. Componentes básicos de la arquitectura SDN

Tomado de Open Network Foundation (2014)

Como se observa en la figura, la arquitectura SDN está compuesta por tres planos. Primero está el plano de aplicación, que es donde se encuentran las aplicaciones SDN las cuales definen el comportamiento de la red. Luego, está el plano de control, donde se encuentra el controlador el cual toma las decisiones del reenvío de paquetes en cada elemento de la red, para luego esas decisiones mandarlas al plano de datos donde se encuentran los elementos de la red como *switches* OpenFlow los cuales se encargan de realizar la acción encomendada por el controlador. Además, se cuenta con la interfaz *southbound,* la cual se encarga de definir el protocolo para la comunicación entre el plano de datos y el plano de control; y la interfaz *northbound*, la cual se encarga de definir la API para el desarrollo de las aplicaciones de la red SDN.

Uno de los principales entornos donde se ha vuelto necesario el uso de SDN es en la nube (Cloud); ya que, para mantener el gran crecimiento de la nube, se deben implementar una gran cantidad de data centers, ubicándolos globalmente para reducir la latencia entre los usuarios globales. Por lo tanto, se está haciendo uso de la infraestructura de la nube para administrar estas instalaciones y conectarlas. Sin embargo, esto no resulta fácilmente implementable ni escalable. En este punto, es donde SDN cumple un rol fundamental ya que permite la escalabilidad y flexibilidad al ritmo que demanda la evolución de la nube, además que ofrece la entrega de información de manera rápida y con bajos costos.

1.1.1 OpenFlow.

OpenFlow, definido como el primer estándar en SDN, es un protocolo no propietario e interfaz que permite la comunicación entre el plano de datos y el plano de control. Además, proporciona un medio para realizar la programación los dispositivos del plano de datos de una red SDN. En la siguiente figura se muestra la arquitectura básica de OpenFlow:

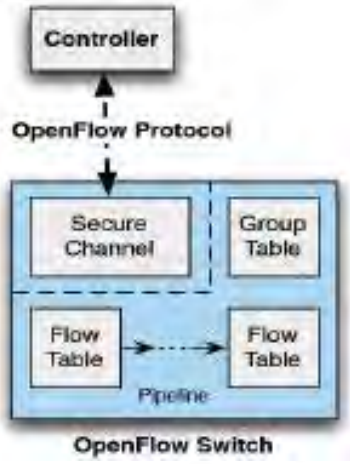

Figura 2. Arquitectura básica de OpenFlow

Tomado de ONF OpenFlow 1.3.0 Switch Specification [3]

Como se puede observar, esta arquitectura está conformado por el controlador, el switch OpenFlow, el canal *switch*-controlador y el protocolo OpenFlow. Estos componentes serán descritos brevemente.

- Controlador

Es el dispositivo de la arquitectura que posee una visión completa de la red y realiza la toma de decisiones. Además, se encarga de la creación de reglas de flujos en los switches.

- *Switch* OpenFlow

Este elemento está conformado por una o más *Flow Tables* y una *Group Tables*; además, se encarga del análisis y reenvío de los paquetes.

- Protocolo OpenFlow

Se basa en un conjunto de mensajes que son enviados desde el controlador hacia el *switch* SDN y un conjunto de mensajes que son enviados en la dirección opuesta a modo

de respuesta; con esto, se permite a el controlador programar al *switch* de acuerdo con los requerimientos de la red [4].

- Canal *Switch*-Controlador

Este canal cumple la función de una interfaz que permite la conexión entre el *switch* OpenFlow y el controlador. Además, a través de este elemento se logra configurar y administrar los *switches*.

1.2 Ciberseguridad en la Actualidad

En la actualidad, la ciberseguridad ha cobrado gran relevancia tanto en el ámbito académico como en la industria. Esto se debe a que año tras año, muchos usuarios maliciosos desarrollen nuevas formas de ataque con el fin de obtener información relevante y ganancias económicas. Uno de los principales objetivos de estos ataques son los entornos en la nube, debido al rápido crecimiento en su uso por parte de empresas y usuarios. Además, uno de los principales usos de las redes SDN es el entorno *cloud* (la nube) como se mencionó anteriormente, por lo tanto, es de interés para este trabajo analizar las amenazas presentes.

A continuación, se mencionan las principales amenazas a las cuales están expuestos los usuarios de la nube diariamente, este top fue realizado por la 'Cloud Security Alliance' (2018) [5]:

1. *Data Breachs*
2. *Misconfiguration and inadequate change control*
3. *Lack of cloud security architecture and strategy*
4. *Insufficient identity, credential, access and key management*
5. *Account hijacking*
6. *Insider threat*

7. *Insecure interfaces and APIs*
8. *Weak control plane*
9. *Metastructure and applistructure failures*
10. *Limited cloud usage visibility*
11. *Abuse and nefarious use of cloud services*

A modo de comprender de forma más clara el alcance y consecuencias que pueden llegar a ocasionar este tipo de ataques se explicarán tres casos relevantes, indicando el tipo de ataque realizado en cada uno:

- MongoDB (2016) - *Data Breach* y *Data Loss*:

Por ejemplo, MongoDB sufrió un ataque a su base de datos, exactamente hubo una vulnerabilidad en el puerto 27017 permitiendo ataques desde redes externas, ya que no se requería una autenticación o control de acceso para acceder al *back-end* de la base de datos de MongoDB; como consecuencia, los usuarios maliciosos pudieron manipular todos los datos (agregar, eliminar, modificar o consultar). En este caso se puede apreciar que hubo incidentes de *'Data Breach'*, ya que se expuso información sensible a todo tipo de usuarios; y *'Data Loss*' ya que la información pudo ser eliminada o manipulada quedando esta inservible [6].

- LinkedIn (2012) - *Data Breach* y *Account Hijacking*:

Otro ejemplo que vale la pena mencionar es el de LinkedIn (2012), en el cual un usuario malicioso robo las credenciales de un empleado de LinkedIn con lo que consiguió el acceso a la base de datos; una vez dentro, se filtró toda la base de datos de usuarios y contraseñas. Entonces, se observa dos principales problemas los cuales son la facilidad para robar las credenciales y que las contraseñas no estaban cifradas mediante funciones Hash. Desde el

punto de vista de ataques se presenta incidentes de '*Data Breach',* ya que se expuso gran cantidad de usuarios con sus respectivas contraseñas; y '*Account Hijacking*' ya que se utilizó la información obtenida de la base de datos para el reúso de contraseñas [7].

- DynDNS (2016) - DDoS:

Como último caso, se menciona el ataque que sufrió DynDNS (2016), en el cual el autor intelectual del ataque utilizó el *malware* Mirai para crear una botnet y realizar un ataque de denegación de servicio (DoS) a Dyn, el cual es un proveedor de DNS. Los clientes de Dyn que no contaban con un proveedor de DNS de respaldo fueron afectados ya que las consultas DNS a sus sitios web no pudieron realizarse. En este incidente se atacó varios dominios de Dyn, donde se registraron tráficos de 1.2 Tbps, con lo cual le resultaba imposible traducir las direcciones IP de manera adecuada, por consiguiente, se mostraban los servicios de Dyn cómo caídos. Cabe mencionar que los ataques DDoS a la nube hace unos años eran bastante constantes; sin embargo, en las últimas investigaciones se ha notado una mejor performance por parte de las organizaciones ante este tipo de ataques [8].

A continuación, se coloca una tabla mencionando los casos analizados en el informe de la CSA en 2018:

Tabla 1. *Casos Analizados con sus respectivas amenazas*

Caso Analizado	**Amenazas detectadas**
LinkedIn	Data Breach, Account hijacking
MongoDB	Data Breach, Data Loss
Dirty Cow	System vulnerability
Zynga	Data Breach

Net Traveler	Data Breach, Data Loss
Yahoo!	Data Breach, Data Loss
Zepto	Data Breach, Data Loss
DynDNS	Denial of Service
Cloudbleed	Denial of Service, Isolation failure

Fuente: "CSA (2018)" [5]

1.3 Descripción de la Problemática

Como se mencionó anteriormente SDN proporciona diversas ventajas; sin embargo, el tema de ciberseguridad en este tipo de redes requiere un análisis, ya que, si bien puede agregar mejoras respecto a este ámbito, también puede generar nuevas vulnerabilidades que ponen en riesgos los datos de los usuarios y empresas.

El controlador al ser el dispositivo que contiene toda la inteligencia lógica de la red puede ocasionar un *bottleneck* (cuello de botella); por ejemplo, se estima que un gran data center compuesto por 2 millones de máquinas virtuales puede generar 200 millones de flujos por minuto, resultando imposible el procesamiento para un controlador [9]. Además, están presentes otros tipos de ataques como es el caso de *man-in-the-middle*, el cual podría consistir en escuchar o modificar la información entre el controlador y los dispositivos del plano de datos; y denegación de servicio (DoS), el cual tiene como objetivo denegar el servicio del controlador.

1.4 Tipos de Ataques Analizados para SDN/OpenFlow

La ciberseguridad en SDN ha resultado ser una promesa debido a la cantidad de soluciones propuestas en los últimos años. Sin embargo, los ataques son cada vez más

complejos y son diseñados de diferentes formas, por lo que se requiere la realización de un análisis sobre las posibles opciones que tienen los atacantes al momento de tener como objetivo una red SDN.

Por lo tanto, se analizarán los posibles ataques que se pueden realizar centrándonos en los posibles elementos objetivos de la red, estos pueden ser el controlador, los *switches* OpenFlow y el canal *switch*-controlador. Además, se plantearán ejemplos para cada tipo de ataque propuesto.

- *Scanning*

  Este tipo de ataque principalmente es el primer paso que realizan los atacantes al momento de elaborar sus ataques, esto se debe a que se puede obtener información relevante sobre el objetivo al atacar. Por ejemplo, en una red SDN se podría obtener la topología, las características de los hosts o los detalles de la comunicación entre los *switches* y el controlador. Cabe mencionar que el *scanning* se puede realizar a nivel de controlador y switches OpenFlow.

- *Spoofing*

  Un ataque del tipo *Spoofing* consiste en hacerse pasar por unos de los elementos de la red; por ejemplo, puede ser el caso de un *switch*, *host* o hasta el mismo controlador. De esta forma se puede manipular los datos que fluyen a través de la red, o incluso en el caso de que sea el controlador el afectado, se podría generar flujos de entrada ocasionando que el atacante tenga control total de la red.

- *Hijacking*

El ataque de tipo *Hijacking*, se puede considerar más peligroso que el *Spoofing* ya que en este caso se tiene control total de un elemento de la red, a diferencia del ataque *Spoofing* donde el atacante se hacía pasar por un elemento de la red. Como se mencionó anteriormente, se puede tener control total de un switch, host o controlador. Si el ataque se realiza al switch, el atacante podría tener el conocimiento de todas las reglas de flujos aplicadas a la red y como se dirige el tráfico de la red. Por otro lado, de darse el caso de un host, este se puede comunicar con otros hosts y obtener información valiosa por parte de estos como pueden ser tokens, contraseñas, etc. Por último, si el ataque es realizado al controlador; es decir, se tiene el control de este, toda la red SDN se vería completamente vulnerada ya que se tendría un control total de la red; por ejemplo, se podría dirigir el tráfico a cualquier destino, modificar información, obtener cualquier información sensible por parte de los hosts, entre otras acciones.

- *Tampering*

Este tipo de ataque consiste en la modificación de información dentro de la red de forma no autorizada. Este ataque podría realizarse mediante la inserción de flujos maliciosos al controlador a través de la falsificación de mensajes de la API *northbound* o los mensajes del *southbound* provenientes de los dispositivos del plano de datos.

- Denegación de servicio (DoS)

Un ataque de DoS consiste en inhabilitar los recursos de una red, para de esta forma hacerla inaccesible a los usuarios finales. En el caso de una red SDN, se podría realizar un envió de varios flujos legítimos hacia un switch OpenFlow, de tal forma que su Flow Table ya no pueda almacenar más flujos provenientes de otros elementos de la red. Por otro lado,

si el ataque se realiza hacia el controlador, se podría dar el caso de un ataque de TCAM *exhaustion*, donde se podría usar una *botnet* en la cual se envían varias solicitudes por segundo al controlador consiguiendo de esa forma saturar la TCAM y reducir en gran manera el funcionamiento completo de la red.

- *Man in the middle* (MITM)

Este tipo de ataque consiste en interceptar la comunicación entre 2 elementos, y a partir de esto obtener, insertar y modificar la información sin que ninguno de los elementos de la comunicación se dé cuenta. En el caso de SDN, un ataque MITM se puede realizar en el canal *switch*-controlador; por ejemplo, se podría mandar información falsificada generando una desconfiguración o mal comportamiento en la red. Otro ejemplo, sería que algún elemento envíe una solicitud y el atacante intercepte la comunicación ocasionando que nunca se obtenga una respuesta generando una parálisis en la red.

- *Repudiation attack*

Este tipo de ataque consiste en que dos elementos fueron participes en una comunicación; sin embargo, luego estos niegan haber participado en esta comunicación. En el caso de SDN, este ataque puede darse a consecuencia de un ataque MITM ya que, al haber interceptado la comunicación, los elementos de la red podrían alegar que nunca fueron participes de la comunicación o transacción que se realizó.

- Otros tipos de ataques

Existen diferentes ataques que pueden generar otras vulnerabilidades a las redes SDN, los cuales pueden atacar tanto a los switches OpenFlow, al controlador y al canal switch-controlador. Por ejemplo, los '*replay attacks'* los cuales consisten en la intercepción de una transmisión de datos a la cual se la retrasa o se la repite. De esta forma, los atacantes pueden

elevarse los privilegios dentro de la red y en conjunto con otros ataques como *spoofing*, *hijacking* o MITM ampliar las vulnerabilidades presentes en la red.

A modo de resumen, se presenta el siguiente cuadro donde se muestra los tipos de ataques, los elementos afectados y una breve definición:

Tabla 2. *Resumen de tipo de ataques*

Tipo de ataque	**Descripción**	**Elementos afectados**
Scanning	Obtención de información relevante de la red	*Switches* OpenFlow, Hosts, Controlador
Spoofing	Hacerse pasar por un elemento de la red	*Switches* OpenFlow, Hosts, Controlador
Hijacking	Tomar el control de un elemento de la red	*Switches* OpenFlow, Hosts, Controlador
Tampering	Modificación de la información de forma no autorizada	*Switches* OpenFlow, Hosts, Controlador
Denegación de Servicio	Inhabilitar un elemento para los usuarios finales	*Switches* OpenFlow, Hosts, Controlador
Man in the middle	Interceptar una comunicación para manipular información	Canal *switch*- controlador
Repudiation attack	Negar la participación en una comunicación	Canal *switch*-controlador

Fuente: [10]

1.5 Objetivos del Trabajo de Investigación

El objetivo de la presente investigación es analizar el rendimiento de diversas soluciones IDS diseñadas para entornos SDN; además, hacer una comparación entre ellas basándonos en los objetivos y resultados de las evaluaciones de cada solución propuesta. Todo con el fin de obtener resultados de que tan eficaces pueden llegar a ser los sistemas de detección de intrusos en las redes definidas por software y determinar que soluciones son las más prometedoras.

1.6 Metodología

En el presente trabajo se va a utilizar una metodología de investigación; es decir, se van a consultar fuentes crediticias como es el caso de documentos proporcionados por revistas o conferencias relacionados con soluciones IDS, ciberseguridad y SDN. Además, los trabajados utilizados como fuentes para la investigación poseen una antigüedad no mayor a 7 años, exceptuando casos donde estos hayan sido referentes en su área.

2. Soluciones IDS en Redes SDN

El presente capítulo contiene la descripción de los conceptos necesarios acerca de sistemas de detección y prevención de intrusos, los cuales son fundamentales para el entendimiento y desarrollo de la investigación del presente trabajo. Además, se presentará una investigación bibliográfica de diferentes soluciones en ciberseguridad haciendo uso de redes SDN e IDPS.

2.1 Concepto de IDPS

IDPS son las siglas de *Intrusion Detection and Prevention System*, estos sistemas son utilizados para la detección y mitigación de diversos ataques dentro las redes mediante el monitoreo del tráfico dentro de estas. Sin embargo, para su correcto entendimiento es necesario

definir previamente los conceptos de IDS (*Intrusion Detection System*) y IPS (*Intrusion Prevention System*).

2.1.1 IDS.

IDS son las siglas de '*Intrusion Detection System'*, el cual es un sistema que se encarga de monitorear la actividad en la red, en busca principalmente de actividad maliciosa o violaciones de políticas de seguridad. Cada actividad sospechosa es reportada e informada al usuario administrador; sin embargo, este sistema solo se limita a detectar y alertar, más no realiza acciones ante actividades maliciosas. El funcionamiento básico de un IDS es comparar el tráfico entrante con firmas de ataques conocidos o comportamientos sospechosos: escaneos de puertos, paquetes malformados, etc. [11]

2.1.2 IPS.

Por otro lado, se tienen los IPS (*Intrusion Prevention System*), el cual es un sistema cuya función es la de detectar y prevenir amenazas identificadas. Los IPS monitorean constantemente la red en busca de actividad maliciosa y capturan información sobre estas. Posteriormente, estos informan sobre las amenazas detectadas al administrador encargado de la red y de esta forma se toman medidas preventivas, como es el caso de cerrar puntos de acceso o configurar reglas en el firewall ante posibles futuros ataques. En resumen, la principal diferencia con el IDS es que los IPS pueden tomar acciones para prevenir el desarrollo de las actividades maliciosas detectadas.[12]

Considerando, el uso conjunto de estos dos sistemas es de donde se surge el término de IDPS el cual son las siglas de '*Intrusion Detection and Prevention System'* el cual brinda al sistema poder operar de forma reactiva y proactiva al momento de filtrar paquetes maliciosos en la red.

2.2 Tipos de IDS

Actualmente existen una gran cantidad de IDPS, los cuales pueden clasificarse según sus metodologías, tecnologías y enfoques de detección.

2.2.1 IDPS según metodologías.

A continuación, se explicarán las diversas metodologías que se utilizan en los IDPS para la detección de actividad maliciosa. Según [11], se dividen en 3 tipos: detección basada en anomalías, detección basada en firmas y *stateful protocol analysis*.

2.2.1.1 Detección basada en anomalías (AD).

Un sistema de detección de intrusos basado en anomalías cumple la función de detectar intrusiones en la red mediante el monitoreo de la actividad en el sistema, con la característica que clasifica esta actividad como normal o anómala. Esta clasificación se realiza en base a reglas, en lugar de usar patrones o firmas, por lo que detecta cualquier comportamiento o actividad maliciosa que produzca un funcionamiento fuera de lo normal en el sistema.

Para poder detectar de forma correcta un ataque, primero es necesario enseñarle al IDS a reconocer la actividad normal del sistema. Un IDS basado en anomalías consta principalmente de dos fases. La primera es la fase de entrenamiento, donde se le construye un perfil de comportamientos normales; y la segunda es la fase de prueba, donde se compara el tráfico en la red con los perfiles creados en la primera fase.

La detección de anomalías se puede realizar de varias formas. Una de estas es mediante técnicas de inteligencia artificial, en el cual se han obtenido buenos resultados en sistemas que usan una red neuronal artificial. Otro método, es mediante el uso de un modelo matemático estricto, en el cual cualquier desviación de este modelo representa una anomalía.

Sin embargo, la principal deficiencia de estos sistemas es que presentan una gran cantidad de falsos positivos y la capacidad de ser engañados por un ataque con las características del perfil definido en la fase de entrenamiento.

2.2.1.2 Detección basada en firmas (SD).

Los sistemas de detección de intrusos basados en firmas se caracterizan por la detección de amenazas mediante la búsqueda de patrones específicos; por ejemplo, secuencias de bytes en el tráfico de red o secuencias de instrucciones usadas por *malwares*. Por lo tanto, firmas se refiere a cualquier patrón ya detectado o conocido, debido a que se ha utilizado anteriormente en distintos tipos de ataques.

Si bien los IDS basados en firmas pueden detectar de forma eficiente ataques conocidos, les resulta también muy difícil detectar nuevos ataques, ya que para estos no existe un patrón disponible que lo identifique.

2.2.1.3 Stateful protocol analysis (SP).

Este tipo de metodología se basa en analizar y comparar perfiles previamente creados para cada protocolo basados en eventos observados. Por lo tanto, se busca identificar desviaciones del estado del protocolo a analizar. Si bien su forma de actuar es similar a los IDS basado en anomalías, la diferencia radica en que este utiliza perfiles universales predeterminados basados en 'definiciones aceptadas de actividades benignas' las cuales son desarrolladas por proveedores y líderes en la industria.

Un ejemplo de este tipo de análisis sería el seguimiento de las solicitudes del protocolo esperando su correspondiente respuesta, cada solicitud debe contener una respuesta predecible, por lo que cualquier respuesta fuera de lo normal serán marcados para ser analizados más a fondo.

Tabla 3. *Comparación entre metodologías de detección*

Metodología	Ventajas	Desventajas
Basado en firmas	- Detecta ataques conocidos mediante patrones o 'strings' - Método simple de detección	- Los ataques nuevos no son detectados a tiempo. - No posee mucho conocimiento de la red
Basado en anomalías	- Detecta eventos que no son acordes con patrones esperados. - Detecta amenazas que no son conocidas	- Nuevas reglas son difíciles de crear - Genera falsos positivos
Stateful Protocol Analysis	- Identifica y distingue secuencias de comandos inesperadas. - Conoce y rastrea los estados del protocolo	- Análisis complejo que involucra una sobrecarga al realizar el seguimiento de muchas sesiones simultaneas. - Incompatibilidad de versiones frente a aplicaciones y sistemas operativos.

Fuente: [11]

2.2.2 IDPS según tecnología.

A continuación, se detallarán las tecnologías que se utilizan en los IDS para la detección de intrusos.

2.2.2.1 Network-based IDS (NIDS).

Un IDPS basado en la red, utiliza sensores para analizar y monitorear la actividad dentro de una red en uno o más segmentos. Los sensores son dispositivos o software cargados a las tarjetas de interfaz de red, las cuales se colocan en modo promiscuo; es decir captura todo el tráfico que circula por la interfaz. Esto permite que el NIC acepte todos los paquetes entrantes

independientemente del destino; por lo que los sensores pueden detectar accesos o ataques no autorizados mediante el análisis de protocolos de red, transporte, aplicación dentro del tráfico de red capturado.

2.2.2.2 Host-based IDS (HIDS).

Un IDPS basado en *host*, hace uso de agentes que residen dentro de los *hosts* en una red. Estos agentes se encargan de analizar los archivos de registro que se crean y almacenan en el *host* y supervisa los procesos en ejecución, el acceso a los archivos y el cambio de las configuraciones. Posteriormente, el agente utiliza la información obtenida del análisis para compararlos con la base de datos de firmas utilizadas por los ataques más conocidos, para de este modo poder detectar posibles amenazas. De esta forma un HIDS puede operar en modo detección y en modo prevención, en el cual toma acciones contra la actividad maliciosa detectada.

Los IDPS pueden usar una combinación de metodologías de detección basadas en firmas y basadas en anomalías para la detección de amenazas.

2.2.2.3 Wireless IDS (WIDS).

Los *wireless* IDS, se encargan de monitorear, analizar e identificar tráfico malicioso en las redes inalámbricas mediante la captura de datos maliciosos provenientes de los *Access Points* (APs).

Un WIDS, compara las direcciones MAC de todos los puntos de acceso inalámbricos en una red con las firmas conocidas de los puntos de accesos previamente autorizados, por lo tanto, si se encuentra alguna diferencia se le informa al administrador. Algunos WIPS de gama alta para evitar la suplantación de MAC utilizan firmas únicas de radiofrecuencia que generan los dispositivos y de esa forma bloquea cualquiera firma desconocida. Principalmente los

intentos de intrusión en los tipos de redes inalámbricas se basan en intentos de obtención de la contraseña, violación del WPS e inundación de paquetes.

2.2.2.4 Network behavior analysis (NBA).

Este tipo de sistema monitorea, analiza y examina el comportamiento de la red en busca de amenazas que generan flujos de tráficos inusuales, como puede ser el caso de ataques de Denegación de servicio (DDoS), ciertos *malware* y violaciones de políticas, entre otras cosas.

Además, permiten el monitoreo y registro de tendencias en el uso de ancho de banda y protocolos. Cabe resaltar, que el análisis del comportamiento de la red es particularmente bueno para la detección de nuevos *malware* o ataques de día cero (*zero day exploits*).

2.2.3 IDPS según enfoque de detección.

Como cada vez la detección de amenazas se hace más compleja se vuelve necesario la división de estas basándonos en enfoques. En [13] proponen clasificar los enfoques de detección en tres categorías: enfoque computacional (*computational approach*), inteligencia artificial y conceptos biológicos (*biological concepts*). Sin embargo, el análisis de estos enfoques resulta bastante complejo debido a todo lo que abarcan estos conceptos. Por lo que, Liao Hung (2013) [11] propone clasificar los enfoques en cinco subclases dando una perspectiva profunda de cada una:

2.2.3.1 Detección basada en estadísticas.

Este enfoque se basa en analizar el tráfico de la red en tiempo real y procesando la información con algoritmos de *machine learning* con el fin de encontrar anomalías en los patrones de tráfico establecidos. Es decir, cada suceso en la red tiene un nivel de anomalía particular, por lo tanto, si una anomalía supera el umbral se lanza una alerta. Sin embargo, este

enfoque se basa únicamente en los patrones de aprendizaje que generan sus propios algoritmos.[14]

2.2.3.2 Detección basada en patrones.

Este tipo de enfoque de detección monitorea los paquetes en la red y hace una comparación con una base de datos de patrones de ataque de conocidas amenazas. Por lo tanto, su funcionamiento: Se captura la data, luego se decodifica esta para poder ser analizada y comparada con la base de datos; finalmente, el resultado del análisis permite brindar la información necesaria para prevenir el ataque y tomar acciones. [15]

2.2.3.3 Detección basada en reglas.

Este enfoque, se basa en la observación de los eventos en el sistema, según los cuales se aplican reglas para la correcta decisión con respecto a patrones de actividades maliciosas detectadas. [16]

2.2.3.4 Detección basada en estados.

En la detección basada en estados, los valores de anomalías son usadas para definir el estado actual de un evento asociado a la red. El funcionamiento detrás de este enfoque era que el detector devolviera un valor muy alto de anomalía cuando se encontrará una transición faltante, de lo contrario se devuelve cero.

2.2.3.5 Detección basada en heurística.

La detección basada en heurística es similar a la detección basada en anomalía, ya que se debe construir un modelo de comportamiento aceptable, al cual el IDS considerará como un

comportamiento adecuado. La heurística funciona de manera similar a un IDS común, ya que aprende con el tiempo que tipos de patrones son comunes en la red.

2.3 Soluciones IDS para Redes SDN

Como se mencionó, la ciberseguridad es un tema que ha cobrado relevancia en los últimos años, a la par del uso de las redes SDN debido a los diversos beneficios que proporciona. Como consecuencia se ha realizado una gran cantidad de investigaciones en el uso de estas redes para mejorar la seguridad informática. Sin embargo, el presente trabajo se va a centrar únicamente en investigaciones que hayan hecho uso de IDS o IPS para proporcionar una mejora en la seguridad de redes SDN. A continuación, se describirán algunos trabajos desarrollados en este ámbito.

2.3.1 BroFlow.

Los autores de BroFlow proponen un sistema capaz de reaccionar contra ataques DDoS en tiempo real, esto lo consiguen mediante el uso de un IDS y una interfaz de programación de aplicaciones OpenFlow. BroFlow es una extensión de la arquitectura Bro ya que se añaden dos módulos, uno para las políticas de seguridad y el otro para la contramedida de mensajes. Una de sus principales ventajas es el uso de aplicaciones reactivas para contrarrestar ataques DDoS; sin embargo, esto no implica que se dé una respuesta efectiva contra este tipo de ataque. Las principales contribuciones de BroFlow son:

- Detección de intrusos mediante simples algoritmos implementados por una flexible y modular arquitectura.
- Respuesta inmediata contra ataques y paquetes maliciosos desde su origen.

- Proporciona un sensor estratégicamente posicionado para la detección de ataques en una red de infraestructura compartida por múltiples usuarios.

Los resultados proporcionados por BroFlow muestra que se garantiza el reenvío de paquetes no maliciosos a la velocidad máxima del enlace y; por otro lado, reduce hasta 10 veces el *delay* máximo de la red causado por un ataque incluso cuando los ataques son provenientes de usuarios legítimos de la red. [17]

2.3.2 SnortFlow.

SnortFlow es uno de los primeros proyectos en juntar un IDS y SDN, el cual tuvo como principal objetivo construir un sistema IPS flexible en entornos de redes virtuales en la nube. Su funcionamiento se basa en el análisis del rendimiento de máquinas virtuales, de modo que ante una actividad anormal se procede a hacer una reconfiguración de la red.[18]

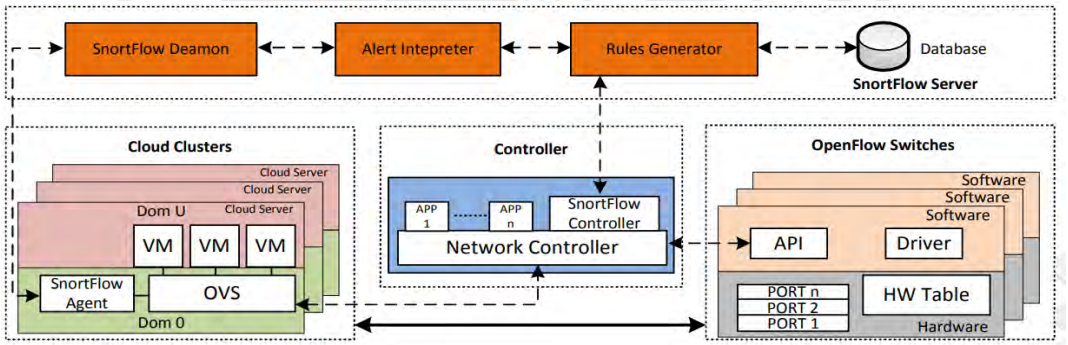

Figura 3: Arquitectura del sistema SnortFlow

Tomado de [18]

La propuesta de este IPS consiste en una arquitectura que soporte una alta flexibilidad y eficiencia mediante la integración de componentes de Snort y OpenFlow. El uso de las funcionalidades de los *switches* OpenFlow permiten que los entornos de red en la nube puedan ser reconfigurados de manera dinámica según el tipo de ataque detectado.

2.3.3 SDNIPS.

SDNIPS es una solución que se encarga de la detección y prevención de intrusos en entornos *cloud*. Su arquitectura se basa en el IDS Snort y Open vSwitch (OVS). Además, las funciones de reconfiguración de red se diseñan e implementan en función del controlador POX para mejorar la flexibilidad de prevención. [19]. En la siguiente figura se muestran los componentes de la arquitectura de la solución:

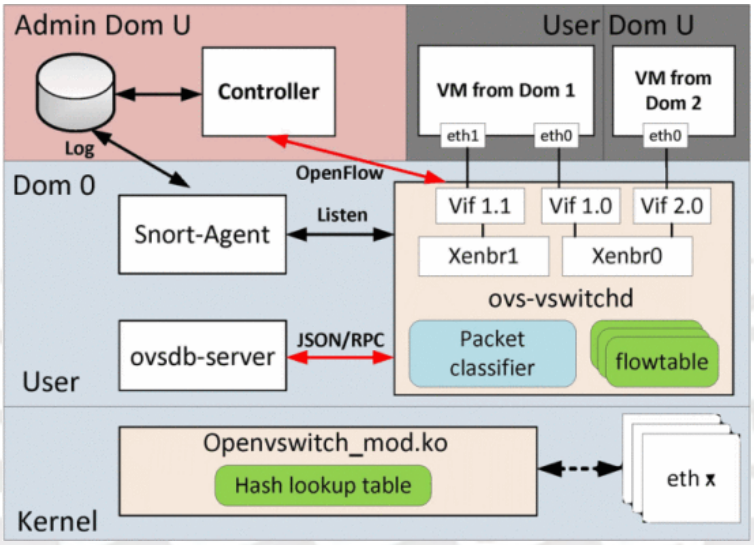

Figura 4: Arquitectura de SDNIPS

Tomado de [19]

2.3.4 IPSFlow.

IPSFlow es una solución de IPS basada en SDN/OpenFlow el cual permite el bloqueo automático de tráfico malicioso. Una de sus principales ventajas es la captura selectiva y distribuido del tráfico en el *switch* para luego realizar un análisis de datos mediante uno o varios IDS. IPSFlow utiliza una aplicación para comunicarse con el controlador para de esa forma poder tomar una decisión ante una amenaza. Sin embargo, el inconveniente es el tiempo de espera de la respuesta del controlador al IDS. [20]

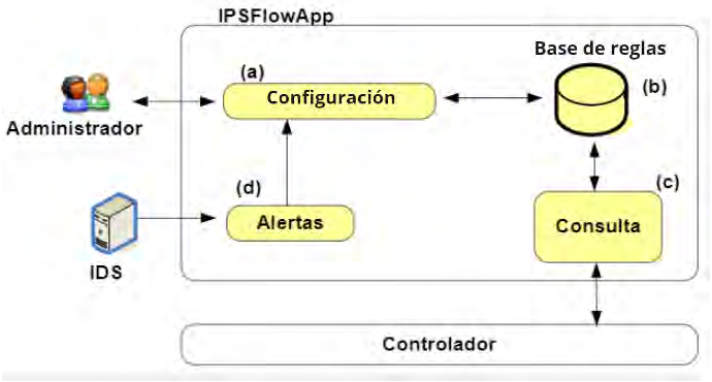

Figura 5: Arquitectura de IPSFlow

Tomado de [20]

La arquitectura está compuesta por el módulo (a) configuración en donde se define las acciones que se deben tomar en los flujos; además, de recibir y gestionar las notificaciones para realizar reconfiguraciones en las reglas. El módulo (b) base de reglas se encarga de almacenar las acciones para cada flujo. El módulo (c) "consulta" se encarga de la comunicación entre el controlador y la aplicación. Por último, el módulo (d) "alertas" se encarga manejar los resultados del análisis de los IDS y notificar al módulo de configuración.

2.3.5 Radware.

Esta es una aplicación comercial SDN comercial, que mejora la seguridad, el rendimiento y la disponibilidad al reenviar de manera óptima el tráfico. Sin embargo, la principal desventaja de este sistema es el alto costo para adquirirlo; además del tiempo que se requiere para aprender a usar esta herramienta adecuadamente. [21]

Figura 6: Radware implementado en la capa de aplicación SDN

Tomado de [21]

En la figura 6, se muestra a Radware como una aplicación SDN la cual brinda una protección DDoS nativo a través de la interfaz *northbound* del controlador SDN. Además, esta solución se basa en el análisis del comportamiento de la red para detectar ataques en tiempo real. [21]

2.3.6 SciPass.

SciPass propone una aplicación OpenFlow para transportar grandes cantidades de datos y enviarlos a sus destinos sin pasar por firewalls y otros dispositivos de red que introducen bajas en el rendimiento. Este sistema mejora la transferencia de datos y reduce la carga en la infraestructura de la red. La principal desventaja de SciPass es que las reglas proactivas son creadas manualmente acorde con los ataques en la red.

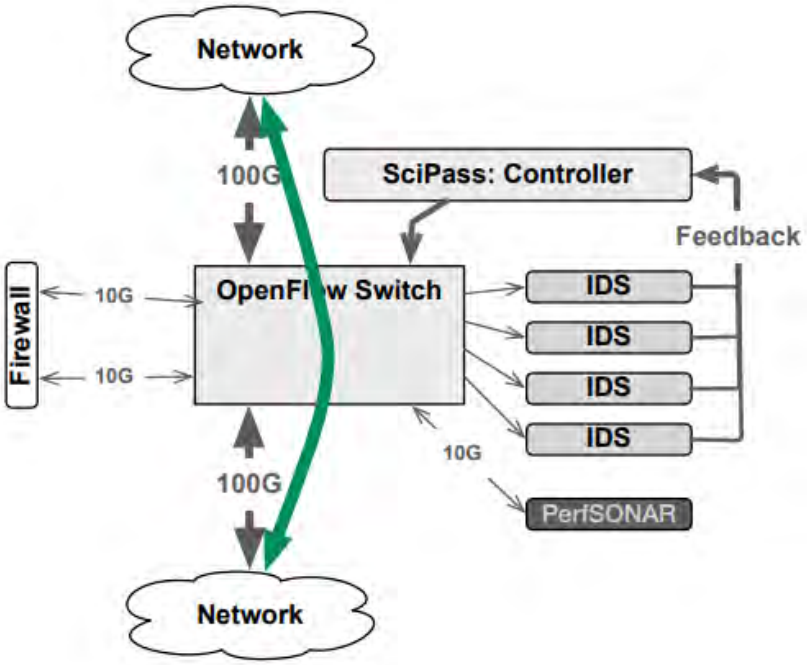

Figura 7: Arquitectura Scipass

Tomado de [22]

Este enfoque que su arquitectura, permite que cuando el sistema determine que los flujos son apropiados para omitir, entonces se agregan reglas OpenFlow al *switch* indicándole que permita a los paquetes asociados a este flujo pasar directamente hasta el otro extremo sin necesidad de atravesar el firewall o los IDS.

2.3.7 IntelliFlow.

IntelliFlow es un novedoso sistema de detección y prevención de intrusos (IDPS) basado en SDN que se apoya en la inteligencia colectiva global de ciberataques libremente disponible (*Cyber Threat Intelligence*, o CTI) y en el sistema IDS de código abierto BroIDS. [23]

Este IDPS ha demostrado ser eficiente en las pruebas que ha sido sometido como es el caso de ataques de fuerza bruta, DoS, escaneo de puerto y mitigación de dominios maliciosos. Sin embargo, IntelliFlow solo ha sido probado en entornos de simulación de tamaño limitado.

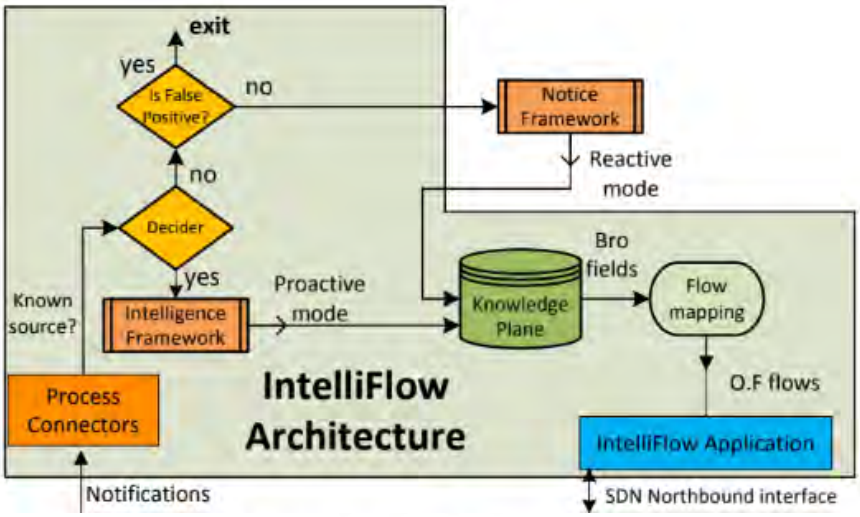

Figura 8: Arquitectura IntelliFlow

Tomado de [23]

En la figura 8 se muestra la aplicación IntelliFlow, la cual se encuentra implementada en la capa de aplicación de SDN. Esta API recibe las notificaciones provenientes del análisis de IDS y de la CTI. Dentro de la aplicación, esta se encarga de determinar si son falsos positivos o no, para posteriormente generar reglas de flujos que permitan mitigar el ataque detectado.

2.3.8 Implementation of SDN-based IDS to protect virtualization server against HTTP DoS attacks.

Los autores de esta publicación proponen una implementación de un IDS basado en SDN para la detección y mitigación de ataques HTTP DDoS en entornos de virtualización. Esta implementación hace uso de un sistema de detección con el método basado en firmas. La arquitectura propuesta es la siguiente.

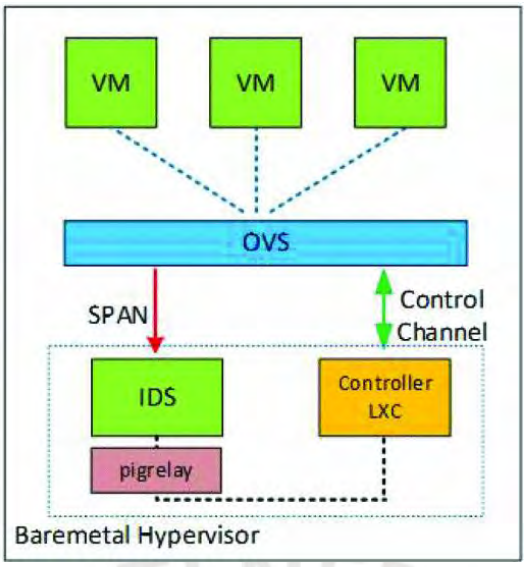

Figura 9: Diseño del sistema para la protección de entornos virtualizados contra ataques DDoS

Tomado de [24]

Su sistema tiene como objetivo que el IDS detecte los ataques DDoS para luego mandar una alerta al controlador SDN. Luego, mediante la agregación de una regla OpenFlow, el controlador mitigará el ataque. [24]

2.3.9 Aplicación para la reducción de tráfico entre switches frente a ataques DoS en SDN.

Esta publicación titulada "*Reduction of traffic between switches and IDS for prevention of DoS attack in SDN*" propone la reducción del tráfico que se envía al IDS a través del *mirroring.* Para esto, se basan en el uso del esquema Hdb, el cual es un historial de incidentes de cada remitente (*host*). La tabla de este esquema está conformada por cuatro campos: el primero es la dirección MAC del host, el segundo es el *switch* al que está conectado, el tercero indica el número del puerto del *switch* conectado y el cuarto indica el porcentaje de incidencia. De esta forma, basándose en el porcentaje de incidencia se puede reducir el tráfico que se envía al IDS cuando se hace uso del *mirroring*.

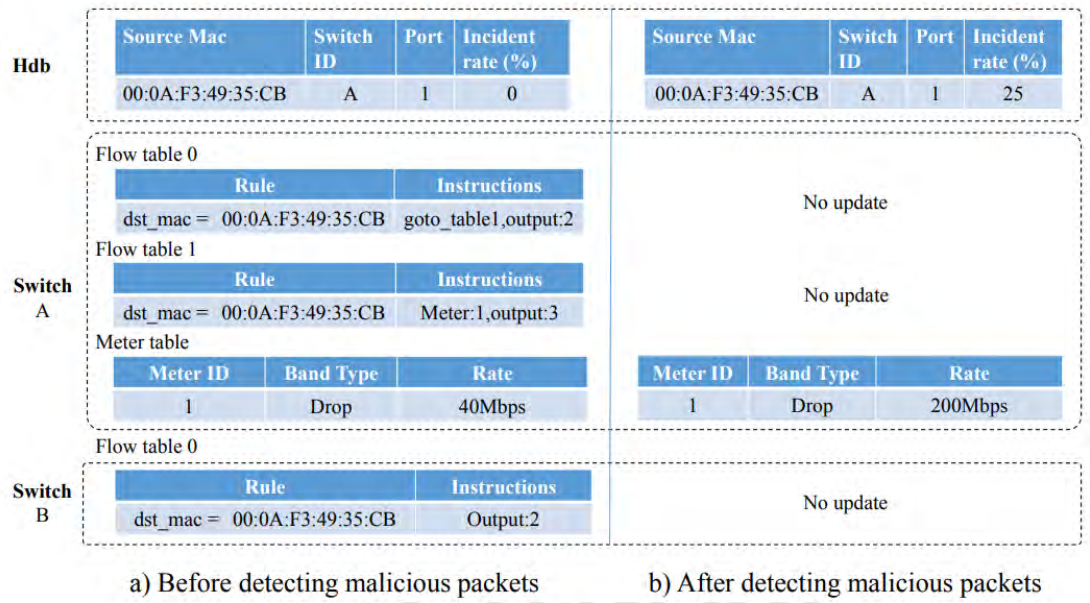

Figura 10: Ejemplo de esquema Hdb de la solución

Tomado de [25]

En general, si el porcentaje de incidencia es 0% se asigna el mínimo tráfico al *port mirror* del IDS. Si el porcentaje es mayor o igual al umbral definido se asigna el máximo tráfico al *mirroring* del IDS. Por último, si el porcentaje es un término medio, este se define en base a la proporción del tráfico. [25]

2.3.10 SDN-Guard.

SDN-Guard es una solución que se enfoca en la protección de las redes SDN frente a ataques DoS, esto se debe a tres principales razones:

- Redirecciona el potencial tráfico malicioso
- Ajusta los tiempos de espera de los flujos.
- Agrega reglas de flujo OpenFlow de manera dinámica

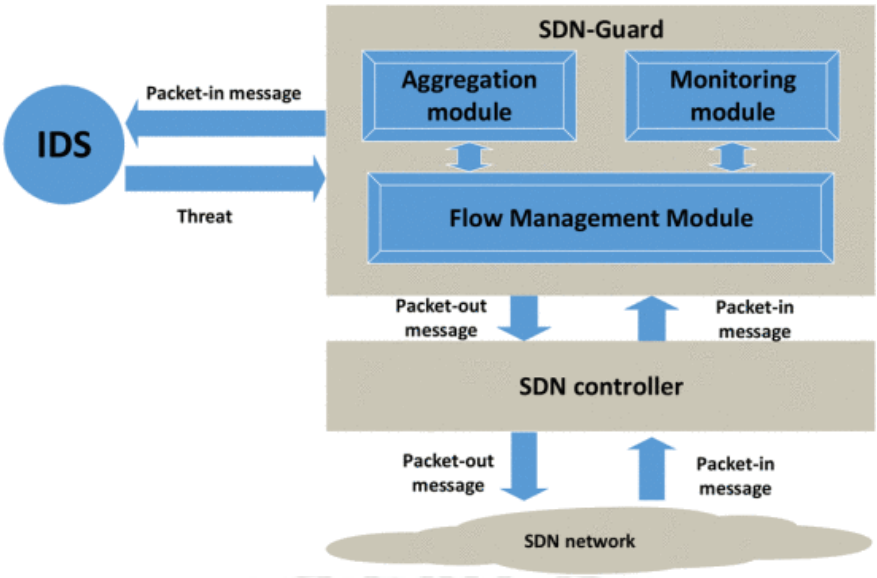

Figura 11: Arquitectura SDN-Guard

Tomado de [26]

En la figura, se muestra los tres módulos con los que cuenta la solución, el Flow Management Module se encarga de seleccionar las rutas de enrutamiento para cada uno de los flujos y decidir el *hard timeout* de sus correspondientes entradas TCAM. Por otro lado, el *Aggregation module* se encarga de agregar los flujos de entrada de trafico de manera que se reduzca la cantidad de entradas usadas en la TCAM. Por último, el *Monitor module* se encarga de recopilar estadísticas de sobre flujos, *switches* y enlaces. [26]

2.3.11 HoneYDSPK.

HoneYDSPK es un diseño de un IDS apoyándose del Cisco One Platform Kit (onePK). Esta solución se encarga de monitorear todo el tráfico de la red e inspecciona cada paquete con el fin de determinar posibles ataques dirigidos a la infraestructura de esta. Además, analiza el comportamiento de la red para detectar patrones de los ciberataques más comunes.

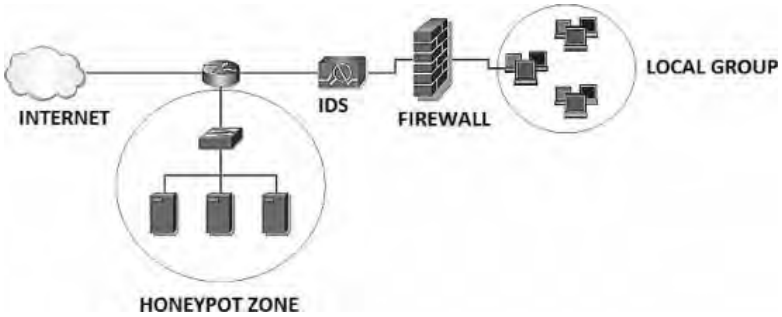

Figura 12: Infraestructura de la red HoneYDSPK

Tomado de [27]

La topología propuesta por esta solución consiste básicamente en dos partes: la *Honeyspot zone* y la *IDS zone*. La primera está compuesta por varios *routers*, los cuales están expuestos y no cuenta con una seguridad por lo cual serán objetivos atractivos para los atacantes de las redes externas. La segunda parte, se encarga de copiar todos los paquetes entrantes y hace una inspección de estos desde la capa 2 hasta la 7 con el fin de encontrar patrones de amenazas. En primera instancia el IDS establece un umbral basándose en el comportamiento normal de la red para lo cual estará analizando una semana aproximadamente este comportamiento. Posterior a esto, todo posible ataque detectado será notificado al controlador para que este agregue nuevas reglas de seguridad al firewall ubicado detrás del IDS [27].

3. Resultados

En el presente capítulo se analizarán los resultados de las evaluaciones realizadas en cada una de las soluciones IDS basadas en SDN propuestas en el capítulo anterior. Además, se hará una comparación basándonos en los resultados obtenidos de la investigación bibliográfica.

3.1 Análisis del Rendimiento de las Soluciones IDS Basadas en SDN

En el capítulo anterior, se explicó de manera breve los objetivos, enfoque y arquitectura de algunas de las diferentes soluciones IDS aplicadas a redes SDN que se han desarrollado en la última década. A continuación, se analizará los resultados que propone cada solución estudiada, cabe mencionar que no se considerarán las soluciones que no tengan una parte experimental como es el caso de IPSFlow; así como también, las soluciones que tengan un costo de pago como es el caso de Radware:

- *BroFlow*

La solución Broflow fue analizado mediante un ataque de DoS mediante la inundación de paquetes SYN, este ataque se realizó mediante un script que permitía la generación de 45 a 55 pps (paquetes por segundo) de manera constante, además se definió como umbral 100 pps. La simulación del ataque se realizó con 3 atacantes mandando paquetes SYN, la alerta del ataque se activaba cuando se superaba 4 veces el umbral. Para el segundo experimento, se hace uso de un ataque de inundación UDP, de manera similar al primer experimento se delimita un umbral y si el tráfico lo sobrepasa 4 veces se lanza una alerta. De esta forma, el controlador recibe la notificación y procede a bloquear el tráfico malicioso en un tiempo no menor a 40 segundos.

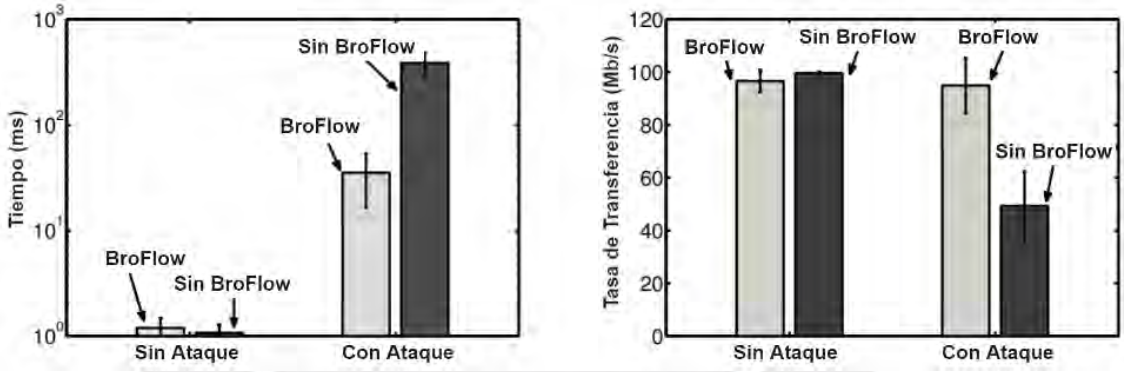

Figura 13: Retraso medio de conmutación de paquetes y comparación de tasa de transferencia

Tomado de [17]

Finalmente, la figura (a) hace una comparación en el retraso medio en la conmutación de paquetes sin el ataque y durante el ataque. Se puede observar que la carga que agrega BroFlow es muy pequeña cuando no se realiza un ataque; por otro lado, al haber un ataque BroFlow disminuye este retraso debido al descarte de paquetes que realiza. También, la figura (b) muestra que BroFlow mantiene la transferencia de datos de manera óptima durante los ataques, mientras que sin BroFlow la transferencia cae hasta un 50%. [17]

- *SnortFlow*

Las pruebas realizadas en esta solución se basan en un servidor en la nube con OVS habilitado en Dom 0. El agente SnortFlow lo instala tanto en el Dom 0 y Dom U. La evaluación se basa en analizar el desempeño del agente en estos 2 escenarios.

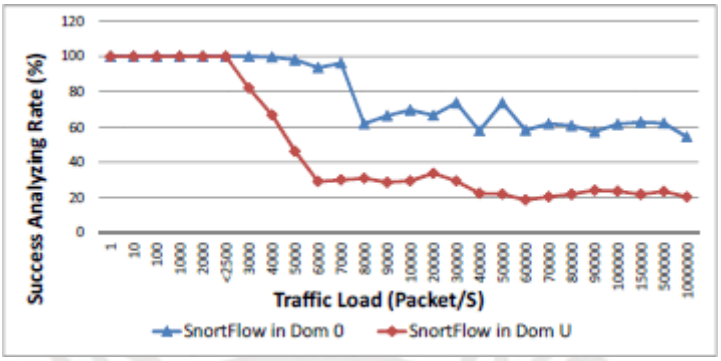

Figura 14: Comparación del rendimiento de la tasa de análisis

Tomado de [18]

En la anterior figura, se muestra el número de paquetes analizados con éxito; es decir, el número de paquetes analizados dividido por el número total de paquetes recibidos. Además, podemos observar que el rendimiento cae a partir de los 2500 pps, esto se debe a que el agente SnortFlow no almacena los paquetes capturados, simplemente los descarta una vez alcanza su máxima capacidad. Por último, se observa que Dom 0 tiene un mejor rendimiento en un 40%.

- *SDNIPS*

Durante las pruebas presentadas, SDNIPS se comparó con una solución IPS tradicional. Uno de los ataques analizados fue del tipo *flooding* "*ICMP flood attack*".

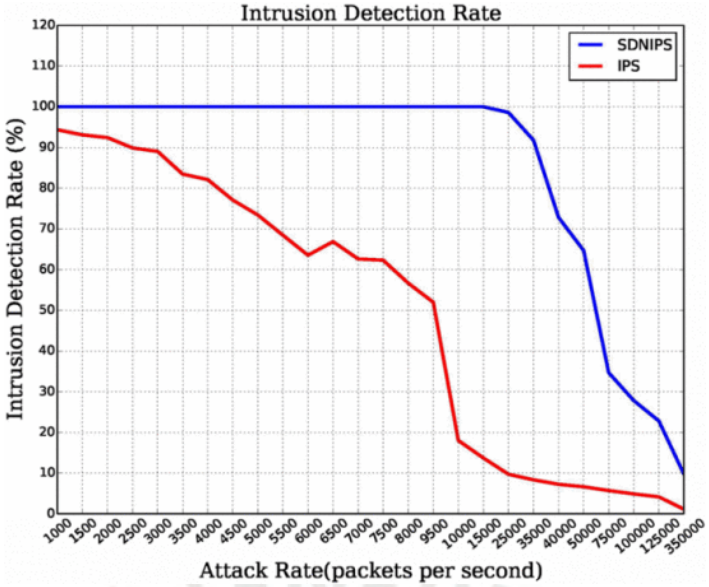

Figura 15: Evaluación de la tasa de detección de intruso

Tomado de [19]

La figura muestra que cuando el ataque ICMP alcanza 15 000 pps, el IPS solo puede generar 13.2% de las alertas del ataque, mientras que el SDNIPS puede generar una alerta por cada uno de los ataques ICMP. Cuando se aumentó la velocidad del ataque ICMP a 30 000 pps se nota un decrecimiento en el rendimiento del SDNIPS; además, cuando se aumentó a 300 000 pps, el agente Snort ya no fue capaz de procesar alertas ya que alcanzo su límite.

- *SciPass*

En la primera prueba realizada a SciPass se buscó evaluar el rendimiento de la transferencia a través del firewall. Para esta prueba se esperó 8 segundos para cambiar la ruta de derivación del firewall, se selecciona 8 segundos ya que asumieron que ese tiempo seria el peor caso posible para realizar una programación de desvío.

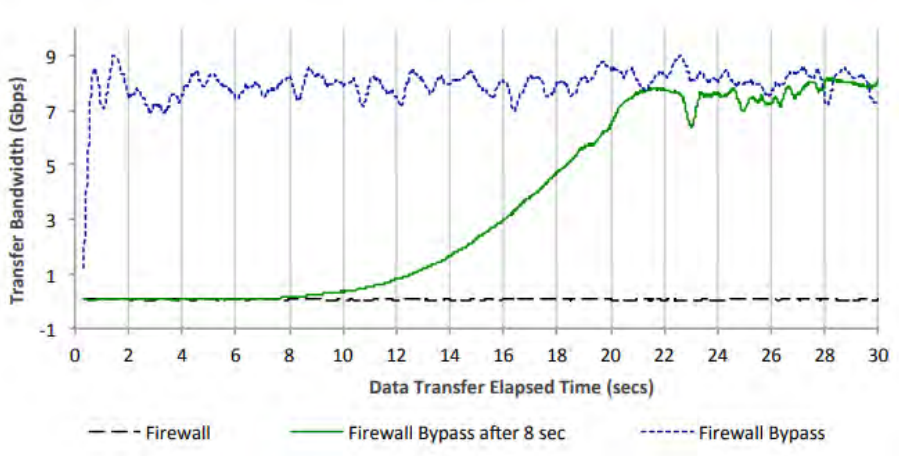

Figura 16: SciPass - Transferencia de datos con Bypass manual luego de 8s

Tomado de [22]

En la figura se muestra como el ancho de banda crece rápidamente a 9Gbs en 1s. Por otro lado, para el caso reactivo se nota que demora aproximadamente 20 segundos en conseguir el máximo ancho de banda.

- *IntelliFlow*

Para las pruebas de IntelliFlow se realizaron 4 pruebas, lo más importante que se analizó en estas pruebas fue el tiempo de reacción:

  - Mitigación de ataque de fuerza bruta: Para el caso de ataques de fuerza bruta, se obtuvo como resultado que para 5 *hosts* maliciosos el tiempo de respuesta al usar Intelliflow es de 0.48s comparado con los 10.77s del método sin usar IntelliFlow. Sin embargo, se observa que cuando los hosts aumentan a más de 10, la solución aumenta su tiempo de reacción haciéndose mayor al tiempo de reacción de la otra metodología.

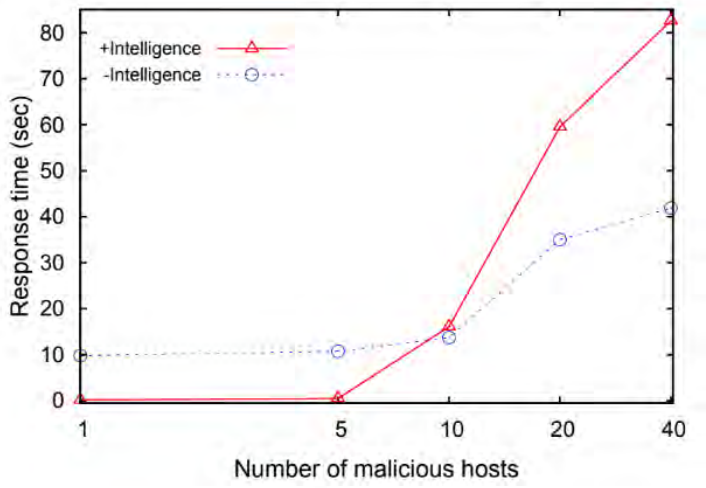

Figura 17: Tiempo de respuesta para diferente cantidad de host

Tomado de [23]

- Mitigación de escaneo: Para esta prueba se determinó que el tiempo para bloquear los ataques de escaneo de puertos es de aproximadamente 1.81 segundos.
- Mitigación de redes de *botnet*: Para el caso de 4000 – 6000 pps, se obtuvo un tiempo de respuesta de 25-30 segundos, mientras que sin hacer uso de IntelliFlow al mismo *rate* se obtiene un tiempo de 45-125 segundos.

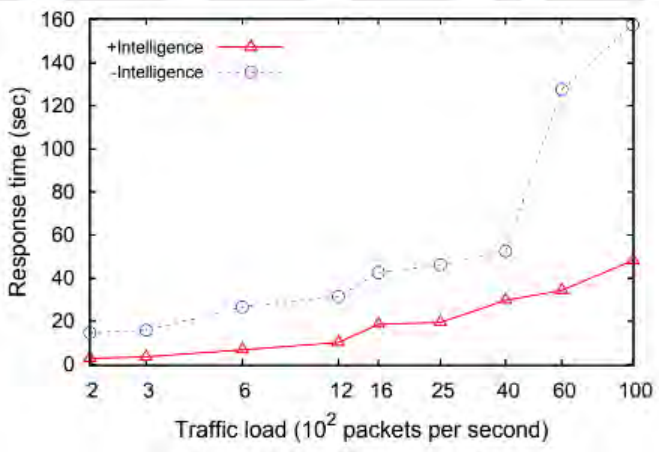

Figura 18: Tiempo de respuesta vs Traffic load

Tomado de [23]

- Mitigación de dominios maliciosos: Para este caso el tiempo de reacción fue de 0.07 segundos, de esta forma se bloquea de forma inteligente los sitios web.

- *Implementation of SDN-based IDS to protect virtualization server against HTTP DoS attacks*

Para la evaluación de esta solución se asumió que los atacantes estaban apuntando a los servicios web usando inundación de solicitudes HTTP.

```
@pve3:~# ovs-ofctl dump-flows vmbr2
kie=0x0, duration=11.147s, table=0, n_packets=3009, n_bytes=231946, priori
00,ip,nw_src=172.16.200.203 actions=drop
kie=0x0, duration=110.838s, table=0, n_packets=929, n_bytes=313076, priori
,in_port=enp6s0,dl_dst=06:a4:33:1f:d1:32 actions=output:veth101i0,output:
002i1
kie=0x0, duration=110.837s, table=0, n_packets=1143, n_bytes=334522, prior
1,in_port=veth101i0,dl_dst=e4:8d:8c:17:bf:be actions=output:enp6s0,output
1002i1
kie=0x0, duration=110.592s, table=0, n_packets=245, n_bytes=137177, priori
,in_port=LOCAL,dl_dst=e4:8d:8c:17:bf:be actions=output:enp6s0,output:tap10
```

Figura 19: Flow table del OVS actualizado al realizar la mitigación del ataque

Tomado de [24]

Como se puede apreciar en la figura 19, al momento que el IDS detecto el ataque de HTTP DoS, este lanzo una alerta al controlador. De esta forma, se puedo actualizar la Flow table del *switch* para agregar una regla que permita la mitigación del ataque.

- *Aplicación para la reducción de tráfico entre switches frente a ataques DoS en SDN*

La evaluación propuesta en esta solución se basa en comparar la reducción del tráfico reflejado a IDS para cada host y el promedio de tráfico reflejado reducido en la red.

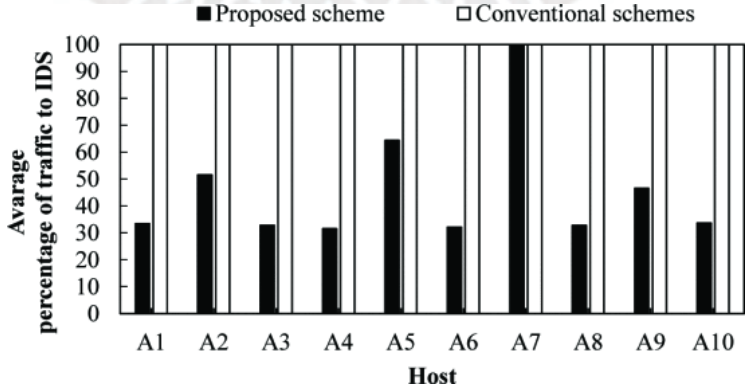

Figura 20: Porcentaje del tráfico reflejado al IDS para cada host

Tomado de [25]

La figura muestra el porcentaje promedio de tráfico reflejado al IDS para cada *host*. Además, se valida que el esquema propuesto por la solución reduce en un 54.1% el tráfico reflejado en comparación con esquemas convencionales.

- *SDN-Guard*

En SDN-Guard el análisis se basó en 4 parámetros principales: *control incoming throughput, the average table size of the switches, the end-to-end throughput and the average Round Trip Time* (RTT). Para analizar estos parámetros se realizó el experimento 2 veces, la primera sin SDN-Guard y la otra con SDN-Guard.

  - Controller Incoming Throughput: Se observa que durante el ataque SDN-Guard logra reducir el *throughput* hasta en un 32%, esto se debe a que agregan tiempos de espera elevados a las reglas de *forwarding* asociadas a los flujos maliciosos.
  - Average Switch Table Size: En la experimentación se obtiene que el número de reglas de flujo en el *switch* disminuyo un 26% a comparación del experimento sin usar SDN-Guard.
  - Throughput from Source to Destination: En este caso, durante el ataque se apreció que hubo una tasa elevada de pérdidas de paquete. Sin embargo, haciendo uso de SDN-Guard solo se pierde un 35% de paquetes a comparación del 40% que se obtiene sin hacer uso de este.
  - Impact on Average Rtt: Finalmente en esta prueba, se muestra que el valor medio de RTT disminuye hasta en un 23% cuando SDN-Guard está activado.

- *HoneYDSPK*

Dentro de las evaluaciones realizadas a HoneYDSPK, se menciona que lograron desviar todos los paquetes entrantes a la aplicación e inspeccionarlos. Además, el controlador pudo modificar los paquetes y enviarlos de vuelta a la dirección IP de origen. Por último, se

demostró que el IDS mandará alertas al controlador apenas detecte un ataque, de esta forma se aplicarán políticas de seguridad que evitarán que los atacantes ingresen a la red.

3.2 Comparación de las Evaluaciones de las Soluciones

A continuación, se procederá a hacer una breve comparación basándonos en los objetivos de cada solución y los resultados obtenidos de sus evaluaciones.

Tabla 4: *Resumen y comparación de las diversas soluciones estudiadas*

Solución IDS basada en SDN	Objetivo	Resultados
BroFlow	Sistema capaz de reaccionar contra ataques DDoS en tiempo real	- La tasa de transferencia haciendo uso de BroFlow no disminuye durante los ataques. - La carga que agrega BroFlow es muy pequeña
SnortFlow	Sistema IPS flexible en entornos de redes virtuales en la nube	- El rendimiento cae a partir de los 2500 pps. - Dom 0 tiene un mejor desempeño en un 40% a comparación de Dom U
SDNIPS	Solución que se encarga de la detección y prevención de intrusos en entornos cloud.	- Muestra un mejor desempeño a comparación de un IPS tradicional - Deja de funcionar al llegar a los 300 000 pps
SciPass	Transportar grandes cantidades de datos y enviarlos a sus destinos sin	- Se consigue el máximo ancho de banda en 1s sin usar

	pasar por firewalls y otros dispositivos de red que introducen bajas en el rendimiento	ningún método reactivo o proactivo. - Modo reactivo: alcanza el máximo ancho de banda en 20s.
IntelliFlow	Solución IDPS basado en SDN que se apoya en la inteligencia colectiva global de ciberataques y en el sistema IDS de código abierto BroIDS	- Mitigación de ataque de fuerza bruta: Para ataques de 10 host o menos se muestra una mejora notable en el uso de la solución - Mitigación de redes botnet: El tiempo de respuesta para 4000-6000 pps es de aprox. 25-30 s - Mitigación de escaneo: Tiempo de respuesta de 1.81s - Mitigación de dominios maliciosos: Tiempo de respuesta de 0.07s
Implementation of SDN-based IDS to protect virtualization server against HTTP DoS attacks	Implementación de un IDS basado en SDN para la detección y mitigación de ataques HTTP DDoS en entornos de virtualización	Actualización de la Flow table del switch para realizar la mitigación del ataque
Aplicación para la reducción de tráfico entre switches frente a ataques DoS en SDN	Reducción del tráfico que se envía al IDS a través del *mirroring*	Reducción en un 54.1% del tráfico reflejado al IDS en comparación con esquemas convencionales
SDN-Guard	Solución que se enfoca en la protección de las redes SDN frente a ataques DoS	- Reducción del throughput hasta en un 32%.

		- Numero de reglas en el switch disminuyo un 26% - Se pierde el 35% de los paquetes durante un ataque - el valor medio de RTT disminuye hasta en un 23%
HoneYDSPK	Diseño de un IDS apoyándose del Cisco One Platfowm Kit. Se encarga de monitorear todo el tráfico de la red	- Desviación de paquetes para su inpección. - Aplicación de políticas de seguridad por parte del controlador para cada ataque detectado

Fuente: Elaboración Propia basado en las fuentes bibliográficas revisadas

4. Aplicabilidad de Solución

En el presente capítulo se tratarán algunas recomendaciones acerca del trabajo de investigación realizado. Además, sobre la aplicabilidad de alguna solución para un futuro trabajo.

4.1 Reflexión

El ámbito de la ciberseguridad va creciendo cada año, de la misma forma el uso de las redes SDN cada vez se hace mayor. Por lo tanto, resulta fundamental el desarrollo de nuevas soluciones que estén a la altura de los nuevos ciberataques que se vienen realizando. También, a lo largo de este trabajo se ha evidenciado que existen diversas soluciones IDS interesantes que buscan aportar una mejora a las redes definidas por software. Este es el caso de mitigación de diversos ataques DoS (HHTP DoS, inundación ICMP, entre otros), mitigación de ataques

de escaneo, mitigación de ataques de redes *botnet*, mitigación de dominios maliciosos, reducción de tráfico reflejado al IDS, protección de entornos *cloud*, etc.

Adicionalmente, es necesario mencionar que estas soluciones no son las únicas en su ámbito, sino que existen una cantidad basta de propuestas, incluso más actuales y ambiciosas que las mencionadas en este trabajo. Sin embargo, la evaluación de tantas soluciones conllevaría a una investigación más extensa. Finalmente, la solución de IntelliFlow es la que se considera la más prometedora por su novedosa arquitectura y modo de operación, considerando también los interesantes resultados mostrados en sus primeras evaluaciones.

4.2 Aplicabilidad y Futuro Trabajo

El presente trabajo da paso a una implementación y evaluación de una de las soluciones IDPS presentadas. De entre todas, la propuesta más interesante es la de IntelliFlow, ya que es un IDPS inteligente y novedoso que hace uso de la CTI y de Bro IDS (actualmente Zeek IDS) para realizar detección y mitigación de ataques en redes SDN. Además de ser una solución de bajo costo en cuanto a su implementación, IntelliFlow solo ha sido probado en entornos de simulación de tamaño limitado y bajo ataques expedicionarios, y por tanto no se ha medido su capacidad para soportar un ataque sofisticado de un adversario con conocimiento de sus especificaciones, como por ejemplo un ataque DoS del tipo “TCAM exhaustion”. También, es necesario mencionar que no se conoce del todo la escalabilidad de esta solución o si generará un aumento de latencia bajo alta carga.

Para esto, se necesitarán realizar pruebas que evalúan diferentes parámetros como es el caso de nivel de tráfico, escalabilidad, capacidad de tráfico, latencia, entre otras. Además, la implementación se apoyará en las máquinas virtuales del testbed hibrido SDN/OpenFlow del GIRA-PUCP. El testbed cuenta con switches SDN físicos y la capacidad de orquestar cientos

de dispositivos virtuales con un tráfico agregado de varios Gbps permitiendo pruebas de stress con máxima fidelidad y realismo.

Conclusiones

- Existen diversas soluciones IDPS basadas en redes SDN que tienen como objetivo reducir la superficie de ciberataques y mejorar la seguridad en la red mediante la mitigación de diversos ataques como: DoS, escaneo de puertos, redes *botnet*, dominios maliciosos, entre otros. Además, proponen funcionalidades como el *bypass* de firewall para mejorar el rendimiento de la red, o disminuir el tráfico reflejado a los IDS para aligerar la carga.
- Los parámetros utilizados en la mayoría de los trabajos analizados para medir la eficiencia de la solución han sido: el tiempo de respuesta ante el ataque, la variación del *throughput* durante un ataque, el impacto en el RTT y porcentaje de paquetes analizados.
- La propuesta más interesante de solución IDPS ha sido IntelliFlow debido a su arquitectura y su modo de operación. Además, los resultados de sus primeras evaluaciones han sido prometedores, ya que demostró una mitigación de ataques DoS, fuerza bruta y escaneo de puertos.

Referencias

[1] M. Jammal, T. Singh, A. Shami, R. Asal, Y. Li.” Software Defined Networking: State of the art and research challenges”, in Computer Networks, 2014. Vol 72, pp.74-98.

[2] G. Cuba, J. Becerra, “Diseño e Implementación de un controlador SDN/OpenFlow para una red de campus académica”, Pontificia Universidad Católica del Perú, Lima, Perú, 2015.

[3] Open Networking Foundation, “OpenFlow Switch Specification”, 2012, pp 26, [Online]. Disponible en: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf

[4] P. Göransson, T. Culver, “The OpenFlow especification”, in Software Defined Networks (Second Edition), 2017, [Online]. Disponible en: https://www.sciencedirect.com/topics/computer-science/openflow-protocol

[5] Cloud Security Alliance. “Top Threat to Cloud Computing: Deep Dive”. 2018. [Online]. Disponible en: https://cloudsecurityalliance.org/press-releases/2019/08/09/csa-releases-new-research-top-threats-to-cloud-computing-egregious-eleven/

[6] P. Paganini, “MongoDB DB containing 93.4 million Mexican voter records open online”, SecurityAffairs.co. Disponible en: https://securityaffairs.co/wordpress/46588/data-breach/mexican-voter-records.html (consultado marzo 9, 2020)

[7] L. Franceschi, “LinkedIn Finally Finished Resetting All the Passwords Leaked in 2012”, Vice.com. Disponible en: https://www.vice.com/en_us/article/53ddqa/linkedin-finally-finished-resetting-all-the-passwords-leaked-in-2012%20 (consultado marzo 9, 2020)

[8] K. York, “Read Dyn’s Statment on the 10/21/2016 DNS DDoS Attack”, Dyn.com. Disponible en: https://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/ (consultado marzo 10, 2020)

[9] A. Tavakoli, M. Casado, T. Koponen, S. Shenker, “Applying NOX to the datacenter.” In: Proceedings of HotNet. ACM Press, New York City, NY, 2009, pp. 1–6.

[10] W. Li, W. Meng, L. Kwok, “A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures”, in Journal of Network and Computer Applications, 2016

[11] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, K.-Y. Tung, “Review: Intrusion detection system: A comprehensive review”. In Journal of Network and Computer Applications, 2013, Vol 36, pp.16-24

[12] SANS. “Understanding IPS and IDS: Using IPS and IDS together for Defense in Depth”. Sans.org. 2004. [Online]. Disponible en: https://www.sans.org/reading-room/whitepapers/detection/understanding-ips-ids-ips-ids-defense-in-depth-1381

[13] P Stavroulakis, M. Stamp, “Handbook of information and communication security”. New York, Springer-Verlag, 2010.

[14] J. Farshchi, “Statistical-Based Intrusion Detection”. Sans.org. 2010. [Online]. Disponible en: https://www.sans.org/security-resources/idfaq/statistic_ids.php

[15] D. D. Kshirsagar, S. S. Sale, D. K. Tagad and G. Khandagale, "Network Intrusion Detection based on attack pattern," 2011 3rd International Conference on Electronics Computer Technology, Kanyakumari, 2011, pp. 283-286, doi: 10.1109/ICECTECH.2011.5942003.

[16] Y. Yang, K. McLaughlin, T. Littler, S. Sezer and H. F. Wang, "Rule-based intrusion detection system for SCADA networks," 2nd IET Renewable Power Generation Conference (RPG 2013), Beijing, 2013, pp. 1-4, doi: 10.1049/cp.2013.1729.

[17] M. A Lopez, U. Figueiredo, A. P. Lobato, O. C. M. B. Duarte, “Broflow: Um sistema eficiente de detecção e prevenção de intrusão em redes definidas por software”. In: CSBC2014. XXXIV Congresso da Sociedade Brasileira de Computação – CSBC 2014. Centro de Convenções Brasil 21, 2014.

[18] T. Xing, D. Huang, L. Xu, C. Chung and P. Khatkar, "SnortFlow: A OpenFlow-Based Intrusion Prevention System in Cloud Environment," 2013 Second GENI Research and Educational Experiment Workshop, Salt Lake City, UT, 2013, pp. 89-92, doi: 10.1109/GREE.2013.25.

[19] T. Xing, Z. Xiong, D. Huang and D. Medhi, "SDNIPS: Enabling Software-Defined Networking based intrusion prevention system in clouds," 10th International Conference on Network and Service Management (CNSM) and Workshop, Rio de Janeiro, 2014, pp. 308-311, doi: 10.1109/CNSM.2014.7014181.

[20] F. Y. Nagahama, F. Farias; E. Aguiar, G. Luciano, L. Granville, E. Cerqueira, A. Antonio. “Ipsflow–uma proposta de sistema de prevençao de intrusao baseado no framework openflow”. In: III WPEIF-SBRC. [S.l.: s.n.], 2012. v. 12, p. 42–47.

[21] Radware. “DefenseFLow: The SDN Application that Programs Networks for DoS Security”. 2015 [online]. Disponible: https://media.arubanetworks.com/sdn-apps/Whitepaper_DefenseFlow.pdf

[22] E. Balas, A. Ragusa, “Scipass: a 100gbps capable secure science dmzusing openflow and bro”, in: Supercomputing 2014 conference (SC14), 2014.

[23] J. Quinto, “IntelliFlow: A Proactive Approach To Add Cyber Threat Intelligence To Software Defined Networking”, M.S. tesis, Dept. Electron. And Comp. Eng., Universidade Estadual de Campinas, Campinas, Brasil, 2015.

[24] S. Usman, I. Winarno and A. Sudarsono, "Implementation of SDN-based IDS to protect Virtualization Server against HTTP DoS attacks," 2020 International Electronics Symposium (IES), Surabaya, Indonesia, 2020, pp. 195-198, doi: 10.1109/IES50839.2020.9231699.

[25] A. M. Quingueni and N. Kitsuwan, "Reduction of traffic between switches and IDS for prevention of DoS attack in SDN," 2019 19th International Symposium on Communications and Information Technologies (ISCIT), Ho Chi Minh City, Vietnam, 2019, pp. 277-281, doi: 10.1109/ISCIT.2019.8905165.

[26] L. Dridi and M. F. Zhani, "SDN-Guard: DoS Attacks Mitigation in SDN Networks," 2016 5th IEEE International Conference on Cloud Networking (Cloudnet), Pisa, 2016, pp. 212-217, doi: 10.1109/CloudNet.2016.9.

[27] R. Trandafir, M. Carabas, R. Rughinis and N. Tapus, "HoneYDSPK: Cisco onePK implementation for anomaly-based IDS and honeypot services," 2016 15th RoEduNet Conference: Networking in Education and Research, Bucharest, 2016, pp. 1-5, doi: 10.1109/RoEduNet.2016.7753221.