SISTEMA DE DETECCIÓN DE ATAQUES INFORMÁTICOS A REDES DE DATOS EMPRESARIALES SOPORTADO EN HONEYPOTS

PROYECTO DE INVESTIGACIÓN
ANTEPROYECTO DE TESIS DE GRADO

Investigadores:

IVÁN DARIO FLOREZ GUERRERO
JESÚS MANUEL QUINTANA MARTÍNEZ

UNIVERSIDAD DE CARTAGENA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
CARTAGENA DE INDIAS, 2018.

SISTEMA DE DETECCIÓN DE ATAQUES INFORMÁTICOS A REDES DE DATOS EMPRESARIALES SOPORTADO EN HONEYPOTS

PROYECTO DE INVESTIGACIÓN
ANTEPROYECTO DE TESIS DE GRADO

Investigadores:
IVÁN DARIO FLÓREZ GUERRERO
JESÚS MANUEL QUINTANA MARTÍNEZ

Director de proyecto:
Ing. JULIO RODRÍGUEZ RIBON

UNIVERSIDAD DE CARTAGENA
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
CARTAGENA DE INDIAS, 2018.

Contenido

TABLA DE ILUSTRACIONES

1. RESUMEN

Un honeypot es una herramienta de “engaño”, diseñada para detectar a un atacante que intenta comprometer los sistemas de información electrónica de una organización. Si se implementa correctamente, un honeypot puede servir como un mecanismo de alerta temprana y un dispositivo avanzado de vigilancia de seguridad. Se puede usar para minimizar los riesgos de ataques a sistemas y redes de TI; sin embargo, actualmente se aplica más como una herramienta con fines investigativos. Este proyecto de grado se propone diseñar y desarrollar un sistema de detección de ataques informáticos a redes de datos empresariales utilizando honeypots para poder analizar, observar y rastrear los ataques de los intrusos o hackers en las redes empresariales, permitiéndole al administrador de TI aportar mejoras a los esquemas de seguridad de la empresa en la cual trabaja.

Este sistema de detección de ataques consulta el principio por la cual fueron creados los honeypots expresado en la frase “CONOCE A TU ENEMIGO”, ya que al identificarlo, estudiarlo, conocer los servicios que más ataca, o los más vulnerables de la red atacada, será posible actuar tomando medidas que permitan mitigar en cierto modo las vulnerabilidades existentes en cualquier entorno de red.

Palabras clave: honeypot, investigación, administrador de TI, red de datos, hacker.

2. ABSTRACT

A honeypot is a cheating tool, designed to lure an attacker into compromising the electronic information systems of an organization. If implemented correctly, a honeypot can serve as an early warning tool and an advanced security surveillance tool. It can be used to minimize the risks of attacks on IT systems and networks but is currently considered a tool for research purposes. The objective of this degree project was to design and develop a system for detecting computer attacks against enterprise data networks using honeypots in order to analyze, observe and track the attacks of intruders or hackers in business networks, allowing the IT administrator to contribute improvements to the security schemes of the company in which he works.

This system of detection of attacks is based on the philosophy by which the honeypots were created, this philosophy is explained with a phrase "KNOW YOUR ENEMY", since, by identifying it, learning from it, knowing the services that most attacks or the most vulnerable services of the attacked network, it will be possible to act taking measures that allow mitigate in some way the existing vulnerabilities in any network environment.

Keywords: honeypot, research, IT administrator, data network, hacker.

3. INTRODUCCIÓN

Las redes de datos en general son un medio de comunicación electrónico muy común en la actualidad, a medida que estas redes y aplicaciones, en particular el internet, van evolucionando, crecen las posibilidades de vulnerabilidad, lo que implica riesgos de ataques, esto se traduce en daños y pérdida para las distintas entidades que la utilizan.

Es una realidad que, en estos últimos años los ataques generados por individuos que buscan perjudicar un sistema electrónico cualquiera, ha aumentado considerablemente. En una encuesta realizada a varias empresas por Business Continuity Institute, la posibilidad de un ataque cibernético fue citada como amenaza de negocio por un 85% de los encuestados, representando la principal amenaza por segundo año consecutivo (Alcantara & Riglietti, 2016). Este factor, unido a las vulnerabilidades existentes en todo tipo de sistemas operativos y aplicaciones, convierte a cualquier organización en una víctima potencial.

Si bien es cierto la seguridad esta finamente ligada a la certeza, es decir, no existe seguridad absoluta, lo que se intenta hacer es minimizar el impacto y el riesgo combinando diferentes herramientas existentes en el medio. Por lo tanto y frente a este panorama, es importante el estudio de nuevas estrategias y técnicas que permitan generar un cierto grado de protección. En la actualidad existen herramientas y mecanismos de defensa que son usados en las redes de computadoras tales como, Firewalls, Sistemas de Detección de Intrusos (Intrusion Detection System IDS), Lista de Control de Accesos (Acces Control List ACL) entre otros. Análisis realizados determinan que el problema con los mecanismos mencionados anteriormente radica en que, muchas veces no están configurados de manera correcta y generan una falsa sensación de seguridad.

Para lograr tener una red segura conviene tomar en cuenta aspectos del entorno como: de qué y de quién se debe proteger, las vulnerabilidades en el hardware o software y los tipos de ataques existentes. Todos estos factores han sido el impulso para que expertos

en el área de seguridad de redes generen nuevas propuestas, dando lugar al surgimiento de una tecnología llamada Honeypot que permite conocer con detalle los ataques y vulnerabilidades que sufren las redes.

En términos básicos en el campo de la seguridad de redes un Honeypot, cuya traducción literal sería "Tarro de miel", se define como un recurso de la red que se encuentra voluntariamente vulnerable para que el intruso pueda examinarla y atacarla, de tal manera que se permita identificar al atacante, los métodos que usa y analizar lo ocurrido para, finalmente, proponer metodologías de prevención o mitigación de las vulnerabilidades encontradas.

A nivel nacional, este término ha captado la atención con fines investigativos de estudiosos del tema, tal y como se evidencia en el artículo *"Honeypot: Ventajas y desventajas como mecanismo para la prevención de intrusos informáticos*" (Eduardo & Daniel, 2013), en el cual se concluye que los honeypots son un complemento global a los sistemas de seguridad ya que proporcionan conocimientos que lo fortalecen y contribuyen a contrarrestar los futuros atacantes que quieran ingresar ilegalmente a las organizaciones

Otra evidencia de la atención captada a nivel nacional por el tema es la monografía llamada "*honeypot, hacia un protocolo de seguridad más eficiente y competitivo"* (Martínez, 2018), en el que se logra identificar, conocer y comprender, la conceptualización de los honeypot, como herramientas usadas dentro del contexto de la seguridad de la información, para la extracción y análisis de los comportamientos de los atacantes en la red; no obstante, dichas investigaciones no han sido sometidas a ensayo y aún se encuentran en la etapa de conocimiento teórico.

En ese orden de ideas, el presente trabajo busca diseñar un sistema para la detección y obtención de información de posibles ataques informáticos a redes de datos empresariales soportado en honeypots, de igual manera ofrecer una herramienta de investigación para que los administradores de la red adopten ideas y soluciones que

podrían ser aplicadas como correctivos que reducirían en cierto modo los riesgos de un ataque, además de generar un aporte a la nueva línea de investigación que se quiere crear en la universidad de Cartagena llamado e-security.

3.1. Descripción del problema

El crecimiento y éxito de las diferentes organizaciones se fundamentan en el manejo de datos e información sensible, la cual es esencial para el desarrollo de las mismas y para los usuarios. De acuerdo a lo expresado por Irwan Se|mbiring dentro de cualquier organización la información sensible fluye día a día, y cada actividad genera más información que puede apoyar las distintas tareas que se llevan a cabo para su buen funcionamiento. (Sembiring, 2016). En la actualidad, esta información se encuentra en riesgo debido a la presencia permanente y el constante aumento de los ciber-ataques; en el año 2016 estos se incrementaron en un 45% según lo reporta la consultora PwC. (Clough & Chaplygin, 2016).

El incremento de estos ataques y su foco en la información de las organizaciones suponen una amenaza, porque si un ataque informático llega a ser exitoso puede dañar significativamente el balance de pérdidas y ganancias, como ejemplo clave la revista Business Continuity Institute público el costo anual de delincuencia cibernética por cada empresa global es de 7,6 millones de dólares, un aumento del 10,4 por ciento comparado con el año 2013 (Bird & Kerr, 2014). Otros impactos de un ataque exitoso son la reputación comercial y la fiabilidad del cliente hacia las organizaciones, cabe recalcar que este último es un pilar del éxito empresarial y para muchas un elemento determinante en su proceso de negocio o actividad (Juan García, 2016).

Ahora bien, existen mecanismos que permiten incrementar los niveles de seguridad de la información como los firewalls, los sistemas de detección de intrusos (IDS), las listas de control de acceso, entre otros; estos elementos forman parte de un todo para la colaboración en aspectos de seguridad de un sistema. Luego, es evidente que las

empresas necesitan aplicar medidas de seguridad necesarias para garantizar un servicio de calidad e integridad de los datos.

Un firewall es una red de sistema de seguridad, ya sea basada en hardware o software, que utiliza las reglas para controlar el tráfico de red entrante y saliente, actúa como una barrera entre una red de confianza y una red no fiable (Qasim Ali, Al-Shaer, & Samak, 2014). Un sistema de detección de intrusos (IDS) es una aplicación de dispositivo o software que alerta a un administrador de seguridad de un incumplimiento , la política de violación o de otro compromiso que pueda afectar negativamente a la tecnología de la información del administrador (TI) de la red (Rajkumar Sethi, Amin, & Schwartz, 2017).

Por tanto, El IDS y el Firewall son herramientas complementarias. Sin embargo, se hace necesario un Sistema que advierta de la presencia de una actividad sospechosa y no autorizada que haya atravesado el filtro del Firewall, un administrador puede creer que su red está protegida cuando realmente no es así. En conclusión, estas medidas son netamente defensivas, por lo cual es necesario que el administrador tenga una visión detallada y objetiva de los tipos de ataque a los que su red es susceptible, para una correcta configuración de las políticas de seguridad.

Otro problema que tienen las organizaciones es la disponibilidad del personal debido a que sus redes deben prestar servicio las 24 horas del día; sin embargo, no es posible que un administrador se encuentre disponible durante tanto tiempo y, posiblemente, en su ausencia se presenten incidentes de seguridad como: accesos no autorizados, manipulación de la información (robo, borrado y alteración), denegación de servicio, escaneo, entre otros. De allí la importancia de implementar un sistema de detección de ataques, que permita registrar información de ataques realizados y mostrarlos de manera simple para que el administrador de TI lo pueda analizar de manera ágil.

Para entidades institucionales como la universidad de Cartagena que manejan una gran cantidad de información privada como la información personal de empleados o estudiantes y no cuentan con personal especializado las 24 horas del día, se hace necesario contar con un método de reporte a bajo coste que brinde información crucial sobre eventos que comprometen la seguridad de estos datos las 24 horas del día y ofrezca soluciones que ayuden a la toma de decisiones para reforzar la seguridad de la información y mitigar las fugas.

Por lo anterior, se evidencia la necesidad de utilizar métodos de defensa actualizadas como son los honeypots. Un honeypot es un método probado que ofrece la oportunidad de adquirir conocimientos sobre las tácticas, técnicas y procedimientos utilizados por los atacantes para comprometer los sistemas sensibles. Sin embargo, dicho método requiere una alta iteración con la red de la organización lo cual hace que su hardware sea de un costo elevado y con una alta dificultad para su implementación en un sistema de seguridad ya establecido (Winn, Rice, Dunlap, Lopez, & Mullins, 2015). En consecuencia, las iniciativas por crear este tipo de software se limitan en la gran mayoría de los casos a una visión académica y abstracta que se lleva a cabo por distintos expertos en seguridad, pero no trascienden a la información relativa a su implementación y funcionamiento.

Por esta razón se requiere diseñar un sistema completo de seguridad que además de utilizar sistemas de señuelos o honeypots alta iteración que simule un servicio utilizando herramientas Opens source, ayude a identificar las amenazas existentes dentro de la red y permita conocer herramientas, motivos, tendencias y tácticas utilizadas por el atacante, con el fin de rediseñar las políticas de seguridad existentes. (Dongxia & Yongbo, 2012).

3.2.Planteamiento del problema

¿Cómo apoyar la labor de los administradores de infraestructura de TI (Tecnologías de la Información), para identificar amenazas a la seguridad del sistema, de tal manera que puedan generar estrategias de protección de forma oportuna?

3.3.Justificación

La humanidad a lo largo de la historia se ha visto influenciada por el desarrollo de las grandes revoluciones científicas, entre ellas la revolución tecnológica, dichos cambios han impactado en la mayoría de organizaciones e instituciones públicas y privadas, porque ayudan a las empresas a potenciar la innovación, incrementar la productividad, disminuir costos entre otras, estas son cada vez más dependientes de la tecnología, por lo cual, un problema que las afecte, puede llegar a comprometer la continuidad de las operaciones (Drnevich & Croson, 2013). Dado que la información ha sido desde siempre un bien invaluable, protegerla es una tarea continua y de vital importancia; a medida que se crean nuevas técnicas para la transmisión de la información, se idean otras que permitan acceder a ella sin autorización.

La seguridad no es sólo una aplicación de un nuevo programa capaz de proteger el sistema, se trata también de minimizar los riesgos asociados al acceso y utilización de determinado sistema de forma no autorizada y en general malintencionada. Es necesario apropiarse del concepto de seguridad para que en cada labor que se desempeñe, se aplique de manera adecuada o se mejore la existente. (Kankanhalli, Teo, C.Y, & Wei, 2013).

Actualmente se cuenta con herramientas complementarias para reforzar la seguridad informática tales como el Firewall, Antivirus, IPS y Antispyware que brindan un mayor soporte a las empresas, pero aun presentan algunas limitaciones como: cada herramienta tiene técnicas de detección limitadas, presentan dudas en cuanto a su efectividad

durante el proceso de mejorar la seguridad, son herramientas netamente defensivas y no se adaptan a los diferentes patrones de ataque de los atacantes.

Por tal motivo, es pertinente la necesidad de utilizar técnicas proactivas de defensa orientadas a la seguridad de redes como son los honeypots. El objetivo de este es mostrar a los atacantes un sistema virtual que aparente el sistema real, con la intención de atraer (como la miel) a los atacantes simulando ser sistemas débiles o con fallas de seguridad, para atraerlos y monitorear todas las actividades que se realizan, de esa manera los ataques se efectuarán sobre ese sistema sin causar ningún daño al sistema real además de obtener información sobre las actividades ilícitas que realiza el atacante.

El presente proyecto propone una nueva alternativa para la detección de intrusos basado en honeypots, dicho sistema brindará información de las actividades de los intrusos que ingresen a la red, lo cual facilitará la toma de medidas preventivas sobre futuros atacantes, además de poder actualizar las políticas de seguridad para evitar replicas o ataques con patrones similares en las organizaciones de nuestra comunidad, facilitando de forma eficiente y eficaz la gestión de la seguridad de la información que se transporta tanto interna como de manera externa por una red de datos.

El desarrollo de este sistema conllevara una cantidad de beneficios para las organizaciones y entre esos se pueden destacar los siguientes:

Beneficios económicos.

Esta solución es viable económicamente porque utiliza diversos tipos de software libre, como parte de un todo, algunos de ellos cobijados bajo licencias GPL, lo cual implica que se encuentran a total disposición sin costo alguno. Además, la información guardada en los honeypot para atraer a los atacantes no afecta a las organizaciones, por tanto, no genera pérdidas económicas para estas.

Beneficios sociales.

El presente proyecto aporta beneficio social porque en primer lugar contiene posibles ataques verdaderamente peligrosos; en segundo, entretiene y desgasta al atacante haciéndole perder el tiempo; y en tercero, analizar los reportes de los ataques para detectar posibles nuevas formas de ataque que se estén llevando a cabo en el sector. Todos los beneficios anteriormente mencionados se traducen ahorro de esfuerzos y preocupaciones al administrador de TI.

Beneficios académicos.

Se justifica en beneficios académicos debido a que permitirá conocer los resultados del desarrollo de un sistema de detección basado en honeypots, para poder examinar ataques informáticos, nuevas herramientas y tipos de ataques a intrusos. Además de incentivar a las empresas a dedicar sus recursos a estudiar nuevas tendencias de ciberseguridad empresarial, analizar las últimas estrategias de cibercrimen y, en definitiva, poder **proteger la seguridad informática de su compañía** de una manera mucho más efectiva, evitando los problemas antes siquiera de que lleguen.

4. MARCO DE REFERENCIA

A lo largo de la investigación, es factible la apropiación de algunos conocimientos que sustentan el manejo, uso y configuración de las distintas herramientas que se utilizan con el objeto de implementar nuestro sistema de forma adecuada. El contexto teórico general esta abordado por temáticas relacionadas con: sistemas de detección de intrusos, honeypots, cortafuegos (firewall) y plataformas web.

4.1. Marco teórico

4.1.1. Antecedentes nacionales e internacionales

Los antecedentes reflejan los avances y el estado actual del conocimiento en un área determinada y sirven de modelo o ejemplo para futuras investigaciones. Para la realización de esta tesis de pregrado se rastrearon investigaciones locales y nacionales, en las investigación realizada destacamos el artículo *"Honeypot: Ventajas y desventajas como mecanismo para la prevención de intrusos informáticos*" (Eduardo & Daniel, 2013). Este artículo se da a conocer el concepto de honeypot, definición, clasificación, ventajas y desventajas, así como de la ubicación de los mismos dentro de una red. También se aborda el tema sobre el impacto que han tenido y los diferentes campos donde se pueden aplicar hoy en día.

La otra investigación destacada es "*Honeypot, hacia un protocolo de seguridad más eficiente y competitivo*" (Martínez, 2018) en el cual además de hondar en los conceptos mencionados en el artículo anterior, realiza una prueba de campo utilizando el honeypot "Kippo", para analizar la viabilidad abordando leyes que rigen el estado colombiano.

En el ámbito internacional nos encontramos con una gran variedad de investigaciones relacionadas con los honeypots, tales como:

- Diseño del prototipo de una honeypot virtual que permite mejorar el esquema de seguridad en las redes de la carrera de ingeniería en sistemas computacionales y networking de la universidad de Guayaquil.
- Captura y análisis de ataques informáticos que sufren las redes de datos de la espol, implantando una honeynet con miras a mejorar la seguridad informática de en redes de datos del ecuador.
- Detección de Intrusiones en redes de datos con captura distribuida y procesamiento estadístico.

4.1.2. Orígenes

Este estudio acerca de la tecnología Honeypot, se basa fundamentalmente en conceptos que tienen más de 20 años de historia, aunque su aplicación formal es más reciente y ofrece poca documentación de sus orígenes.

Históricamente las primeras referencias a un sistema de monitorización de intrusos aparecen ya en la bibliografía sobre los años 90, de la mano del astrónomo y escritor Estadounidense Clifford Stoll (Stoll, 1989); en esta obra narra su experiencia personal en la búsqueda y captura de un hacker, el cual irrumpió la seguridad de una computadora en el Laboratorio Nacional Lawrence Berkeley en Estados Unidos. Sin embargo los líderes en la investigación y desarrollo del concepto de Honeypot se agrupan en el Honeynet Project que junto a una serie de publicaciones denominadas "Know Your Enemy" (Project, 2001), fueron las que hicieron crecer el valor de la tecnología Honeypot.

El responsable de esta idea es Lance Spitzner, un consultor y analista informático experto en temas de seguridad. Construyó a comienzos del año 1999 una red de seis ordenadores en su casa diseñada para estudiar el comportamiento y formas de actuación de los hackers. Fue de los primeros en adoptar la idea, hoy es uno de los mayores expertos en Honeypots, precursor del proyecto Honeynet, en marcha desde 1999 y autor del libro "Honeypots: Tracking Hackers". Fue entonces cuando octubre del año 1999

cuando se formó un grupo de personas que buscaban aprender más acerca de ataques, amenazas y vulnerabilidades.

Este grupo estaba compuesto en sus comienzos por: Martin Roesch (desarrollador del sistema de detección de intrusos llamado Snort), Cris Brenton, J.D Glazer, Ed Skoudis y Lance Spitzner (autor de: The Honeynet Project), autodenominados "Wargames mail list", trabajaban en la construcción de computadoras que eran utilizadas para vulnerarse unas con otras, desarrollando de esta manera habilidades tanto para el ataque como metodologías de análisis para comprender como habían sido atacadas. Finalmente, este grupo fue creciendo y se convirtió en lo que se conoce hoy como "The Honeynet Project".

En junio de 2000 y por espacio de tres semanas, el Honeypot del proyecto fue atacado y comprometido por un famoso grupo de hackers, lo que permitió el estudio del comportamiento de este grupo en "real" así como demostrar la viabilidad y utilidad de esta nueva herramienta de seguridad. Este conocido incidente catapultó mediáticamente el concepto de Honeypot como la última tendencia en seguridad de redes convirtiendo su libro en un best-seller de lectura obligatoria para todos los profesionales de la seguridad (Project, 2001).

A inicios de 2001 se convirtió en una organización sin ánimo de lucro dedicada al estudio de los hackers; esta organización actualmente está compuesta por más de 30 miembros permanentes, además se han fundado capítulos (Chapters) o grupos de desarrolladores de diversos países, que colaboran en conjunto para evolucionar esta tecnología e ir creando nuevas soluciones basadas en Honeypots, y también proporcionan ayuda a los interesados en esta herramienta.

Cerca del año 2002 se empieza hacer uso de Honeypots para la captura de información con el objetivo de estudiar la actividad de gusanos informáticos y muchas organizaciones adoptaron los Honeypots como un medio de investigación y detección de ataques informáticos. Para el desarrollo de la presente investigación motivo de esta tesis,

se planea adoptar la guía de uno de los miembros del Honeynet Proyect pertenecientes al Spanish Chapter, cuyo líder y fundador es el Ingeniero en Telecomunicaciones y Telemática Diego González Gómez.

4.2.Definición de honeypot

El concepto de Honeypot no fue extraído o inventado de la nada, sino que es fruto de la realización de varios estudios en el campo de la seguridad de redes. Se define a una Honeypot como:
Un recurso de red que simula ser un objetivo real, pero destinado a ser atacado, de tal forma que un intruso pueda ingresar, examinarla y comprometerla. Las Honeypot no tienen en ningún caso la finalidad de resolver o arreglar fallos de seguridad en nuestra red. Son los encargados de proporcionarnos información valiosa sobre los posibles atacantes en potencia a nuestra red antes de que comprometan sistemas reales. (Spitzner, 2002)

Como se hace mención en la definición anterior, el valor real de la Honeypot reside en ser atacada, examinada y vulnerada. Dicha condición permite:

- Obtener pruebas del ataque al sistema.
- Conocer nuevas vulnerabilidades.
- Capturar nuevos tipos de ataques.
- Descubrir riesgos de sistemas.
- Despistar al atacante sobre los servidores en producción.
- Implementar mejoras a la seguridad global de la red.

Este nuevo enfoque a la seguridad de redes rompe muchos paradigmas clásicos que se daban como axiomas en la seguridad informática “clásica”. Debido a que en lugar de evitar a toda costa el ataque, se incita al atacante a ingresar a la red, presentándole un nivel de complejidad adecuado para atraerlo, no exagerado para no desalentarlo, y ofrecerle una serie de archivos y programas atractivos para los intrusos, mientras el personal de seguridad del sistema monitorea, registra y observa todas las acciones para

aprender todas las herramientas de ataque empleadas por el intruso, y así usar lo aprendido para mejorar la seguridad y adicionalmente lograr desviar la atención del sistema real.

La Honeypot es un sistema muy controlado, considerado como un sistema trampa y ha sido diseñada para interactuar con el atacante, imitando el comportamiento de un sistema que pueda ser de interés para el intruso, donde todo el tráfico entrante y saliente es detectado y capturado. De este modo se puede llevar a cabo un examen en profundidad del atacante, durante y después del ataque a la Honeypot. La información recogida por un Honeypot es crucial para la detección y protección de las amenazas a las que nos enfrentamos diariamente.

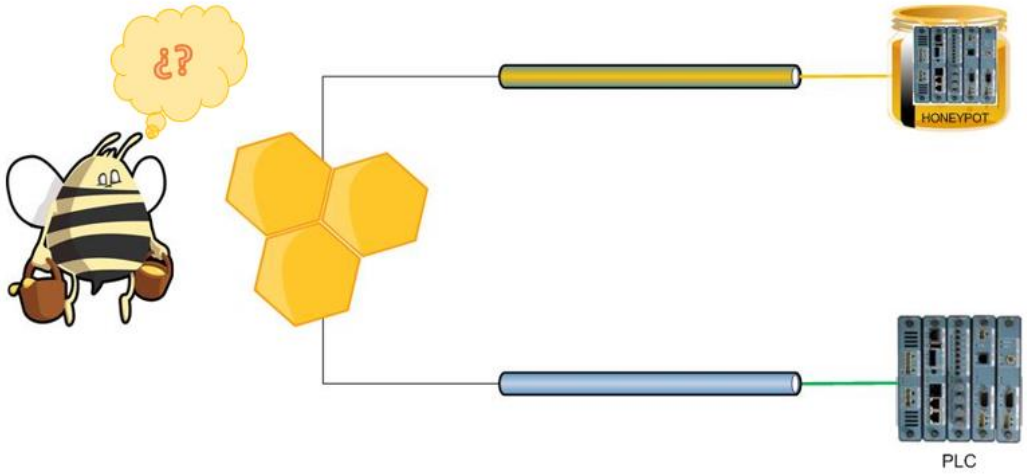


Ilustración 1. Ejemplo grafico de un honeypot (INCIBE, 2017)

Cabe aclarar que un Honeypot no es un sistema de detección de intrusiones, aunque puede utilizarse como una herramienta de apoyo, y puede ayudar a mejorar los métodos de detección y aportar al conocimiento de nuevos patrones de ataque. En resumen, es un sistema diseñado para engañar a los intrusos, poder estudiar sus actividades y así aprender de sus métodos. Se basa en la idea de "conocer al enemigo" para poder combatirlo (Project, 2001).

4.2.1. Ventajas de los Honeypot

Las Honeypot son un concepto increíblemente simple, los cuales ofrecen una fortaleza muy poderosa. Podemos observar sus ventajas en los siguientes puntos:

- Genera un volumen pequeño de datos, al contrario que los sistemas clásicos de seguridad que generan cientos de megas de ficheros de logs con todo tipo de información, las Honeypots generan muy pocos datos y de altísimo valor, recogen pequeñas cantidades de información sólo cuando el atacante interactúa con ellas.

- Necesita unos recursos mínimos, a diferencia de otros sistemas de seguridad, las necesidades de un Honeypot son mínimas. No consume ni ancho de banda ni memoria o CPU extra. No necesita complejas arquitecturas o varios ordenadores centralizados, cualquier ordenador conectado a la red puede realizar este trabajo.

- Inexistencia de Falsas alarmas, estas herramientas de seguridad sólo deben recibir únicamente actividades sospechosas. Esto reduce significativamente el 77 número de falsos positivos (alarma cuando no existe ataque) y falsos negativos (omisión de alarma cuando hay verdaderamente un ataque).

- Encriptación en IPv6, Uno de los problemas que presentan algunas herramientas de seguridad es que no soportan el protocolo IPv6, sucesor del actual IPv4 ampliamente utilizado en Internet. Utilizar IPv6 a través de túneles sobre IPv4, como hacen algunos atacantes, puede imposibilitar la detección por parte de muchos sistemas de detección. No obstante, las Honeypot registran toda la actividad ocurrida, por lo que se pueden identificar este tipo de ataques.

- Reutilización, la mayoría de los productos de seguridad necesitan mantener al día sus mecanismos de detección y defensa para mantener su efectividad. Si no se renuevan, dejan de ser útiles. Pero las Honeypot, debido a su propia naturaleza, siempre serán de ayuda independientemente del tiempo que pase, siempre habrá

atacantes dispuestos a comprometer estos sistemas de una u otra forma, mostrando el nivel de actividad de este sector y los métodos que utilizan.

- Información, Pueden recopilar información de manera detallada a diferencia de otras herramientas de análisis de incidentes de seguridad.

- Distraer al atacante, detener o entretenerlo con sistemas en los que no puede causar daño, protegiendo así las redes y sistemas en producción.

- Universalidad, este tipo de sistemas sirven tanto para posibles atacantes internos como externos. De esta forma, obviamente, se ha de evitar poner a las máquinas nombres como "Honeypot" o "attack-me". Su objetivo es pasar desapercibidas en una red como una máquina más.

- Simplicidad, uno de los puntos más importantes a favor de las Honeypot es su sencillez. No utilizan complicados algoritmos de análisis, ni rebuscados métodos para registrar la actividad de los intrusos. Por el contrario, sólo hay que instalarlos y esperar. Algunos sistemas trampa de desarrollo pueden poseer mayor nivel de complejidad, pero no comparable a otros enfoques (Gonzalez, 2003).

4.2.2. Desventajas de las Honeypot

Como cualquier otra tecnología, las Honeypot también tienen debilidades inherentes a su diseño y funcionamiento. Esto se debe a que éstos no reemplazan a las tecnologías actuales, sino que trabajan con las tecnologías existentes (Kaur, Malhotra, & Singh, 2014) . Entre las desventajas que tienen las honeypot se encuentran:

- Son elementos totalmente pasivos, de esta forma, si no reciben ningún ataque no sirven de nada.

- Fuente potencial de riesgo, debido a la atracción de atacantes, se debe tener cuidado en la configuración, y convertirlo en un entorno cerrado y controlado 79 (jailed

envinronment), para evitar que se utilice como fuente de ataque a otras redes e incluso a la propia.

- No resuelven fallos de seguridad, las Honeypot son herramientas empleadas para el análisis de ataques, son usadas para la búsqueda de mejoras y soluciones a los posibles problemas que presenten los métodos de seguridad implementados en una red.

- No detienen a un atacante, al contrario, lo atraen con el fin de permitir estudiar sus técnicas de ataque para un posterior análisis.

- Fingerprint, es la identificación local o remota de un sistema o servicio. Es posible que la deficiente implementación de la Honeypot la delate, haciéndola reconocible ante un intruso, lo que la volverá inútil, o se lo puede utilizar para desviar la atención de la administración de seguridad.

- Visión Limitada, solo pueden rastrear y capturar actividad destinada a interactuar directamente con ellas. No capturan información relacionada a ataques destinados hacia sistemas vecinos, a menos que el atacante o la amenaza interactúe con la Honeypot al mismo tiempo.

4.2.3. Clasificación de las Honeypot

Las Honeypot son una tecnología nueva con enorme potencial para la comunidad informática, la taxonomía de los diferentes tipos de Honeypot depende de la bibliografía consultada puesto que, como todo nuevo concepto, su estandarización es compleja y aún no ha sido universalmente aceptada.

El autor Lance Spitzner (fundador del Honeypot Proyect) y Martin Roesch (creador de Snort) afirman que las Honeypot se pueden dividir de acuerdo a tres aspectos

fundamentales: según su Ambiente de Uso, según su Nivel de Interacción y según sus recursos.

4.2.3.1. Según Ambiente de Uso

- **Honeypot de producción (Production Honeypot System):** Llamados así por su ubicación junto a la red de producción en una organización. Su principal objetivo es el de mitigar el riesgo de un ataque informático a la red productiva de una institución o empresa. De esta forma, una Honeypot de producción simula diferentes servicios con el único fin de ser atacada y obtener información sobre las técnicas empleadas para tratar de vulnerar los sistemas que componen dicha infraestructura.

  Los Honeypots de producción aportan un gran valor específico para asegurar sistemas y redes con la prevención, el engaño y la disuasión de los atacantes, desviándolos de su objetivo real hacia el señuelo. Como respuesta a una intrusión se toman medidas oportunas en contra de cualquier ataque hacia la red real (denegando cualquier acceso con un origen determinado, limitando las capacidades de un servicio o paralizando servicios momentáneamente en el caso de ser posible).

  Las Honeypot al no ser sistemas en producción reales pueden ser apagados y puestos para un análisis forense post ataque, lo cual puede proporcionar más información sobre ataques realizados, esto las convierte en una herramienta poderosa para complementar la capacidad de reacción de un administrador de red al tener un detalle de los métodos, herramientas usadas por los atacantes en los sistemas (Kalma Rahmatullah, Michrandi Nasution, & Azmi, 2016).

- **Honeypot de investigación (Research Honeypot System):** En este caso, el principal objetivo es la recopilación de la mayor cantidad de información que permita al investigador poder analizar las nuevas tendencias en los métodos de

ataque, así como los principales objetivos perseguidos y los distintos orígenes de los ataques. El resultado de este análisis es recogido en informes cuyo objetivo es respaldar la toma de decisiones en la implantación de las medidas de seguridad preventivas.

Los honeypot de investigación también han sido diseñadas para ser comprometidas al igual que los de producción, sin embargo, no añaden ninguna capacidad extra de seguridad o mitigación de los ataques. La principal ventaja de situar el Honeypot en una red independiente, dedicada únicamente a la investigación, es la separación del sistema vulnerable del resto de sistemas productivos y evitar así la posibilidad de sufrir un ataque a través del propio Honeypot. Por el contrario, el inconveniente es la cantidad de recursos necesarios.

4.2.3.2. Según su Nivel de Interacción

- **Honeypot de baja interacción:** Suelen ser creadas y gestionadas por organizaciones dedicadas a la investigación del fraude en Internet, o cualquier tipo de organización que necesite investigar sobre las nuevas amenazas en la red. Estas Honeypot trabajan únicamente emulando servicios y sistemas operativos. La actividad del atacante se encuentra limitada al nivel de emulación del Honeypot.

  Por ejemplo, la emulación de un servidor HTTP podría responder tan sólo a peticiones de un fichero en particular e implementar sólo un subconjunto de las especificaciones HTTP. El nivel de interacción debe ser el justo y suficiente para engañar al atacante o a una herramienta automatizada, tal como un worm que está buscando un fichero concreto para comprometer al servidor.
  La ventaja de una Honeypot de Baja Interacción radica principalmente en su simplicidad y fácil mantenimiento, ya que estas tienden a ser fáciles de utilizar y mantenerse con un riesgo mínimo. Normalmente, basta con implementar la Honeypot de baja interacción y dejarla recolectar datos por sí sola. La información

puede tratar sobre propagación de gusanos o worms en la red o el escaneo causado por spammer de las transmisiones abiertas en la red.

La desventaja de este tipo de Honeypot es la limitada cantidad de información recogida, al no permitirle un mayor nivel de interacción hacia el atacante, este queda limitado en su ataque y sólo muestra quizá lo que sería uno de sus primeros pasos dentro de la bitácora planificada para su ataque. En el ejemplo de la emulación del servicio HTTP, con un Honeypot de baja interacción sólo se podría registrar intentos del atacante de entrar al sistema por medio de alguna vulnerabilidad en este servicio, pero nunca se sabría cuáles son las intenciones reales de ingreso.

Las Honeypot de baja interacción se dedican primordialmente a recolectar datos y recoger información de alto nivel sobre los patrones de ataque. Además, pueden ser usadas como un tipo de detección de intrusiones en el sistema a modo de aviso. Asimismo, pueden ser utilizados para atraer a los intrusos y alejarlos de las máquinas reales, aquellas que realmente son útiles.
Entre los más comunes Honeypots de baja interacción están (Jordao da Silva Vargas & Kleinschmidt, 2013):

- **Honeyd:** Quizás uno de los Honeypots más sencillos y populares. Es un demonio que crea hosts virtuales en una red. Los anfitriones pueden ser configurados para ejecutar servicios arbitrarios, y su comportamiento puede ser adaptado para que simule estar en ejecución en ciertos sistemas operativos.

- **HoneyC:** El objetivo de este Honeypot es la identificación de servidores web maliciosos en la red. Para ello emula varios clientes y recaba la mayor cantidad posible de información de las respuestas de los servidores cuando estos contestan a sus solicitudes de conexión. HoneyC es ampliable de diversas formas: pueden utilizarse diferentes clientes, sistemas de búsqueda y algoritmos de análisis.

- **Nephentes:** Honeypot de baja interacción que pretende emular vulnerabilidades conocidas para recopilar información sobre posibles ataques. Nephentes está diseñado para emular vulnerabilidades que los gusanos utilizan para propagarse y cuando estos intentan aprovecharlas, captura su código para su posterior análisis.

- **Honeytrap:** Este Honeypot está destinado a la observación de ataques contra servicios de red. En contraste con otros Honeypots, que se suelen centrar en la recogida de malware, el objetivo de Honeytrap es la captura de exploits.

- **Glastopf:** Emula miles de vulnerabilidades para recopilar datos de los ataques contra aplicaciones web. La base para la recolección de información es la respuesta correcta que se le ofrece al atacante cuando intenta explotar la aplicación web. Es fácil de configurar y una vez indexado por los buscadores, los intentos de explotación de sus vulnerabilidades se multiplican.

ϖ **Honeypot de alta interacción:** Este tipo de Honeypots constituyen una solución compleja, ya que implica la utilización de sistemas operativos y aplicaciones reales montados en hardware real sin la utilización de software de emulación e involucrando aplicaciones reales que se ejecutan de manera normal, muchas veces en directa relación a servicios como bases de datos y directorios de archivos compartidos.

Teniendo en cuenta lo dicho anteriormente, todo esto constituye una solución mucho más compleja, son más difíciles de implementar y mantener. Retomando el ejemplo del servicio HTTP, en este caso no se emularía dicho servicio, ahora se instalaría un sistema operativo Windows o Unix, al cual se le instalará el servidor HTTP verdadero, al ponerlo en línea en algunos casos estará en la misma red de otros sistemas en producción, y brindará al atacante un nivel real de interactividad con el servicio.

Este tipo de Honeypot se trata de un sistema de computadora convencional. El sistema no tiene tareas en la red o actividad de usuarios regulares. Así, esta máquina no debe tener ningún proceso inesperado ni generar ningún tipo de tráfico en la red, excepto el propio de un sistema que está activo (demonios o servicios corriendo en el sistema). Estas presunciones facilitan el proceso de detección del ataque: toda interacción con la Honeypot resultará sospechosa y se convertirá en un punto de mira para una posible acción maliciosa.

De este modo, todo el tráfico de la red hacia o desde la Honeypot es registrado, así como toda la actividad del sistema, que será grabada para un análisis posterior. También se puede combinar el uso de varios Honeypots en una misma red, configurando de esta manera una Honeynet. Por lo general, las Honeynet consisten en varias Honeypot de diferentes tipos en cuanto a plataformas y/o sistemas operativos. Esto permite recolectar datos sobre distintos tipos de ataques simultáneamente.

Las ventajas de este tipo de Honeypot son dos:

Por un lado, se tiene la posibilidad de capturar grandes cantidades de información referentes al modus operandi de los atacantes debido a que los intrusos se encuentran interactuando frente a un sistema real. De esta manera, se está en posibilidad de estudiar la extensión completa de sus actividades: cualquier cosa desde nuevos Rootkit, zero days, hasta sesiones internacionales.

Por otro lado, las Honeypot de Alta Interacción no asumen nada acerca del posible comportamiento que tendrá el atacante, proveyendo un entorno abierto que captura todas las actividades realizadas y que ofrece una amplia gama de servicios, aplicaciones y depósitos de información que pueden servir como blanco potencial para aquellos servicios que específicamente se desean comprometer. Esto permite a las soluciones de alta interacción conocer comportamientos no esperados.

Sin embargo, esta última capacidad también incrementa el riesgo de que los atacantes puedan utilizar estos sistemas operativos reales para lanzar ataques a sistemas internos que no forman parte de las Honeypot, convirtiendo una carnada en un arma. En consecuencia, se requiere la implementación de una tecnología adicional que prevenga al atacante el dañar otros sistemas que no son Honeypot o que prive al sistema comprometido de sus capacidades de convertirse en una plataforma de lanzamiento de ataques.

Otra de las ventajas que se obtiene al montar esta solución es la gran cantidad de información que se puede recoger del atacante, según la complejidad de la Honeypot, podemos ser capaces de conocer exactamente todos los pasos del intruso, sus técnicas y sus herramientas. Aunque este tipo de Honeypot es, tal y como se ha comentado, sumamente útil como herramienta de seguridad e investigación, se ha de tener en cuenta la cantidad de recursos que consumen, siendo ésta su principal desventaja.

Entre las Honeypot de altas interacciones más usadas se pueden destacar las siguientes (Segobia, 2010):

- **HI-HAT** (High Interaction Honeypot Analysis Toolkit): Herramienta que transforma aplicaciones PHP en aplicaciones honeypot de alta interacción. Además, ofrece una interfaz web que permite consultar y monitorizar los datos registrados.

- **HoneyBow:** Herramienta de recopilación de malware que puede integrarse con el Honeypot de baja interacción Nephentes para crear una herramienta de recolección mucho más completa.

- **Sebek:** Funciona como un HIDS (Host Intrusion Detection System) permitiendo capturar una gran variedad de información sobre la actividad en un sistema ya que actúa a muy bajo nivel. Es una arquitectura cliente-servidor,

con capacidad multiplataforma, que permite desplegar Honeypots cliente en 88 sistemas Windows, Linux, Solaris, etc., que se encargan de la captura y el envío de la actividad recopilada hacia el servidor Sebek.

- **Capture-HPC:** Del tipo cliente, como HoneyC, identifica servidores potencialmente maliciosos interactuando con ellos, utilizando una máquina virtual dedicada y observando cambios de sistema no previstos o autorizados.

4.2.3.3. Según su implementación

Otra clasificación para las Honeypot se basa en su implementación (The Honeynet Proyect, 2010). Se distinguen dos tipos: Honeypot Físicas y Honeypot Virtuales.

- **Honeypot Físicas:** Las Honeypot Físicas son implementadas en una máquina física real, convirtiéndola en una Honeypot de alta interacción la cual puede ser comprometida totalmente. Como constituyen una máquina real, normalmente son más caras y complejas en su implementación.

- **Honeypot Virtuales:** En este caso, se trata de utilizar máquinas virtuales para la implementación de la Honeypot. Este método resulta mucho más sencillo de mantener que una Honeypot física y con una escalabilidad mucho mayor. Así, sería posible tener miles de estos señuelos en una sola máquina física. Además, resultan de muy bajo coste económico, por lo que son accesibles a prácticamente cualquier persona.

  Para la virtualización de un honeypot, se suele hacer uso de programas como VMware o UserMode Linux. Ambas aplicaciones habilitan a una máquina la simulación de un sistema completo o no, que responde al tráfico de red enviado. Para cualquier trabajo con las Honeypot, es necesario que el sistema pueda acceder a Internet, así como que Internet pueda acceder al sistema. La mayoría de personas están conectadas a Internet vía DSL o módems. Para una experimentación prudente,

es necesario tener un proveedor ISP que suministre una conectividad IP real completa.

4.2.4. Arquitectura de las Honeypots

La ubicación de una Honeypot es fundamental para maximizar su efectividad, debido a su carácter pasivo, una ubicación de difícil acceso eliminará gran parte de su atractivo para potenciales atacantes. Por otro parte, si su ubicación es demasiado artificial u obvia cualquier experimentado atacante la descubrirá y evitará todo contacto con ella.

Además, hay tener en cuenta que debe integrarse con el resto del sistema que se encuentran implantados (servidores web, servidores de ficheros, DNS, etc.), y asegurarse de que no interfiere con las otras medidas de seguridad que puedan ya existir en la red. La bibliografía consultada establece tres puntos básicos para albergar una Honeypot que se adapte a las diversas necesidades (Levine, Owen, Grizzard, Lee, & Dagon, 2010).

- **Delante del Firewall:** Al colocarlo delante del Firewall, hace que la seguridad de nuestra red interna no se vea comprometida en ningún momento ya que el Firewall evitara que el ataque vaya a la red interna. Como este se encuentra fuera de la zona protegida por el firewall, puede ser atacado sin ningún tipo de peligro para el resto de nuestra red.

  Esta ubicación permite tener un acceso directo a los atacantes, puesto que el firewall ya se encarga de filtrar una parte del tráfico peligroso o no deseado, obteniendo trazas reales de su comportamiento y estadísticas muy fiables sobre la cantidad y calidad de ataques que puede recibir la red.

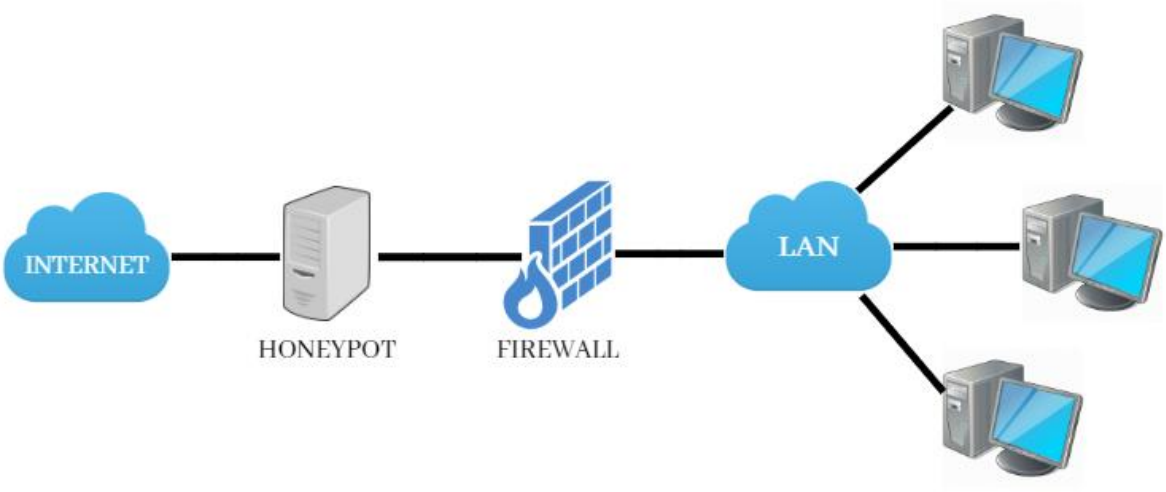


Ilustración 2. Implementación de Honeypot delante del Firewall.

(Fuente: Elaboración propia)

Además, con esta configuración se evitan las alarmas de otros sistemas de seguridad de la red al recibir ataques en la Honeypot. Sin embargo, existe el 91 peligro de generar mucho tráfico debido precisamente a la facilidad que ofrece al Honeypot para ser atacado.

Las dificultades al usar este método son:

- El ancho de banda que se consumiría, ya que al estar en el exterior del Firewall no abra dificultad en acceder a él.
- A él estar fuera de nuestro Firewall, no podremos controlar los ataques internos.

ϖ **Detrás del Firewall:** En esta posición, la Honeypot queda afectado por las reglas de filtrado del firewall. Por un lado, hay que modificar las reglas para permitir algún tipo de acceso a la Honeypot por posibles atacantes externos, y por el otro lado, al introducir un elemento potencialmente peligroso dentro de la red se puede permitir a un atacante que gane acceso a la Honeypot un pase directo a la red interna.

La ubicación tras el firewall permite la detección de atacantes internos, así como firewalls mal configurados, máquinas infectadas por gusanos o virus e incluso atacantes externos.

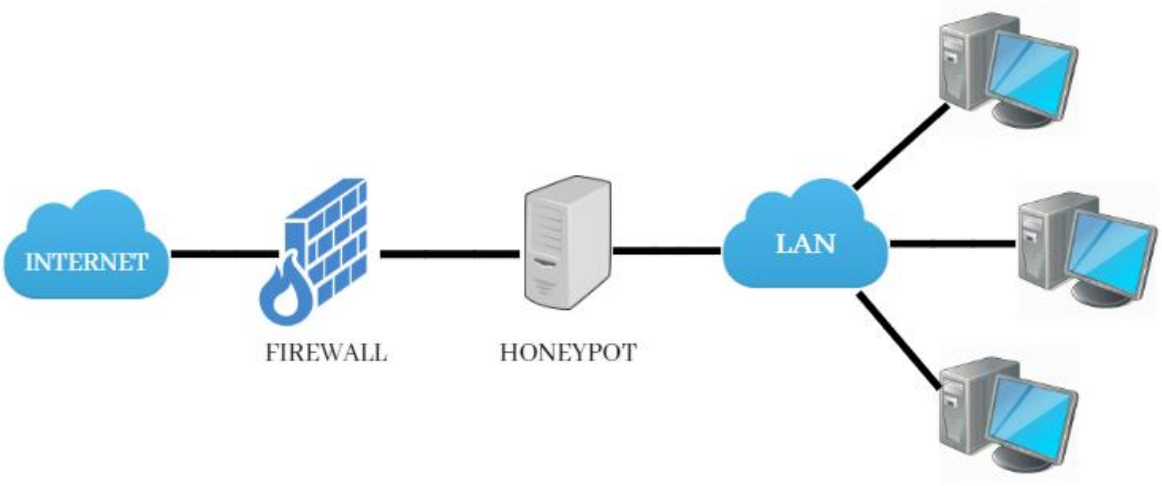


Ilustración 3. Implementación de Honeypot detrás del Firewall.

(Fuente: Elaboración propia)

Sin embargo, las contrapartidas más destacables de esta arquitectura son la gran cantidad de alertas que generarán otros sistemas de seguridad de la red, por lo que existe la necesidad de asegurar el resto de la red interna contra la Honeypot mediante el uso de firewalls extras o sistemas de bloqueo de acceso, ya que si un atacante logra comprometer el sistema tendrá vía libre en su ataque a toda la red.

- En una zona desmilitarizada: Al posicionarlo aquí se hace posible la separación del Honeypot de la red interna y la unión con los servidores, esta posibilidad permite detectar tanto ataques internos como externos con una pequeña reconfiguración del Firewall, debido a que se encuentra en una zona pública.

  Además, se elimina las alarmas de otros sistemas internos de seguridad y el peligro que supone para la red interna al no estar en contacto directo con esta.

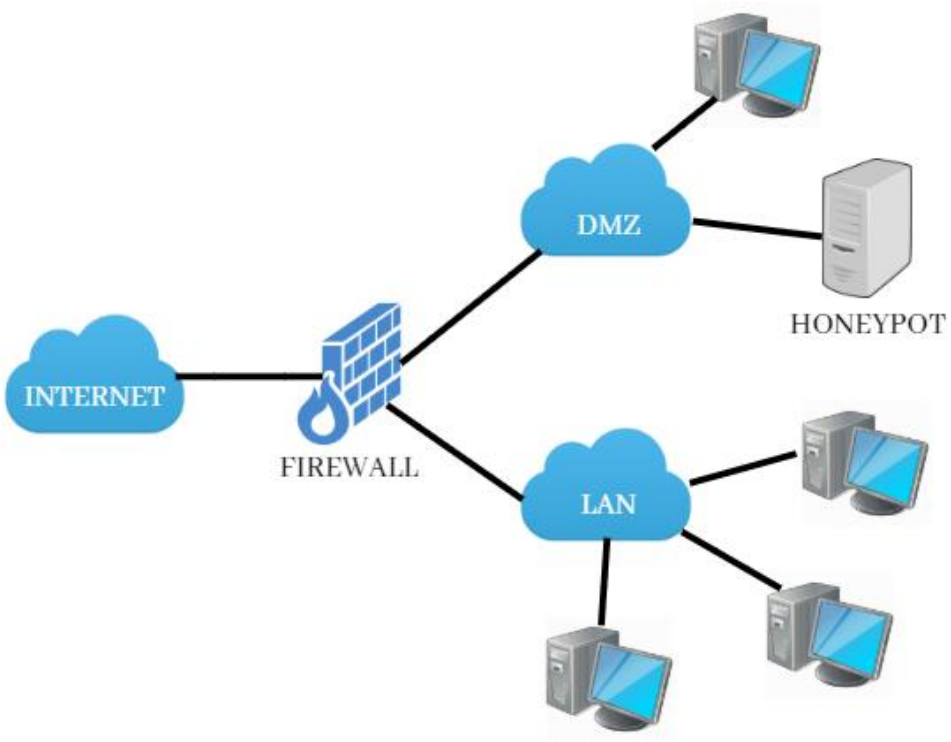


Ilustración 4. Implementación de Honeypot en una zona desmilitarizada.
(Fuente: Elaboración propia)

4.2.5. Aplicaciones prácticas de las Honeypot.

Cuando son utilizados con propósitos productivos, las Honeypot proveen protección a la organización mediante prevención, detección y respuesta a un ataque. Cuando son utilizados con propósitos de investigación, éstos recolectan información que depende del contexto bajo el cual hayan sido implementados.

Algunas organizaciones estudian la tendencia de las actividades intrusivas, mientras otras están interesadas en la predicción y prevención anticipada. Las Honeypot pueden ayudar a prevenir ataques en varias formas:

- **Defensa contra ataques automatizados:** Estos ataques son basados en herramientas que aleatoriamente rastrean redes enteras buscando sistemas vulnerables. Si un sistema vulnerable es encontrado, estas herramientas automatizadas atacaran y tomaran el sistema (con gusanos que se replican en la víctima). Uno de los métodos para proteger de tales ataques es bajando la

velocidad de su rastreo para después detenerlos. Llamados “Sticky Honeypots”, estas soluciones monitorean el espacio IP no utilizado. Cuando los sistemas son analizados, estas Honeypots interactúan con él y disminuyen la velocidad del ataque. Esto es excelente para disminuir la velocidad o para prevenir la diseminación de gusanos que han penetrado en la red interna.

- **Protección contra intrusos humanos:** Este concepto se conoce como engaño o disuasión. La idea de esta contramedida es confundir al atacante y hacerle perder tiempo y recursos mientras interactúa con la Honeypot. Mientras ese proceso se lleva a cabo, se puede detectar la actividad del atacante y se tiene tiempo para reaccionar y detener el ataque.

- **Métodos de Detección Precisa:** Tradicionalmente, la detección ha sido una tarea extremadamente difícil de llevar a cabo. Las tecnologías como los Sistemas de Detección de Intrusos y sistemas de logueo han sido deficientes por diversas razones: Generan información en cantidades excesivas, grandes porcentajes de falsos positivos (o falsas alarmas), no cuentan con la habilidad de detectar nuevos ataques y/o de trabajar en forma encriptada o en entornos IPv6.

- **Las Honeypots son excelentes en el ramo de la detección, solventando muchos de los problemas de la detección clásica:** Reducen los falsos positivos, capturan pequeñas cantidades de datos de gran importancia como ataques desconocidos y nuevos métodos de explotación de vulnerabilidades (Zero Day) y trabajan en forma encriptada o en entornos Ipv6.

- **Labor Ciber-Forense:** Una vez que un administrador de red se da cuenta que uno de sus servidores fue comprometido ilegalmente, es necesario proceder inmediatamente a realizar un análisis forense en el sistema comprometido para realizar un control de daños causados por el atacante. Sin embargo, hay dos problemas que afectan a la respuesta al incidente: frecuentemente, los sistemas comprometidos no pueden ser desconectados de la red para ser analizados y la

cantidad de información que se genera es considerablemente extensa, de manera que es muy difícil determinar lo que hizo el atacante dentro del sistema.

Las Honeypots ayudan a solventar los problemas anteriormente mencionados, ya que son excelentes herramientas de análisis de incidencias que pueden rápida y fácilmente ser sacados de la red para un análisis forense completo, sin causar impacto en las operaciones empresariales diarias. La única actividad que guardan las Honeypots son las relacionadas con el atacante, ya que no son utilizadas por ningún otro usuario, excepto los atacantes. La importancia de las Honeypot, es la rápida entrega de la información, analizada en profundidad previamente, para responder rápida y eficientemente a un incidente.

4.2.6. Definiciones conceptuales

- **Red de datos:** Se denomina red de datos a aquellas infraestructuras o redes de comunicación que se han diseñado específicamente a la transmisión de información mediante el intercambio de datos. La red de datos, también conocida como red de ordenadores o red informática, es un conjunto de equipos conectados por medio de cables, señales, ondas o cualquier otro método de transporte de datos, que comparten información, recursos y servicios.

- **Hardware:** Término inglés que hace referencia a cualquier componente físico tecnológico, que trabaja o interactúa de algún modo con la computadora. No sólo incluye elementos internos como el disco duro, CD-ROM, disquetera, sino que también hace referencia al cableado, circuitos, case, etc. E incluso hace referencia a elementos externos como la impresora, el mouse, el teclado, el monitor y demás periféricos.

- **Software:** El software, en sentido estricto, es todo programa o aplicación programada para realizar tareas específicas. El término "software" fue usado por primera vez por John W. Tukey en 1957. En una definición más amplia, se

conoce como software al equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos 107 necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos del sistema, llamados hardware.

- **Sistema Operativo:** Un Sistema Operativo (SO) es el software básico de una computadora que provee una interfaz entre el resto de programas del ordenador, los dispositivos hardware y el usuario. Las funciones básicas del Sistema Operativo son administrar los recursos de la máquina, coordinar el hardware y organizar archivos y directorios en dispositivos de almacenamiento.

- **Host:** Se refiere a las computadoras conectadas a la red, que proveen o utilizan los servicios que ofrece. Los usuarios deben utilizar hosts para tener acceso a la red. En general, los hosts son computadoras mono o multiusuario que ofrecen servicios de transferencia de archivos, conexión remota, servidores de base de datos, etc.

- **Equipos de comunicación:** Son los dispositivos de hardware, sean estos internos o externos, que permiten la conexión y comunicación entre los elementos que conforman una red.

- **Protocolo:** Es un conjunto de reglas usadas por las computadoras y equipos de comunicación, para establecer una conexión o comunicarse unas con otras a través de una determinada red. Un protocolo es una convención o estándar que 108 controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales.

- **Daemon o Demonio:** Programa o proceso que se ejecuta en segundo plano en los sistemas UNIX/Linux, es decir, se ejecuta sin intervención del usuario, mientras este ejecuta o trabaja con otras aplicaciones.

- **IDS:** El término IDS (Sistema de detección de intrusiones) hace referencia a un mecanismo que, sigilosamente, escucha el tráfico en la red para detectar actividades anormales o sospechosas, y de este modo, reducir el riesgo de intrusión.

- **Auditoria Informática:** La auditoría informática es el proceso de recoger, agrupar y evaluar evidencias para determinar si un sistema de información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización, utiliza eficientemente los recursos, y cumple con las leyes y regulaciones establecidas. También permiten detectar de forma sistemática el uso de los recursos y los flujos de información dentro de una organización y determinar qué información es crítica para el cumplimiento de su misión y objetivos, identificando necesidades, duplicidades, costes, valor y barreras, que obstaculizan flujos de información eficientes.

- **Herramientas, Mecanismos o Esquema de seguridad de redes:** Conjunto de técnicas o instrumentos sean estos, de hardware, software, reglas o normas que son usadas para proteger un grupo de computadoras que formen una red dentro o fuera de una organización o empresa, y cuya comunicación se desea proteger de elementos no deseados tales como intrusos y software maliciosos (malware).

- **Intruso o Atacante:** Es una persona o individuo que intenta acceder a un sistema informático sin autorización, con el fin de obtener información del sistema o sabotear al mismo. En contraste con los hackers, los intrusos tienen a menudo malas intenciones y suelen disponer de muchos medios para introducirse en un sistema.

- **Riesgo:** En el contexto de la seguridad informática, la palabra riesgo casi siempre nos hace pensar en las amenazas que pueden atentar contra la seguridad de nuestros recursos, nuestra información o nuestra empresa. En un sentido más

estricto se considera riesgo al conjunto de circunstancias que pueden afectar el desempeño de todo sistema informático.

- **Vulnerabilidad:** A nivel informático, es considerada un defecto de hardware o software. Es el resultado de un fallo o deficiencia durante el proceso de creación de programas o también una falta de atención a los detalles mientras se instala algún tipo de hardware. Estos defectos si no se corrigen a tiempo harán que se vea comprometidos los sistemas de una organización o empresa y como resultado, estará propensa a recibir ataques y por sobre todo a perder información valiosa.

- **Amenaza:** En el ámbito de la seguridad informática se considera amenaza a un quebrantamiento de las normas que van en contra de la seguridad informática, es decir, burla los mecanismos de seguridad poniendo en riego sistemas enteros provechándose de las vulnerabilidades de estos con fines perjudiciales. En algunos países es considerada un delito e incluso es penado por la ley.

- **Puertos de Red:** Un puerto de red o puerto TCP/IP hace referencia a una interfaz de comunicación no física utilizada para que dos ordenadores intercambien datos haciendo uso de un servicio particular. El servicio que se utilice quedará representado por un número seguido del protocolo que se utilice para la comunicación. A los puertos se les asigna una numeración de 2 bytes (16 bits), por lo que existen 65535.

- **TCP (Transmission Control Protocol):** Protocolo de Control de Transmisión, provee un flujo de bytes confiable de extremo a extremo sobre una Internet no confiable. TCP puede adaptarse dinámicamente a las propiedades de Internet y manejar fallas de muchas clases. Este protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina.

- **IP (Internet Protocol):** Protocolo de Internet, es uno de los protocolos de Internet más importantes ya que permite el desarrollo y transporte de datagramas de IP (paquetes de datos), aunque sin garantizar su "entrega". En realidad, el protocolo IP procesa datagramas de IP de manera independiente al definir su representación, ruta y envío.

- **ICMP (Internet Control Message Protocol):** Protocolo de Control de Mensajes de Internet, notifica errores del Protocolo de Internet (IP). Como tal, se usa para 112 enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado. ICMP difiere del propósito de TCP y UDP ya que generalmente no se utiliza directamente por las aplicaciones de usuario en la red. La única excepción es la herramienta ping y traceroute, que envían mensajes de petición Echo ICMP (y recibe mensajes de respuesta Echo) para determinar si un host está disponible, el tiempo que le toma a los paquetes en ir y regresar a ese host y cantidad de hosts por los que pasa.

- **UDP (User Datagram Protocol):** Protocolo Datagrama de Usuario, proporciona muy pocos servicios de recuperación de errores, ofreciendo en su lugar una manera directa de enviar y recibir datagramas con datos del host a través una red IP.

- **FTP (File Transfer Protocol):** Protocolo de Transferencia de Archivos, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

- **SMTP (Simple Mail Transfer Protocol):** Protocolo Simple de Transferencia de Correo, protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.

- **SSH (Secure Shell):** Intérprete de órdenes segura, es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos. Se puede decir que es más seguro que Telnet.

- **Telnet (Telecommunication Network):** Es el nombre de un protocolo de red y del programa que lo implementa, que sirve para acceder mediante una red a otra máquina para manejarla remotamente. A diferencia de SSH, Telnet es un protocolo poco seguro, debido a que Telnet no encripta la información que envía por la red.

- **NetBIOS (Network Basic Input/Output System):** Es una especificación de interfaz para acceso a servicios de red, es decir, una capa de software desarrollado para enlazar un sistema operativo de red con hardware específico. De forma sencilla, NetBIOS, permite a las aplicaciones 'hablar' con la red. Su intención es conseguir aislar los programas de aplicación de cualquier tipo de dependencia del hardware. También evita que los desarrolladores de software tengan que 114 desarrollar rutinas de recuperación ante errores o de enrutamiento o direccionamiento de mensajes a bajo nivel.

4.3.Estado del arte

El campo de la seguridad informática está experimentando, en los últimos años, un crecimiento constante, el aumento de los ciber-ataques ha hecho que las empresas cada vez requieran más de perfiles especializados en la seguridad de la información y contraten servicios de seguridad. Además, cada vez son más las personas interesadas en este campo debido a la gran cantidad de información que se puede localizar en Internet.

Internet está desempeñando un papel muy importante en la sociedad moderna: el número de usuarios que acceden a la red aumenta continuamente, así como el número y el tipo de aplicaciones disponibles. Aunque originalmente fue diseñado para fines militares y de investigación, Internet ahora se utiliza para la entrega de correo rápido y confiable, transacciones comerciales y financieras, telemedicina, entretenimiento, telefonía y para controlar / acceder a laboratorios remotos y una variedad de dispositivos, sensores, pequeños procesadores e instrumentos médicos portátiles.

Al mismo tiempo, Internet se caracteriza por un número creciente de ataques de red dirigidos a conceder accesos no autorizados a las computadoras, perturbando el tráfico de la red, dañando servicios e interceptando datos (León, Hernández-Serrano, & Soriano, 2010) . El hecho de que este campo esté muy ligado a Internet hace que el día de hoy podamos encontrar mucha información de todo tipo, tanto de expertos de la seguridad de la información tratando temas complejos, como de personas que se inicia en este campo y quiere realizar sus aportaciones, investigaciones, creación de herramientas propias, modificaciones de herramientas, entre otros.

El campo de la seguridad informática evoluciona cada día, por un lado, los atacantes intentan mejorar sus métodos para así evitar las medidas de seguridad, y por otro lado las empresas de seguridad y toda la comunidad de la seguridad intentan mejorar las herramientas de protección para hacer frente a todas las amenazas. Para poder hacer frente a las amenazas es necesario entender muy bien cómo funcionan los ataques y cómo piensan los atacantes.

Dicho lo anterior, es en este punto donde las honeypots juegan un papel muy importante, ya que con el uso de estos podemos obtener información sobre los atacantes, de igual manera como fue usado para la realización del trabajo hecho en la universidad tecnológica nacional que tuvo como objetivo, determinar a través de servicios publicados en ambientes informáticos controlados, los mecanismos de ataques actuales, vulnerabilidades de servicios y la generación de políticas de seguridad que

permitan la protección de acceso no autorizado a la información utilizando honeypots(Gaspar et al., n.d.).

Entre otras tecnologías de seguridad, como cortafuegos y sistemas de detección de intrusos, las honeypots ocupan actualmente sólo un pequeño nicho(Chuvakin, 2003). Una honeypot imita las características de un sistema operativo. Las características más realistas producen una honeypot más complejo que puede recolectar datos más detallados. Este es inútil si es evitado por los atacantes. Por lo tanto, una honeypot debe ser segmentable: debe tener una presencia lo suficientemente grande como para atraer a los atacantes(Winn, Rice, Dunlap, Lopez, & Mullins, 2015)._Sin embargo, son capaces de proporcionar información de ataque única, inalcanzable por cualquier otro medio.

A día de hoy existen multitud de programas que simulan aplicaciones con el único objetivo de ser atacadas para obtener información, existen además distribuciones completas con diversas herramientas que nos permiten la puesta en marcha de servicios con el único objetivo de ser atacados. Además, podemos encontrar proyectos dedicados a las honeypots, como puede ser; el desarrollaron un sistema honeypots para teléfonos inteligentes, el cual presento problemas en el funcionamiento del teléfono relacionado con el deterioro de la vida útil del mismo, sin embargo, la honeypot podría ayudar a generar estadísticas reales sobre el comportamiento de ataque que se puede compartir con otros usuarios o la seguridad del sistema para evitar la propagación del malware(Ahmed, Hassan, & Fahad, 2017).

A pesar de todos los programas dedicados a simular servicios y aplicaciones, las distribuciones y la documentación que podemos encontrar en Internet, no existe un manual de cómo poner en marcha una honeypot adaptado a las necesidades de una empresa, ni existe un software que se adapte de forma fácil a las características de una empresa. Esto es así, en parte, porque el estudio de las honeypots es muy complejo, de hecho, su estudio y aplicación aboca más a las grandes empresas, aunque con el tiempo ha evolucionado en su uso(Salazar, 2006).

La tarea de poner en marcha una honeypot en una empresa se complica aún más cuando la infraestructura es compleja, como lo es la de un operador de telecomunicaciones. Durante este proyecto se pretende dar solución a este problema, por esto no solo se centrará en la instalación y configuración de un sistema honeypot sino en todo el análisis previo de la infraestructura a simular, qué distribuciones, herramientas o programas se adaptan mejor y como la información obtenida de este honeypot nos aporta valor.

Es importante resaltar que la sola implementación de honeypots, por más eficaz que este pueda ser, no garantiza la seguridad de una red, ya que la seguridad que este brinda depende del análisis de la información que logre recopilar y del conocimiento del analista. Honeypots como "Maya" creado por S. Sharma el cual posee un robusto mecanismo de registro, es una herramienta basada en web para administrar la supervisión de Honeypot y el cual tiene un nuevo conjunto de reglas para el análisis de ataques a nivel de aplicación(Sharma, 2016). Son capaces de brindar una gran cantidad de información que en manos de expertos puede convertirse en posibles cambios que mejoren la seguridad de una red.

5. OBJETIVOS Y ALCANCE

5.1. Objetivo general

Implementar un sistema para simplificar el análisis de la información obtenida de ataques informáticos a redes de datos empresariales soportado en honeypots.

5.2. Objetivos específicos

- Identificar los requisitos de un sistema de detección de ataques informáticos a redes de datos empresariales soportado en honeypots, a partir del análisis de la información recopilada.
- Diseñar una arquitectura para el sistema de detección de ataques informáticos a redes de datos empresariales soportado en honeypots.
- Implementar un sistema de detección de ataques informáticos a redes de datos empresariales soportado en honeypots.
- Realizar pruebas de funcionalidad para comprobar el correcto funcionamiento del sistema de detección de ataques informáticos.

5.3. Alcance de la investigación

La Universidad de Cartagena ha desarrollado investigaciones con el fin de generar nuevos conocimientos con un impacto a nivel local, regional, nacional e internacional.

Con base en lo anterior, las líneas de investigación de la e-security proyectan la posibilidad de desarrollar un sistema para la detección y obtención de información de posibles ataques informáticos a redes de datos empresariales soportado en honeypots, dicho estudio aportar una base de información que contendrá de manera clara las lecciones aprendidas de éxito y de fracaso durante todo el proceso, desde la generación de la base de datos para almacenar la información de los diferentes ataques, el desarrollo del sistema y en si el proceso en general, generando un aporte a la comunidad investigadora para fortalecer las diferentes áreas de la e-security.

La universidad de Cartagena brinda la posibilidad de crear los escenarios adecuados en un ambiente controlado para la simulación y análisis de las funcionalidades, evitando riesgos como comprometer información importante de la institución. Los resultados

arrojados por las pruebas serán analizados y servirán como evidencia que garantice la fiabilidad del sistema.

Por otra parte, esta investigación amplía el acervo documental de la comunidad científica y de las empresas de desarrollo de sistemas de seguridad, en relación con la aplicación de nuevas técnicas en el diseño de productos a partir de resultados obtenidos que puedan ser utilizados para definir procesos y desarrollar otros productos alternativos e innovadores.

6. METODOLOGIA

6.1. Tipo de investigación

La metodología que permitió el desarrollo de este proyecto investigativo tiene carácter aplicado, bibliográfico y experimental. La investigación tiene un componente aplicado, pues el proyecto busca el diseño y desarrollo de un sistema de información, teniendo en cuenta conocimiento previo obtenido a lo largo de la carrera, además se han aplicado conceptos propios del contexto del problema principal que es la seguridad informática.

El carácter bibliográfico es evidente en el primer objetivo específico, debido a que mucha de la información que comprende esta investigación se halla en documentos, textos, guías, revistas, internet; lo que permitirá no solo determinar la utilidad de este proyecto de tesis, sino, que se podrá analizar resultados obtenidos basados en otras investigaciones, construir un modelo de demostración y generar conclusiones.

Por último, fue de vital importancia comprobar la funcionabilidad del sistema en un ambiente real, en este caso la red de datos de los laboratorios de la Universidad de Cartagena. Aquí deberemos ponerlas en marcha el sistema de detección y tras esto realizar ataques controlados para ver qué información obtenemos. Además de los ataques controlados deberemos estudiar la información generada por ataques no

controlados que nos ayudarán a decidir en un futuro si la empresa necesita nuevas medidas de seguridad, esto le otorgó al proyecto un carácter experimental.

6.2.Diseño utilizado

El proyecto de grado está enmarcado en el desarrollo de una plataforma web que recibe alertas de honeypots vinculados a este para complementar la seguridad en las redes de datos empresariales, para esto fue necesario, primero la recopilación y contextualización de información referente a las redes de datos de las empresas, lo cual permita determinar datos como: políticas de una compañía, misión, visión, aspectos técnicos : topología de red, servidores de red, servidores, firewalls, equipos, etc. Por otra parte, el proceso de contextualización proporciona un claro enfoque para realizar un estudio previo y planeación de los recursos (tiempo, dinero, tecnología, etc..) requeridos para realizar los pasos posteriores de la tesis de grado.

Paralelamente se llevó a cabo una profundización por parte del grupo acerca de la seguridad informática, honeypots y las técnicas de ataques a redes de datos, para tener en cuenta las pautas necesarias para realizar el trabajo.

Tras obtener la información que necesita el grupo de trabajo, se desarrollaron todos los puntos que conlleven a los objetivos específicos. Luego se realizaron pruebas funcionales del sistema desarrollado, para esta prueba se eligió como el escenario donde se producen y obtienen datos un centro de educación superior como lo es la Carrera de Ingeniería en Sistemas de la Universidad de Cartagena; específicamente La red de datos de los laboratorios de sistemas. Se activó el sistema por el lapso de tiempo de 1 hora y durante esa hora se realizaron 2 ataques controlados que fueron un escaneo de puertos con nmap y una conexión al servicio TELNET utilizando la herramienta putty. Finalmente se analiza el sistema, en aras de describir los hallazgos y resultados encontrados pertinentes del caso, que permitan posteriormente realizar recomendaciones de remediación.

6.3. Metodología para desarrollar el sistema

Para cumplir los objetivos específicos que comprenden la construcción del sistema se empleó la metodología de desarrollo de software R.U.P. Esta es una metodología parte de una implementación del desarrollo en espiral. Se caracteriza por dividir el ciclo de vida del desarrollo en etapas y fases (IBM, 1998)

En concreto existen 4 fases en R.U.P:

Fase inicial: En esta fase se realizó el análisis y comprensión del problema, se determinó el límite, alcance y riesgos asociados al proyecto. Esta fase nos dio como resultados los casos de uso del modelo de negocio con su respectiva descripción y una visión general de la arquitectura.

Fase de elaboración: En esta fase se diseñaron los diferentes modelos y diagramas a partir de los requisitos establecidos en la fase anterior para el desarrollo de un sistema de detección de ataques informáticos a redes de datos empresariales, soportado en honeypots y se seleccionaron los patrones arquitectónicos que se ajustan a las necesidades del componente software a desarrollar.

Fase de construcción: En esta etapa se realizó el desarrollo de las funcionalidades, implementación de la estructura de datos, procedimientos, documentación técnica de la integración del componente software que permitió realizar un sistema de detección de ataques informáticos a redes de datos empresariales soportado en honeypots a partir de los diagramas, modelos y patrones arquitectónicos establecidos en la fase anterior.

Fase de transición: En esta última fase se realizaron las pruebas de funcionalidad y se verifico la adaptabilidad del sistema en combinación con el sistema de detección de

intrusos y cortafuegos, además se realizó la evaluación del resultado de esta investigación de acuerdo a las pruebas de ataques realizadas a la red.

7. RESULTADOS Y DISCUSIÓN

Como resultado del proceso de investigación que se llevó a cabo durante el desarrollo del trabajo de grado, se construyó un sistema de detección de ataques a redes de datos empresariales utilizando honeypots, que permita minimizar los riesgos de ataques a sistemas y redes de TI

El sistema anteriormente mencionado se describe en las cuatro secciones siguientes: Requisitos, que describe el contexto del mundo real del proyecto y además se explica los requerimientos funcionales y no funcionales que cumple el software desarrollado; Arquitectura del sistema, donde se describe la arquitectura por medio del modelo 4+1 view; la implementación del sistema, donde se detalla el proceso de desarrollo, haciendo énfasis en las decisiones que permitieron obtener la versión final del sistema de detección de ataques; por último se muestre el apartado de las pruebas, donde se anotan los resultados obtenidos de someter al sistema de detección de ataques al caso de prueba desarrollado para examinar su funcionamiento.

7.1. Requisitos de un sistema de detección de ataques informáticos a redes de datos empresariales soportado en honeypots

En el presente inciso se encuentran definidas las especificaciones funcionales para la implementación y construcción de un sistema de detección de ataques informáticos a redes de datos empresariales soportado en honeypots realizado en la Universidad de Cartagena.

7.1.1. Ámbito del sistema

El sistema de detección de ataques informáticos a redes de datos empresariales soportado en honeypots, es una aplicación que da apoyo a los procesos que gestionan la seguridad en las redes de empresas. Mediante el análisis de la información de una honeypot, el sistema debe ser capaz de:

- Vincular honeypot.
- Reportar ataques dirigidos hacia la honeypot.
- Analizar alertas.
- Generar estadísticas.
- Gestionar cuentas de usuario.

Ya que la información de cada reporte realizado con el tiempo se convertirá en las vigas que sostendrán la arquitectura de seguridad de las redes empresariales, el sistema brindara protección a estos datos mediante la gestión de cuentas de usuarios.

7.1.2. Modelo de dominio

El producto software es un sistema que administra la información de una honeypot y brinda soluciones que representaran un valor añadido a la seguridad de las redes de una empresa.

La honeypot se trata como un software independiente del sistema de administración, ya que su posición como señuelo garantiza que sea víctima de ataques informáticos contantemente, podemos considerar el entorno de ejecución de este como un ambiente no seguridad, lo que hace indispensable separar la base de datos y el servidor del sistema, del ambiente en donde se esté ejecutando la honeypot. La relación entre el sistema de administración y la honeypot es muy estrecha, solo se limita a hacer reportes de los ataques y el protocolo de vinculación.

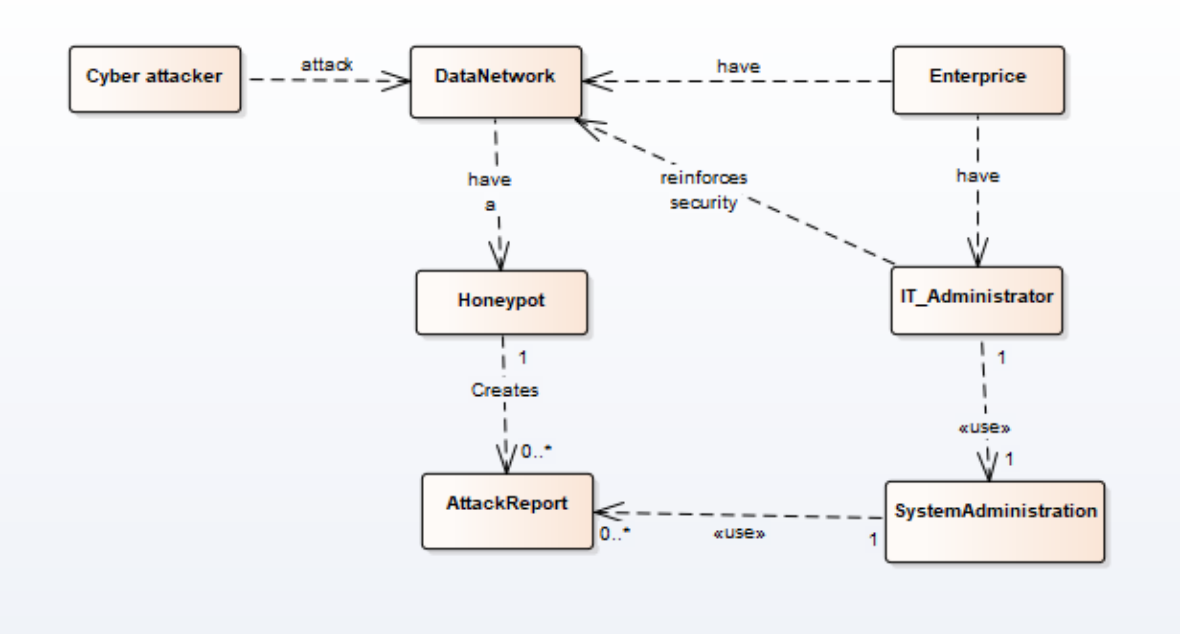


Ilustración 5. Modelo de dominio

(Fuente: Elaboración propia)

En la *Ilustración 5*, se muestra el modelo de dominio del sistema de seguridad y se puede apreciar la interacción de los entes principales. Un administrador de tecnologías de la información puede tener más de una red empresarial bajo su dominio con un honeypots instalado en cada una de ellas, ya que analizar la información de varios honeypot es una tarea complicada, el sistema de administración se encarga de tomar los informes de ataques dirigidos a los honeypot y unirlos en una sola base de datos, de manera que el administrador de tecnologías de la información pueda fácilmente analizar la información y tomar decisiones tomando como apoyo las sugerencias del sistema , que le permitan mejorar la seguridad de sus redes.

Por otra parte, aunque el sistema de administración puede realizar su función principal automáticamente, necesita de un administrador para cumplir con las funciones que tienen que ver con la protección de la información y vinculación de honeypot al sistema. Desde la perspectiva de administrador de tecnologías de la información se ve la información grupal e individual de los honeypots instalados por medio de estadísticas e informes detallados que tienen los datos provenientes de cada ataque individualmente.

7.1.3. Diagrama de casos de uso

En este apartado se encuentran descritas las funcionalidades de producto basándose en los usos que se evidencian en la Ilustración 5. En principio, los actores descritos en la ilustración representan los principales entes que interactúan con el sistema, especificando los usos de relevancia para cada uno.

- **Administrador del sistema:** Este rol se encarga principalmente de la instalación y configuración de las honeypots, también de la configuración de las cuentas de los usuarios para la vinculación de honeypots a su cuenta. Este rol implica tener conocimientos de manejo e instalación de servidores y un completo conocimiento del funcionamiento de la aplicación ya que este representa el soporte técnico del sistema.

- **Administrador de tecnología de la información**: Este rol le corresponde al usuario final del sistema, dichos administrador tienen a su disposición 1 caso de uso principal relacionado con la revisión de estadísticas y que tiene implícito los reportes de ataques dirigidos al honeypot lo cual le permite hacer modificaciones en el esquema de seguridad de la empresa para hacerla más rígida contra futuros ataques.

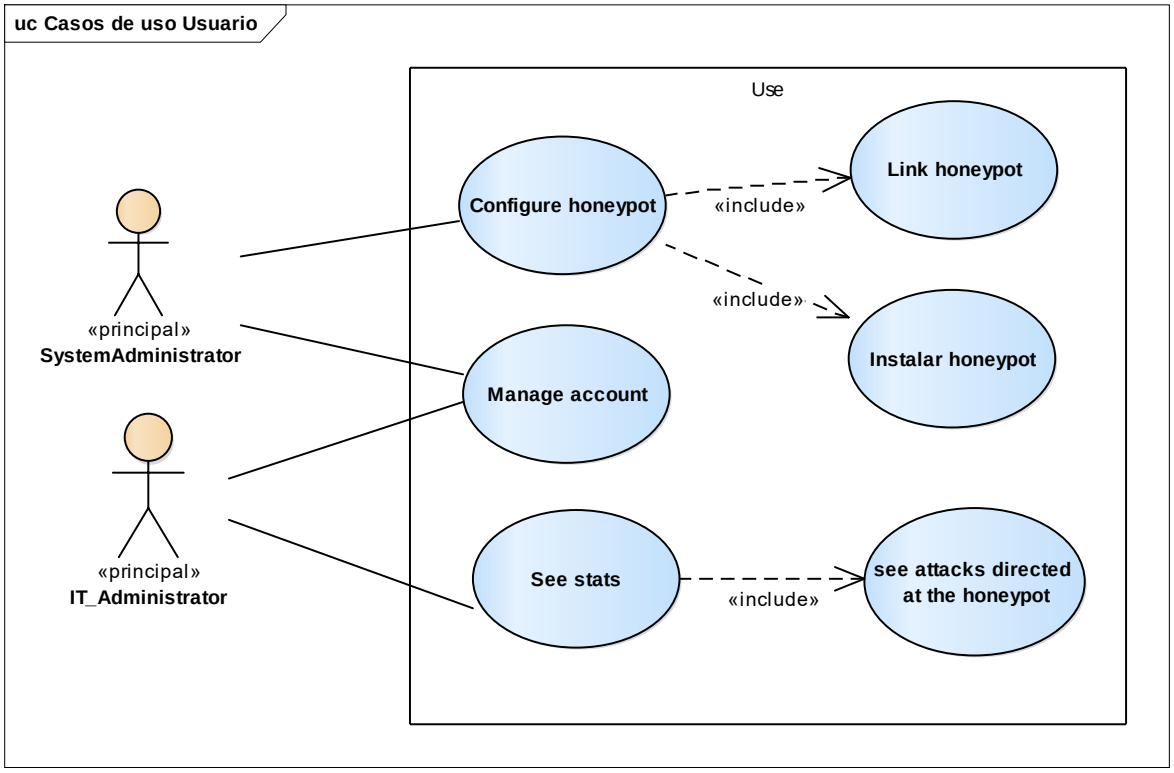


Ilustración 6. Casos de uso

(Fuente: Elaboración propia)

7.1.3.1. Configurar honeypot (Configure honeypot)

La vinculación del honeypot se hace al momento de configurar su instalación. En la instalación, se le suministra al honeypot el enlace que permite el consumo del servicio para enviar los datos de los ataques al sistema de administración. Cuando se finaliza el proceso de instalación y configuración del honeypot, se hace un escaneo inicial para comprobar su funcionamiento, con esto el honeypot entra en una fila de no vinculados dentro del sistema de administración y queda en espera que el administrador del sistema lo vincule mediando su cuenta de administrador a una cuenta de usuario.

7.1.3.2. Analizar alertas (See attacks directed at the honeypot)

Cuando se genera un ataque y luego de ser reportado al servidor web, este desglosa la información del archivo recibido para crear un nuevo formulario que tendrá la información pertinente para el usuario.
Para generar el nuevo formulario es necesario realizar los siguientes pasos:

- Extraer la ID del honeypot; de esta manera podremos saber a quién va dirigido el nuevo formulario que se está creando.

- Procesar la información que proporcionó el honeypot acerca del ataque y compararla con la información de la base de datos.

- Posteriormente, se hace nueva consulta a otra base de datos que contiene recomendaciones de cómo protegerse ante ese tipo de ataques y se realiza un informe con detalles del ataque y recomendaciones resultantes de la consulta.

- Y, por último, consolidar la información y crear el formulario para enviarlo al usuario respectivo.

7.1.3.3.Ver estadísticas (See stats)

El usuario podrá ver estadísticas referentes a los tipos de ataques que han sido dirigidos a su honeypot de manera que pueda determinar el número de ataques con relación al tipo y tiempo.

7.1.3.4. Gestionar cuenta (Manage account)

La protección de la información es un punto muy importante para toda empresa por lo que el sistema de administración cuenta con un sistema de cuentas de usuarios, permitiendo la creación de una cuenta de administrador y de usuario que contienen la

permitiendo la creación y eliminación de cuentas de usuario por medio de una cuenta de administrador.

7.2. Arquitectura del sistema

Para expresar la arquitectura del sistema de se eligió el modelo 4+1 view. A continuación, se listará cada vista con sus respectivos artefactos UML que la expresan.

7.2.1. Vista lógica

En la *Ilustración 7*, se muestra la interacción entre componentes del sistema que se detallaran a continuación.

- **functionsController:** Este controlador representa un conjunto de las siguientes funcionalidades: guardar información de los ataques en la base de datos, extraer la información de los ataques de la base de datos y crear archivos detallados de los ataques y sus recomendaciones para guardarlos en la base de datos.

- **reportController:** Este controlador controla la estructura de las entradas al sistema y convierte el formato suministrador por le honeypot en un formato leíble para functionsController.

- **accountsController:** Este controlador tiene las funcionalidades de gestionar la información de las cuentas de usuario y la vinculación de los honeypot.

- **routesController:** Este controlador de encarga de disponer las rutas de los métodos y funcionalidades que se exportar de los controladores accountsController y functionsController.

- **honeypot:** Representa a la instalación remota del honeypot en una red empresarial.

- **userApplication:** Es un controlador que representa la capa de presentación de los datos.

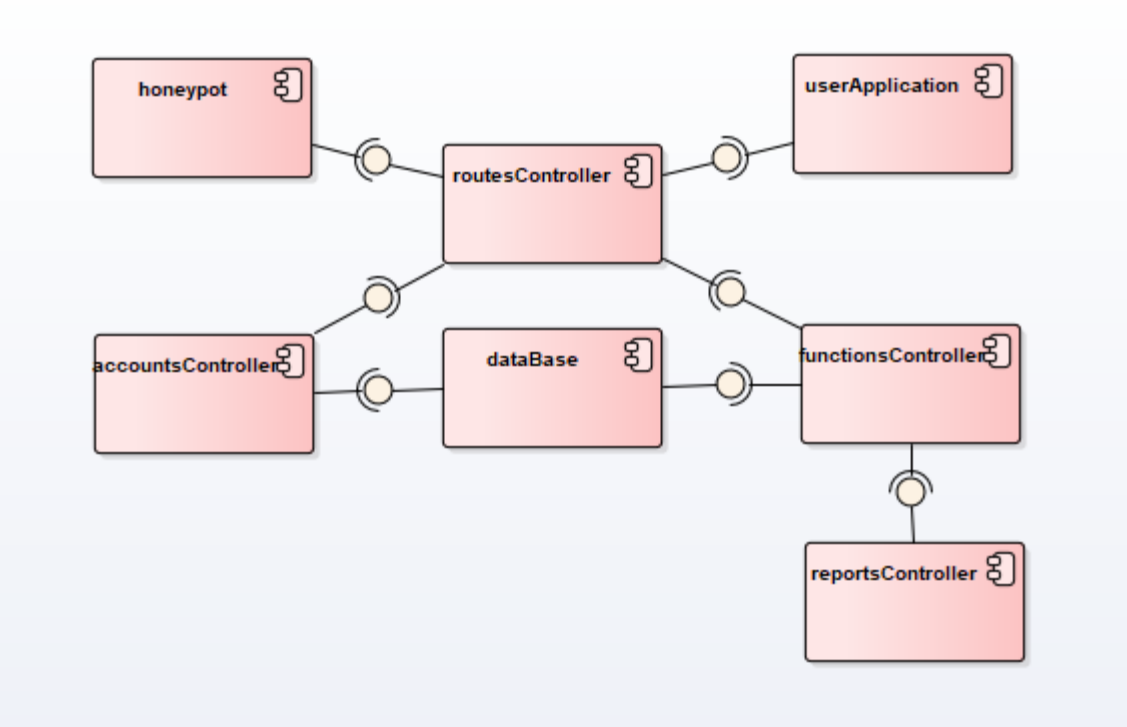


Ilustración 7. Diagrama de componentes

(Elaboración propia)

7.2.2. Vista de despliegue

Los componentes necesarios para la creación de un sistema de análisis de la información obtenida de ataques informáticos a redes de datos empresariales soportado en honeypots son: honeypot, un componente de análisis de datos, un componente de presentación y base de datos.

Correspondientemente los componentes que representan los descritos anteriormente son:

- **Honeypot:** Este componente es quien captura la información de los ataques informáticos. Par exportar sus datos el honeypot consume los servicios del

sistema de administración, enviando mediante un archivo JSON los datos de cada ataque en el momento en que se ejecutan.

- **Sistema de administración:** Este componente se encarga del análisis de los datos suministrados por la capa de presentación, honeypot y persistencia. Este componente expone servicios que permiten recolectar datos y es el único componente que consume los servicios del componente de base de datos.
  Desde este componente se realizan las tareas de analizar alertas y dispone la información para el resto de funcionalidades del sistema como son: la gestión de cuentas de usuarios, vincular honeypot, generar estadísticas, gestionar cuentas de usuario.

- **Aplicación de usuario:** este componente tiene la funcionalidad de mostrar los datos al usuario, representa un conjunto de controladores que gestionan la parte grafica del sistema y el tratamiento de los datos suministrados por la capa de análisis de datos de manera que sea fácil de analizar por el usuario.

- **Base de datos:** este componente se encarga de la permanencia de los datos del usuario con respecto al tiempo, se guarda toda la información sensible que sea administrada por el sistema de administración.

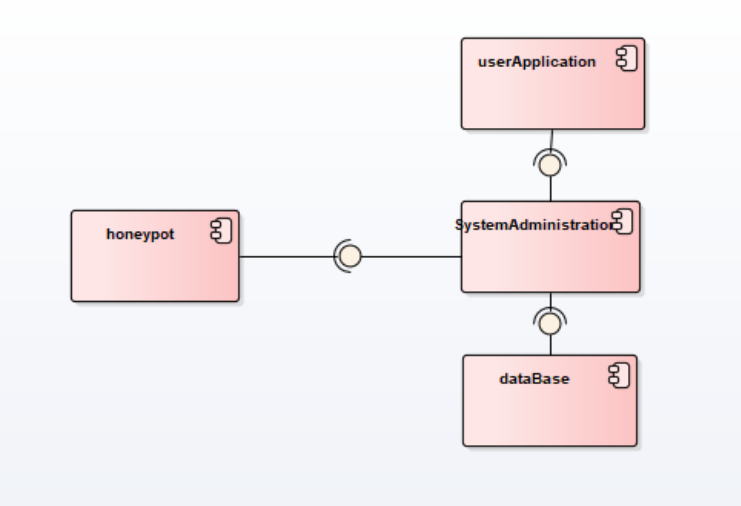


Ilustración 8. Diagrama de componentes

(Fuente: los autores)

7.2.3. Vista de procesos

El sistema de administración está pensado para simplificar las actividades que el usuario debe llevar a cabo para suplir su necesidad. Las funcionalidades que se detallaron en el inciso 7.1.3 Diagrama de casos de uso, corresponden a partes de procesos mayores que suplen la necesidad del usuario. En la *Ilustración 9* e *Ilustración* 10 se puede apreciar una vista simplificada de los procesos que contribuyen a cumplir los objetivos de este proyecto, incluyendo dentro de estos las funcionalidades del producto de las que se habló anteriormente y mostrados desde una perspectiva de negocio.

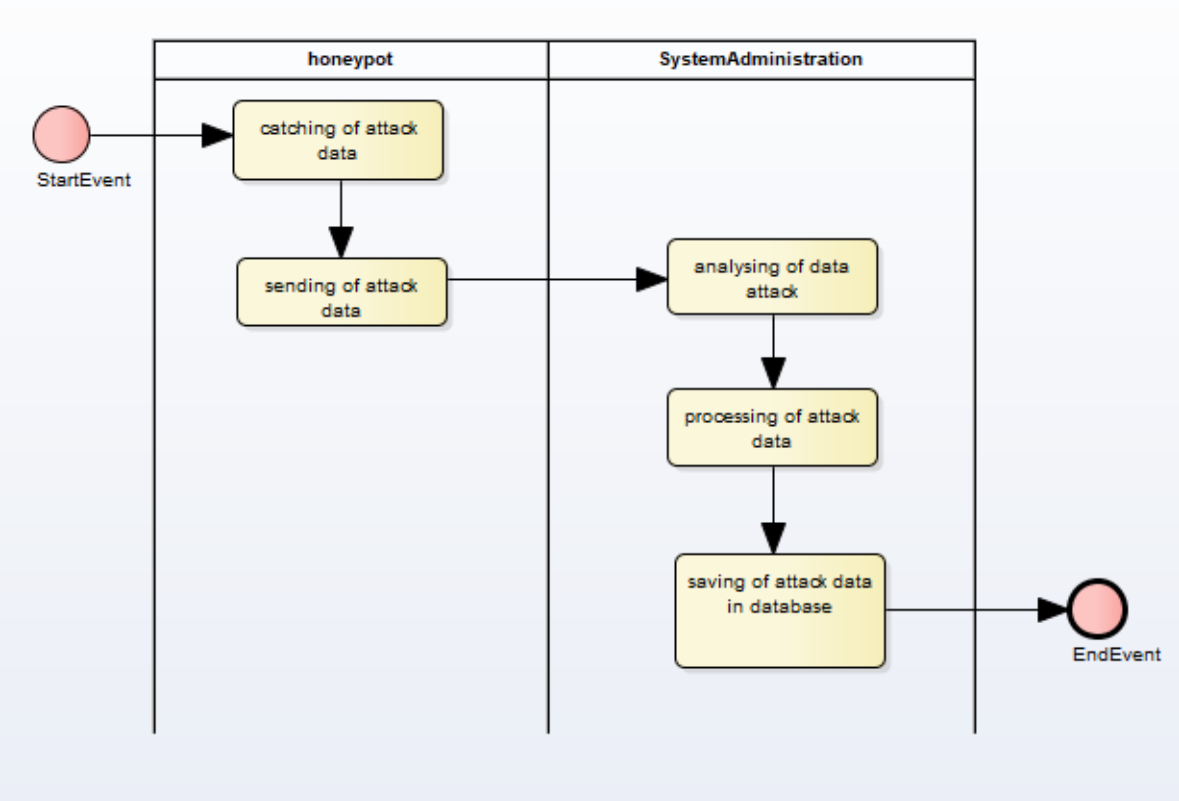


Ilustración 9. Diagrama de actividades de reportes de ataques

(Fuente Elaboración propia)

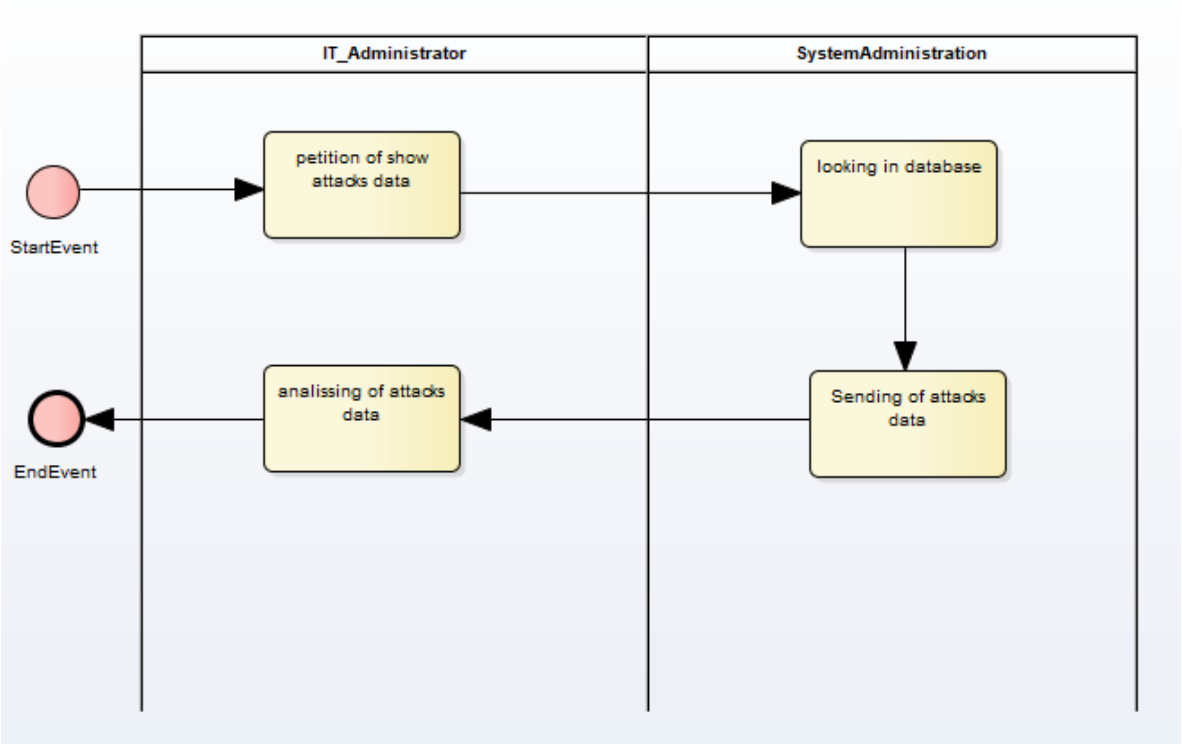


Ilustración 10. Diagrama de actividades administrador de tecnologías de la información

(Fuente: Elaboración propia)

7.2.4. Vista Física

En la *Ilustración 11* se muestra un diagrama de despliegue que evidencia las dependencias físicas que consume el sistema para su ejecución. En la imagen se aprecian tres nodos que contienen partes diferentes del sistema, uno de ellos representa el servidor en donde se estará ejecutando el honeypot y los otros dos corresponden a servidores y nodos en donde se hace el análisis y lectura de datos correspondientemente. Estos tres nodos se comunican por medio de peticiones realizadas al servidor web, los datos de entrada y salida deben tener un formato JSON.

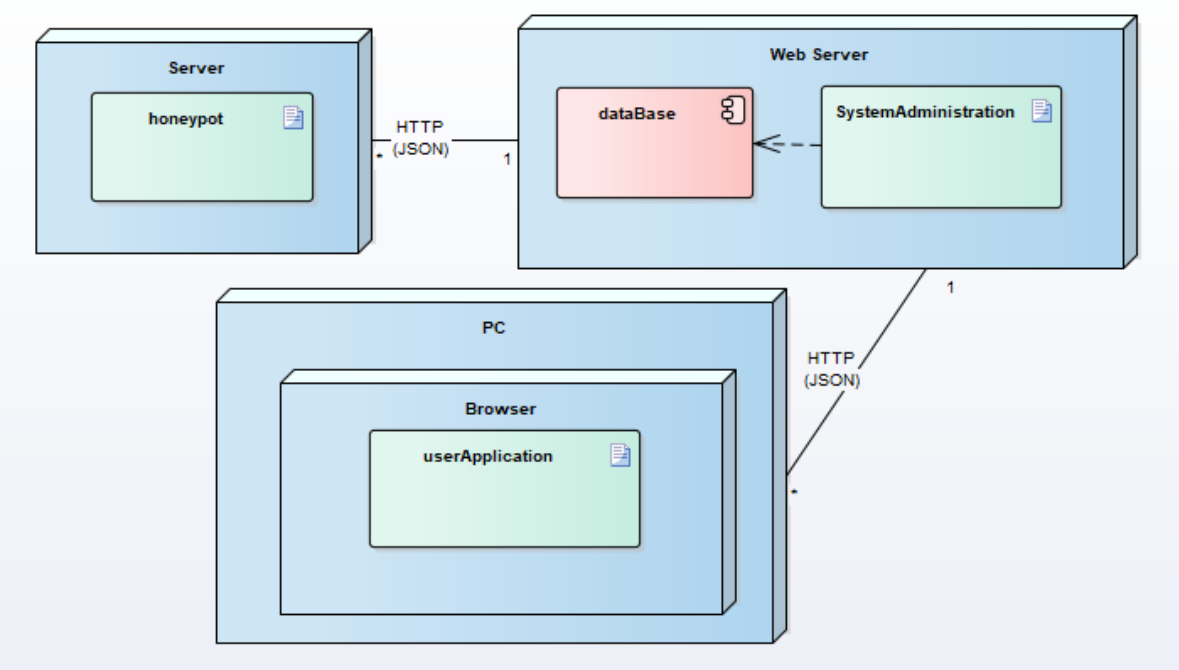


Ilustración 11. Diagrama de despliegue

(Fuente: Elaboración propia)

7.2.5. Vista de escenario

Esta vista está representada por el modelo de casos de uso que se puede apreciar en la *Ilustración 6*, en donde se muestran los siguientes casos de uso y sus relaciones:

- **Configurar honeypots:** Este caso de uso representa la vinculación e instalación tanto física como lógica de las honeypots. Como se muestra en el apartado 6.1.4 Funciones del producto, este caso de uso debe estar relacionado con el administrador del sistema y con el honeypot en cuestión.

- **Gestionar cuentas:** Este caso de uso está compuesto por la necesidad de protección y separación de la información relevante para los administradores de sistemas y usuarios finales. Este caso de uso es de suma importancia para los administradores del sistema y los usuarios ya que les permite cumplir sus funciones y suplir sus necesidades de manera segura.

- **Ver estadísticas:** Este caso de uso representa la finalidad del sistema, ya que, mediante el análisis de los datos suministrados por el honeypot, el usuario final puede apreciar estadísticas e informes que lo ayudaran a tomar las decisiones que mejoraran la seguridad de sus redes.

- **Reportar ataque:** Este es un caso de uso únicamente de interés para la honeypot, ya que obliga al establecimiento de reglas e interfaces que le permitan exportar sus resultados y ser analizados a mayor escala.

7.3. Implementación y desarrollo del sistema

En este punto se describirá las herramientas y frameworks utilizados para el desarrollo del sistema de detección de ataques a redes empresariales; fueron elegidos por su facilidad de uso, estabilidad, seguridad y rendimiento.

7.3.1. Selección del honeypot

El objetivo principal de esta sección es mostrar opciones que se pensaron y el honeypot seleccionado para integrarlo al sistema de detección de ataques. A día de hoy nos encontramos con multitud de herramientas que permiten simular puertos y crear honeypots, muchas de ellas están enfocadas a simular un solo servicio, por ejemplo, Wordpot es una herramienta en Python que simula un Wordpress. En lugar de tener una herramienta diferente por cada servicio, fue interesante tener el mínimo de herramientas que nos permitan simular el máximo de servicios, ya que esto nos facilitará trabajar con los resultados, debido a que cada herramienta nos ofrecerá unos resultados distintos que luego deberemos procesar.

A continuación, se detallaremos herramientas se consideraron candidatas para el sistema de detección de ataques:

- **HoneyPy:** Dentro de los Honeypots de baja interacción, pero que nos permiten simular varios servicios, nos encontramos con HoneyPy. HoneyPy es honeypot escrito en Python que nos permite emular servicios basados en TCP o UDP mediante el uso de plugins, la interacción con el atacante será mayor o menor según el plugin. El hecho que HoneyPy funcione en python y permita añadir plugins nos da la oportunidad de simular todos los procesos que deseemos utilizando un plugin existente o creando nuestro propio plugin. Actualmente podemos encontrar plugins para simular servicios web, DNS, SMTP, Telnet etc. Este honeypot por defecto nos escribirá toda la actividad en un fichero, pero a día de hoy puede ser configurado para enviar la actividad a una base de datos Elasticsearch, a honeydb, splunk, twitter etc.

- **Kippo:** Es un honeypot de interacción media que simula un servicio SSH. El objetivo principal es registrar ataques de fuerza bruta y registrar todos los movimientos que el atacante realiza dentro de la Shell. Dentro de la Shell el atacante observará un sistema de ficheros como el de una instalación de Debian 5.0 y podrá ver el contenido de algunos ficheros. Si el atacante descarga ficheros a través de wget estos se guardarán para su posterior análisis.

- **Mailoney:** Honeypot escrito en Python utilizado para simular un servicio SMTP. Este Honeypot puede registrar los emails que se intentan enviar estando el servicio de email configurado como open relay. También permite registrar credenciales de intentos de inicio de sesión.

- **Sippot:** Herramienta escrita en bash que tiene como objetivo detectar intentos de inicio de sesión en sistemas Asterisk. Sippot convierte un sistema Asterisk en un honeypot SIP y bloqueará todos los intentos de registro.

- **Honeyd:** Es un Honeypot que nos permite crear máquinas virtuales en una red, cada una de estas máquinas virtuales puede ser configurada para tener diferentes servicios y para simular diferentes sistemas operativos. En un mismo servidor, honeyd puede

estar a la escucha en diferentes IPs con el objetivo de simular las diferentes máquinas virtuales. Honeyd es uno de los proyectos para honeypots que cuenta con más herramientas, entre ellas podemos localizar, HoneyView: que nos permite presentar los datos de los ficheros de log de manera gráfica, Honeyd2MySQL que nos permite extraer los datos de los logs de honeyd y cargarlos en una base de datos MySQL o HoneydViz que nos permite crear gráficas a partir de los datos de Honeyd.

- **Glastopf:** Honeypot escrito en python que hace de servidor web y en este emula diversos tipos de vulnerabilidades. Dentro de las vulnerabilidades que emula este honeypot encontramos las inyecciones SQL, la inserción remota de ficheros (RFI) y la inserción local de ficheros (LFI).

- **HiHAT:** Es una herramienta escrita en Java que nos permite transformar una aplicación web PHP en un Honeypot de alta interacción. Con HiHAT podemos tener una instalación cualquiera de una aplicación web, con todas sus funcionalidades pero que será utilizada únicamente para adquirir información y monitorear su uso. HiHAT además cuenta con una interfaz gráfica que nos facilita la monitorización.

- **MysqlPot:** Es de los pocos honeypots, por no decir el único, que simula un servicio MySQL. Según indican los autores se encuentra en una fase inicial, de todos modos, este proyecto no ha sido actualizado desde octubre de 2012, por lo que no se espera que siga en desarrollo.

- **KFSensor:** Es un honeypot de uso comercial con 12 años de antigüedad en el mercado diseñado para sistemas Windows. KFSensor nos permite tener varios sensores (honeypots) en diversas instalaciones y realizar una gestión centralizada. Este Honeypot escuchará en todos los puertos TCP y UDP para detectar ataques en todos los servicios. KFSensor cuenta con un sistema de alertas por correo y permite la integración con sistemas de administración de eventos y seguridad (SIEM).

Además, incorpora un módulo de reportes y gráficas para ayudar al tratado de los ataques.

A continuación, se muestra una tabla resumen de las aplicaciones anteriormente analizadas:

	Servicios	Opciones de configuración	Documentación	Open Source	Continuidad	Plataforma
HoneyPy	Varios configurables	SI	SI	SI	SI	Linux
Kippo	SSH	NO	SI	SI	Solo pequeños cambios	Linux
MailOney	SMTP	SI	SI	SI	SI	Linux
Sippot	SIP	NO	SI	SI	NO	Linux
Honeyd	Varios configurables	SI	SI	SI	NO	Linux
Glashtopf	Web	NO	SI	SI	Solo pequeños cambios	Linux
HIHAT	Web	NO	SI	SI	NO	Linux
MysqlPot	MySQL	NO	NO	SI	NO	Linux
KFSensor	Varios configurables	SI	SI	NO	SI	Windows

Tabla 1. Comparación de honeypots

Además de todas las herramientas indicadas y de otras que podemos localizar en Internet, podemos instalar aplicaciones, que no están orientadas a ser un honeypot, pero que pueden ser configuradas de tal manera que su único objetivo sea ofrecernos información de quien las intenta utilizar y de qué manera, obteniendo así información sobre posibles ataques a este tipo de aplicaciones.

La herramienta que se escogió para simular los servicios será HoneyPy. Los motivos por los cuales se escogió esta herramienta como se comenta en el apartado anterior nos permite, mediante el uso de plugins, poner a la escucha cualquier puerto y además añadir plugins personalizados para hacer que, para servicios determinados, el grado de interacción con el atacante sea mayor.

Además, nos permite modificar su configuración y cambiar los servicios a la escucha de forma rápida, por lo tanto, nos aporta flexibilidad para añadir nuevos servicios en caso de que sea necesario en un futuro. Otro punto a favor por el cual optamos por HoneyPy es que puede soportar la plataforma open source Linux.

Las ventajas que nos ofrece utilizar recursos open source son, disminuir a dependencia a vendedores de código propietario como por ejemplo las licencias de Microsoft, no hay que presupuestar el coste de mantenimiento del software y su personal ya que sus licencias y actualizaciones son gratuitas, soporte por parte de una comunidad de usuarios, acceso al código y posibilidad de modificarlo según las necesidades, entre otros.

Por contra, al ser una herramienta de honeypot de baja y media interacción nos aportó menos información sobre los ataques realizados. De todos modos, el objetivo principal en un operador de telecomunicaciones con pocos recursos es bloquear los ataques y posteriormente realizar un análisis con la información que provee el sistema de detección de ataques del tipo de ataques o del posible malware utilizado. Además, siempre tendremos la posibilidad de añadir plugins al honeypot que se integró al sistema de detección de ataques lo cual nos permite una mayor interacción con el atacante o utilizar software adicional en el servidor para simular servicios específicos.

7.3.2. Configuraciones del honeypot

7.3.2.1. Configuración de registro datos

La configuración de HoneyPy en sí es muy simple y directo. El archivo de configuración está en **<el directorio HoneyPy seleccionado> /etc/honeypy.cfg**. Al ejecutar HoneyPy sin cambiar dicho archivo, los ataques a la red de datos solo se registraban en un archivo de texto plano que se guardaba de manera local, además el archivo de texto plano no es el mejor archivo para leer como se muestra en la *ilustración 12* lo cual ralentiza la comprensión del texto al momento de revisar los ataques, y es por eso que se configuro HoneyPy por otro "registrador" que nos permita obtener los reportes y procesarlos en la plataforma.

El conjunto actual de registradores que cuenta HoneyPy es:

- **Twitter** - Eventos de Twitter en Twitter.
- **HoneyDB:** publica eventos en HoneyDB (futura publicación de blog en HoneyDB por venir).
- **Slack:** publicar eventos en un canal Slack.
- **Elastic search**: publicar eventos directamente en Elasticsearch.
- **Notificaciones POST:** Enviar reportes a API externo por método medio de método POST.

Se escogió utilizar notificaciones POST debido a que la plataforma web del sistema de detección que se está implementando tendrá una arquitectura cliente- servidor.

```
                                post_logstash(honeypy_config.get('honeypy', 'useragent'), honeypy_config.get('logstash', 'host'
), honeypy_config.get('logstash', 'port'), parts[0], time_parts[0], parts[0] + ' ' + time_parts[0], time_parts[1], parts[4], parts[5],
parts[6], parts[7], parts[8], parts[9], parts[10], parts[11], parts[12])

                        # Elasticsearch integration
                        if 'Yes' == honeypy_config.get('elasticsearch', 'enabled'):
                                from lib.honeypy_elasticsearch import post_elasticsearch

                                if 'TCP' == parts[4]:
                                        if 11 == len(parts):
                                                parts.append('') # no data for CONNECT events

                                        post_elasticsearch(honeypy_config.get('honeypy', 'useragent'), honeypy_config.get('elasticsearc
h', 'es_url'), parts[0], time_parts[0], parts[0] + ' ' + time_parts[0], time_parts[1], parts[3], parts[4], parts[5], parts[6], parts[7]
, parts[8], parts[9], parts[10], parts[11])
                                else:
                                        # UDP splits differently (see comment section above)
                                        if 12 == len(parts):
                                                parts.append('') # no data sent

                                        post_elasticsearch(honeypy_config.get('honeypy', 'useragent'), honeypy_config.get('elasticsearc
h', 'es_url'), parts[0], time_parts[0], parts[0] + ' ' + time_parts[0], time_parts[1], parts[4], parts[5], parts[6], parts[7], parts[8]
, parts[9], parts[10], parts[11], parts[12])
```

Ilustración 12. Logs o notificaciones de ataques predeterminados de HoneyPy

(Fuente: Elaboración propia)

7.3.2.2. Configuración de puertos y servicios

El archivo de configuración de servicios de HoneyPy se tiene modifico se le realizan modificaciones dependiendo del operador, pero nosotros haremos una configuración de servicios similar a un computador normal ya que la que esta predeterminada como se mencionó anteriormente es muy sencillo o poco creíble. El archivo se encuentra en **< directorio HoneyPy seleccionado> /etc/services.cfg**, y este es el archivo que le dice a HoneyPy qué servicios ejecutar y en qué puertos ejecutarlos. En el archivo **services.cfg**, se muestran varias entradas que se parecen al ejemplo a continuación. Cada entrada representa un servicio que HoneyPy lanzará cuando se inicie.

```
[<nombre_del_servicio>]
plugin = <plugin que utilizaremos>
low_port = <tcp/udp>:<puerto>
port = <tcp/udp>:<puerto>
description = <descripción>
enabled = <Yes/No>
```

Nombre del servicio: Será el nombre que identificará al servicio, no debe contener espacios y aparecerá en los logs de salida.

Plugin: le dice a HoneyPy qué complemento cargar y usar para la emulación del servicio. Todos los complementos disponibles viven en el directorio de Plugins y son lo que emula un servicio, por ejemplo, Telnet, FTP, SSH, etc.

Low_port: En esta línea se deberá indicar el puerto y el protocolo si se escuchará en un puerto por debajo del 1024, en caso contrario deberá ser igual que la siguiente línea de configuración.

Port: Se debe indicar el puerto y el protocolo en el cual HoneyPy escuchará.
Description: Esta línea será utilizada para que describamos el servicio y así poder entender la configuración.

Enable: Se deberá indicar si el servicio está activo o no.

La configuración del honeypot para el operador tendrá en cuenta los puertos detectados como importantes y los que no. Para los puertos no importantes solo se tendrá en cuenta si son TCP o UDP y se configurarán con el plugin MOTD para puertos TCP y MOTD_udp para puertos UDP. Estos plugins siempre devolverán el mismo contenido a todas las conexiones, pero nos servirán para ver quien intenta establecer conexiones a estos servicios.

Por otro lado, para los puertos importantes se intentarán utilizar plugins específicos que nos aporten una respuesta lo más parecida posible al servicio expuesto y en la medida de los posible un mayor grado de interacción. A continuación, se presenta una tabla de los puertos que marcamos como importantes y que plugin de los existentes se adapta más al servicio:

Puerto	Servicio	Plugin
80	HTTP	Web
443	HTTPS	Web
5060-5080	SIP	Echo_udp
8080	HTTP	Web
25	SMTP	SmtpExim
53	DNS	DnsUdp
69	TFTP	Echo_udp
22	SSH	Random
23	TELNET	TelnelUnix
3306	MySQL	Random

Tabla 2. Servicios y plugins utilizados

A continuación, se muestra un fragmento del fichero services.cfg con la configuración para los puertos detectados como importantes:

```
[SMTP]
plugin = SmtpExim
low_port = tcp: 25
port = tcp:10025
description = Simula servicio de envió de correos (SMTP).
enabled = Yes



[DNS]
plugin = DnsUdp
low_port = udp: 53
port = udp:10053
description = Un oyente DNS falso muy simple que repite datos de
consulta DNS.
enabled = Yes
[TFTP]
plugin = Echo_udp
```

```
low_port = udp: 69
port = udp:10069
description = Simula un servicio TFTP.
enabled = Yes
```

7.3.3. Implementando contenedores con Docker

Para ofrecer una mayor flexibilidad y portabilidad al honeypot que se ha modificado, se utilizara Docker que es una herramienta diseñada para facilitar la creación, implementación y ejecución de aplicaciones mediante el uso de contenedores.

Los contenedores nos permitieron empaquetar al honeypot con todas las partes que necesita, como bibliotecas y otras dependencias, y enviarlo como un solo paquete. Este enfoque reduce el consumo de recursos, ya que cada contenedor contiene solo dependencias relacionadas. Además, gracias al contenedor, podemos estar seguros de que el honeypot se ejecutara en cualquier otra máquina Linux, independientemente de las configuraciones personalizadas que la maquina pueda tener que puedan diferir de la máquina que usamos actualmente.

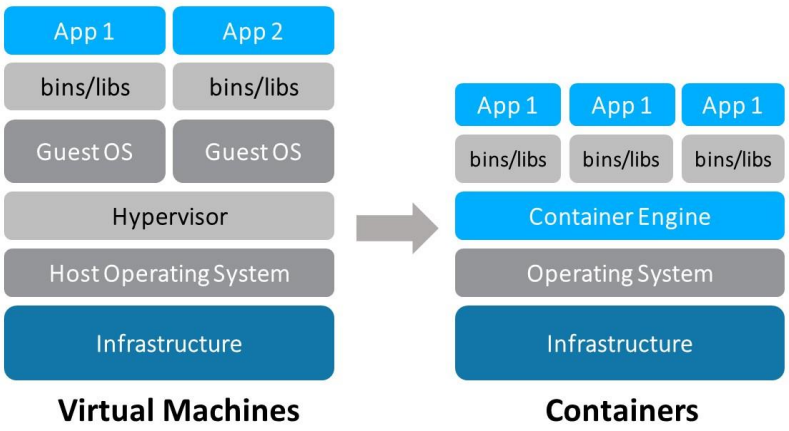


Ilustración 13. Contenedores vs máquinas virtuales

(Fuente: (Sobrado, 2017))

Como se muestra en la *ilustración 13* los contenedores proporcionan una mejor utilización de los recursos informáticos al eliminar el hipervisor, al tiempo que mantienen las tareas de separación y aislamiento sin utilizar un sistema operativo. A continuación, una serie de ventajas y capacidades:

- Las abstracciones de contenedores reducen la complejidad. Los contenedores no requieren dependencias en la infraestructura de la aplicación y, en consecuencia, no hay necesidad de una interfaz compleja con los servicios de la plataforma.

- Los contenedores proporcionan capacidades avanzadas de computación distribuida. Los componentes de aplicaciones creados en contenedores se pueden ejecutar en diferentes plataformas de la nube. Las empresas pueden elegir proveedores en la nube en función del costo y el rendimiento.

- Los contenedores pueden aprovechar la automatización para maximizar la portabilidad.

- Los contenedores brindan una mejor seguridad y gobernanza. Los servicios de seguridad y gobernanza son específicos de la plataforma y no de la aplicación. Al colocar la seguridad y el gobierno fuera del contenedor, se reduce la complejidad de una manera significativa.

- Los contenedores pueden proporcionar servicios de automatización que hacen uso de la optimización basada en políticas. Una capa de automatización puede ubicar una plataforma adecuada para ejecutar contenedores y migrar automáticamente a la plataforma respectiva.

7.3.4. Aplicación web de una sola pagina

En lugar de mostrar varias páginas HTML se optó por crear una aplicación de una sola página o simple page application (SPA) ya que las aplicaciones web con varias páginas vuelven a cargar toda la página y muestran la nueva cuando el usuario

interactúa con la aplicación web. Cada vez que se intercambian datos de ida y vuelta, se solicita una nueva página al servidor para mostrar en el navegador web. Este proceso lleva tiempo para generar las páginas en el servidor, enviarlas al cliente y mostrarlas en el navegador, lo que afecta la experiencia de usuario.

En una aplicación SPA como se muestra en la *Ilustración 14*, después de que se carga la primera página, todas las interacciones entre el cliente y el servidor ocurren con las llamadas AJAX (Asynchronous JavaScript and XML), que a su vez envían datos en formato JSON (o XML). Luego, el navegador usa los datos JSON para actualizar la página dinámicamente, sin recargar ninguna página. De esta manera se proporciona al administrador de TI una interfaz rica y una experiencia de usuario fluida. Además, los SPA hacen que los usuarios sientan que interactúan con una aplicación de escritorio.

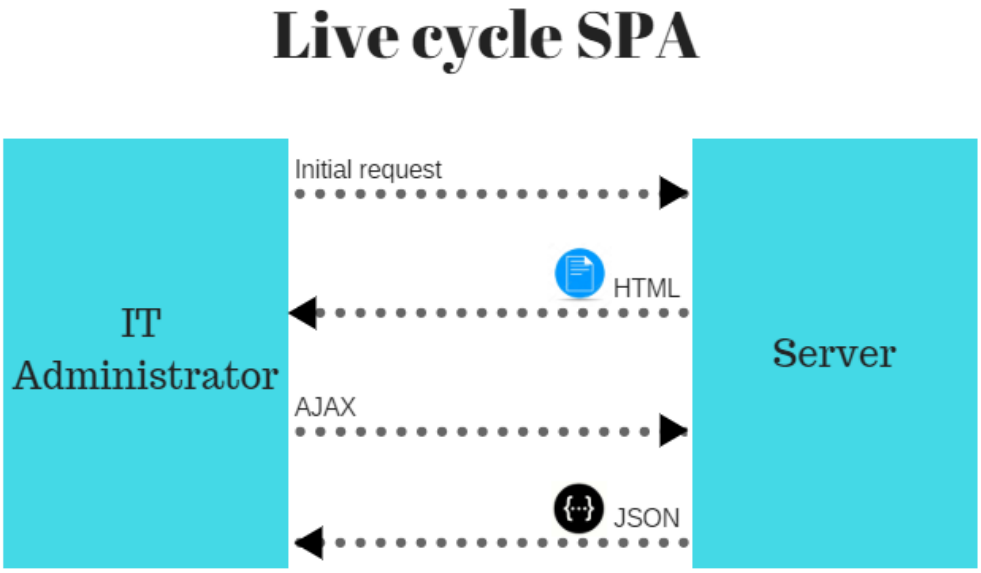


Ilustración 14. Ciclo de vida de una aplicación web SPA

(Fuente: Elaboración propia)

7.3.5. MEAN stack

En este apartado se mencionarán los frameworks y herramientas utilizadas para el desarrollo para el componente de la base de datos, el back-end y el Font-end de la plataforma; fueron elegidos por su uso, estabilidad y rendimiento:

- **MongoDB:** base de datos NoSQL.
- **ExpressJS:** framewotk para crear aplicaciones web de una o varias páginas en Node.js
- **AngularJS:** framework para el desarrollo rápido de front-end.
- **Node.js:** entorno de ejecución de JavaScript del lado del servidor creado en el tiempo de ejecución de JavaScript V8 de Google Chrome.

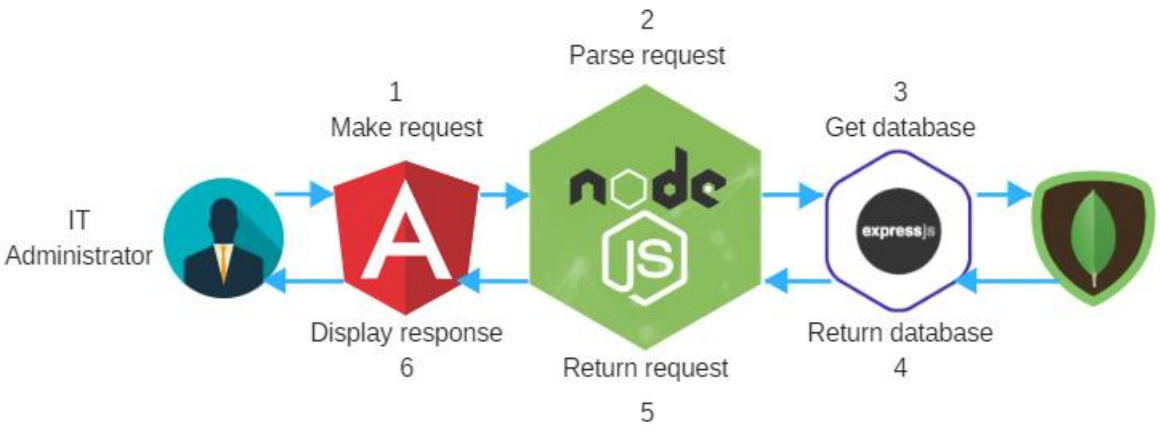


Ilustración 15. Interacción Mean Stack

*(*Fuente: Elaboración propia)

Se eligió MongoDB, pues ha sido la solución para construir bases de datos sin una idea básica de un diseño de software. Permite una prueba y un error sencillos ya que no es necesario definir esquemas para diseñar la base de datos. Esto nos ayudó a almacenar y administrar datos con facilidad. Además, dado que MongoDB admite la transferencia de datos a través del formato JSON, la transferencia de datos desde la aplicación web es fácil y económica. JSON también permite la transmisión de datos del servidor cliente

fácil. Además, MongoDB le permite utilizar un solo idioma para toda la aplicación web en lugar de fragmentar entre múltiples lenguajes de programación.

En cuanto a tecnología del lado del servidor se optó por Express dado que esta herramienta maneja cada parte del servidor y oculta la mayor parte del funcionamiento interno de Node, los desarrolladores no necesitan prestar atención a la tecnología de servidor adicional. Los desarrolladores pueden elegir las bibliotecas que necesiten para una tarea específica que les proporcione adaptabilidad y una gran personalización. En MEAN, Express nos ayudó como un medio para transferir solicitudes del cliente a la base de datos y envía respuestas de la base de datos al cliente como se muestra en la *Ilustración 15*.

Para el front-end se decidió utilizar Angular JS que es un framework del lado del cliente mantenido por Google. Dicho framework nos proporcionó toda la funcionalidad para manejar la información del cliente en el navegador, controlar la información en el lado del cliente y manejar cómo se muestran los componentes en la vista del navegador.

Node js es el entorno en tiempo de ejecución que se eligio, ya que es un entorno multiplataforma creado en el motor de JavaScript V8 de Chrome. Además, utiliza un modelo de E / S no bloqueado y controlado por eventos que lo hace liviano y eficiente. Su ecosistema de paquetes, npm, es el ecosistema más grande de bibliotecas de código abierto en el mundo. En la pila MEAN, Node actúa como una plataforma del lado del servidor que es similar al sistema Apache que se ejecuta con un lenguaje de programación de servidor para crear aplicaciones.

7.3.6. Las vistas

Las vistas son el conjunto de interfaces que el usuario puede visualizar al usar la aplicación accesible desde un navegador web. Las interfaces del software fueron diseñadas para brindar una experiencia amigable e intuitiva que le permita al usuario una fácil navegación por el aplicativo. A continuación, se mostrarán las distintas

interfaces de usuario diseñadas para la aplicación web además de las distintas funcionalidades que estas tienen.

Interfaz de login

La primera interfaz que se ejecuta en el aplicativo web es la de inicio de sesión, en la que el usuario debe proporcionar su id y contraseña suministrada por el administrador. Como es un servicio para clientes específicos no se implementó una interfaz de registro y se le delego esa función a una única cuenta de administrador.

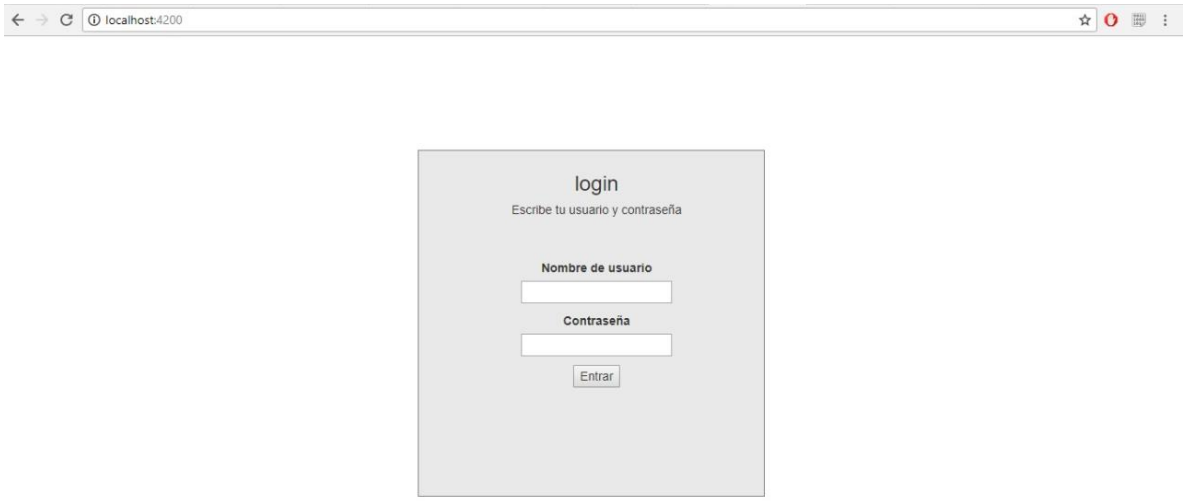


Ilustración 16. Interfaz de login

(Fuente: Elaboración propia)

Ilustración 17. Interfaz de inicio

(Fuente: Elaboración propia)

Interfaz de reportes

Para revisión de los reportes se escoge el icono de carpeta en la parte superior, la interfaz de la *ilustración 18* cuenta con una tabla donde el usuario podrá observar información general de los ataques reportados por los honeypots que la cuenta de administrador le asigno, en caso de querer observar más a detalle. Además, cuenta con filtros en parte superior de la tabla para realizar búsquedas por nombre del honeypot. Esta interfaz está disponible para las cuentas de usuario como la cuenta de administrador.

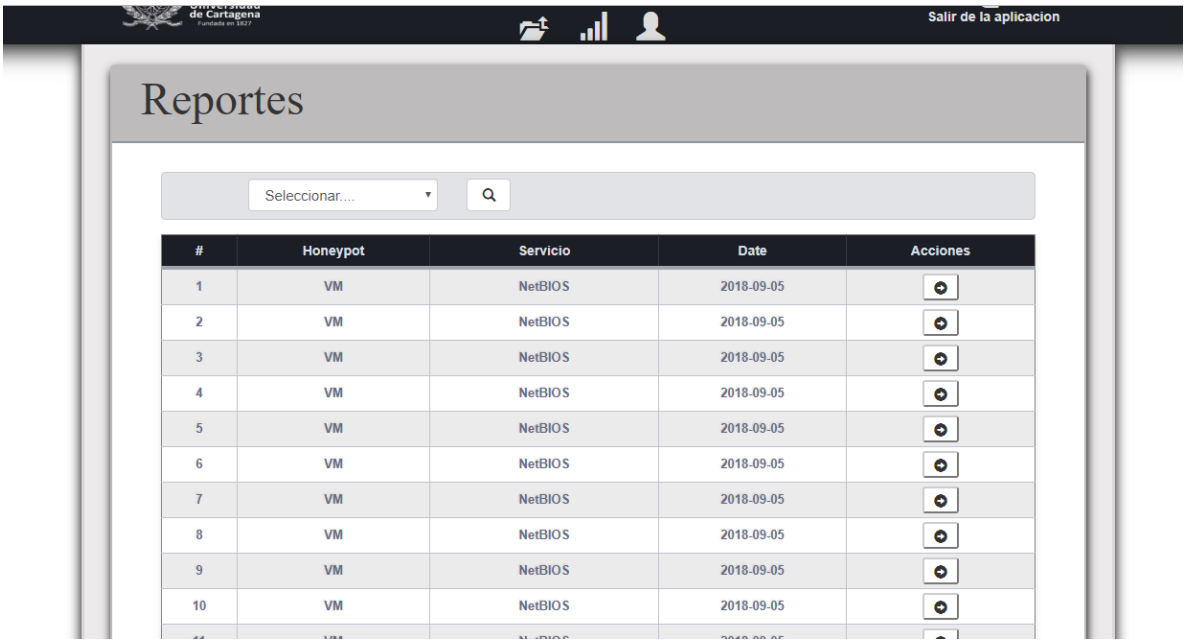


Ilustración 18. Interfaz de reportes

(Fuente: Elaboración propia)

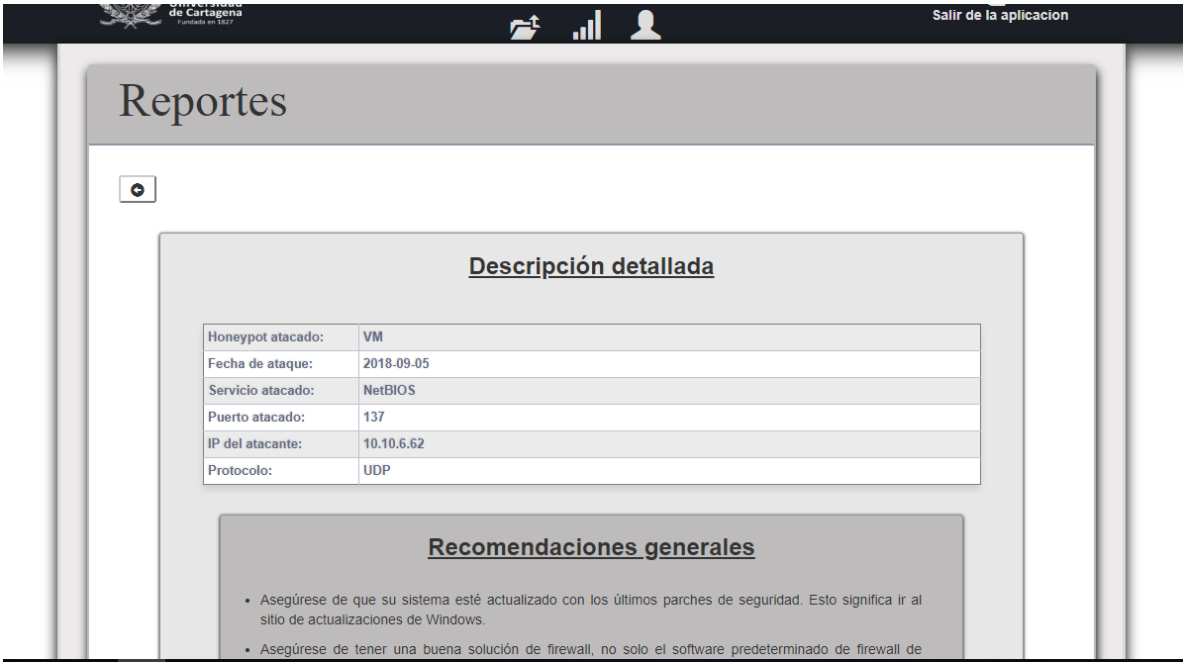


Ilustración 19. Interfaz de reporte individual

(Fuente: Elaboración propia)

Interfaz de estadísticas

Desde la interfaz que se muestra a continuación el usuario puede revisar estadísticas relacionadas con el servicio más ataca o los hosts remotos que más ataques envían. Esta interfaz está disponible para las cuentas de usuario como la cuenta de administrador.

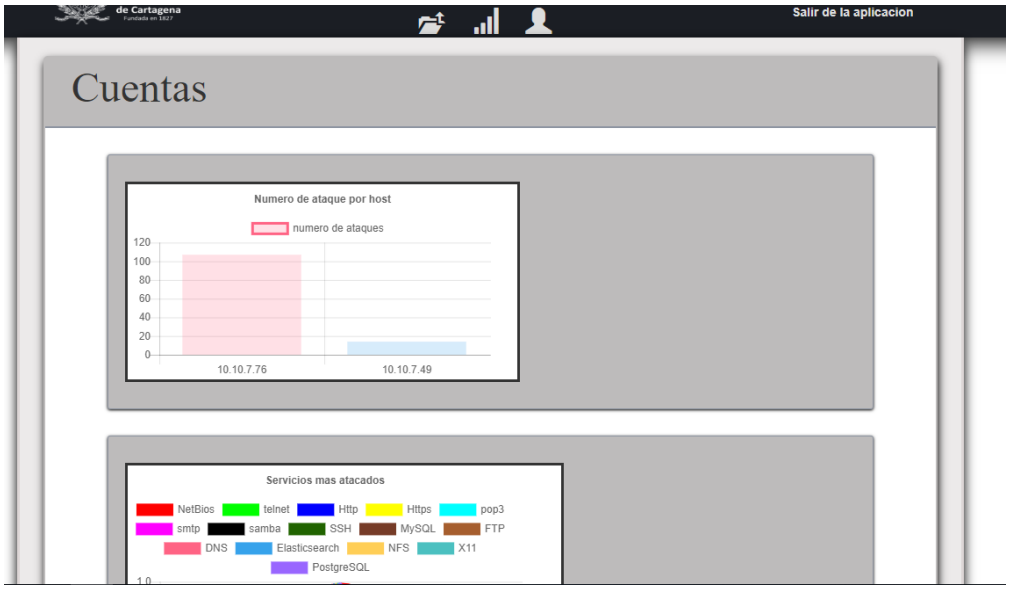


Ilustración 20. Interfaz de estadísticas

(Fuente: Elaboración propia)

Interfaz de administración de cuentas

A continuación, se muestra una interfaz que es solo disponible por la cuenta de administrador y permite la creación y eliminación de cuentas de usuarios además de podés asociar los honeypots conectados al sistema a un usuario específico para que pueda ver los reportes de dicho honeypot.

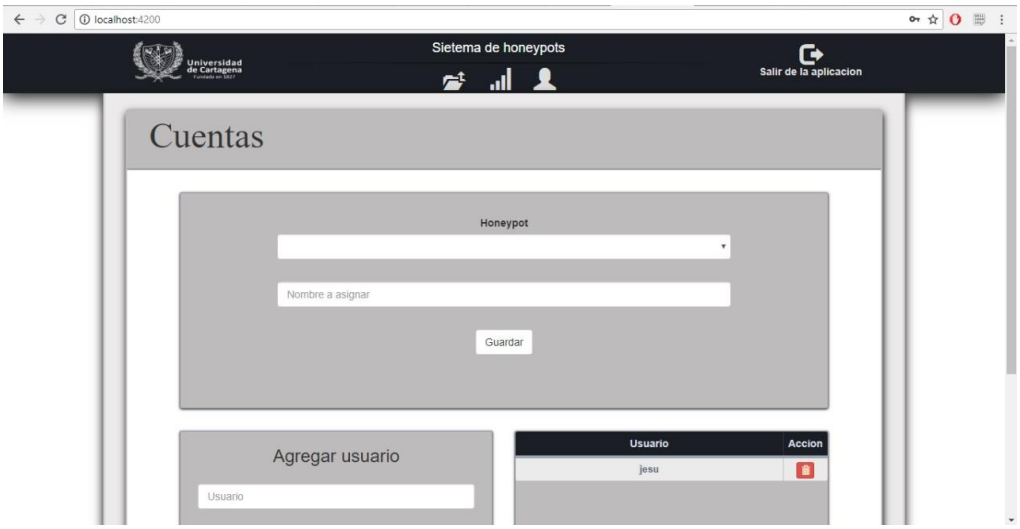


Ilustración 21. Interfaz de manejo de cuentas y honeypots

(Fuente: Elaboración propia)

7.4. Pruebas de funcionalidad

Este capítulo describe el cumplimiento del objetivo específico 4. Se realizaron dos ataques a servicios del honeypot para evaluar el sistema de detección a redes de datos empresariales que permitan determinar si los pasos que se han realizado anteriormente están correctos, si los componentes instalados trabajan bien en conjunto y en general, comprobar si la Honeypot funciona adecuadamente.

7.4.1 Escenario de pruebas

Para evaluar la eficiencia del sistema de detección se realizó pruebas de concepto para verificar que, si realizamos un ataque al ordenador que contiene el honeypot ya configurado y funcionando, este presentara la información en la plataforma web y pueda advertir al encargado de seguridad de cualquier red de datos empresarial.

Dichos ataques se realizarán en la Red LAN de los laboratorios de la universidad de Cartagena, utilizando 2 ordenadores proporcionados por los autores, uno con sistema operativo Windows 10 (dirección IP: 10.10.7.40) y será el encargado de correr el Back-end del sistema y también revisar los ataques reportados por el sistema de detección y el

otro ordenador tiene un sistema operativo Linux (Debían 9, Dirección IP:10.10.7.49) y será el encargado de ejecutar el honeypot y realizar los ataques.

A continuación, se mencionarán los ataques realizados al sistema y una ilustración del escenario de ataque:

1. Escaneo de puertos con la herramienta Nmap
2. Conexión al servicio Telnet por medio de la herramienta Putty.

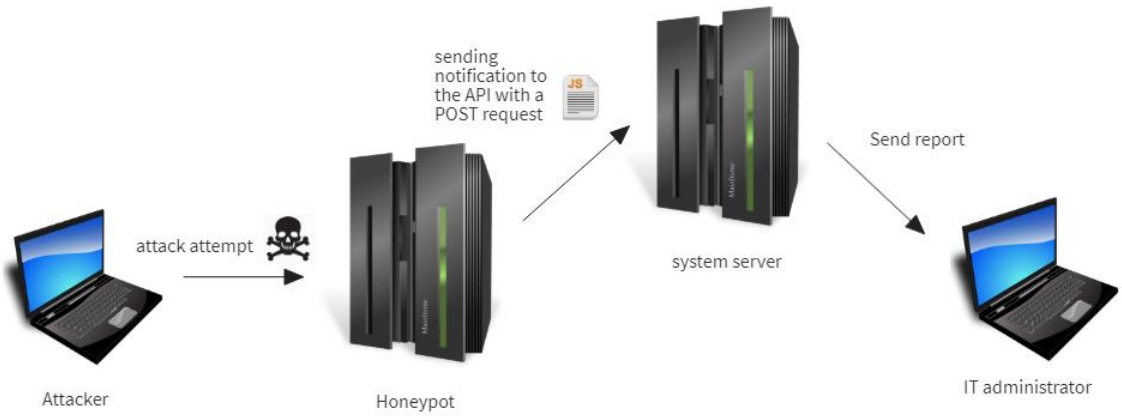


Ilustración 22. Esquema general del escenario de prueba

(Fuente: Elaboración propia)

7.4.2. Ataques realizados

En primer lugar, se realiza una configuración del honeypot utilizando las indicaciones del Anexo 1, seguidamente se inicia el honeypot, al ser primera vez que se inicia se envía de manera automática una petición de vinculación al sistema de detección, el cual se utiliza como mecanismo de seguridad para que el administrador del sistema pueda verificar si es un honeypot que él instalo con antelación.

La IP del honeypot que realizo la petición es agregada a una lista de pendientes por aceptar, y hasta que el administrador no acepte la petición de vinculación los reportes que esta manda no serán mostrados en la plataforma web, de tal manera que la primera acción que realizamos en la plataforma del sistema es aceptar la petición de vinculación.

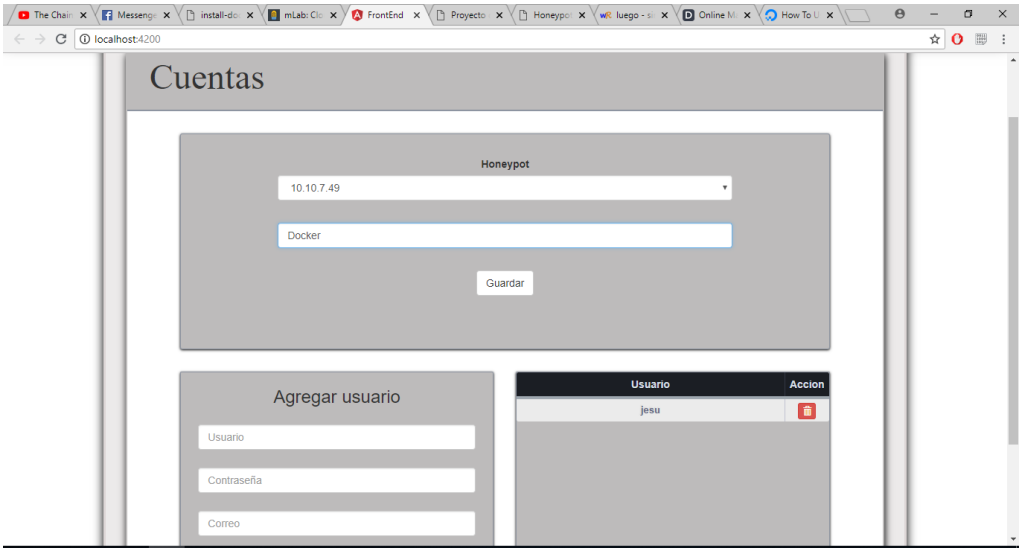


Ilustración 23. Vinculación de honeypot al sistema

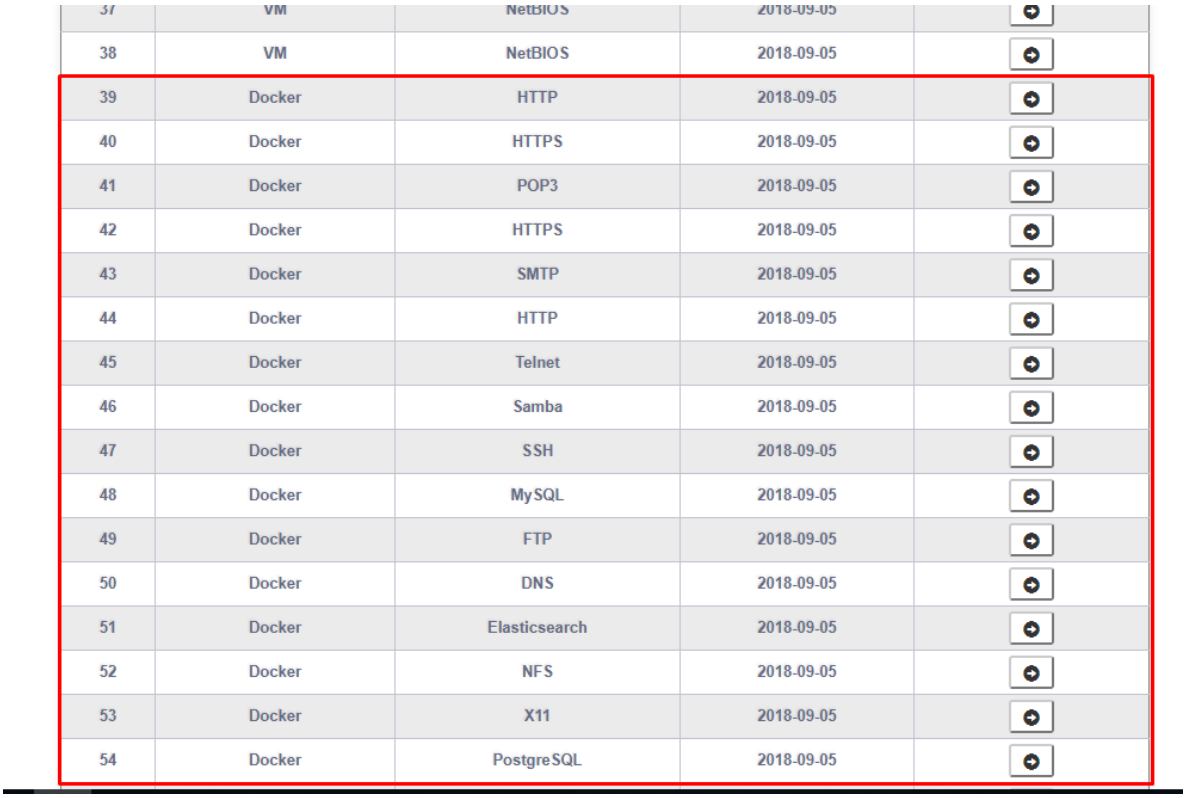
(Fuente: Elaboración propia)

Como se muestra en la *Ilustración 23*, para hacer efectiva la vinculación buscamos la IP en la lista de honeypots y le asignamos un nombre o alias para que sea más fácil de reconocer, en este caso el nombre que asignamos es “Docker”. A continuación, realizamos un ataque de prueba utilizando la herramienta Nmap.

Ilustración 24. Resultado mapeo de puertos abiertos con protocolo TCP del honeypot

(Fuente: Elaboración propia)

Nmap envía una solicitud de vinculación a todos los puertos que utilicen protocolo TCP y los que manden una respuesta son los que se muestran en los resultados de la *ilustración 24.* Tras realizar el ataque, procedemos tras esto a verificar que la conexión ha sido reportada a la plataforma web. Para esto buscamos en la sección de reportes que ataques se realizaron.



37	VM	NetBIOS	2018-09-05	
38	VM	NetBIOS	2018-09-05	
39	Docker	HTTP	2018-09-05	
40	Docker	HTTPS	2018-09-05	
41	Docker	POP3	2018-09-05	
42	Docker	HTTPS	2018-09-05	
43	Docker	SMTP	2018-09-05	
44	Docker	HTTP	2018-09-05	
45	Docker	Telnet	2018-09-05	
46	Docker	Samba	2018-09-05	
47	Docker	SSH	2018-09-05	
48	Docker	MySQL	2018-09-05	
49	Docker	FTP	2018-09-05	
50	Docker	DNS	2018-09-05	
51	Docker	Elasticsearch	2018-09-05	
52	Docker	NFS	2018-09-05	
53	Docker	X11	2018-09-05	
54	Docker	PostgreSQL	2018-09-05	

Ilustración 25. Reportes resultantes del de la conexión a los servicios del honeypot

(Fuente: Elaboración propia)

Como podemos ver, las conexiones han sido indexadas al sistema de detección de ataques y presentadas en la plataforma web. Con esto finaliza el primer ataque ahora al honeypot y procedemos realizamos el segundo ataque, nos conectarnos con el servicio telnet mediante la herramienta Putty y hacemos login con cualquier nombre de usuario y contraseña “root”.

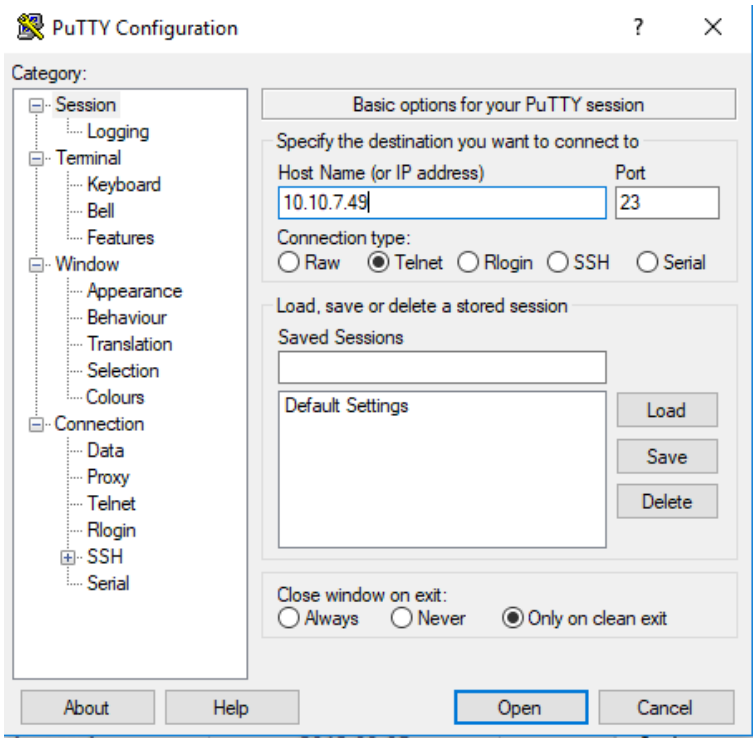


Ilustración 26. Realizando conexión telnet con el honeypot

(Fuente: Elaboración propia)

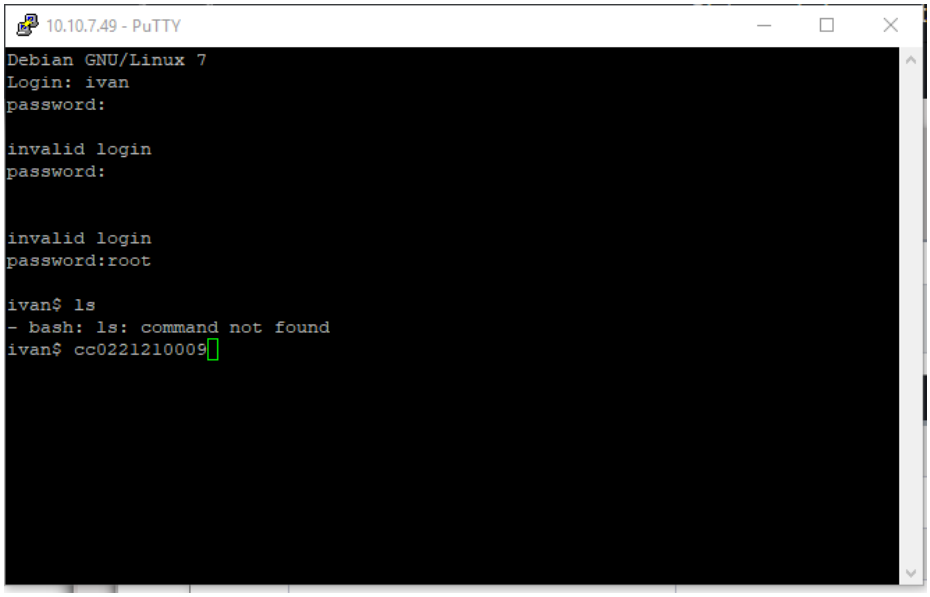


Ilustración 27. Conexión exitosa al servicio telnet

(Fuente: Elaboración propia)

Con esto habríamos simulado una conexión telnet de un atacante, ya que nadie debería conectarse al telnet de la máquina del honeypot. A continuación, revisaremos si el ataque fue reportado.

53	Docker	X11	2018-09-05	
54	Docker	PostgreSQL	2018-09-05	
55	Docker	Telnet	2018-09-05	
56	Docker	Telnet	2018-09-05	
57	Docker	Telnet	2018-09-05	
58	VM	NetBIOS	2018-09-05	
59	VM	NetBIOS	2018-09-05	
60	VM	NetBIOS	2018-09-05	
61	VM	NetBIOS	2018-09-05	
62	VM	NetBIOS	2018-09-05	
63	VM	NetBIOS	2018-09-05	
64	VM	NetBIOS	2018-09-05	
65	VM	NetBIOS	2018-09-05	
66	VM	NetBIOS	2018-09-05	
67	VM	NetBIOS	2018-09-05	
68	VM	NetBIOS	2018-09-05	
69	VM	NetBIOS	2018-09-05	
70	VM	NetBIOS	2018-09-05	

Ilustración 28. Reportes de ataque servicio telnet

(Fuente: Elaboración propia)

La *ilustración 28* nos muestra que los reportes fueron registrados con éxito en la plataforma web, pero también se puede apreciar que otros ataques provenientes del honeypot con el nombre "VM" (virtual machine) ha registrado varios ataques al servicio NetBIOS, lo cual es un evento inesperado y que vale la pena investigar.

Procedemos a revisar más detalles de los reportes y las estadistas para saber cuántos ataques a este servicio se han realizado.

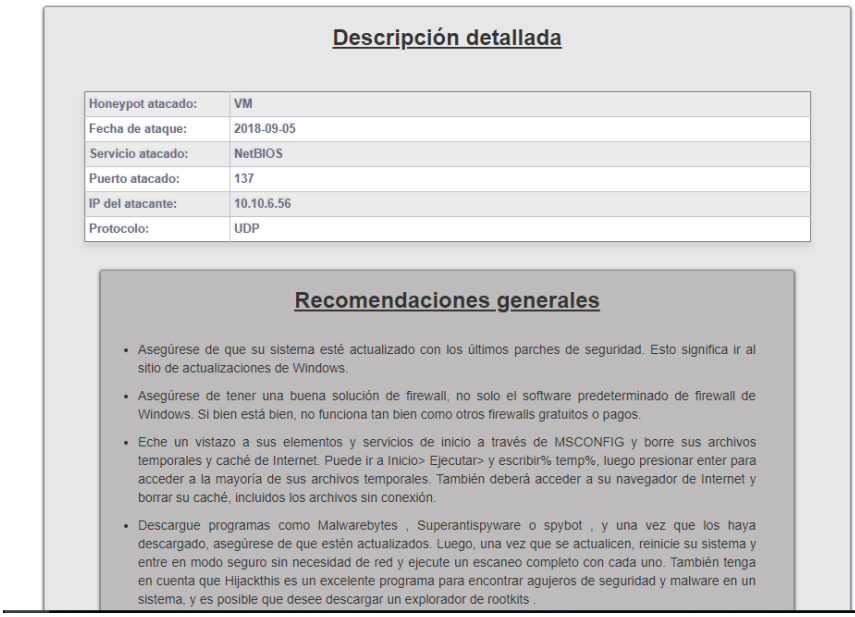


Ilustración 29. Reporte de conexión a servicio NetBIOS del honeypot VM

(Fuente: Elaboración propia)

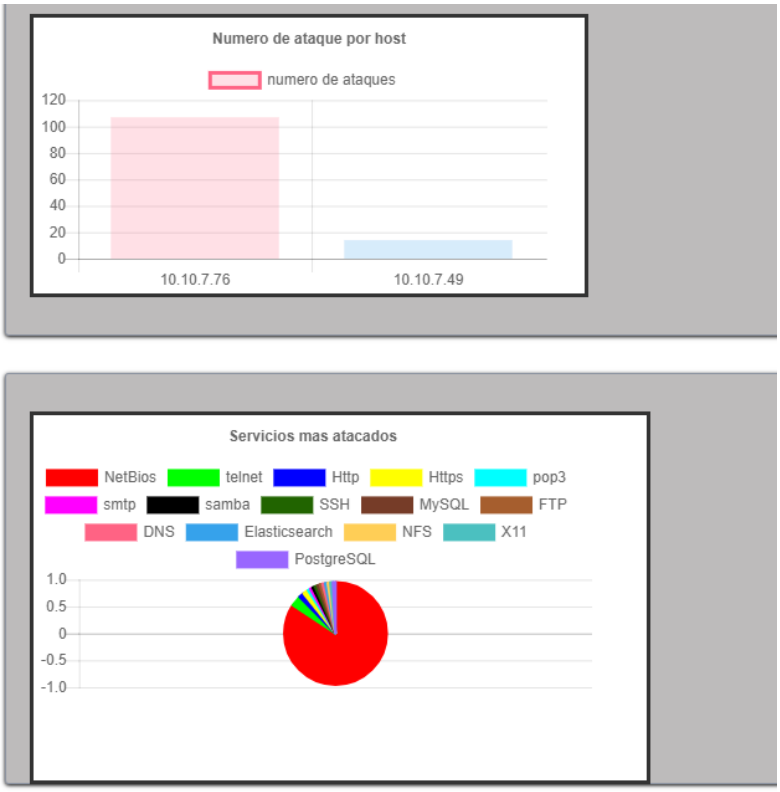


Ilustración 29. Estadísticas de ataques

(Fuente: Elaboración propia)

Las estadísticas de la *ilustración 29* en primer lugar, si analizamos el servicio más atacado podemos observar como de manera clara el servicio NetBIOS es el más atacado, con un 84.29% del tráfico registrado, el resto del trafico fueron ataques realizados en la prueba.
En segundo lugar, procedemos a analizar el origen de los ataques podemos observar que es el host 10.10.7.76, al revisar esa máquina con el host mencionado se logra ver que un ordenador con sistema operativo Windows y que te tiene activada la opción de "compartir redes", esto hace que constantemente escanean la red usando mensajes broadcast y estos deben responder con información como el nombre del equipo, IP, servicios que comparte, etc. El honeypot que teníamos instalado recibió dicha petición para obtener información del ordenador lo cual conllevo a que se realicen los respectivos reportes.

Para terminar esta sección de pruebas, queremos destacar la diferencia de analizar un reporte presentado por el honeypot y un reporte presentado por el sistema de detección.

Ilustración 30. Resultado de reporte de ataque filtrado y detallado.

(Fuente: Elaboración propia)

Ilustración 31. Resultado de reporte del honeypot

(Fuente: Elaboración propia)

8. CONCLUSIONES

La información como fuente primaria de conocimiento y principal activo de la sociedad, las organizaciones y cualquier ente en general, representan un tesoro de gran valor por el cual deben implementarse cualquier tipo de estrategias que permitan, además de aprovechar y explotar al máximo su contenido dentro de los límites establecidos, evadir y prevenir daños sobre la información.

La cantidad de amenazas y vulnerabilidades que están presentes en el medio y a las cuales es altamente susceptible la información, deben mitigarse y limitarse, de manera tal que se procure la erradicación y eliminación de cualquier fuga o mala manipulación de esta. Por lo anterior se manifestó esta idea de proyecto de grado cuyo objetivo fue identificar falencias que se presentan en los sistemas de información y posterior a ello ahondar en los esfuerzos por proteger la información contra cualquier ente indeseado con posibles fines criminales, como aquellos que en la historia han dejado evidencia de su mal aprovechamiento.

El proceso inicio con un estudio previo, la herramienta en la cual soportamos el sistema de detección llamado honeypot el cual constituye el marco de referencia de nuestro proyecto, el estudio consistió en una investigación de los orígenes del honeypot, sus antecedentes, su clasificación, su arquitectura, su configuración y búsqueda de los sistemas que ya poseían estas implementaciones como por ejemplo Honeydrive. Este

estudio previo dio como resultado los requisitos funcionales del sistema cumpliendo así el primer objetivo específico de la investigación.

Luego de identificados los requisitos, se propuso la arquitectura del sistema siguiendo el modelo 4+1 view; se realizaron las 4 vistas principales (lógica, despliegue, procesos y física), dentro de este paso se definieron también las tecnologías a utilizar, teniendo en cuenta cuales eran las más convenientes y la facilidad al implementarlas permitiendo esto que se pensara más en cumplir los requisitos y no en la implementación; dando así por cumplido el segundo objetivo específico.

Siguiendo el modelo RUP se realizó la implementación del sistema (tercer objetivo específico), desarrollando la plataforma por iteraciones de acuerdo a cada funcionalidad planteada en el diseño; las iteraciones tomaban una semana cada una e iban en paralelo tanto la de la plataforma web como la configuración del honeypot, de manera que así se fueran encontrando y corrigiendo errores mientras se realizaba la interacción web-honeypot.

En cuanto a los resultados obtenidos, hay que decir que no solo se han obtenido resultados esperados, sino que también se han obtenido resultados inesperados como los evidenciados en la prueba de funcionabilidad, varios reportes cuya procedencia no venía de los ataques controlados que realizo el equipo de trabajo sino que provenía de equipos de la misma red para conectarse al servicio NETBIOS, lo cual se llegó a la conclusión que no es un ataque ya que el protocolo NETBIOS hace que todos los computadores de la red manden mensajes broadcast a todos los computadores para conocer los equipos a los que están conectados.

El resultado inesperado anteriormente mencionado tiene que ver con lo expuesto en uno de los artículos mencionados en esta investigación "*Capture and Analysis of Malicious Traffic in VoIP Environments Using a Low Interaction Honeypot*" (Jordao da Silva Vargas & Kleinschmidt, 2013) , el articulo menciona que las conexiones que se realizan

al honeypot pueden ser consideradas como un ataque, ya que este dispositivo no debe recibir ninguna conexión y por lo tanto el riesgo de tener un falso positivo es nulo.

Este sistema de detección muestra una visión innovadora ya que brindara información procesada, más simple y grafica de las actividades de los intrusos que ingresen a la red, lo cual facilitara la toma de medidas preventivas sobre futuros atacantes , además de poder actualizar las políticas de seguridad para evitar replicas o ataques con patrones similares en las organizaciones de nuestra comunidad; facilitando de forma eficiente y eficaz la gestión de la seguridad de la información que se transporta tanto interna como de manera externa por una red de datos.

Dando respuesta a la pregunta principal de la investigación: ¿Cómo apoyar la labor de los administradores de infraestructura de TI (Tecnologías de la Información), para identificar amenazas a la seguridad del sistema, de tal manera que puedan generar estrategias de protección de forma oportuna? Proponiendo la utilización de la tecnología Honeypot como complemento, con el fin de que esta herramienta proporcione la información necesaria para aprender de los intrusos y ayude a los administradores de TI a determinar las posibles vulnerabilidades, para posteriormente emprender mejores estrategias de protección.

Por lo tanto, podemos decir que se este trabajo ha cumplido los objetivos y propuestas establecidas en un principio, llegando a ser útil para darnos cuenta de la importancia de la seguridad en una red.

9. RECOMENDACIONES.

Si bien la percepción general de la implementación del sistema de detección de ataques informáticos a redes de datos empresariales fue positiva, queda claro que el producto final que arroja el proyecto de grado es solo un producto mínimo viable. A lo largo de este proceso se han detectado varios puntos que no han podido gestionarse en este pero

que podrán desarrollarse en un futuro. A continuación, se listarán estos puntos para posibles trabajos futuros:

- Mejorar los plugins para incrementar la interacción con los atacantes: Muchos de los servicios tienen una interacción baja con los atacantes, por lo que será interesante desarrollar los plugins para obtener más información sobre los ataques. El primero de los plugins que se deberían desarrollar es el plugin web, ya que actualmente ofrece una interacción muy baja y es uno de los servicios que nos pueden ofrecer más información.

- Mejoras del plugin SIP: El plugin SIP es muy importante debido al alto número de servicios SIP abiertos por el operador. Mejorando este podríamos además de realizar bloqueos por IPs, obtener información de destinos de las llamadas para aplicar reglas sobre estos, por ejemplo, limitar el tráfico a estos destinos por sospecha de tráfico fraudulento.

- Añadir más recomendaciones a la base de datos: Ahora mismo contamos con recomendaciones universales pero los atacantes todos los días van actualizando sus esquemas de ataque a redes empresariales por tanto mejorar e incrementar la información de las recomendaciones base de datos.

- Protección antiDDoS: Al crear un servicio con tantos puertos susceptibles de ser atacados, aumentamos el riesgo de sufrir un ataque de denegación de servicio distribuido. Por esta razón, sería conveniente configurar un sistema de detección que en caso de detectar una cantidad muy alta de tráfico lanzara de manera automática una configuración BGP que envíe a los operadores de entrada la IP del honeypot para ser configurada en el blackhole y así detener el ataque. En caso de activarse este sistema de seguridad, el honeypot desaparecería de Internet.

10. REFERENCIAS

Alcantara, P., & Riglietti, G. (2016). Horizon Scan Report 2016. *Business Continuity Institute*, 32.

Bird, L., & Kerr, H. (2014). Horizon Scan 2014. *Business Continuity Institute*, 36.

Clough, T., & Chaplygin, R. (2016). The Global State of Information Security® Survey 2016. *PWC*, 30.

Dongxia, L., & Yongbo, Z. (2012). An Intrusion Detection System Based on Honeypot Technology. *2012 International Conference on Computer Science and Electronics Engineering* (pág. 4). China: Computer Science and Electronics Engineering.

Drnevich, P., & Croson, D. (2013). INFORMATION TECHNOLOGY AND BUSINESS-LEVEL STRATEGY: TOWARD AN INTEGRATED THEORETICAL PERSPECTIVE . *MIS Quarterly*, 28.

Eduardo, A., & Daniel, L. (2013). *Honeypot: Ventajas y desventajas como mecanismo para la prevencion de intrusos informaticos.* Bogota.

Gonzalez, D. (2003). *Sistema de deteccion de intrusos.* madrid: the freesoftware fondation.

IBM. (1998). *IBM*. Obtenido de IBM: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

INCIBE. (23 de 03 de 2017). *certsi*. Obtenido de certsi: www.certsi.es/blog/honeypots-industriales

Jordao da Silva Vargas, I. R., & Kleinschmidt, J. H. (2013). Capture and Analysis of Malicious Traffic in VoIP Environments Using a Low Interaction Honeypot. *IEEE Latin America Transactions*, 32.

Kalma Rahmatullah, D., Michrandi Nasution, S., & Azmi, F. (2016). Implementation of low interaction web server honeypot using cubieboard. *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)* (pág. 14). Bandung: IEEE.

Kankanhalli, A., Teo, H.-H., C.Y, B., & Wei, K.-K. (2013). An integrative study of information systems. *PERGAMON*, 154.

Kaur, T., Malhotra, V., & Singh, D. (2014). Comparison of network security tools- Firewall Intrusion Detection System and Honeypot. *International Journal of Enhanced Research in Science Technology & Engineering*, 5.

Levine, J., Owen, H., Grizzard, J., Lee, W., & Dagon, D. (2010). HoneyStat: Local Worm Detection Using Honeypots. *Georgia Institute of Technology*, 20.

Martínez, K. (2018). *HONEYPOT, HACIA UN PROTOCOLO DE SEGURIDAD MÁS EFICIENTE Y COMPETITIVO.* Cartagena de indias.

Project, T. H. (2001). *Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community*. nueva york: Paperback.

Qasim Ali, M., Al-Shaer, E., & Samak, T. (2014). Firewall Policy Reconnaissance: Techniques and Analysis. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 2*, 13.

Rajkumar Sethi, A., Amin, S., & Schwartz, G. (2017). Value of intrusion detection systems for countering energy fraud. *American Control Conference 2017* (pág. 12). Seattle: IEEExplore.

Sembiring, I. (2016). Implementation of Honeypot to Detect and Prevent Distributed Denial of Service Attack. *Conf. on Information Tech., Computer, and Electrical Engineering (ICITACEE)*, (pág. 6). Semarang.

Sobrado, P. (02 de 05 de 2017). *emergya*. Obtenido de emergya: https://www.emergya.com/blog/qa/2017/05/02/execute-appium-on-docker/

Spitzner, L. (2002). *Honeypots: Tracking Hackers.* chicago: Addison Wesley.

Stoll, C. (1989). *THE CUCKOO'S EGG.* Nueva york: Doubleday.

Winn, M., Rice, M., Dunlap, S., Lopez, J., & Mullins, B. (2015). Constructing cost-effective and targetable industrial control system honeypots for production networks. *international journal of critical infrastructure protection*, 51.

11. ANEXOS

11.1. Instalación del honeypot

La instalación del honeypot es recomendable realizarla en un ordenador con sistema operativo Debían versión 9. Se comienza la instalación del honeypot agregando la librería "curl" el cual es in interprete de comandos orientados a la transferencia de archivos.

Al abrir el intérprete de comandos se comprueba si el comando comienza con el símbolo #, lo cual significa que los comandos se están ejecutando como usuario root, de no ser así se procede a acceder como usuario root y se escribe el siguiente comando:

```
# apt-get install curl -y
```

Y después se procede a instalar Docker:

```
# curl -fsSL https://get.docker.com -o get-docker.sh
# sh get-docker.sh
# systemctl enable docker
# systemctl start docker
```

Una vez instalado Docker se procede a instalar Docker-compose el cual es una herramienta para definir y ejecutar Docker de múltiples contenedores (La versión 1.22.0 es la usada ahora).

```
# curl –L
https://github.com/docker/compose/releases/download/1.22.0/docker-
compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
# chmod +x /usr/local/bin/docker-compose
```

A continuación, se descomprime el archivo que contiene el honeypot y entra en su repositorio.

```
# unzip HoneyPy-Docker.zip -d .
# cd HoneyPy-Docker
```

Aquí se debe editar etc/honeypy.cfg con la IP, puerto y URL del api para que quede así:

```
[elasticsearch]
enabled = Yes
# Elasticsearch url should include ":port/index/type
# example: http://localhost:9200/honeypot/honeypy
es_url  = http://10.10.7.40:4100/api/catcherofreports
```

Aquí iniciamos el honeypot, vemos sus logs (Para salir, presionar Control+C) y detenemos el honeypot, **IMPORTANTE** Antes de iniciar el honeypot, verificar que el servidor no está utilizando ninguno de los puertos del honeypot descritos en `etc/services.cfg`; Esto puede verse al ejecutar `docker-compose up` como root. Si no aparece nada, no hay conflictos y puede terminarse el proceso con `Control+C`.
En caso de haber conflictos, quitar el servicio que usa el puerto en conflicto en el servidor o de la lista de servidos en el archivo `etc/services.cfg`.

```
# docker-compose up -d
# ./log.sh
# docker-compose down
```