Arquitectura DMZ Perimetral: Una implementación corporativa

Danis Azizov, Universidad Autónoma de Barcelona, Bellaterra

Resumen—La seguridad informática es un elemento esencial en el campo empresarial. Disponer de una infraestructura robusta y segura es un requisito importante debido a la naturaleza hostil de Internet. En este artículo se explorará uno de los modelos de arquitectura de red más conocidos, que será complementada con herramientas que permitan mejorar la seguridad. Veremos cómo gracias a estos componentes, junto a una infraestructura adecuada, se propone una solución asequible a aquellas empresas emergentes que establecen el primer contacto con la red global.

Abstract— Computer security is an essential element in the business field. Having a strong and secure infrastructure is an important requirement due to the hostile nature of the Internet. This article will explore one of the best-known network architecture models, which will be complemented by tools that improve security. It will be seen how thanks to these components and an adequate infrastructure, an affordable solution is proposed to those emerging companies that establish the first contact with the global network.

Índice de Términos—Seguridad, Arquitectura de red, HoneyPot, Cortafuegos, Sistema de detección de Intrusos, reglas de Cortafuegos, Snort.

1 Introducción

En la web se producen numerosos ataques constantemente, por lo que la seguridad informática se ha vuelto un componente esencial tanto el entorno personal como empresarial. Los ataques son cada vez más sofisticados y numerosos, tal que como respuesta se llevan a cabo el desarrollo de métodos, plataformas y sistemas de seguridad cada vez más complejos y avanzados para contener estas amenazas, que se ha convertido en una característica de la propia red.

Todas las empresas que posean plataformas online propias son afectadas por estos ataques, por lo que parte de su presupuesto debe dedicarse a la seguridad en la red. Aunque existan soluciones denominadas *Cloud-based Web Applitacion Firewall* donde se proporciona la seguridad como un servicio, su coste puede ser elevado para algunas empresas. En este trabajo se plantea una solución, llegando a implementar una infraestructura necesaria para que el coste principal no sea monetario sino de mantenimiento.

Un diseño de red bien escogido, planteado y definido, que será acompañado con diversos dispositivos de seguridad como Cortafuegos, Honeypots y Sistemas de Detección de Intrusos (*Intrusion Detection System*) nos pueden proporcionar la seguridad necesaria empleando no más de los recursos necesarios para desplegarnos en internet.

El resto de este documento está organizado de la siguiente manera. La sección dos explica el escenario propuesto y los objetivos de este proyecto. En La Sección tres se sitúa el contexto del trabajo. La sección cuatro analiza los componentes que se usarán en el proyecto y las arquitecturas más usadas. La Sección cinco detalla la planificación llevada a cabo en el proyecto. La Sección seis detalla la implementación de la infraestructura llevada a cabo. En la Sección siete se muestra la evaluación de la implementación. La Sección ocho presentan la discusión y las conclusiones del documento. Y finalmente en la Sección nueve se realiza una breve propuesta para continuar el proyecto.

- *E-mail de contacto: danis.160296@gmail.com*
- *Mención Realizada: Tecnologías de Información y Comunicación*
- *Trabajo Supervisado por: Aitor Alsina*
- *Año 2019/2020*

2 Análisis de Requerimientos

El primer paso es determinar el escenario y el dominio del problema ante cual nos encontramos. No existe una resolución universal, por lo que se deben determinar algunos de los factores, entre los cuales están los recursos necesarios para desplegar la red, la carga del tráfico y el tiempo disponible para llevar a cabo el proyecto.

2.1 Planteamiento

El escenario para el cual se desea plantear nuestra solución está enfocado hacía el sector corporativo, sobre todo aquellas empresas creadas recientemente, con capital y recursos limitados. Estos, generalmente tienen la necesidad de implementar una red corporativa o *intranet,* y se contempla la posibilidad de despliegue de un servidor web propio, sin recurrir al alojamiento en empresas de terceros. Siendo una oficina única, concentraremos las comunicaciones dentro de la red, sin la necesidad de implementar métodos para la comunicación y acceso externo.

2.2 Objetivos

Nuestro objetivo principal es el despliegue de una infraestructura de red que disponga de las características

necesarias para ser aplicable a nuestro escenario, siendo viable en un entorno real. Para ello, se plantean los siguientes objetivos a cumplir, en orden de prioridad:

1. Determinar la Arquitectura necesaria para el proyecto.
2. Determinar la composición de la Arquitectura, que elementos de seguridad serán integrados.
3. Familiarizarse con el entorno y las herramientas que se usarán.
4. Determinar las reglas para asegurar el entorno, monitorizar el tráfico e implementar dispositivos señuelo.
5. Verificar y Comprobar el diseño mediante pruebas de funcionalidad y simulaciones de ataque.

3 ESTADO DEL ARTE

Existen múltiples alternativas que puedan simplificar u omitir por completo el trabajo de implementación de arquitecturas de forma manual. En ciertos escenarios, donde interesa un despliegue rápido, con alta disponibilidad y seguridad, además de disponer de suficientes recursos económicos, podemos contratar servicios de alojamiento o servicios de computación en la nube (*Cloud as a Service* [1]). Corporaciones como Amazon, Google y Microsoft dan abasto a ese mercado, ofreciendo un servicio flexible, versátil con una disponibilidad casi completa, hasta un 99.95% [2]. Sin embargo, muchos no confían en alojar su negocio en manos de terceros comprometiendo los datos empresariales. Por lo que éstas optan por el desarrollo de sus propias infraestructuras, compuestas por múltiples capas y mecanismos más complejos del tratado y filtrado de tráfico. Aunque la arquitectura que se lleva a cabo en este proyecto es más simple comprada con aquellas que se llevan a cabo en entornos reales, ésta representa la base del concepto de todas las arquitecturas modernas: separación de redes. Una práctica que consiste en separar la red en múltiples capas, donde en cada una se realizan distintas operaciones. De esta forma, la vulnerabilidad de una capa no compromete al resto. Este es el concepto principal que se pretende desarrollar, junto a la implementación de dispositivos de seguridad que nos ayudarán a reforzar nuestro entorno.

4 ANÁLISIS

El siguiente paso es diseñar una infraestructura acorde a los objetivos que se han planteado anteriormente. En este apartado se repasará las características principales de un cortafuegos, y por qué es necesario elaborar las políticas de seguridad. Mas adelante se verá cuáles son las arquitecturas que podemos implementar en nuestro proyecto, así como sus ventajas y desventajas. Por último, los dispositivos de seguridad que se integrarán.

4.1 Cortafuegos

A grandes rasgos, los cortafuegos pueden ser implementados tanto en hardware, software u ofreciendo una combinación de ambos. En todos ellos, se mantiene el principio básico: todo el tráfico entrante es examinado bajo distintos parámetros, asegurando que solo las comunicaciones autorizadas puedan atravesar el cortafuegos. Las acciones de este pueden ser simples como verificar reglas, intentos de acceso a archivos o servicios, o pueden ser más complejas y profundas, donde se analizan los datos internos de los propios paquetes. Un cortafuegos ofrece una capa de aislamiento entre una *red hostil*, como internet, y una *red limpia*, así como también son capaces de separar subredes y mantener el control de comunicaciones entre ellos.

Una conclusión errónea a la hora de implementar cortafuegos es que todas las amenazas tendrán origen en el exterior de la red. Esto es erróneo, debido a que puede haber un fallo de configuración o uso malintencionado de usuarios internos puede provocar brechas en la seguridad de nuestra red. Además, debemos tener en cuenta que cortafuegos mal configurado es peor que la ausencia de uno, ya que nos da una falsa sensación de seguridad.

Antes de integrar el cortafuegos en nuestra red, debemos realizar un *análisis de seguridad* en el cual, dado el escenario, debemos determinar nuestras *políticas de seguridad*. Estas políticas son representadas en un formato de documento de alto nivel, en las que se define el concepto de seguridad en nuestra red, que está permitido y que no, como debemos actuar ante las amenazas presentes y mitigar daños y consecuencias. Tal y como se indica en el libro ***Handbook of Information Security*** [3] para poder elaborar nuestras políticas de seguridad, podemos basarnos en las siguientes preguntas:

- Identificación de activos - ¿Qué es lo que estamos tratando de proteger?
- Análisis de Vulnerabilidades – ¿Cómo la información puede ser accesible por fuentes no autorizadas?
- Análisis de amenazas - ¿Quién puede usar la información de forma malintencionada? ¿Existen individuos internos con esas características?
- Análisis de riesgos - ¿Cuál es la probabilidad de ser atacado a través de una vulnerabilidad?
- Análisis de medidas protectoras - ¿Qué se puede hacer para mitigar el riesgo dado?
- Coste - ¿Cuáles son las alternativas a las infraestructuras basadas en cortafuegos? ¿Cuál es su coste?

Una vez determinadas las políticas de seguridad, gracias a las cuales implementaremos elementos, como reglas en nuestro cortafuegos, se puede pasar al siguiente apartado: determinar la infraestructura adecuada para el proyecto.

4.2 Arquitectura

Uno de los conceptos que se incorporará dentro de nuestro diseño es la *Defensa en Profundidad* o defensa elástica. Esta propone múltiples capas de seguridad basadas en el valor de la información que queremos salvaguardar. A mayor valor, mayor número de capas de seguridad estarán

dispuestas que tendrán que ser superadas. Además, debemos contemplar escenarios donde los ataques no solo pueden provenir desde fuera sino también desde dentro. Para ello, nuestras reglas de acceso al entorno deben estar correctamente configuradas y debidamente probadas. A continuación, se presentan algunos de los modelos de arquitectura más comunes que se pueden encontrar en distintos entornos corporativos.

Arquitectura de Cortafuegos en Trípode

Esta probablemente sea una de las arquitecturas más extendidas. Como se ve en la fig. 1, consiste en colocar un filtro de paquetes en la parte exterior o *frontera* de nuestra red.

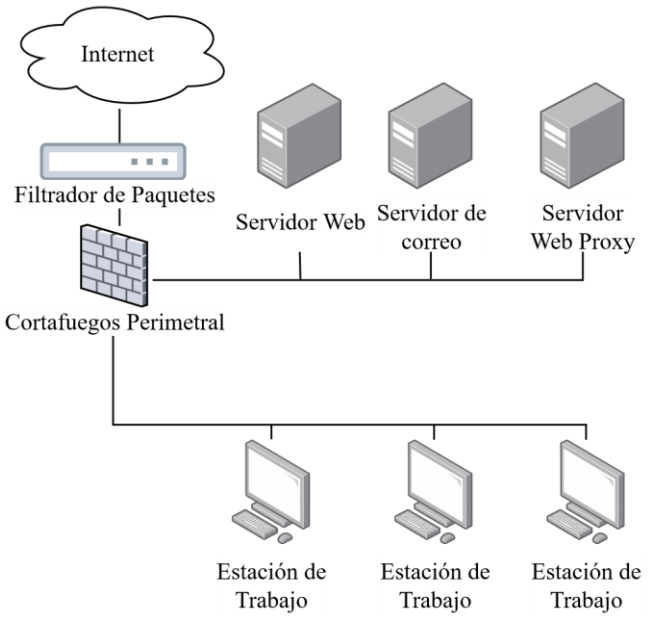

Fig. 1 Arquitectura de Cortafuegos en Trípode.

Este filtro cumple la función de descartar paquetes que ni siquiera deberían llegar hacia los cortafuegos. Algunos de estos paquetes son: con un tamaño anormal, fragmentos de paquetes, o protocolo ICMP [4]. Así, aliviamos la carga computacional que puedan sufrir los cortafuegos, evitando colapsarlo mediante ataques de denegación de servicio (DoS). El cortafuegos es encargado de separar la red interna de la zona desmilitarizada o *DMZ* [5], de allí el nombre de la arquitectura al tener 3 redes distintas conectadas al mismo cortafuegos.

Una de las ventajas de esta arquitectura es que es simple de implementar, configurar y gestionar. Pero, al haber un único punto de seguridad, solo hay una oportunidad, donde un fallo es suficiente para comprometer la seguridad de nuestra red. Este fenómeno es conocido como "*Punto de Fallo Único*" (SPoF).

Arquitectura DMZ Perimetral

Siendo una variación del diseño anterior, esta vez proporcionamos dos cortafuegos que separan las dos redes (fig. 2). Este ofrece una mayor seguridad al haber dos puntos de acceso, o control para acceder al área más restringida. Además, esta arquitectura provee de una gran flexibilidad para las aplicaciones de internet como e-mail, servicios web o e-commerce, a la vez que mantenemos protegida la red interna. Esto conlleva por otra parte, al aumento de la complejidad de configuración y elaboración de reglas de acceso, por lo que las probabilidades de errores aumentan. Otro de los factores es que es probable que aplicaciones alojadas en la DMZ necesiten acceso a la red interna para poder funcionar, pudiendo ocasionar una vulnerabilidad en el entorno.

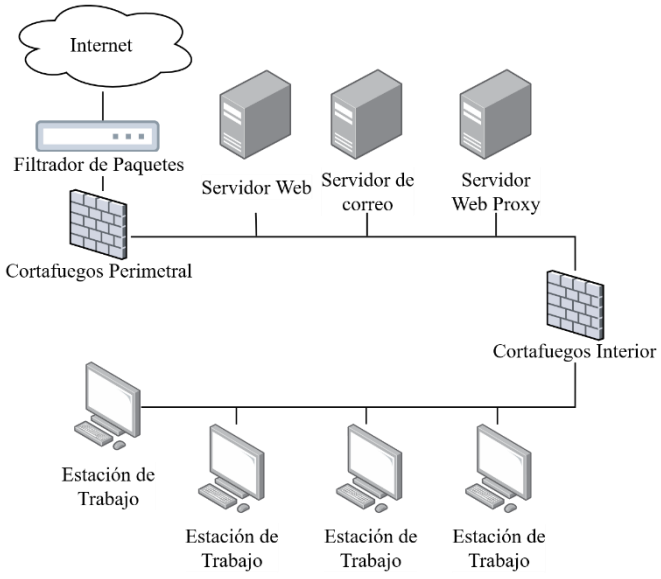

Fig. 2 Arquitectura DMZ Perimetral.

Arquitectura DMZ Distribuida

A medida que el comercio electrónico (e-commerce) se establecía en internet ganando cada vez más popularidad y aumentando el manejo de transacciones monetarias, surge una mayor necesidad de seguridad en nuestras infraestructuras.

Nuestro escenario, ahora requiere la implementación de capas adicionales a nuestra red Perimétrica. A modo de ejemplo, observando la fig. 3, tengamos en cuenta una estructura de cuatro capas: la primera podría ser encargada de anunciar el servicio y recoger las peticiones; el acceso a la segunda capa estaría protegido por un cortafuegos donde solo algunos paquetes esenciales tendrían acceso; en la tercera podríamos alojar la logística de nuestro negocio; En la última capa alojaríamos una base de datos que se encarga de procesar las transacciones que nos lleguen y, además podríamos alojar la red corporativa. Esta infraestructura provee una protección excelente y disminuye el riesgo a cada capa que profundicemos en la red. Sin embargo, es excesivamente compleja de mantener y tiene costes de implementación elevados. Cada capa adicional que podamos añadir aumenta las probabilidades de tener un error de configuración.

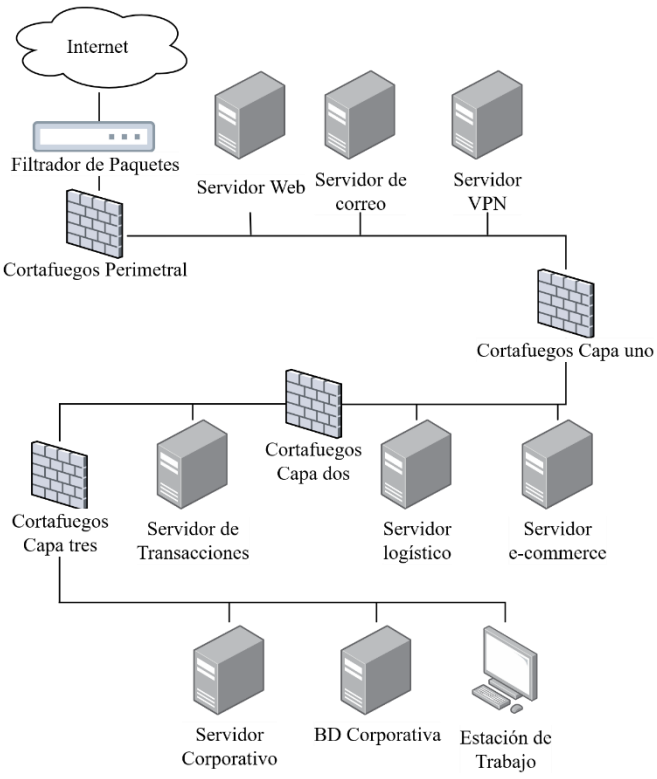

Fig. 1 Arquitectura DMZ Distribuida.

Éstas son algunas de las arquitecturas más comunes que nos podemos encontrar. Ninguna de estas opciones ofrece la mejor solución, ya que depende del escenario, recursos de los que se dispone y el conocimiento para configurar y administrar el sistema.

4.3 HoneyPot

Los honeypot son herramientas de seguridad que se utilizan con el fin de desviar ataques dentro de una red, teniendo este el rol de *cebo* que, a primera vista parece vulnerable. Estos, además, integran servicios que recolectan la información sobre los atacantes y las técnicas que utilizan.

Existen dos grandes categorías de Honeypots, de baja y alta interacción. Uno de baja interacción se limita a simular los componentes y servicios deseados de forma básica. Su diseño le permite recopilar información de los ataques automatizados. Los de alta interacción, están diseñados de manera más elaborada donde imitan casi por completo a los sistemas reales, llegando hasta tener actividad propia dentro del sistema. En las arquitecturas más avanzadas y complejas, podemos incluso llegar a observar *Honeynets*, un conjunto de Honeypots de alta interacción que simulan una red entera. Su implementación y mantenimiento es especialmente más complejo, pero posee un gran impacto dentro de la seguridad de la red.

En la red perimétrica de nuestro proyecto, está situado un servidor web. Este servicio puede representar una vulnerabilidad en nuestra infraestructura ya que debemos permitir acceso a él desde la red interna. Si el servidor es comprometido, este escenario puede ser explotado de forma malintencionada. Sin embargo, con la integración de un Honeypot en la DMZ, dividimos el factor riesgo en una probabilidad teórica del cincuenta por ciento. Hipotéticamente, a más honeypots presentes en la red, menor es el riesgo de comprometer un sistema real. Todo y que el atacante pueda percatarse de la presencia de estos componentes en la red y deba replantear su estratagema, se ha ganado un tiempo valioso en el que el ataque tiene que ser detectado para poder actuar.

4.4 Sistema de Detección de Intrusos

También llamado IDS, tiene como finalidad de incrementar la seguridad de un sistema capaz de avisar o anticipar los ataques que pueden estar ocurriendo o estén a punto de suceder. Al ser una herramienta que monitoriza la red, es capaz de detectar cualquier tipo de actividad anómala que se produzca, sea su origen desde fuera o dentro. Debemos tener en cuenta que las violaciones de políticas de seguridad pueden producirse también por parte de los usuarios internos, sea cuando intente realizar una operación no autorizada, intente acceder a lugares que, a priori no deberían o simplemente por errores de configuración de sistemas. Un IDS se encuentra en constante monitorización de la red en busca de cualquier actividad que puede ser sospechosa o anómala.

Este elemento puede estar compuesto por uno o un conjunto de módulos que pueden desde recoger datos de monitorización históricos hasta tomar decisiones sobre la red limitando el tráfico entrante y saliente ante una amenaza detectada. Un ejemplo que cabe destacar en nuestro proyecto, que a la vez será integrado en nuestra arquitectura, es el software *Snort* [6]. Además de ser un esnifador de red, Snort es también considero un Sistema de detección de intrusos. Cuenta con la gran ventaja de que se puede complementar con módulos desarrollados de forma independiente mencionados anteriormente, que le permiten añadir características adicionales. Esto es posible gracias a que Snort es un software libre y su interfaz de programación o *application programming interfaz* (API) está disponible de forma gratuita.

5 METODOLOGÍA Y PLANIFICACIÓN

5.1 Metodología

El desarrollo del proyecto se ha dividido en tres fases, cuales son dependientes entre sí. Teniendo en cuenta las dimensiones del proyecto, se decidió que el *Desarrollo en Espiral* es el modelo más adecuado a seguir en este proyecto. Esta metodología es utilizada en proyectos grandes, en los cuales los costes de error son muy elevados. La metodología consiste en cuatro etapas que siguen el siguiente orden: *Planificación, Análisis de Riesgo, Implementación y Evaluación* donde el conjunto de estas etapas se denomina *ciclo.* Estos ciclos empiezan siendo muy sencillos y rápidos, pero a medida que avanza el proyecto, se producen más iteraciones que nos permiten escalar y adaptar el proyecto a nuevas funcionalidades incorporadas. Esta metodología conlleva a ciertas ventajas en relación con los riesgos y el posterior mantenimiento

que se pueda conllevar. Al repetir y escalar las 4 etapas con cada ciclo, minimizamos los posibles riesgos que puedan surgir mitigándolos en etapas iniciales. Unificando el desarrollo con el mantenimiento, nos permite escalar el proyecto de una forma más segura.

5.4 Fases del Proyecto

El Proyecto está dividido en dos fases, en las cuales se establece un escenario, se integran elementos de seguridad adicionales y se realizan pruebas de funcionalidad y ataque contra la infraestructura. En la Fase uno se simula un escenario empresarial compuesto por una intranet, una DMZ y conexión a internet. En la Fase dos, se implementa el programario compuesto por dispositivos de seguridad.

- *Definir Objetivos y Requerimientos*: proceso en el que se lleva a cabo el planteamiento del propio trabajo incluyendo el caso de uso, escenario y cuál es el aporte social del proyecto.
- *Recopilación de Herramientas*: Definición del estado del arte del proyecto. Comporta la investigación y búsqueda de cuáles son los propios medios para llevar las tareas a cabo. En este proceso se determinan las herramientas con las que se trabajará durante todo el proyecto.

Fase Uno: Desarrollo de infraestructura

Una vez determinados los objetivos y herramientas de trabajo, se preparará el escenario de trabajo donde se implementará el paquete de seguridad.

- *Conexión exterior:* Con el simulador de redes, se configurará el enrutador para que pueda proporcionar conexión a internet a toda la red.
- *Intranet:* Se implementarán un conjunto de máquinas simples formando una intranet empresarial.
- *Cortafuegos:* Configurar un firewall sencillo que no permita pasar al tráfico no autorizado a la red e intranet.
- *Pruebas de Integración:* Una vez implementados los módulos, se realizarán pruebas de integración en las que se pueda verificar la conexión a internet, comunicación entre máquinas de la intranet y su no-conexión desde fuera.

Fase Dos: Integración de los componentes de Seguridad

Consiste en añadir elementos gracias a los cuales se podrá mejorar la seguridad e integrar la monitorización en la red.

- *Propuesta de los componentes:* Especificación amplia del conjunto programario que se añadirá en la infraestructura. Se estudiará si dichos elementos mejoran la seguridad del entorno.
- *Implementación del HoneyPot:* se llevará a cabo la incorporación del señuelo con el objetivo de detectar el ataque en el sistema. El software específico será elegido en la fase anterior.
- *Implementación del IDS:* Consiste en implementar un sistema de detección de intrusos en el que se recopilará la información del atacante.
- *Implementación del DMZ:* Implementación de un servidor web simple que simulará la página web de la empresa.
- *Implementación de Cortafuegos:* Desarrollo de un cortafuegos adicional.
- *Pruebas de Integración:* Una vez finalizados los módulos, se ejecutarán para verificar su correcto funcionamiento entre ellos.

Inicialmente en el proyecto, se había planificado una tercera fase en la que estaba previsto realizar varios ataques a distintas partes del sistema como DMZ, Servidor Web o la Red Interior. Esta parte finalmente ha sido descartada y reemplazada con pruebas de funcionalidad del sistema.

5.3 Herramientas de Soporte

Simulador de Redes

El objetivo es poder analizar y observar la viabilidad que tiene la implementación de los componentes de seguridad que queramos dentro de nuestra estructura. Para ello, nuestro entorno debe tener la máxima similitud con la realidad. Por lo que debemos optar por un simulador y no un emulador de redes. Todo y que la diferencia es sutil, la diferencia reside en que un emulador recrea el comportamiento de los componentes de manera que parezcan reales. El ejemplo más representativo de un emulador sería *Packet Tracer*, de Cisco [7]. Un simulador, sin embargo, virtualiza el hardware de dicho componente para que podamos ejecutarlo como lo haríamos en un entorno de producción. Esto, por supuesto, conlleva a un coste computacional mayor que un emulador.
Se ha optado por la elección del *GNS3*. Es un simulador de código abierto que, a diferencia de otros simuladores, permite combinar dispositivos reales, a través de imágenes de formato ISO. Grandes corporaciones como la Agencia Espacial Estadounidense *NASA*, empresa petrolera *Exxon* o conglomerado empresarial de telecomunicaciones *AT&T* trabajan con este producto [8].

Analizador de Tráfico

Una de las ventajas de *GNS3,* es que incluye de forma nativa el analizador de tráfico *WireShark*. Este programario nos incluye múltiples opciones de organización y filtrado de información que obtenemos. Teniendo en cuenta que no necesitamos la instalación de componentes adicionales, así como experiencia previa de uso, se ha optado por su uso.

6 DESARROLLO

En esta sección describiremos los puntos más importantes llevados a cabo en el desarrollo del proyecto. Se explicará cuales son los factores que determinaron la arquitectura implementada, así como sus cortafuegos asociados. Se mostrará cual es la política de seguridad asociada al entorno y algunas de las reglas más destacadas. Presentaremos adicionalmente los elementos más comunes que podemos encontrar en estas arquitecturas simulando un entorno real de trabajo. Finalmente explicaremos como son los componentes que

proporcionan seguridad adicional y cuál es la labor de cada uno.

6.1 Topología

Nuestro simulador de redes trae algunos de los componentes más esenciales para poder conectar nuestro entorno de trabajo hacia el exterior, o *Wide Area Network*. Uno de estos elementos es llamado *Cloud*. Este no es más que una interfaz tipo "Puente Virtual" [9] que permite la interconexión de máquinas virtuales para formar una red virtual. Esta interfaz es visible fuera del simulador, en el propio equipo se denomina "*virbr0*". Esto es importante ya que es nuestra puerta de entrada y salida a nuestra red. Todas las conexiones entrantes y salientes pasan por la interfaz.

Para que podamos ejecutar los componentes que no están integrados en GNS3, debemos descargar las plantillas llamadas "*appliances*" para que se puedan configurar y abrir. Estas plantillas pueden ser obtenidas tanto desde la página web del simulador como aquellos que han sido creados por los usuarios. Una vez integrada la plantilla, el siguiente paso es descargar la imagen ISO del propio componente (también podemos integrar máquinas virtuales o "hosts" en nuestro simulador, sincronizando GNS3 con el software de virtualización).

En el simulador, se han integrado los siguientes elementos para poder construir la construir la red:

- *Enrutador:* Mikrotik 6.44.5
- *Navegador Web:* Firefox 31.1.1
- *Firewall:* PfSense 2.4.4
- *Servidor Web:* Toolbox

Cabe destacar que todos estos componentes fueron elegidos con el criterio de estar disponibles en forma de libre uso no comercial, y algunos como *Mikrotik* o *Firefox*, por la experiencia adquirida en su uso.

El siguiente paso es decidir cuál es el tipo de arquitectura que emplearemos en nuestra red. En la Sección cuatro hemos visto cuales son los distintos tipos de arquitectura más comunes a los cuales podemos basarnos. Cada una tiene sus ventajas y desventajas que ya hemos mencionado con relación al coste de mantenimiento e implementación, y la seguridad que ofrece. Después de estudiar cada una de las opciones, se ha elegido la *Arquitectura DMZ Perimetral*. Para nuestros objetivos, este diseño base ofrece un balance adecuado para las necesidades de seguridad que tenemos, su coste de mantenimiento e implementación. Las arquitecturas más simples que constan de un punto de seguridad no proporcionan la seguridad necesaria para nuestro entorno de trabajo, caracterizados por el factor SpoF mencionado anteriormente.

Por otro lado, todo y que las arquitecturas multicapa o distribuidas, son atractivas al proporcionar un nivel de seguridad superior. Sin embargo, su complejidad de implementación y mantenimiento es desproporcional a las necesidades de seguridad de nuestra red. Este tipo de arquitecturas están comúnmente enfocadas a negocios relacionados con *e-commerce*, lo cual se encuentra fuera del dominio del problema de este trabajo. Sin embargo, cabe destacar que la arquitectura perimetral ofrece la escalabilidad necesaria para añadir capas que hagan falta, si en el futuro surge la necesidad de implementar esta arquitectura.

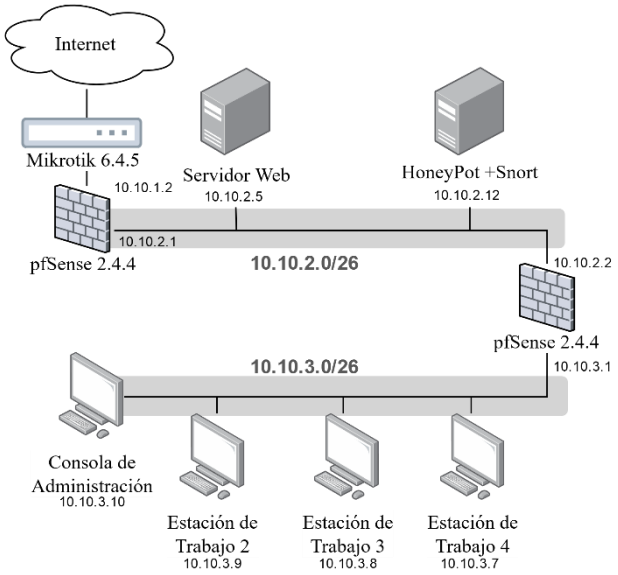

Fig. 4 Topología de Red Desarrollada

En la fig. 5 podemos observar los elementos presentes en nuestra red. Está compuesta principalmente por dos cortafuegos, donde el primero separa la conexión exterior de la zona perimetral, y el segundo protege la red corporativa. Además, podemos observar los elementos de seguridad incorporados en la red perimetral como el servidor web, las estaciones de trabajo de la red interior, y el filtrador de paquetes ubicado en la frontera.

6.2 Políticas de Seguridad

Paralelamente, es importante definir pautas, reglas o planes de acción en nuestro entorno empresarial. Éstos pueden cubrir desde las buenas prácticas en relación con mantenimiento de contraseñas, hasta protocolos complejos de actuación en caso de que se comprometa la seguridad. Es importante definir estas políticas ya que también pueden servirnos de guía para poder definir las reglas específicas de cada elemento de seguridad, como puede ser un cortafuegos, enrutador, estación de trabajo e incluso hardware que implique acceso a las instalaciones. Unas buenas políticas de seguridad pueden determinar incluso quien puede acceder a que equipos o salas de la oficina.

Todo y que se deba elaborar un documento formal donde reunir todas esas variables, estas reglas se han simplificado para este proyecto, ya que nuestro dominio se encuentra únicamente en la red. Las políticas de seguridad de la red se muestran a continuación:

1. Se permite acceder únicamente al Servidor Web desde la WAN.
2. Tráfico destinado a consolas de administración está prohibido.
3. No se permite el paso de paquetes ICMP a través de redes.

4. Únicamente los usuarios autorizados pueden acceder a internet desde la red interna. Éste solo permitirá navegar y consultar bandejas de correo remotas.
5. El tráfico que no sea de respuesta no puede acceder a la red interna.
6. Únicamente se habilitarán protocolos necesarios para la navegación en internet, el resto de los protocolos serán deshabilitados por defecto.
7. La política por defecto será "Denegar por defecto".

Estas políticas las usaremos de respaldo a la hora de implementar las reglas que veremos a continuación, en cada uno de los dispositivos de seguridad. Cabe destacar que la política número 7, "Denegar por defecto" es muy común, ya que indica que, todo lo que no esté permitido explícitamente, está prohibido.

Filtrador de paquetes

Es el primer elemento que está en contacto con la WAN, siendo el enrutador de frontera. Su función principal es eliminar todo tráfico que ni siquiera tiene que llegar al cortafuegos. Este proceso es relativamente sencillo pudiendo ser ejecutado por un hardware de bajo coste y fácil de configurar. Las llamadas "chains" o reglas, más destacables que se han implementado son:

- *No permitir fragmentación de paquetes:* La fragmentación, además de ser asociada con ataques de denegación de servicio (DDoS), puede ocasionar excesivas retransmisiones ya que es común observar pérdida de paquetes en esta práctica. Además, esto conlleva a un coste computacional en los enrutadores y cortafuegos.
- *No permitir valores TTL inusuales:* Debido a que los paquetes que viajan por internet no realizan más de 30 saltos, a partir del valor TTL establecido por defecto de sistemas operativos más usados, podemos extraer cuales son los valores menos frecuentes que podríamos encontrar. Siendo TTL=255 en Linux, 128 en Windows y 64 en sistemas operativos MAC, determinamos que:

  $1 < X \leqq 30$, o $64 < X \leqq 98$, o $128 < X \leqq 225$
  $30 < Y \leqq 64$, o $98 < Y \leqq 128$, o $225 < Y \leqq 255$

  Sea Y el valor de TTL normal y X un TTL anormal. Por desgracia, una de las limitaciones tanto del enrutador Mikrotik como de nuestro cortafuegos es que no se pueden establecer rangos de valores TTL, por lo que únicamente se deniegan los paquetes con un TTL menor a 30.

Permitir protocolo ICMP

El protocolo ICMP tiene múltiples vulnerabilidades asociadas [10], por lo que una solución posible sería evitar el tráfico entrante en la red. Sin embargo, esta práctica puede conllevar muchos problemas, y una de ellas es que puede conllevar a la inevitabilidad de fragmentación IP. Al utilizarse este protocolo para determinar la unidad máxima de transferencia (Maximum Transfer Unit) [11], su bloqueo podría provocar la fragmentación de paquetes al no poder determinar el MTU. Además, como decidimos filtrar los paquetes fragmentados, es muy posible que los paquetes legítimos nunca consigan pasar o que ocasione una tasa de pérdida de paquetes muy alta. Por eso mismo, el protocolo ICMP es permitido hasta el cortafuegos perimetral.

¿Rechazar o Denegar?

La diferencia reside en que una denegación (*DROP*) descarta los paquetes sin notificar al emisor. El rechazo (*REJECT*) sí que genera un paquete de respuesta notificando el descarte. Desde el punto de vista de seguridad, un *DROP* es preferible al rechazo, forzando al atacante esperar una respuesta, ralentizando el proceso del ataque. Todo y que un atacante pueda escanear múltiples puertos al mismo tiempo, no responder siempre es mejor, evitando que se recoja la mayor información disponible.

Cortafuegos

En nuestra infraestructura existen dos cortafuegos: perimetral e interior. Todo y que algunas de las funciones que realizan pueden ser distintas, tienen que cohesionar en la red para poder minimizar los errores que pueden producirse, tanto de configuración como conflictos entre las reglas. Para ello, debemos establecer cuáles de las funciones que realizan son ambos iguales y cuáles no. Por ejemplo, ambos realizan una conversión de direcciones IP (Network Address Translation) para pasar de una red a otra. De esta forma, enmascaramos la red en la que estamos del exterior. Por el otro lado, nos facilita la gestión ya que, al ser el cortafuegos quien traduce las direcciones internas hacia el exterior, solo su dirección IP es visible al resto.
Tal y como vemos en la Figura, si queremos restringir la conexión a internet a la Estación de Trabajo 1, solo hace falta configurar una única regla en el enrutador interior rechazando paquetes provenientes. En caso de que no efectuáramos NAT, podríamos introducir la redundancia de replicar la regla en el cortafuegos perimetral, pero implicaría manejo de mayor número de reglas totales de la red, lo que aumentaría la probabilidad de errores de configuración interiores.

Filtrado con estado y sin estado

También denominado como *stateful y stateless* respectivamente, son técnicas de evaluación de paquetes en las que stateful realiza un seguimiento de estado de la conexión, mientras que el filtrado stateless evalúa todos los paquetes de forma independiente. Este último quiere decir que, independientemente si es una conexión nueva o existente, el paquete será evaluado según las reglas establecidas por el cortafuegos. Esto implica que, debemos crear una regla para cada dirección de tráfico, teniendo en cuenta tanto la entrada de paquetes, como la salida de una misma conexión.
El filtrado stateful, sin embargo, fue creado para conocer el estado de la conexión y en base a ello decidir la legitimidad del paquete, evitando así ataques de suplantación. Para ello se crea una tabla de estados en la que cada conexión, anteriormente validad por las reglas del cortafuegos, es instanciada. Posteriormente, antes de iniciar el proceso de

evaluación de reglas, el cortafuegos stateful comprueba los estados guardados en la tabla. Si existe una conexión asociada, el cortafuegos acepta el paquete sin la comparación de reglas. En el proyecto fue incorporado el filtrado con estado ya que, además existe una reducción del coste computacional, al no tener que procesar las reglas individualmente para cada paquete.

Reglas

Pfsense divide las reglas según la interfaz a la que son asignadas. Por lo tanto, si nuestro cortafuegos está conectado a dos redes distintas, tendremos dos categorías donde asignar las reglas. Su simple propósito es la organización y asociación de reglas a una interfaz específica, evitando así ataques *IP Spoofing*. Existe también una tercera categoría llamada "Flotante" que es la primera categoría en ser verificada ante una comprobación de reglas. Acto después si no se han encontrado coincidencias, se evalúan las reglas de la categoría en función de entrada del paquete al cortafuegos; si evaluamos un paquete entrante por la interfaz 1, las reglas asociadas a esa categoría serán primeras en evaluarse.

TABLA 1
REGLAS DE CORTAFUEGOS PERIMÉTRICO E INTERIOR

Acción	Protocolo	Origen	Puerto	Destino	Puerto
Permitir	TCP	*	>1024	S.web	80,443
Permitir	TCP/UDP	C. Interior	>1024	C	53
Permitir	TCP	C. Interior	>1024	*	80,443
Permitir	TCP	C. Interior	>1024	C	8920
Denegar	*	*	*	*	*
		Cortafuegos Interior			
Permitir	TCP	C. Admin	>1024	C.Interior	4360
Permitir	TCP	C. Admin	>1024	C	8920
Permitir	TCP	Est. Trabajo	>1024	*	80,443
Permitir	TCP/UDP	Est. Trabajo	>1024	C.Interior	53
Permitir	TCP	Est.Trabajo	>1024	S.web	80,443
Denegar	*	*	*	*	*

** = cualquier dirección IP, S.web= servidor web, C. Admin = consola administrativa, C = cortafuegos, C. Interior = cortafuegos interior, Est. Trabajo=Estaciones de Trabajo; S. web = 10.10.2.5/26, C. Admin = 10.10.2.10/26, C = 10.10.2.1/26, C. Interior = 10.10.2.2/26.*

La tabla 1 muestra una simplificación de estas reglas. El primer conjunto pertenece al cortafuegos perimétrico y el segundo, al interior. La primera regla permite el tráfico entrante al servidor web, mientras que las dos siguientes permiten la salida de tráfico del cortafuegos interior. Y la última regla concede permisos de acceso al propio cortafuegos.

En el cortafuegos interior, disponemos de reglas que permitan la conexión al servidor web; reglas de acceso a ambos cortafuegos; reglas asociadas a las estaciones de trabajo que tienen permitida la conexión a internet. Cuando un ordenador de la red interior tiene la intención de consultar una página web externa, envía una petición de DNS al cortafuegos. Si el ordenador está autorizado, esta petición es reenviada al cortafuegos perimetral, debido a que localmente no hay ningún servidor DNS habilitado. El cortafuegos perimetral tampoco la puede resolver y vuelve a reenviarla, esta vez a servidores públicos asignados. Esta respuesta es procesada y reenviada de vuelta al cortafuegos, que lo reenvía a su vez al emisor. Cuando el ordenador intenta conectarse a la página web solicitada, se realiza NAT tanto en el cortafuegos interior como en el perimetral.

Cabe destacar que, en ambos cortafuegos, no se realizan las resoluciones de nombres de dominio (Domain Name System) todo y que reciban peticiones por el puerto 53. Estos, simplemente reenvían la petición a servidores de terceros como Google o CloudFlare.

Snort

Para poder monitorizar la red perimetral de la infraestructura, se ha decidido implementar un sistema de detección de intrusos *Snort.* La razón principal para implementarlo fuera de la red interior es que la mayor parte de actividad es ocurrida en la DMZ, añadiendo el hecho de que la mayoría de los ataques esperan provenir del exterior, pasando por esta red. Las reglas de detección más básicas pueden ser elaboradas de forma manual, añadiéndolas al archivo correspondiente e indexándolo en la configuración de arranque de este servicio. Anteriormente se ha mencionado que algunas de las reglas relacionadas con valores de TTL no podían ser implementadas en el Filtrador de Paquetes. Bien, con las reglas de Snort podremos elaborar sistemas de alerta cuando se detecten paquetes con TTL inusuales.

El cortafuegos PfSense posee un paquete de compatibilidad con el IDS Snort, integrado en el. Una vez configurada la interfaz en la que decidamos usarlo, podremos arrancarlo. Además, Snort cuenta con un gran número de plug-ins desarrollados por la comunidad, lo que nos permite personalizarlo para el entorno en el que se use. En este proyecto, uno de los plug-ins incorporados es el de detección de amenazas en la capa de Aplicación, lo que nos permite analizar datos que se ejecutarán en el propio sistema posteriormente.

HoneyPot

Gracias a una distribución de Linux dedicada exclusivamente al manejo de HoneyPots llamada *HoneyDrive*, se ha añadido una instancia dentro de la Red Perimetral. Para nuestro entorno se ha decidido implementar el software Kippo y Dionaea. Ambos considerados honeypots de baja interacción en las que su función es soportar ataques de fuerza bruta mientras que recopilan toda la información del atacante. Kippo simula

un servidor SSH, mientras que Dionaea ofrece soporte a una mayor variedad de protocolos, indicados en la Tabla 2. El conjunto de puertos abiertos de Dionaea simula el comportamiento de Microsoft Windows Server. De esta forma, se espera que esta máquina sea capaz de desviar ataques que se produzcan contra esta red. Además, estos servicios incluyen herramientas gráficas de monitoreo para que podamos ver los datos obtenidos de manera ordenada. Algunos de los datos que se muestran son el número de conexiones, cantidad de paquetes recibidos, procedencia de IPs, etc.

TABLA 2
PUERTOS Y SERVICIOS DE LOS HONEYPOTS

Puerto	Servicio	Honeypot
21	FTP, Transferencia de Archivos	Dionaea
22	SSH, Acceso remoto cifrado	Kippo
80	HTTP, Transferencia de hipertexto	Dionaea
135	RPC, redirección de tráfico	Dionaea
443	HTTPS, Transferencia segura de hipertexto	Dionaea
445	SMB, Compartición de archivos	Dionaea
1433	MSQLS, Servidor SQL de Microsoft	Dionaea
5060	SIP, Negociación de llamadas VoIP	Dionaea

FTP = File TransferProtocol, SSH=Secure Shell, HTTP = Hypertext Transfer Protocol, RPC = Remote Procedure Call, HTTPS =Hypertext Transfer Protocol Secure, SMB = Server Message Block, MSQS = Microsoft SQL Server, SIP = Session Initiation Protocol, VoIP = Voice over IP.

7 PRUEBAS

La realización de pruebas en el sistema es esencial para detectar vulnerabilidades, errores de configuración o posibles planteamientos erróneos que pudiéramos desarrollar durante el proyecto. Esta sección está dedicada a documentar aquellos tests realizados a los distintos componentes del sistema. Primero analizaremos la configuración de los propios cortafuegos en el sistema. Luego comprobaremos las reglas configuradas manualmente en nuestro Sistema de Detección de intrusos, y, por último, aprovecharemos la vulnerabilidad de puertos abiertos del Honeypot para realizar un ataque y comprobar si éste se ha registrado correctamente.

7.1 Configuración de Infraestructura

Para analizar las configuraciones del sistema, se han añadido múltiples estancias a cada una de redes de tal forma que, estas no están contempladas por las reglas de configuración de los cortafuegos. Respecto a los elementos como estaciones de trabajo o servidores web, se realizaron varias pruebas de conectividad a la WAN.

Red Perimétrica

Las primeras pruebas de conectividad fueron realizadas al servidor Web. En nuestro caso, realizamos varias peticiones desde el navegador de la maquina Host, por lo tanto, fuera de la red simulada, hacía el servidor web.
Para ello, necesitamos añadir rutas estáticas a nuestra tabla de enrutamiento. Las redes que queremos añadir son: 10.10.1.0/24 perteneciente a la red frontera, 10.10.2.0/26 perteneciente a la red perimétrica y 10.10.3.0/26 perteneciente a la red interior. El *gateway* indicado para cada ruta estática es la dirección IP de la interfaz exterior del enrutador frontera, que a la vez actúa de filtrador de paquetes.
Gracias al analizador de redes Wireshark incorporado, podremos obtener información acerca de cada una de las redes por las que viajan las peticiones. En estas pruebas, se habían encontrado errores de configuración en el cortafuegos debido a la categoría propia a la que pertenece la regla. Al llegar los paquetes de la interfaz WAN del cortafuegos, es el lugar adecuado para establecerla. Este error fue muy común en el establecimiento de las reglas ya que el concepto cortafuegos con estado, ligado a separación de reglas por categorías puede parecer caótico al interactuar por primera vez con estos sistemas.
La siguiente prueba realizada fue crear una estancia de un servidor web junto a una estación de trabajo. Se les había asignado una IP por el servidor DHCP presente en el cortafuegos, aunque este podría ser deshabilitado si lo quisiéramos. Una vez configurados y arrancados, se realizaron intentos de conexión, a través de las estaciones de trabajo tanto hacia el exterior como hacia la red interior. En ambas pruebas, los cortafuegos no dejaron pasar los paquetes y fueron registrados en el *log*. Respecto al servidor web instanciado, se realizaron intentos de conexión dirigidos hacía el, y tampoco se ha recibido respuesta, lo que indica que el cortafuegos impidió la entrada de estos paquetes.

Red Interior

Al haber presentes 4 estaciones de trabajo, realizamos pruebas de conexión a internet con cada una de ellas. En esta parte, hemos detectado varios problemas de configuración relacionadas con la resolución de nombres de dominio. Ambos cortafuegos, interior y exterior, realizaban intentos de resolución de nombre sin éxito. Estos han sido puestos en modo "*Forwarder*", y para el cortafuegos interior, el servidor de resolución indicado es el cortafuegos interior, únicamente. Mientras en el cortafuegos exterior, fueron indicadas las direcciones IPs del servidor DNS de Google y CloudFlare. Una vez configurados los cortafuegos, seguimos teniendo problemas de conexión ya que, los paquetes enviados por el enrutador interior no llegaban al enrutador interior. Esto se debía a que las reglas fueron configuradas únicamente bajo el protocolo TCP, mientras que las peticiones DNS también pueden viajar por el protocolo de transmisión UDP. Las reglas afectadas fueron corregidas en ambos cortafuegos. Esta vez sí, al comprobar la conectividad de las estaciones de trabajo, estas fueron capaces de enviar y recibir datos desde internet. Se realizó una última prueba

en la que una de las direcciones IP de las estaciones de trabajo fueron eliminadas del cortafuegos para comprobar si está aún pudiera mantener su conexión a internet. Esta comprobación es importante ya que no haría falta en nuestro caso bloquear direcciones interiores para restringir su conexión, sino únicamente borrar la IP existente de la regla.
Gracias a ello, tanto en la red perimétrica como interior, si se incorporan nuevos dispositivos, estos no pueden tener conexión al exterior, ya que deben ser debidamente aprobados por el administrador de sistemas.

Snort

Para comprobar que el sistema de alertas Snort funciona de manera correcta, hemos creado varias alertas que quedan registradas en el sistema. Todo y que el mayor potencial de Snort lo obtenemos a través de las reglas creadas y mantenidas por la comunidad, en las que podemos detectar presencia de exploits en el contenido de los paquetes, las reglas que se han creado son más sencillas, pero pueden ser más adaptables a nuestro entorno. Las reglas que fueron creadas son los intentos de conexión al enrutador a múltiples puertos, acceso a dominios específicos fuera de la red y paquetes que tienen dirección origen o destino de la red 10.10.3.0/26 interior. Dado que es el cortafuegos interior quien enmascara la IP bajo la NAT, estos paquetes no deberían existir en la red perimétrica.
Para ello, hemos realizado distintas peticiones desde la consola de administración, presente en la red perimétrica, para lanzar estas peticiones y comprobar si realmente Snort es capaz de detectarlo. El resultado de todas estas pruebas fue satisfactorio, en el que se han generado las alertas correspondientes.

Honeypot

Gracias a la herramienta *nmap*, realizamos un escaneo hacia todos los puertos abiertos que tiene el honeypot, visualizados en la Tabla 2 de la sección anterior. En esta prueba, esperamos que el ataque quede registrado debidamente, para que pueda ser posteriormente analizado.
En este caso, hemos elegido el puerto 445, perteneciente al servicio SMB de Microsoft. Con la herramienta Metasploit, en el que se recopilan vulnerabilidades de seguridad conocidas, usaremos el exploit asociado. Al ser un servicio de Microsoft, el atacante podrá interpretar que se trata una maquina Windows, cuando en realidad es un Linux.
El resultado del uso de exploit fue negativo, dado que ninguna sesión fue creada al usarlo. Mientras que, el ataque fue registrado en los archivos *log*, el extracto que se puede consultar en el Apéndice 1.

Para visualizarlo de la forma más cómoda, disponemos de herramientas gráficas de cada uno de los honeypots, para poder analizar estos datos. En ello podemos visualizar el servicio *pcap*, por el cual se había intentado acceder. Podemos comprobar también el numero 6395 conexiones realizadas y el puerto de origen 39862 a través del cual se realizó el ataque.

8 CONCLUSIONES

El enfoque principal de las pruebas realizadas ha sido demostrar la viabilidad y la funcionalidad de los componentes integrados en el sistema. La cohesión de los diversos sistemas y componentes en el sistema ha sido el mayor desafío para poder llevar a cabo el proyecto. La exploración del sistema de cortafuegos integrado ha sido definitivamente el elemento más explorado de todo el sistema, donde se han estudiado diversas herramientas, tanto integradas como opcionales que las componen. Los elementos de seguridad adicionales como el Sistema de detección de Intrusos y los Honeypot se han incorporado en el sistema como prueba de concepto que representan, que podrán llegar a ser desarrollados más a fondo aprovechando todas las ventajas que se nos ofrecen. Sin embargo, estos componentes tienen un gran potencial, *Snort* posee una gran versatilidad en la que podremos desarrollar reglas complejas que pueden ser adaptadas a nuestras facilidades del entorno. Posee una gran cantidad de módulos que extienden las posibilidades del software brindándole la capacidad de tomar decisiones en la propia red. HoneyPot es un sistema extraordinario que nos permite desviar los ataques que pueden ser producidos contra red. Su incorporación a la red perimétrica de la infraestructura nos ayuda a desviar un 50% teórico de los ataques hacia el sistema, donde el administrador de sistemas adquiere la ventaja más importante para lidiar con los ataques, el tiempo.
En este trabajo se ha visto como a partir del análisis de arquitecturas de red planteadas, se ha llevado a cabo una implementación completa simulando el entorno de trabajo. Gracias a los componentes como HoneyPot nos permite recoger datos de los posibles ataques que se dirijan hacía el dispositivo, desviando y mitigando parcialmente los daños que pueden causar. Gracias al sistema de detección de intrusos Snort, podemos monitorizar la actividad que se produce en la red, pudiendo detectar anomalías, uso malintencionado de sistemas, errores e incluso identificar posibles ataques que estén a punto de ocurrir.
Todo ello, junto al concepto de defensa en profundidad o separación por capas, es posible ofrecer servicios públicos sin comprometer la seguridad de la red interior. Y con la incorporación de elementos de seguridad adicionales, se permite monitorizar mejor la actividad en la red y desviar los ataques que pueden ser producidos.

9 TRABAJO FUTURO

Tal y como se ha desarrollado este proyecto, se ha demostrado que los dispositivos Snort y honeyPot representan una buena prueba de concepto a la hora de integrarlos en la red. La implementación exhaustiva de Snort nos permitiría monitorizar la red con más precisión; múltiples instancias de HoneyPot repartidos podrían absorber mayor cantidad de ataques. Además, centralizar la administración de dispositivos podría repercutir en una simplificación de la gestión de la red, por lo tanto, se podrían reducir los costes de mantenimiento.
Otra posible vía de estudio y desarrollo de este proyecto es

la implementación de una red entera compuesta por HoneyPots de alta interacción llamada HoneyNet. Estos simulan por completo tanto la actividad de cada dispositivo como la propia comunicación entre ellos mismos, siendo en algunos casos prácticamente indistinguibles de un entorno real.

REFERENCIAS

[1] "What is the Cloud?", Cloudflare, 2020. [Online]. Available: https://www.cloudflare.com/learning/cloud/what-is-the-cloud/. [Accessed: 23- May- 2020].

[2] "Amazon EC2 Service Level Agreement – Amazon Web Services", Amazon Web Services, Inc., 2020. [Online]. Available: https://aws.amazon.com/es/ec2/sla/historical/. [Accessed: 17- Apr- 2020].

[3] H. Bidgoli, Handbook of information security, 3rd ed. Hoboken, N.J: John Wiley & Sons, 2006, pp. 510-540.

[4] "RFC 792 - Internet Control Message Protocol", Tools.ietf.org, 2020. [Online]. Available: https://tools.ietf.org/html/rfc792. [Accessed: 29- Apr- 2020].

[5] "What is a DMZ (networking)? | Barracuda Networks", Barracuda.com, 2020. [Online]. Available: https://www.barracuda.com/glossary/dmz-network. [Accessed: 10- Mar- 2020].

[6] "Snort - Network Intrusion Detection & Prevention System", Snort.org, 2020. [Online]. Available: https://www.snort.org/. [Accessed: 13- Apr- 2020].

[7] "Cisco Packet Tracer - Networking Simulation Tool", Netacad.com, 2020. [Online]. Available: https://www.netacad.com/courses/packet-tracer. [Accessed: 17- Feb- 2020].

[8] "GNS3 | The software that empowers network professionals", Gns3.com, 2020. [Online]. Available: https://www.gns3.com/. [Accessed: 06- Feb- 2020].

[9] "Virtual Networking", People.gnome.org, 2020. [Online]. Available: https://people.gnome.org/~markmc/virtual-networking.html. [Accessed: 14- May- 2020].

[10] F. Gont, "ICMP Attacks Against TCP", OWASP, pp. 10-16, 2006. Available: https://owasp.org/www-pdf-archive/ICMP_Attacks.pdf. [Accessed 28 June 2020].

[11] "RFC 4821 - Packetization Layer Path MTU Discovery", Tools.ietf.org, 2020. [Online]. Available: https://tools.ietf.org/html/rfc4821#section-5.2. [Accessed: 31- May- 2020].

[12] "RedIRIS - Implementación práctica de políticas de seguridad: La S.G.T.I. del MEC", Rediris.es, 2020. [Online]. Available: https://www.rediris.es/difusion/publicaciones/boletin/38/ponencia1.html. [Accessed: 09- Feb- 2020].

[13] "RedIRIS - Cortafuegos: Casos de estudio", Rediris.es, 2020. [Online]. Available: https://www.rediris.es/cert/doc/unixsec/node24.html. [Accessed: 16- Feb- 2020].

[14] "Bastion host", Es.wikipedia.org, 2020. [Online]. Available: https://es.wikipedia.org/wiki/Bastion_host. [Accessed: 01- Mar- 2020].

[15] "About Us | Project Honey Pot", Projecthoneypot.org, 2020. [Online]. Available: https://www.projecthoneypot.org/about_us.php. [Accessed: 24- Feb- 2020].

[16] M. Traver Codina, "Honeypots. L'art de la Guerra", p. 14, 2020. [Accessed 26 February 2020].

[17] H. Espinosa, "Implementation of a Web Application Firewall for a High Availability front end", p. 10, 2020. [Accessed 1 March 2020].

APÉNDICE 1

Extracto del registro de HoneyPot de Dionaea

Conexión establecida

```
[28062020 14:08:28] connection connection.c:4337-mes-
sage: connection 0x96afe00 accept/tcp/none
[192.168.203.129:445->192.168.203.1:39872] state: none-
>established

   [28062020 14:08:28] connection connection.c:4337-message:
connection 0x945cdd0 connect/tcp/shutdown [un://-
>un:///tmp/p0f.sock] state: shutdown->close

   [28062020 14:08:28] SMB dionaea/smb/smb.py:651-critical:
Traceback (most recent call last):
     File "/opt/dionaea/lib/dionaea/python/dionaea/smb/smb.py",
line 648, in handle_io_in
       p = DCERPC_Header(data)
     File "/opt/dionaea/lib/dionaea/python/dionaea/smb/in-
clude/packet.py", line 96, in __call__
       i.__init__(*args, **kargs)
     File "/opt/dionaea/lib/dionaea/python/dionaea/smb/in-
clude/packet.py", line 166, in __init__
       self.dissect(_pkt)
     File "/opt/dionaea/lib/dionaea/python/dionaea/smb/in-
clude/packet.py", line 439, in dissect
       s = self.do_dissect(s)
     File "/opt/dionaea/lib/dionaea/python/dionaea/smb/in-
clude/packet.py", line 416, in do_dissect
       s,fval = f.getfield(self, s)
     File "/opt/dionaea/lib/dionaea/python/dionaea/smb/in-
clude/fieldtypes.py", line 101, in getfield
       return  s[self.sz:], self.m2i(pkt, struct.un-
pack(self.fmt, s[:self.sz])[0])
   struct.error: unpack requires a bytes object of length 4
```