Entrenamiento de un sistema de detección de intrusos

Cindy Paola Castellanos Rodríguez
Nelson Javier García Suarez

Universitaria Agustiniana
Facultad de Ingenieras
Programa Ingeniería en Telecomunicaciones
Bogotá. D.C.
2020

Entrenamiento de un sistema de detección de intrusos

Cindy Paola Castellanos Rodríguez
Nelson Javier García Suarez

Director
Francisco Clemente Valle Díaz

Trabajo de grado para optar al título de Ingeniero en Telecomunicaciones

Universitaria Agustiniana
Facultad de Ingenieras
Programa Ingeniería en Telecomunicaciones
Bogotá. D.C.
2020

Resumen

En el presente documento veremos cómo se puede entrenar un sistema de detección de intrusos por medio de reglas específicas, las cuales nos sirven para alertar cualquier actividad maliciosa o vulnerabilidad de nuestra red, a lo largo del proceso de investigación y ejecución del trabajo se utilizaron varias herramientas, la principal y el eje de nuestra investigación fue el IDS SNORT el cual nos ayudó a conseguir nuestro objetivo además de obtener conocimientos base en seguridad informática un concepto que con el paso del tiempo y el avance de la tecnología se vuelve casi indispensable en nuestra vida laboral.

Palabras clave: IDS, Sistema de detección de intrusos, reglas, entrenamiento, vulnerabilidad.

Abstract

In this document we will see how an intrusion detection system can be trained by means of specific rules, which help us to alert any malicious activity or vulnerability of our network, throughout the investigation process and execution of the work they were used in several tools, the main and axis of our research was the IDS SNORT which helped us achieve our objective in addition to obtaining basic knowledge in computer security, a concept that with the passage of time and the advancement of technology becomes almost indispensable in our working

Keywords: IDS, Intrusion Detection System, rules, training, vulnerability.

Tabla de contenidos

Lista de figuras

Introducción

Los sistemas de detección de intrusos son cada vez más necesarios en todo tipo de entornos, con el constante cambio hacia nuevas tecnologías van apareciendo a su vez fallas en la seguridad, lo que nos lleva a comprender la necesidad de proteger nuestra información o equipos, ya que tienden a estar expuestos y a veces no somos conscientes del daño que nos pueden ocasionar si no protegemos nuestro entorno informático. Es por esto que un IDS puede ser una de las herramientas más efectivas ya que por medio de alertas configurados podemos identificar vulnerabilidades en nuestra red.

Planteamiento del problema

Conforme avanza la tecnología en nuestro siglo se tienen nuevos conceptos de ciberseguridad que le sorprenden a cada uno de nosotros. Diferentes métodos y amenazas están apareciendo paulatinamente lo que nos lleva a pensar en los procesos de ataque a una red, llevándonos en el momento a establecer cuentas de seguridad y mejoras de diseño para los procesos de ciberseguridad, teniendo en cuenta las amenazas de este tipo, identificamos que estas metodologías han existido durante años.

Un Sistema de Detección de Intrusos IDS es una herramienta para el monitoreo al interior de una red en búsqueda de actividades sospechosa, incorrectas o anómalas. En la actualidad existen variedad de IDS, Snort es uno de ellos de acceso libre el cual permite detectar, registrar y alertar ataques que estén previamente establecidos.

Justificación

La finalidad del presente proyecto es utilizar una herramienta que nos ayude a automatizar la identificación de comportamientos anómalos al interior de una red con el fin de proteger nuestro sistema, por medio de la creación de reglas en una herramienta específica, es importante establecer que la seguridad informática es un factor importante relacionado al entorno de nuestra carrera de Ingenieros en telecomunicaciones. Dentro de nuestro proceso investigativo se afianza conceptos de análisis de red, firmas o ataques conocidos, entre otros que pueden tener afectaciones directa o indirectamente.

Con el avance tecnológico de este siglo hemos visto como la mayoría de las empresas han dejado a un lado los documentos en físico para transfórmalos en formato digital, han subido la aplicaciones de uso diario a la nube, todo esto ocasiona un gran problema originado con el manejo de la información ya que presenta un alto riesgo de pérdida, puede ser muy vulnerable a cualquier tipo de amenaza y abre una ventana enorme en nuestra vida profesional en la cual nos podremos desenvolver y aplicar nuestros conocimientos.

Objetivos

Objetivo general

Automatizar el proceso de detección de comportamientos anómalos en una red de prueba mediante la ejecución de ataques al interior de la misma por medio del entrenamiento de un IDS.

Objetivos específicos

Adquirir los conocimientos necesarios para el manejo de las herramientas NMAP, Metasploit, Wireshark

Configurar un entorno de red para pruebas de vulnerabilidad sobre la LAN 10.0.2.0/24.

Crear reglas en sistema de detección de intrusos IDS Snort que permita visualizar alertas ante diferentes actividades maliciosas realizadas al interior de una red como lo son el escaneo de puertos o la explotación de vulnerabilidades.

Marco referencial

Estado del arte

En la investigación de técnica de detección de intrusos, realizada por, Rivero Perez (2014);

> El desarrollo de sistemas de detección de intrusos en redes de computadoras constituye un reto, debido a que, con el crecimiento de las redes de computadoras, aparecen, constantemente nuevos ataques basados en contenido. En este artículo además de hacer una descripción de los enfoques de detección de intrusos basados en firmas y en anomalías, constituye una revisión de las diferentes técnicas de aprendizaje automático a aplicar en las etapas de procesamiento de los datos para la detección temprana de cualquier tipo de anomalía. (p.52).

Por otro lado, (Brown, Anwar, Dozier, 2017);

> "Inspirados en el sistema inmunitario humano, exploraron el desarrollo de un nuevo sistema inmune artificial con sistema de detección múltiple (mAIS) para la detección de malware móvil basado en los flujos de información en las aplicaciones de Android. Los mAIS difieren de los AIS convencionales en que los conjuntos de detectores múltiples evolucionan simultáneamente a través de una selección negativa. Normalmente, el primer conjunto de detectores está compuesto por detectores que coinciden con los flujos de información asociados con las aplicaciones maliciosas, mientras que el segundo conjunto de detectores está compuesto por detectores que coinciden con los flujos de información asociados con las aplicaciones benignas. AIS ha presentado en este documento incorpora la selección de características junto con una técnica de selección negativa conocida como el método detector de división (SDM). Este nuevo mAIS se ha comparado con una variedad de AIS y mAIS convencionales utilizando un conjunto de datos de flujos de información capturados desde aplicaciones de Android maliciosas y benignas. Los resultados preliminares muestran que el mAIS de nuevo diseño supera a los AIS convencionales y los mAIS en términos de precisión y tasa de falsos positivos de detección de malware. Este documento finaliza con una discusión sobre cómo los mAIS se pueden usar para resolver problemas dinámicos de ciberseguridad, así como una discusión de nuestra investigación futura. Este enfoque logró una precisión del 93.33% con una tasa de falsos positivos del 0.00%." (p.1).

Este documento finaliza con una discusión sobre cómo los mAIS se pueden usar para resolver problemas dinámicos de ciberseguridad, así como una discusión de nuestra investigación futura.

Según, (Gutierrez del Moral, 2014);

“Investigadores de las universidades Politécnica de Valencia, Burgos y Salamanca han desarrollado una nueva herramienta de ayuda a la detección precoz de ataques informáticos denominada RT-MOVICAB-IDS (Real-Time Mobile Visualisation Connectionist Agent-Based IDS). Su principal avance reside en la acotación temporal del proceso de detección de intrusiones mediante la visualización grafica de tráfico en una red de ordenadores. El sistema utiliza para ellos técnicas de Inteligencia Artificial: la herramienta genera un informe visual que permite al administrador de red detectar de forma sencilla y rápida un posible ataque y, a partir de ahí, iniciar todo el protocolo de protección del servidor.”(p. 518).

Dentro de los procesos de análisis en la configuración de un sistema de detección de intrusos, se debe contemplar significados como lo expone, (Martinez Lopez, 2010) indicando temas como ataques pasivos, activos, detección de anomalías. (pp. 9-23).

En el artículo relacionado por (Amador, Arboleada, & Bedon, 2006); habla sobre las vulnerabilidades que puede presentar un red interna, con ello utilizar sistemas de detección de intruso opens source como Snort, adicional realizan procesos de integración con inteligencia artificial. (p. 1)

La aplicación NMAP es una de tantas aplicaciones para el escaneo de puertos y servicios que usaremos principalmente, tiene un sistema de scripts que nos ayudara a realizar el proceso, los script que maneja la aplicación es una de las características más potentes, en cual permite a usuarios escribir los comandos para automatizar una amplia variedad de tareas de red.

Metasploit, es una aplicación relacionada con un curso de offensive security para temas de ética hacking, gratuito, online llamado Metasploit Unleashed.

Snort es una herramienta open source de sistema detección de intrusos, en la página web se puede interactuar escenarios de community en cual uno puedo ver reglas probadas por usuarios y funcionales para su uso, para ese proceso se debe registrarse y suscripción, procesos o videos interactivos de practica y enseñanza para afianzar los conocimientos de uso y mejora continua, toda su página se basa en reglas y procesos dirigidos a ataques específicos.

Wireshark es una analizado de captura de trafico de red, con esta aplicación podemos realizar filtros para para identificar los datos que sean requeridos para nuestro análisis. Los logs que genera Snort con compatibles con esta aplicación, Por tal motivo, el uso de Wireshark es de uso visual de los logs de Snort que generemos con la aplicación Snort el modo sniffer y NIDS.

Marco teórico

Es importante definir la priorización de mantener nuestros sistemas actualizados, sea a empresas pequeñas o medianas, enfocados en la información que circula por la red y confidencialidad e integración que manejemos realizando el uso de un sistema de detección de intrusos.

Sistema de detección de intrusos.

Es un sistema que realiza procesos de monitoreo de protocolos de una red interna, dice (Gomez Lopez, 2009).

> Las definiciones de la detección de intrusos, indicada anteriormente, es la propuesta por el NIST (Intitute of Standard and Technology)[BAC04], que la define como el proceso de monitorización de eventos que suceden en un sistema informáticos o red y análisis de dichos eventos en busca de signos de intrusiones. Estos sistemas están continuamente supervisando los componentes de la red y las personas o intrusos que están intentando entrar ilegalmente a ella.(p. 3).

Ciberseguridad.

Proceso que se relaciona con las prácticas para la de detección de amenazas de computadores, servidores, redes. Abarcando diferentes contextos según categorías comunes. Como indica Relacionado con seguridad de red, aplicaciones, información y operativa (kasperky, 2020).

Superficie de ataque.

Es la identificación de un entorno susceptible a una vulnerabilidad de red relacionada con componentes como firewall, routers, computadoras, servidores, aplicaciones, entre otros, como lo relaciona para perjudicar empresa, instituciones o personas (Romero Castro, et al, 2018 p.36)

Concepto de seguridad informática y seguridad de la información.

Los dos conceptos como seguridad informática y seguridad de la información relaciona un punto de vista diferente; seguridad informática se basa el procesos y técnicas, seguridad de la información todo aquello que contenga información delicada, como lo expresa (Romero Castro, et al, 2018 pp. 13-14).

Marco legal

Congreso de la Republica (5 enero de 2009). Por medio de la cual se modifica el Código Penal, se crea un nuevo bien jurídico tutelado - denominado “de la protección de la información y de los datos”- y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones [LEY 1273 DE 2009].[Diario Oficial No. 47.223 de

5 de enero de 2009]. Recuperado de: http://www.secretariasenado.gov.co/senado/basedoc/ley_1273_2009.html

Congreso de la Republica (18 Octubre de 2012). Artículo 4. *PRINCIPIOS PARA EL TRATAMIENTO DE DATOS PERSONALES. En el desarrollo, interpretación y aplicación de la presente ley, se aplicarán, de manera armónica e integral* [Ley 1581 de 2012]. [Diario Oficial No. 48.587 de 18 de octubre de 2012]. Recuperado de: http://www.secretariasenado.gov.co/senado/basedoc/ley_1581_2012.html.

Organización Internación de Normalización (octubre 2005). Permite la gestión y control de los riesgos de la seguridad de la información en las organizaciones para las cuales la información y la tecnología son activos importantes de su negocio.[ISO 27001]. Recuperado de: https://www.isotools.org/normas/riesgos-y-seguridad/iso-27001/

Metodología

Nuestro proyecto “Entrenamiento de un sistema de detección de intrusos”, está enfocado en la automatización de detección de actividades maliciosas al interior de una red por medio de un acceso no autorizado, basándonos en el análisis detallado del tráfico de red o uso inadecuado de algún dispositivo. La metodología cualitativa nos permite buscar la información, manipularla y aplicarla en diferentes fases del proyecto, ejecutando un proceso de múltiples vectores de ataque, utilizando una herramienta disponible de forma gratuita en internet en la cual nos permita realizar un diseño de red y así poder demostrar el desarrollo de un intento de actividad intrusiva en donde se podrá evidenciar las posibles fallas de una red y lo vulnerable que puede ser un equipo.

Cronograma

En la columna de actividades se relacionan las tareas realizadas para este proyecto.

ACTIVIDAD	INICIO DEL PLAN	DURACIÓN DEL PLAN	INICIO REAL	DURACIÓN REAL	Periodo resaltado: PORCENTAJE COMPLETADO
Investigacion base de Sistema de deteccion de intrusos	1	3	1	5	90%
Investigacion Herramientas de Ciberseguridad	1	5	2	5	90%
Investigacion de amenazas y metodos de ataque	2	3	3	6	95%
Consulta de proyectos relacionados	1	8	1	8	95%
Vulnerabilidades de la red 10.0.2.0/24	6	2	6	3	95%
Instalacion Snort	6	1	6	1	100%
Identificar puertos abiertos en una red	7	1	7	1	100%
Analizar trafico evidenciado	7	2	7	3	100%
Creacion de reglas	9	5	9	5	100%
Implementacion de red pruebas de ataque	4	10	5	9	100%
Pruebas funcionales y resultados	12	3	12	3	100%

Figura 1. Cronograma de actividades. Elaboración propia.

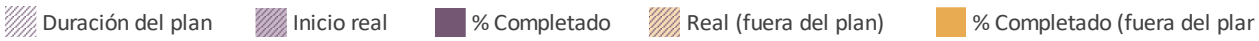

Figura 2. Referencias de tiempo cronograma. Elaboración propia.

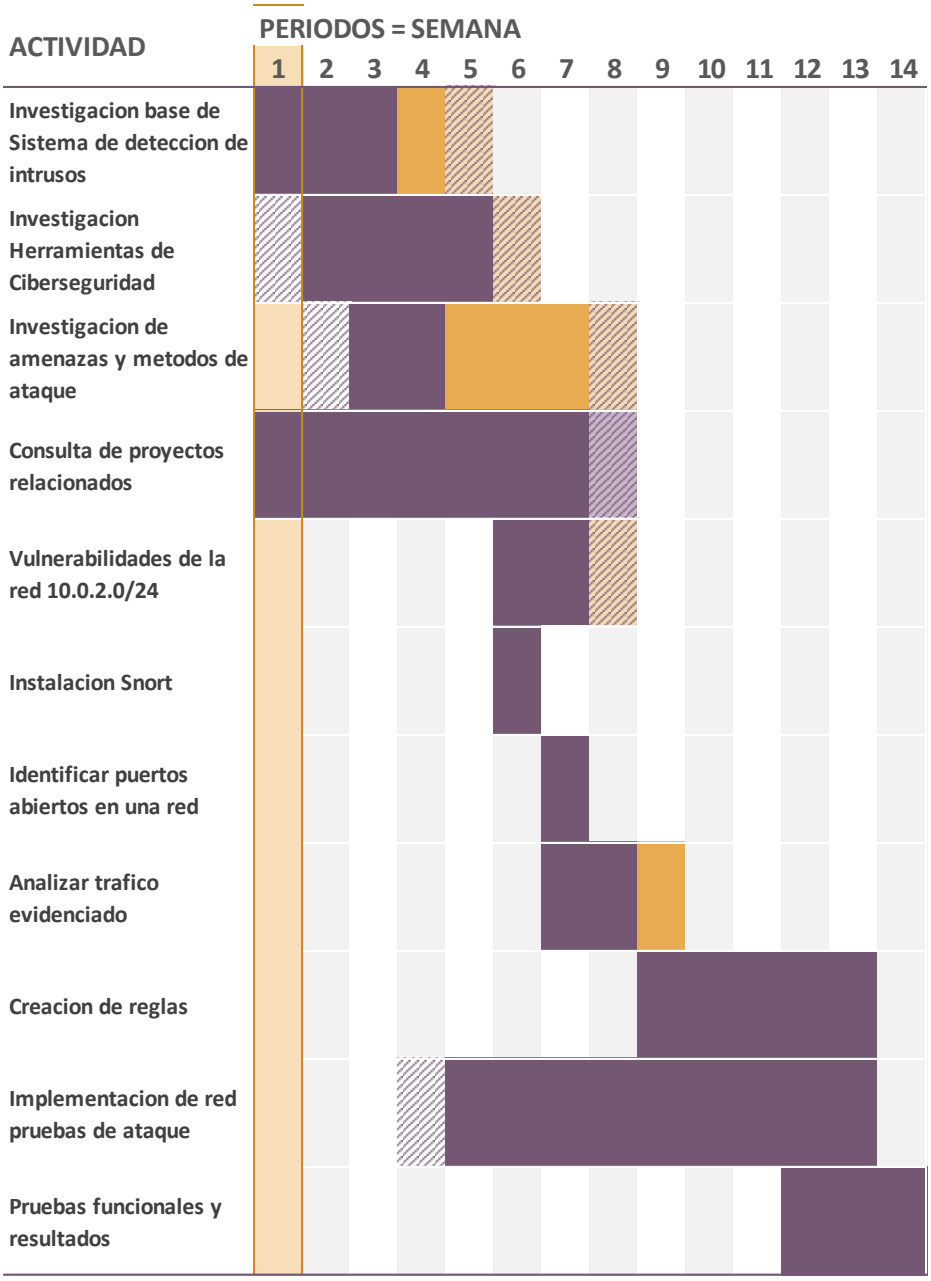

Figura 3. Diagrama Gantt. Elaboración propia.

Procedimiento

Estructura de red

En el proceso de implementación se diseñó una red local con dos máquinas físicas y en ellas incluimos en cada una 2 máquinas virtuales como se evidencia en la (figura 4).

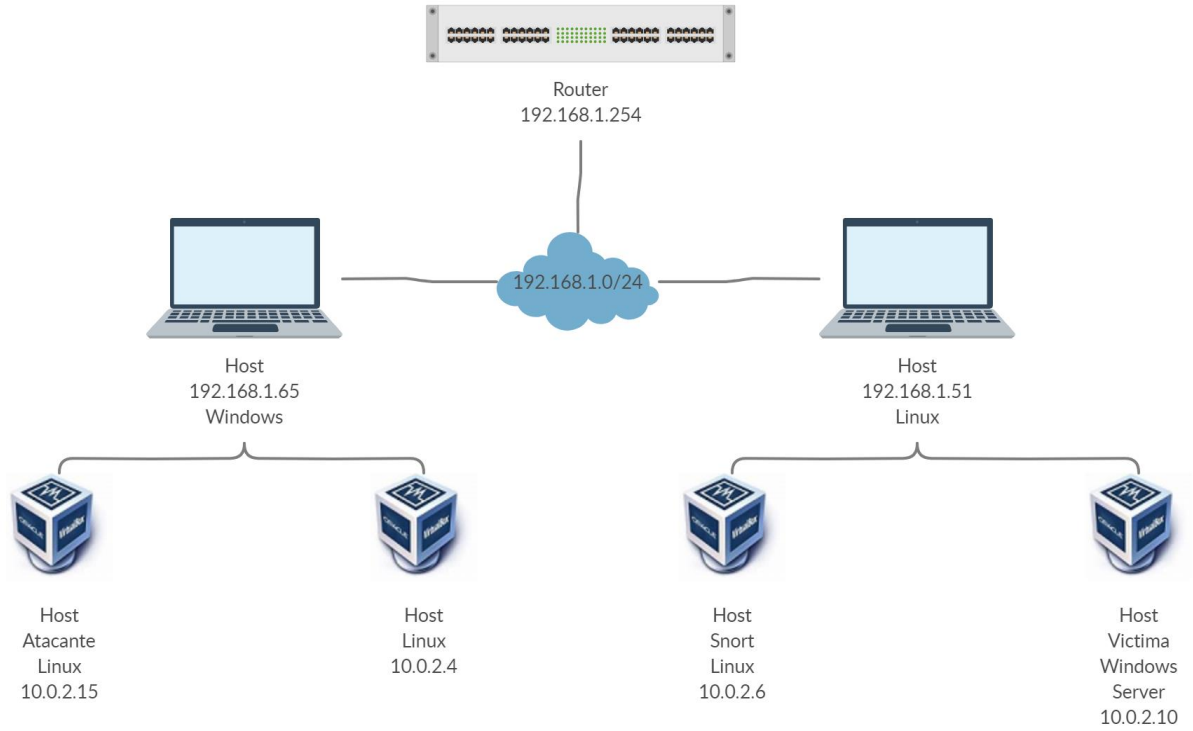

Figura 4. Diagrama de red. Elaboración propia.

Para el desarrollo del proyecto se han utilizado los sistemas operativos GNU Linux (Kali Linux), Windows server 2012 y las siguientes aplicaciones

Nmap: Efectúa rastreo de puertos

Metasploit: Nos ayuda a investigar vulnerabilidades de seguridad

Whireshark: Analizador de trafico

Snort: Identifica tráfico malicioso en la red

Kali Linux: Software para auditoria y seguridad informática general

NMAP

Nmap ("Network Mapper") es un código abierto y gratuito (licencia) utilidad para el descubrimiento de redes y la auditoría de seguridad. Muchos administradores de sistemas y redes también lo encuentran útil para tareas como el inventario de la red, la gestión de programas de

actualización del servicio y la supervisión del tiempo de actividad del host o del servicio. Nmap utiliza paquetes de IP sin procesar de formas novedosas para determinar qué hosts están disponibles en la red, qué servicios (nombre y versión de la aplicación) ofrecen esos hosts, qué sistemas operativos (y versiones de SO) están ejecutando, qué tipo de filtros de paquetes / firewalls están en uso y docenas de otras características. Fue diseñado para escanear rápidamente redes grandes, pero funciona bien con hosts únicos. Nmap se ejecuta en todos los principales sistemas operativos de computadoras, y los paquetes binarios oficiales están disponibles para Linux, Windows y Mac OS X. Además del ejecutable clásico de línea de comandos Nmap,Zenmap), una herramienta de depuración, redirección y transferencia de datos flexible (Ncat), una utilidad para comparar resultados de escaneo (Ndiff) y una herramienta de análisis de respuesta y generación de paquetes (Nping). (NMAP, s.f.)

Metasploit framework

El Metasploit Framework (MSF) (Universidad Nacional Autonoma de Mexico, 2018) La versión gratuita y limitada de Metasploit framework Community es una herramienta que permite ejecutar y desarrollar exploits contra sistemas objetivos. Actualmente se encuentra integrado con Kali Linux, una distribución de Linux con diversas herramientas orientadas a la seguridad y es ampliamente utilizado para realizar pruebas de penetración.

Wireshark

(The Wireshark team, 1990) Es el analizador de protocolos de red más importante y más utilizado del mundo. Le permite ver lo que está sucediendo en su red a un nivel microscópico y es el estándar en muchas empresas comerciales y sin fines de lucro, agencias gubernamentales e instituciones educativas. El desarrollo de Wireshark prospera gracias a las contribuciones voluntarias de expertos en redes de todo el mundo y es la continuación de un proyecto iniciado por Gerald Combs en 1998.

Kali Linux

Es una herramienta para el uso de técnicas de hacking ético, destinada para pruebas de penetración y auditoria, contiene varios cientos de herramientas incluidas orientadas a diversas tareas de seguridad de la información, como pruebas de penetración, investigación de seguridad, informática forense e ingeniería inversa, este software es financiado por Offensive Security. (Kali Linux, 2013)

Snort

(Cisco Systems, 1998) Snort es el sistema de prevención de intrusiones (IPS) de código abierto más importante del mundo. Snort IPS usa una serie de reglas que ayudan a definir la actividad de red maliciosa y usa esas reglas para encontrar paquetes que coincidan con ellas y genera alertas para los usuarios. También se puede implementar en línea para detener estos paquetes. Snort tiene tres usos principales: como rastreador de paquetes como tcpdump, como registrador de paquetes, que es útil para la depuración del tráfico de red, o puede usarse como un sistema de prevención de intrusiones en la red en toda regla. Snort se puede descargar y configurar tanto para uso personal como comercial.

Instalación

Kali Linux.

Kali Linux nos ofrece tres tipos de imágenes diferentes Installer, NetInstaller y Live cada uno para las arquitecturas de 32-bit y 64-bit. Ejecutamos la instalación de Kali Linux utilizando la imagen Installer y para no realizar dos procesos de instalación utilizamos un paquete pre configurado llamado OVA.

En el paquete Installer se realizó la instalación de la aplicación Snort, el ova es para nuestra maquina atacante. Los requerimientos para la imagen installer son:

CPU: procesador ARM, i386 o x64
Espacio de almacenamiento: 15 GB o mas
Memoria RAM: 1 GB como mínimo para versión de escritorio y 2 GB recomendado
Tarjeta de red: Cableada o WI-FI para las pruebas

Figura 5. Características de configuración Kali Linux. Elaboración propia.

Para el proceso de instalación de kali Linux no nos centraremos desde la parte de inicial de la instalación de nuestro sistema operativo, para el proceso de implementación con software de

virtualización para arquitecturas x86/adm64, se encuentra múltiples procesos en internet relacionado con Virtual Box y Vmware, para su respectiva instalación.

En la figura (6), seleccionamos la opción Graphical Installer, esta es una versión GUI o entorno gráfico.

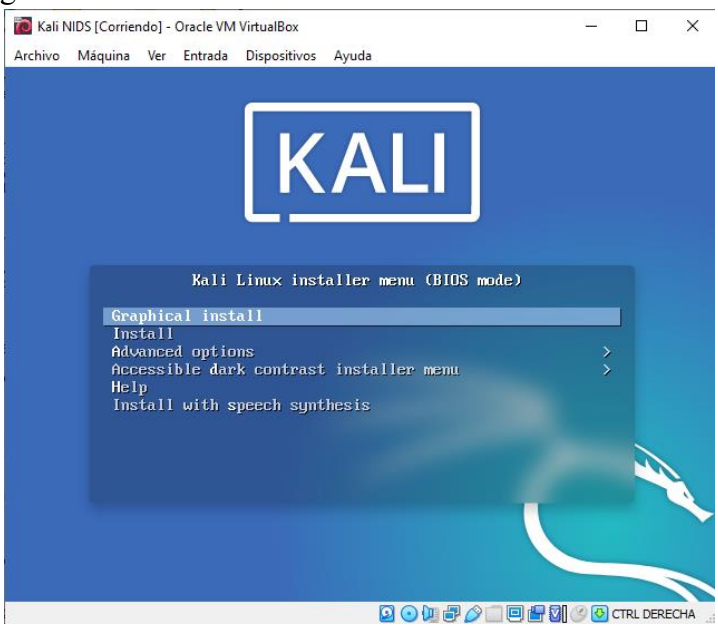

Figura 6. Instalación Kali Linux - Interface Grafica. Elaboración propia.

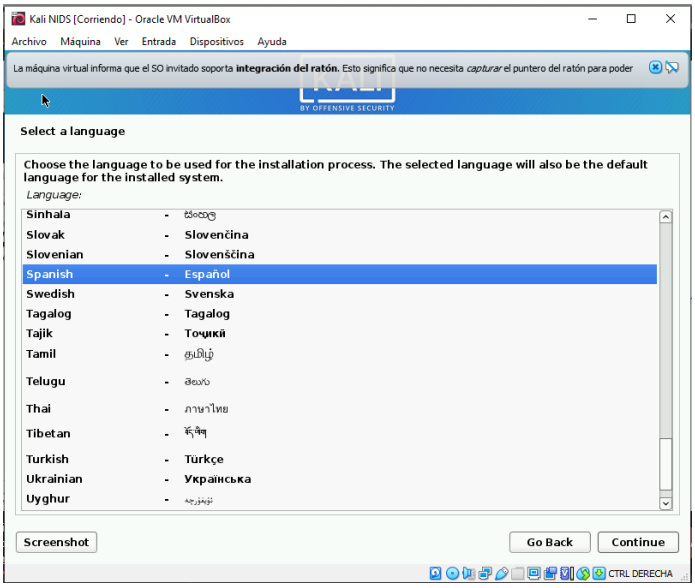

Figura 7. Instalación Kali Linux - Selección lenguaje. Elaboración propia.

Después de la asignación del lenguaje, solicita la asignación de ubicación, nombre de usuario, contraseña, no es necesario especificar estos pasos ya que son muy comunes con las instalaciones de software, continuamos desde el proceso de partición de disco, seleccionando todo el disco.

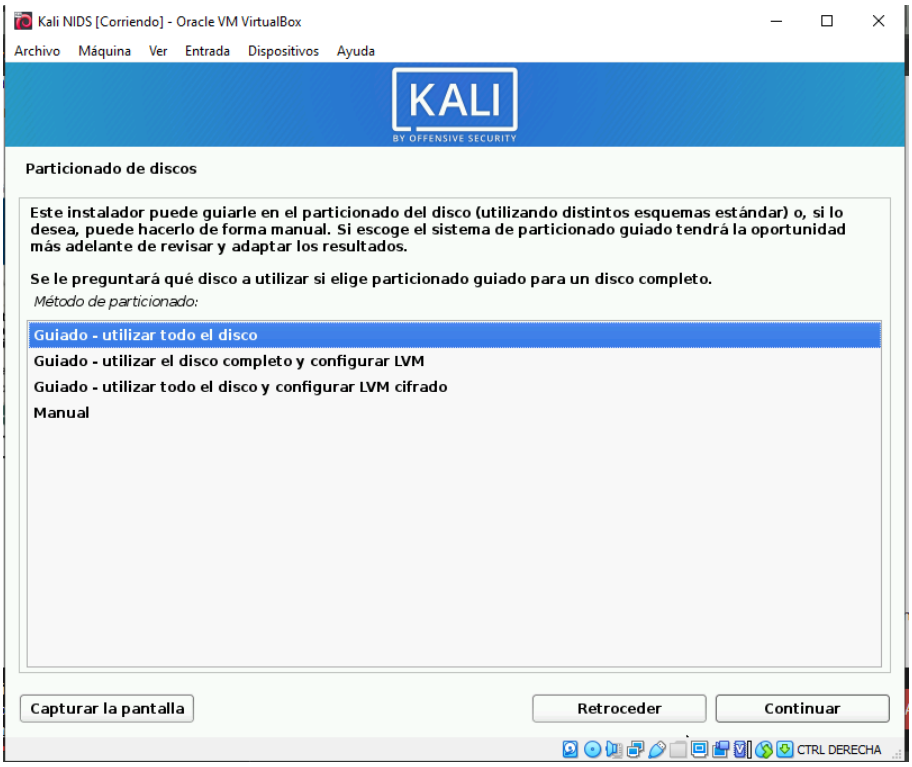

Figura 8. Instalación Kali Linux - Partición de discos. Elaboración propia.

En el proceso de continuación de partición de discos seleccionaremos la opción todos los ficheros en una partición, no tenemos la necesidad de explorar las demás opciones contempladas ya que la finalidad de este proyecto es la implementación de un NIDS. Por lo tanto es la recomendada para la ejecución de nuestro sistema.

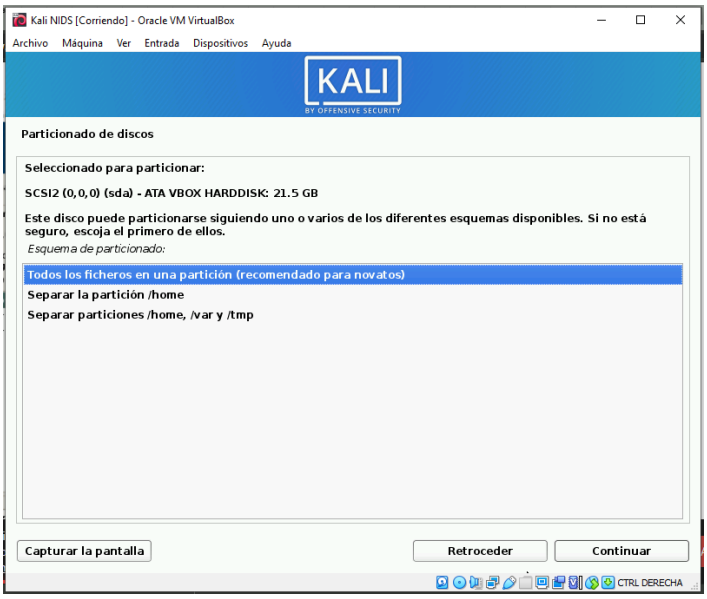

Figura 9. Instalación Kali Linux - Ficheros de partición. Elaboración propia.

En el siguiente paso preguntara el menú si deseamos hacer una réplica de la red, generalmente para las actualizaciones de los programas, elegimos la opción sí. Dejamos que el sistema instale las aplicaciones por defecto, terminado esto se podrá ejecutar para su uso.

La instalación de la versión ova, es un proceso más sencillo. En la aplicación de Virtual Box en la opción archivo seleccionamos importar servicio virtualizado, en ese paso indicamos la ruta donde se encuentra nuestro archivo correspondiente.

En el momento de dar siguiente nos genera una vista de la configuración que realizara automáticamente en la aplicación de Virtual Box, como memoria RAM, tamaños de disco duro, total de núcleos, entre otros. Clic el importar.

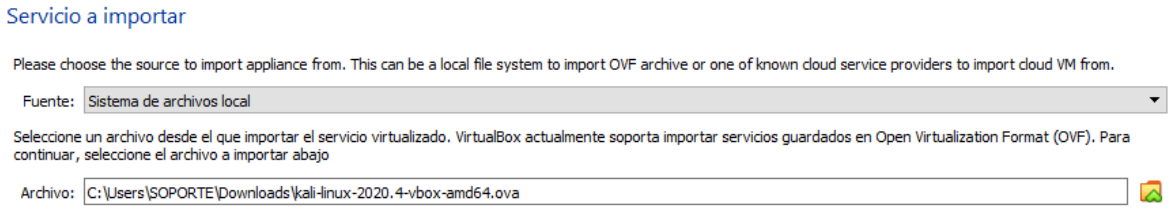

Figura 10. Instalación Kali Linux OVA. Elaboración propia.

Con el comando grep VERSION /etc/os-release identificamos la versión instalada de nuestro sistema operativo, este proceso se ejecuta como apoyo de identificación de versión.

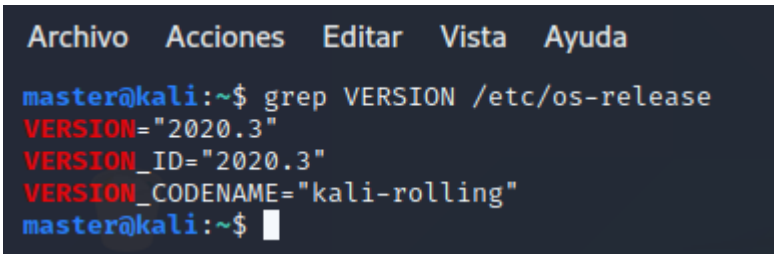

Figura 11. Versión Kali Linux. Elaboración propia.

Nmap,Metasploit, Wireshark

Las aplicaciones Nmap, Metasploit y Wireshark ya se encuentran instaladas en el sistema operativo de Kali Linux, se relaciona la versión instalada de cada una de ellas.

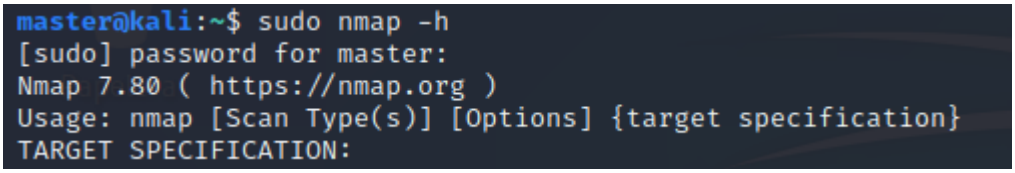

Figura 12. Versión Nmap. Elaboración propia.

Figura 13. Versión Metasploit. Elaboración propia.

Figura 14. Versión Wireshark. Elaboración propia.

Snort.

Snort es herramienta basada tcpdump, un capturados de paquetes en sistemas UNIX en cual obtiene información de la capa enlace de datos a través del uso de la librerías lipcap. La cual se añadió la posibilidad de analizarlos en búsqueda de actividad maliciosa conocida. En la versión 2.9, se introduce la librería de adquisición de datos conocida como DAQ, cuyo objetivo es reemplazar las llamadas directas a las funciones a la librería lipcap por una capa extracta. (Cantos Polaino, 2012).

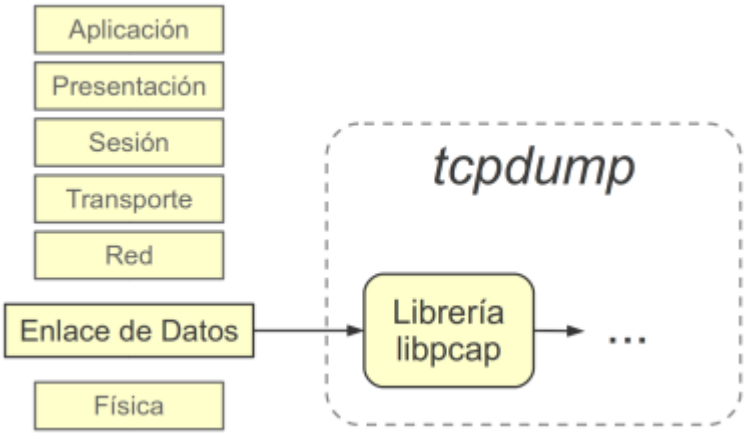

Figura 15. Modelo OSI paquetes tcpdump tomado de (Cantos Polaino, 2012)

Los componentes configurados para Snort están encargados de realizar tareas específicas capturadas por el DAQ y reenvía a Snort. En la investigación realizada por (Cantos Polaino, 2012) van en el siguiente orden:

"1. Packet Decode Engine (motor de descodificación de paquetes o simplemente descodificador de paquetes): Una vez que el módulo DAQ envía el paquete a Snort, este pasa por el descodificador de paquetes (Packet Decoder), que se encarga de almacenar toda la información de cada paquete que llegue de la red, como por ejemplo protocolos, IPs de origen y destino, etc., en una estructura de datos para su posterior procesamiento.

2. Preprocessor (preprocesador): Los preprocesadores fueron introducidos en la versión 1.5 de Snort. Gracias a ellos se pueden ampliar las funcionalidades de Snort permitiendo a usuarios y programadores crear módulos (plug-ins) dentro de Snort de una manera sencilla. Una vez que se descodifica el paquete, se ejecutan los preprocesadores, que pueden analizar e incluso modificar el paquete en cuestión, dependiendo del objetivo de cada preprocesador. También, pueden lanzar alertas, clasificar o descartar un paquete antes de enviarlo al motor de detección (Detection Engine), que cuenta con un alto coste computacional.

3. Detection Engine (Motor de detección): Este componente es considerado como el corazón de Snort. Toma información del Packet Decoder y de los preprocesadores e inspecciona el contenido del paquete para compararlo a través de su módulo, o plug-in, de detección (Detection) con los patrones de la base de firmas.

4. Output (Salida de Snort): Cuando se ha detectado un paquete sospechoso, ya sea porque preprocesador lo ha decidido o porque cumple con una regla concreta, este módulo de salida genera una alerta, en el formato que se especifique en el archivo de configuración de Snort."

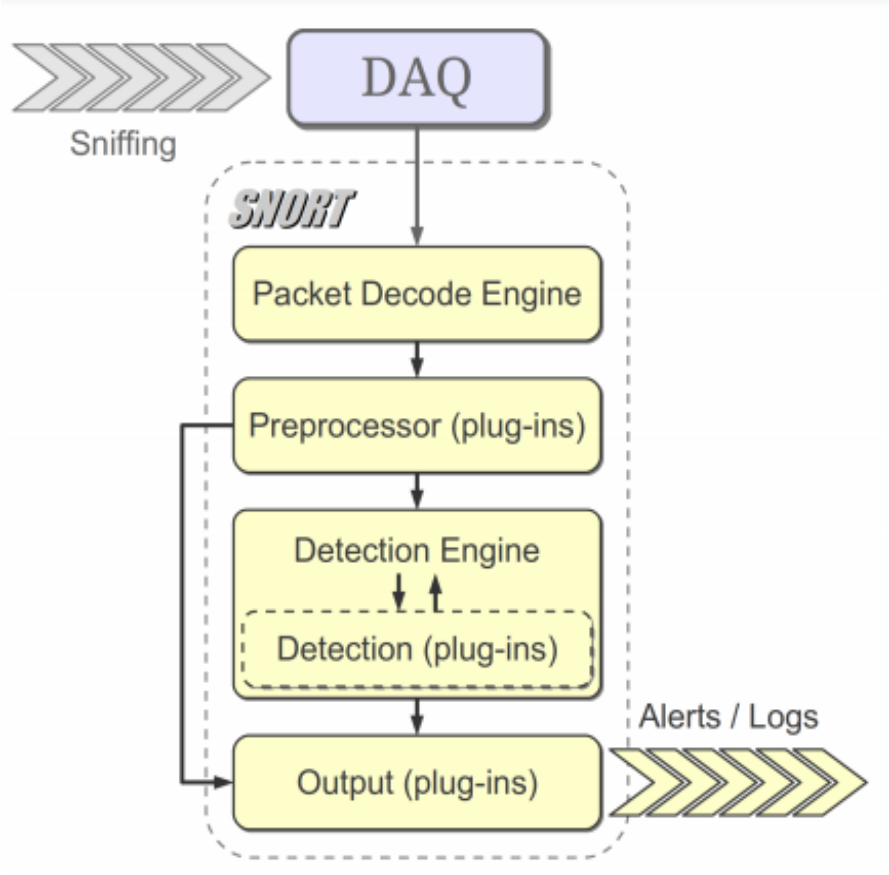

Figura 16. DAQ y componentes Snort tomado de (Cantos Polaino, 2012)

Con base en la anterior información procedemos en la implementación del aplicativo Snort, realizando los siguientes pasos en un equipo Kali Linux ya configurado.

Se debe ir directamente a la página de Snort https://www.snort.org/ para descargar los paquetes relacionados con DAQ y Snort,

Crear carpetas repositorio

```
[sudo] password for kali:
┌──(root💀kali)-[/home/kali]
└─# mkdir snort_source
```

Figura 17. Creación carpeta repositorio. Elaboración propia.

Instalar bison flex

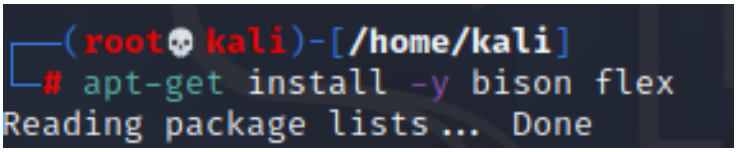

Figura 18. Instalación paquetes bison flex. Elaboración propia.

Descargar archivo Snort y DAQ, en la sección Get Started seleccionar la versión del sistema operativo a instalar.

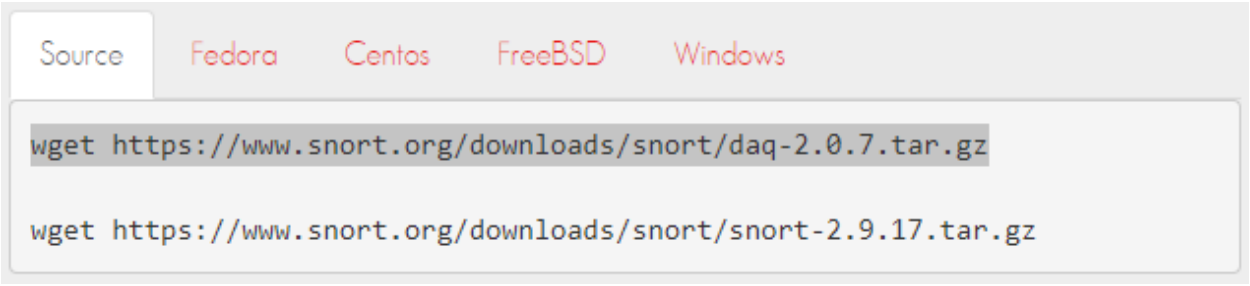

Figura 19. Instaladores DAQ y Snort Sistema operativo Debian. Elaboración propia.

En comando wget realiza el proceso de descarga de los paquetes, para nuestro entorno ejecutamos el comando dentro de la carpeta creada anteriormente.

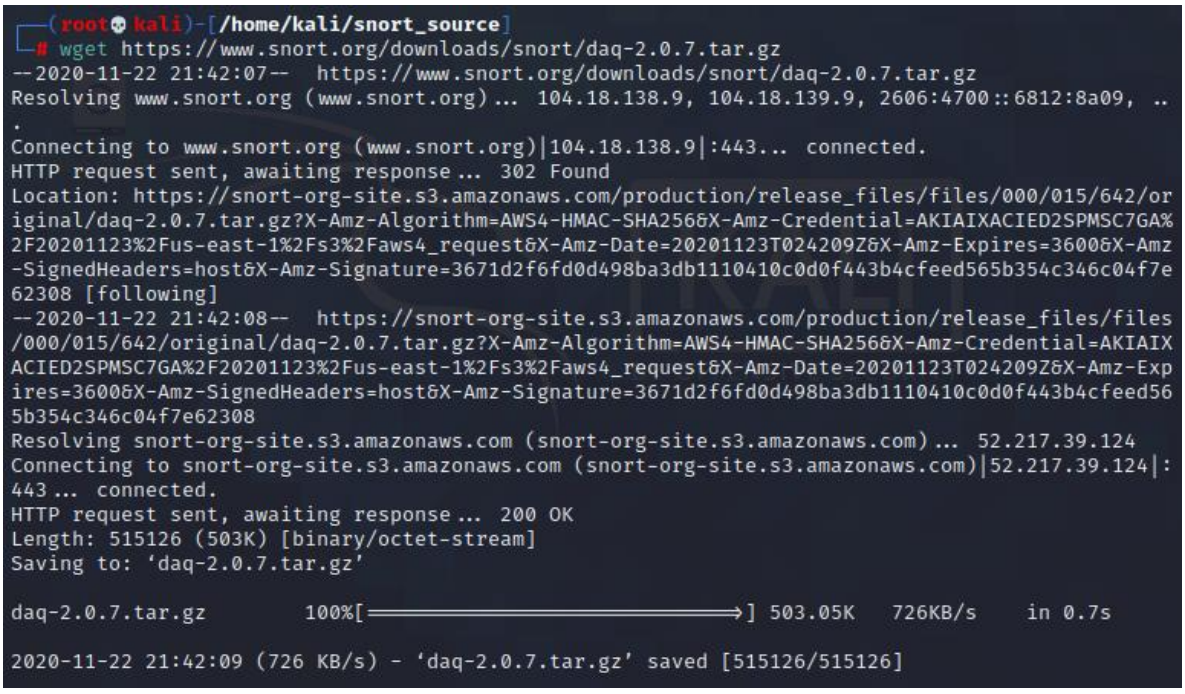

Figura 20. Muestra descarga de paquete DAQ. Elaboración propia.

Descomprimir archivos

Figura 21. Descomprir paquetes DAQ. Elaboración propia.

Figura 22. Descomprimir paquete Snort. Elaboración propia.

Instalar, este proceso se ejecuta para el DAQ y Snort

```
┌──(root💀 kali)-[/home/kali/snort_source]
└─# ./configure && make && sudo make install
```

Figura 23. Comando instalación DAQ. Elaboración propia.

```
┌──(root💀 kali)-[/home/kali/snort_source]
└─# ./configure --enable-sourcefire && make && sudo make install
```

Figura 24. Comando instalación Snort. Elaboración propia.

Se recomienda que el Kali Linux tenga las últimas actualizaciones de paquete o el su defecto se realizar un upgrade de las librerías lipcap.

Después de las ejecutar los parámetros de instalación del aplicativo, realizamos el proceso de verificación con el comando "Snort –V", el cual nos muestra la versión instalada, como se visualiza en la (figura 2.).

```
root@kali:/home/master# snort -v
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "eth0".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.16.1 GRE (Build 140)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2020 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=1525)
```

Figura 25. Versión Snort. Elaboración propia.

Análisis de información Aplicaciones

En el proceso de análisis de información, recopilamos datos asociados al funcionamiento de los aplicativos NMAP, METASPLOIT, los cuales como se dijo anteriormente, ya se encuentran incluidos en software Kali Linux. En conjunto creamos un entorno local para la puesta en marcha de las aplicaciones referenciadas con el fin de ejecutar los análisis correspondientes basados en los objetivos.

ICMP

Internet Control Messanging Protocol (protocolo de mensajes de control de internet), definido en el RFC 792, este protocolo IP no proporciona mecanismos de control, por eso fue necesario su creación para identificar fallos y problemas como mecanismos de control. Los mensajes ICMP envía la siguiente información. (Redes locales y globales).

Confirmación de host (Request / Response): Determina si un nodo está en funcionamiento, envía una petición Echo request, el nodo recibe un Echo response, estos comando son la base de comandos ping y traceroute.

Destino inalcanzable (Destino Unreachable): este mensaje notifica a un equipo que el destino o servicio es inalcanzable.

En el estudio realizado se identificaron varios aspectos relacionados con ataques de reconocimiento, entre ellos estaba el anteriormente conocido como ping de la muerte que son múltiples envíos de paquetes ICMP para la denegación de un servicio, los nuevos sistemas en la actualidad utilizan un tipo de bloqueo externo. Es necesario relacionar que si los sistemas no se encuentran parchados pueden presentar vulnerabilidades las cuales pueden ser utilizadas por los atacantes.

Indica (PUREVPN), sobre ping flood como una inundación de servicio mediante el protocolo ICMP dejando el objetivo hasta que no sea funcional, este proceso se puede ejecutar a varias máquinas dentro de una misma red, se puede efectuar por medio de botnet, el cual está basado en un conjunto de robots informáticos que pueden controlar los servidores u ordenadores infectados.

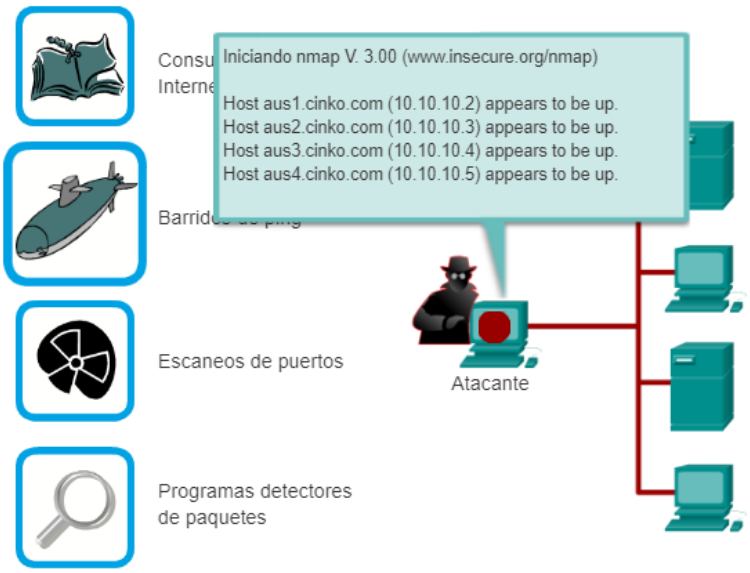

Figura 26. Ataque de reconocimiento, barrido de ping (Tecnologico Nacional de Mexico)

Escaneo de puertos

El escaneo de puertos es una parte fundamental para el desarrollo de auditorías de red, permitiéndonos identificar puertos abiertos en los hosts. Dentro de la aplicación metasploit está incorporado el escaneo de puertos con el comando db_nmap, el cual nos permite realizar ciertas consultas predeterminadas (Martinez López, 2010).

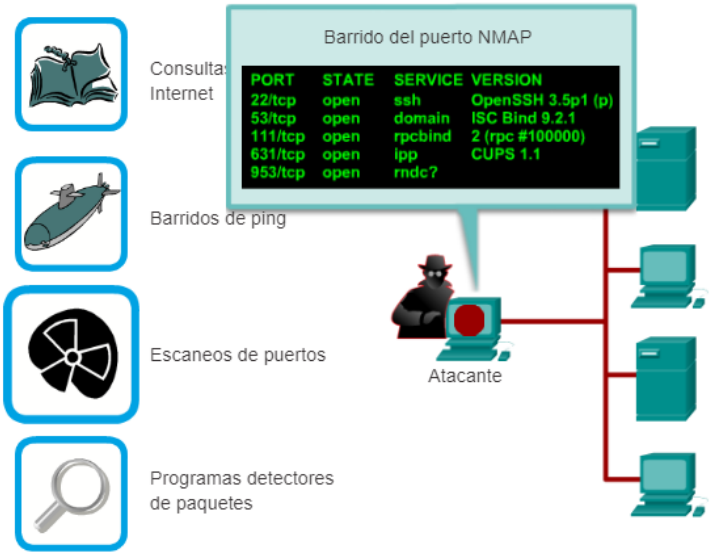

Figura 27. Ataque de reconocimientos, escaneo de puertos (Tecnologico Nacional de Mexico)

El parámetro –sV Interroga al conjunto de puertos abiertos detectados para tratar de descubrir servicios y versiones en puertos abiertos. El parámetro -P Especifica el rango de puertos a analizar (Lyon, 1997).

```
msf5 > db_nmap -sV 10.0.2.7 -Pn
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-25 20:02 -05
[*] Nmap: Nmap scan report for 10.0.2.7
[*] Nmap: Host is up (0.88s latency).
[*] Nmap: Not shown: 986 closed ports
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 135/tcp   open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 1433/tcp  open  ms-sql-s     Microsoft SQL Server 2012 11.00.3128; SP1+
[*] Nmap: 2383/tcp  open  ms-olap4?
[*] Nmap: 8009/tcp  open  ajp13?
[*] Nmap: 8080/tcp  open  http-proxy   Apache-Coyote/1.1
[*] Nmap: 49152/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49156/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49158/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49163/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 172.92 seconds
msf5 >
```

Figura 28. Escaneo puertos Sistema operativo WS. Elaboración propia.

```
msf5 > db_nmap -sV 10.0.2.4 -Pn
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-25 19:57 -05
[*] Nmap: Nmap scan report for 10.0.2.4
[*] Nmap: Host is up (0.00073s latency).
[*] Nmap: Not shown: 977 closed ports
[*] Nmap: PORT     STATE SERVICE     VERSION
[*] Nmap: 21/tcp   open  ftp         vsftpd 2.3.4
[*] Nmap: 22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
[*] Nmap: 23/tcp   open  telnet      Linux telnetd
[*] Nmap: 25/tcp   open  smtp        Postfix smtpd
[*] Nmap: 53/tcp   open  domain      ISC BIND 9.4.2
[*] Nmap: 80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: 111/tcp  open  rpcbind     2 (RPC #100000)
[*] Nmap: 139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 512/tcp  open  exec        netkit-rsh rexecd
[*] Nmap: 513/tcp  open  login       OpenBSD or Solaris rlogind
[*] Nmap: 514/tcp  open  tcpwrapped
[*] Nmap: 1099/tcp open  java-rmi    GNU Classpath grmiregistry
[*] Nmap: 1524/tcp open  bindshell   Metasploitable root shell
[*] Nmap: 2049/tcp open  nfs         2-4 (RPC #100003)
[*] Nmap: 2121/tcp open  ftp         ProFTPD 1.3.1
[*] Nmap: 3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
[*] Nmap: 5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
[*] Nmap: 5900/tcp open  vnc         VNC (protocol 3.3)
[*] Nmap: 6000/tcp open  X11         (access denied)
[*] Nmap: 6667/tcp open  irc         UnrealIRCd
[*] Nmap: 8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
[*] Nmap: 8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 12.82 seconds
```

Figura 29. Escaneo de puertos Sistema operativo Linux. Elaboración propia.

El parámetro –script vuln descubre las vulnerabilidades más conocidas; El parámetro –F Limita el análisis a los 100 puertos más comunes, Por defecto, se usan los 1000 los más comunes (Lyon, 1997).

```
msf5>db_nmap -F --script vuln 10.0.2.4
```

Figura 30. Escaneo de vulnerabilidad con límite de puertos. Elaboración propia.

El resultado de la ejecución del comando anterior nos arroja las siguientes vulnerabilidades

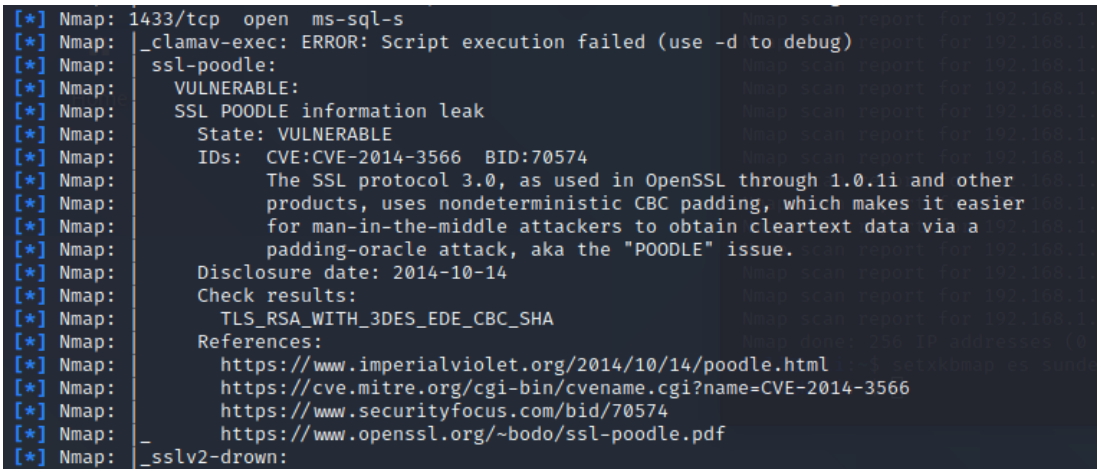

Figura 31. Vulnerabilidad Windows server 10.0.2.7. Elaboración propia.

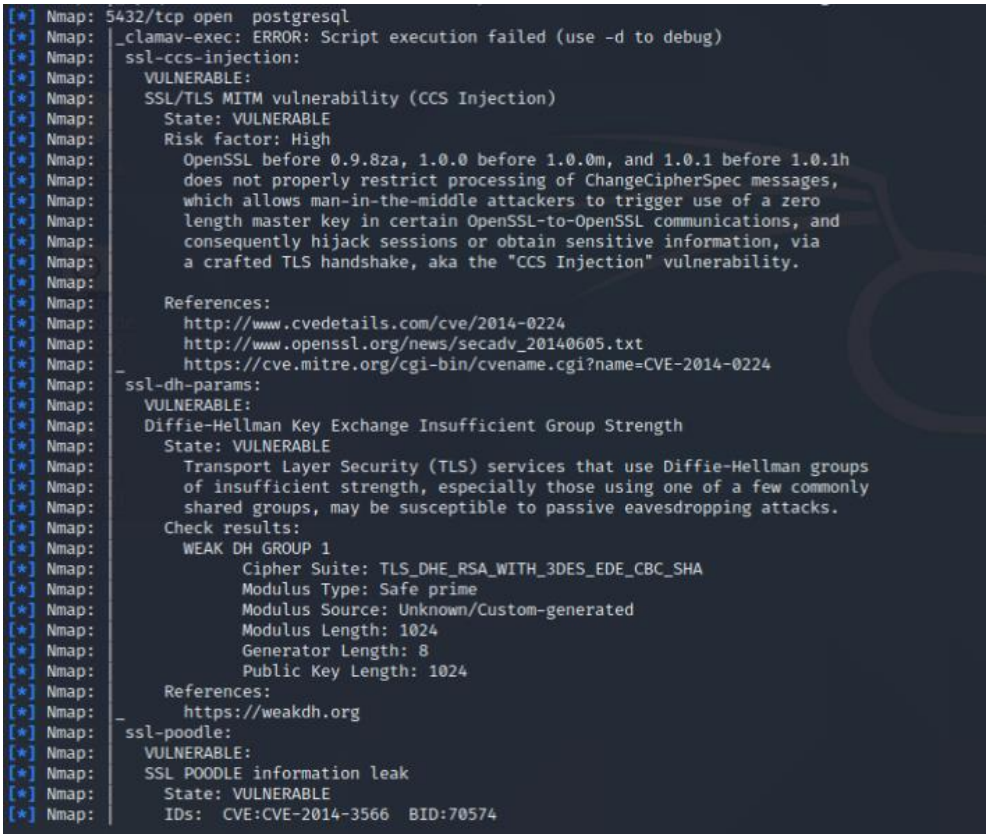

Figura 32. Vulnerabilidad Linux 10.0.2.4. Elaboración propia.

Denegación de servicios

Buscar que un servicio informático deje de funcionar, este proceso puede ser relacionado con servicios como correo electrónico, servidores web o cualquier otro tipo. Inicialmente el proceso de

un DOS puede ser directo a una maquina o muchas máquinas, el este proceso se puede utilizar botnet, existen dos técnicas:

DOS: múltiple generación de peticiones a un servicio o máquina, consumiendo al máximo los recursos que ofrece el servicio.

DDOS: Distribución de un ataque utilizando n cantidad de ordenadores o direcciones IP, ejecutándose al mismo servidor o dirección IP.

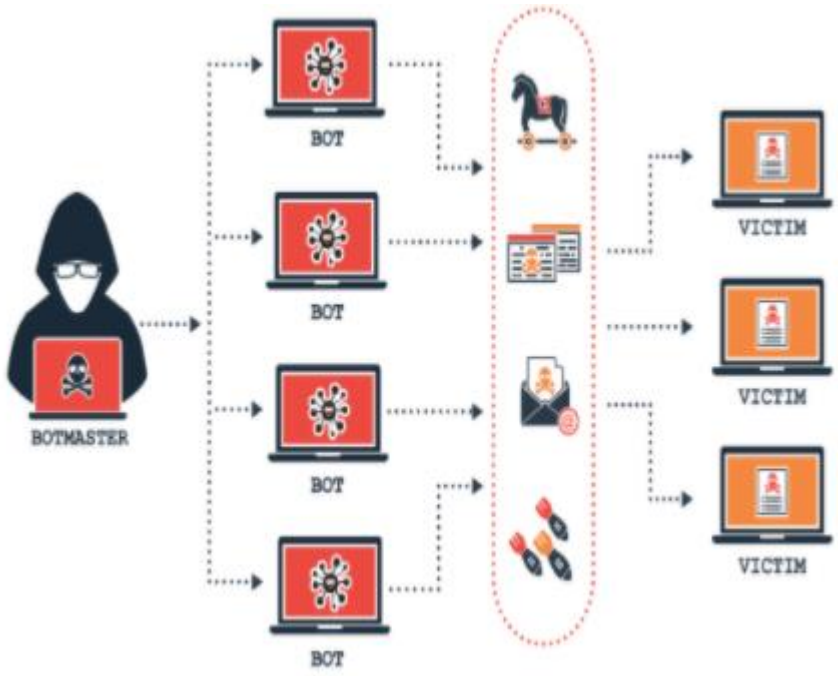

Figura 33. Denegación de Servicio DDOS. (ICMQ) (2019)

Identificar vulnerabilidad

En las vulnerabilidades obtenidas en el equipo 10.0.2.4, encontramos una relacionada con postgresql la cual podemos explotar con la aplicación Metasploit.

```
msf5 > search postgresql type:exploit

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   1  exploit/multi/http/manage_engine_dc_pmp_sqli          2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
   2  exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20  excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   3  exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   4  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
```

Figura 34. Búsqueda de exploit relacionado con Postgresql. Elaboración propia.

Utilizamos el exploit “exploit/Linux/postgres/postgres_payload” en cual relaciona algunas instalaciones Linux predeterminadas de PostgreSQL, la cuenta de servicio de postgres puede escribir en el directorio / tmp y también puede obtener bibliotecas compartidas UDF desde allí, lo que permite la ejecución de múltiples códigos. Este módulo compila un archivo de objeto

compartido de Linux, lo carga en el host de destino mediante el método UPDATE pg_largeobject de inyección binaria y crea una UDF (función definida por el usuario) a partir de ese objeto compartido. Debido a que la carga útil se ejecuta como el constructor del objeto compartido, no es necesario que se ajuste a versiones específicas de la API de Postgres. (WonderHowTo Network, 2008).

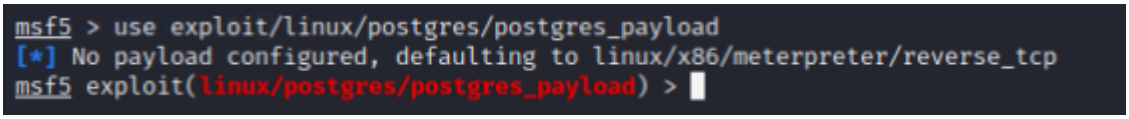

Figura 35. Utilización del exploit Linux/postgres/postgres_payload. Elaboración propia.

Para realizar la explotación, debemos configurar el parámetro RHOST de la maquina objetivo

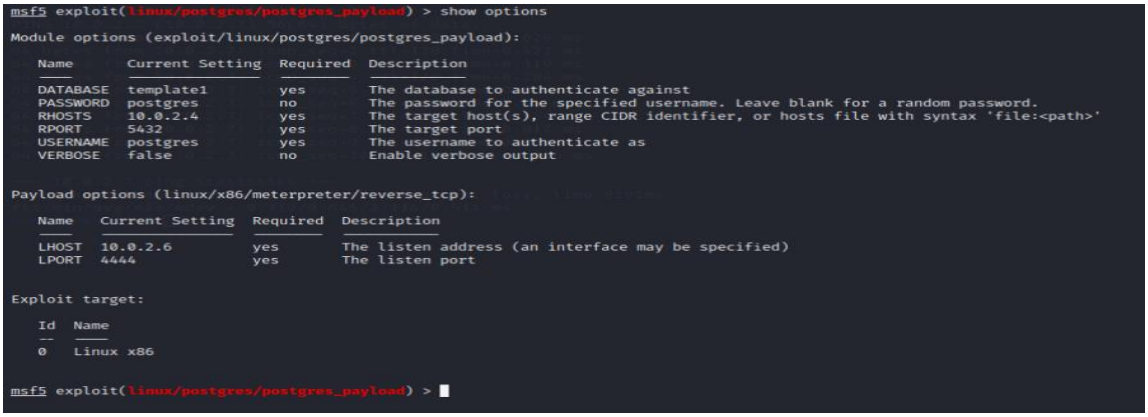

Figura 36. RHOST Configurado. Elaboración propia.

Ejecución del exploit, y apertura de sesión

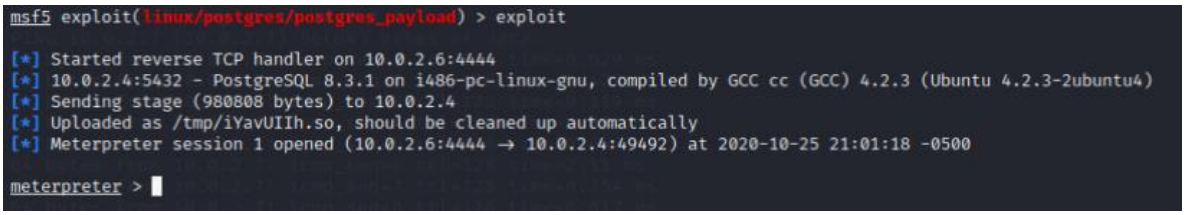

Figura 37. Exploit ejecutado. Elaboración propia.

Configuración de reglas

Para la realización del proceso de configuración de las reglas de automatización en Snort, se debe aplicar el siguiente proceso de Hearder format.

Header Format						
Action	Proto	SRC	SRC Port	Direction	DST	DST Port

Action	Function	Proto	Direction	Meaning
alert	alerts and logs event	IP	->	from SRC to DEST
log	logs event	TCP	<>	in either direction
pass	ignores event	UDP		
drop	drops packet and logs event	ICMP		
reject	TCP reset of session or ICMP Type 3 Code of UDP	Source/Destination Port		
sdrop	drops packet without logging	A.B.C.D		
activate	drops packet without logging	A.B.C.D/xx		
dynamic	alerts and activates a dynamic rule			

Figura 38. Tabla estructura reglas Snort. Elaboración propia.

Esquema de alerta

En el proceso realizado para nuestro servicio de NIDS, basado en la alerta configurada, relacionamos un diagrama según pruebas funcionales ejecutadas para identificar el alcance que puede tener cada una de ellas.

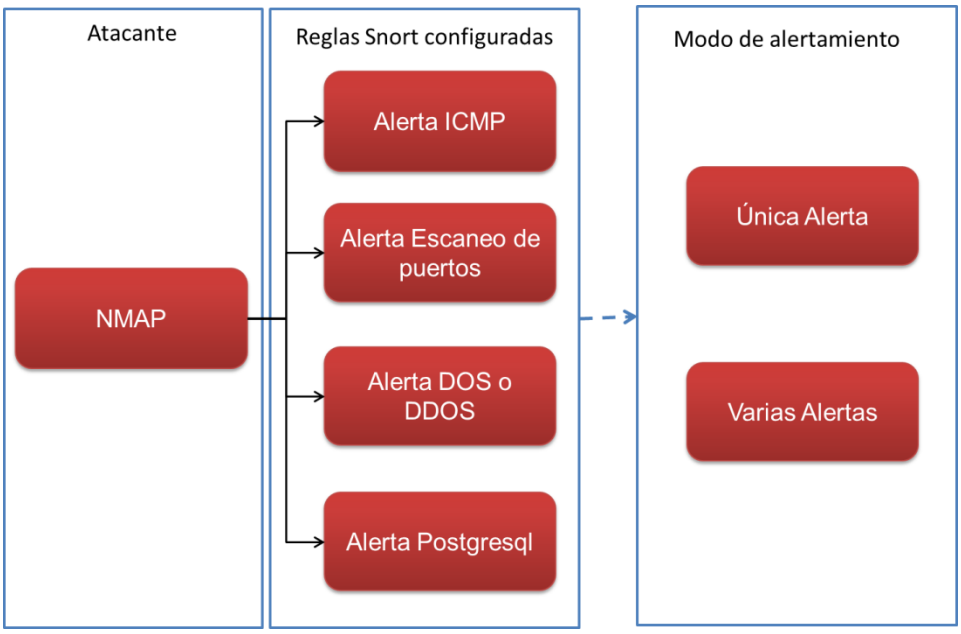

Figura 39. Modo alerta. Elaboración propia.

En las pruebas funcionales ejecutadas en el entorno con diferentes scripts de Nmap, nos generaron múltiples resultados en la activación de las alertas configuradas en nuestro sistema. Identificamos que nuestras reglas trabajan en conjunto según sea el proceso de escaneo de vulnerabilidad realizado por los aplicativos Nmap o metasploit.

Única alerta: en la ejecución de ciertos scripts de Nmap solo nos muestra nuestro sistema una única alerta.

Varias alertas: en la ejecución de ciertos scripts de Nmap, nos muestra varias alertas de acuerdo al script que estemos ejecutando en el sistema Nmap.

Alerta ICMP.

Dentro del análisis realizado se identificó que este protocolo es utilizado para el descubrimiento de equipos activos en una red. En Snort, se realizó la configuración de la regla relacionado en la figura 19, para la alerta de múltiples intentos de escaneo Ping.

```
alert icmp any any -> 10.0.2.0/24 any (msg:"Escaneo de Red";
    threshold: type threshold, track by_dst, count 10 , seconds 30;
    sid:0000001; rev:001;)
```

Figura 40. Regla ICMP, configurada. Elaboración propia.

```
        --== Initialization Complete ==--

        -*> Snort! <*-
        Version 2.9.16.1 GRE (Build 140)
        By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
        Copyright (C) 2014-2020 Cisco and/or its affiliates. All rights reserved.
        Copyright (C) 1998-2013 Sourcefire, Inc., et al.
        Using libpcap version 1.9.1 (with TPACKET_V3)
        Using PCRE version: 8.42 2018-03-20
        Using ZLIB version: 1.2.11

Commencing packet processing (pid=2125)
11/18-19:52:45.314447  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.324936  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.344453  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.344705  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.350761  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.361375  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.365060  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.376291  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.385347  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.392872  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.396397  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.403816  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.409303  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.412581  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.416830  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.419273  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
11/18-19:52:45.423875  [**] [1:1:1] Escaneo de Red [**] [Priority: 0] {ICMP} 10.0.2.3 → 10.0.2.6
```

Figura 41. Alerta Snort ICMP. Elaboración propia.

Alerta escaneos puertos.

Creamos una regla para alertar el escaneo de puertos para identificar el tráfico inusual bidireccional.

```
alert tcp any any -> 10.0.2.0/24 any (flags: A; msg: "Escaneo puertos TCP";
    threshold:type threshold, track by_dst, count 4 , seconds 60;
    sid:10000002; rev:002;)
```

Figura 42. Regla Escaneo de puertos, configurado. Elaboración propia.

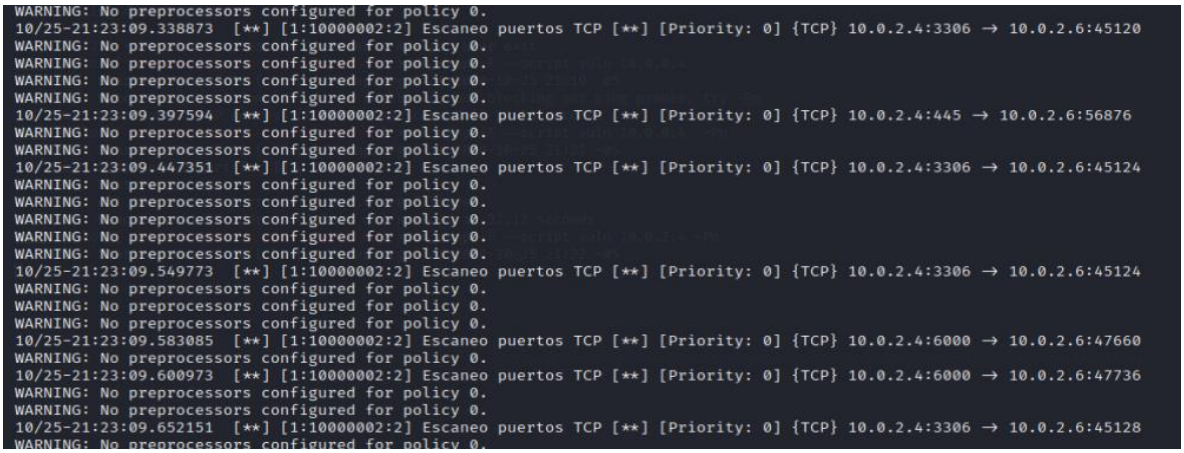

Figura 43. Alerta Snort Escaneo de puertos. Elaboración propia.

Alerta intento DOS.

Creamos una regla para alertar un intento de denegación de servicios, de esta manera identificamos las múltiples peticiones a un equipo de nuestra red.

```
alert TCP any any -> 10.0.2.0/24 any (msg:"Intento DDOS";
    flags:S; threshold: type threshold, track by_dst, count 1000 ,
    seconds 60; sid:5000002;)
```

Figura 44. Regla intento DOS. Elaboración propia.

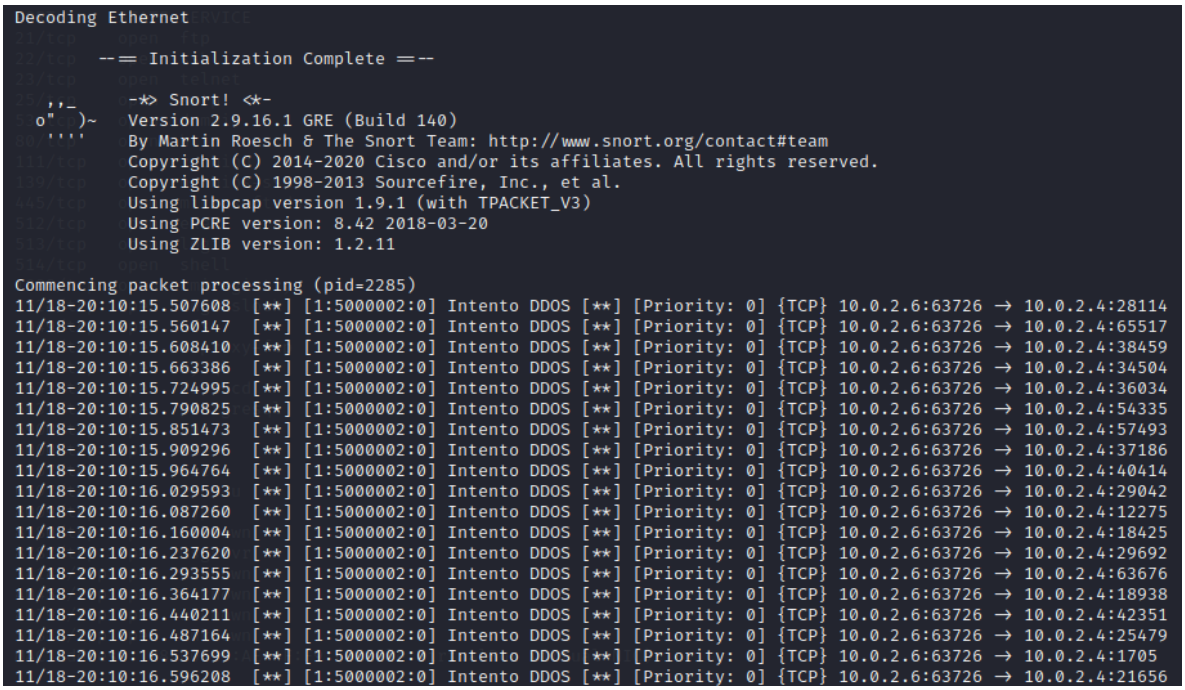

Figura 45. Alerta intento DOS. Elaboración propia.

Alerta vulnerabilidad Postgresql.

Creamos una regla para la detección de proceso de ingreso sobre el sistema, especificando la búsqueda coherente que contenga la palabra “user” y “postgre”, identificados como cuentas supe usuarios en el sistema.

```
alert tcp any any -> 10.0.2.0/24 any (flags: AP; msg:"linux/postgres/postgres_payload";
    content:"user"; content:"postgre"; nocase; sid:99999902; rev:1; priority:8;)
```

Figura 46. Regla vulnerabilidad postgresql. Elaboración propia.

```
10/30-21:13:04.404098  [**] [1:99999902:1] linux/postgres/postgres_payload [**] [Priority: 8] {TCP} 10.0.2.4:5432 → 10.0.2.6:35229
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
```

Figura 47. Alerta Snort identificación postgresql. Elaboración propia.

Preprocesador PortScan.

Los preprocesadores de Snort tienen la funcionalidad de identificar los paquetes antes de llegar el motor de detección, es decir nos da una muestra de información. SfPortScan, identifica los puertos que están siendo analizados poder tomar decisiones, esto no implica que sea un método cien por ciento confiable, ya que hay script de Nmap que no genera identificación. (Cantos Polaino, 2012).

“Por tanto, antes de enviar cada paquete entrante al motor de detección, el preprocesador se Encargará de analizar cada uno de ellos contra cada plug-in que haya sido activado previamente en el archivo de configuración de Snort. Estos plug-ins buscan un cierto tipo de comportamiento por parte del paquete y una vez que se determina, éste se envía al motor de detección. Por ejemplo, si por cualquier razón no se quiere analizar el tráfico RPC que circula en la red, se puede desactivar el plug-in correspondiente en el archivo de configuración de Snort y dejar el resto de plug-ins activos. El hecho de que el usuario tenga la posibilidad de poder activar y desactivar los módulos (plug-ins) que desee es una gran ventaja para un IDS.

El concepto de preprocesador nació en la versión 1.5 de Snort. La principal idea que perseguía la inclusión de preprocesadores fue la de incorporar la posibilidad de permitir inspeccionar los paquetes antes de que alcanzaran el motor de detección, para así generar alertas sobre los paquetes, omitirlos del análisis si cumplieran, o no, con ciertas características y también tener la posibilidad de modificar el contenido de los mismos antes de reenviarlos al motor de detección” (p. 73).

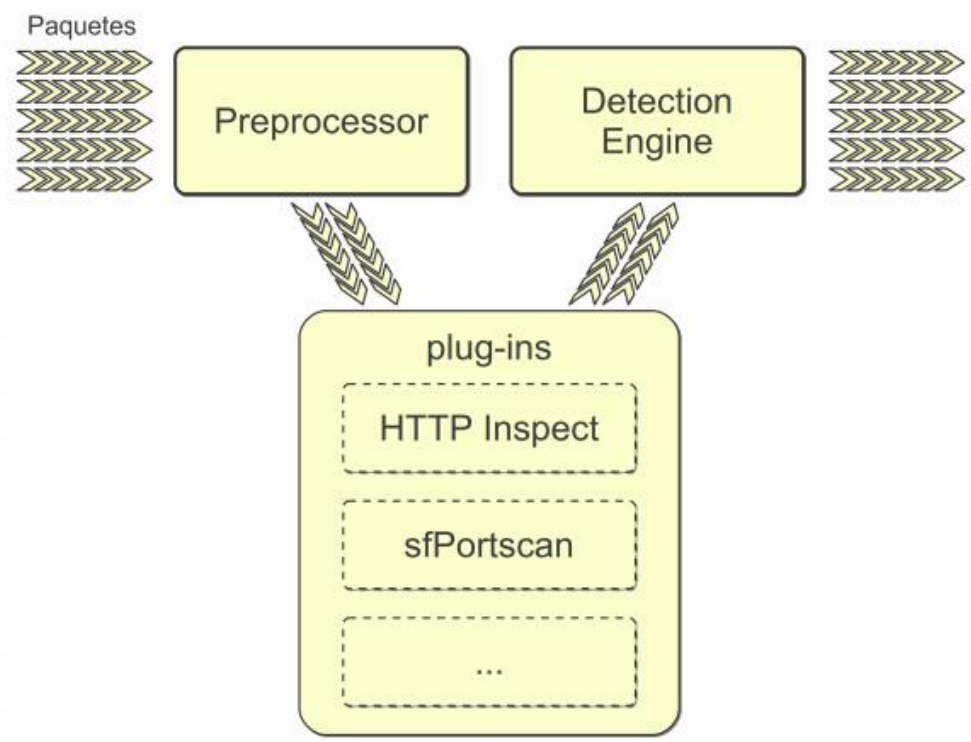

Figura 48. Funcionamiento preprocesador Snort (Cantos Polaino, 2012)

En el presente proyecto el funcionamiento del preprocesador sfPortscan que se configuro el Snort fue utilizado como repositorio para capturar en log y así identificar una parte varios de los script de Nmap, para comprender la diferencia que pueda presentarse con las alertas creadas. Se tuvo en cuenta que este proceso de configuración como alerta se puede generar por un llamado de evento y asociarlo a una regla en particular.

```
17 # preprocessor flow: stats_interval 0 hash 2 \
18     preprocessor sfportscan:\
19         proto { all } \
20         scan_type { all } \
21         sense_level { medium } \
22         logfile { portscan.log }
23
```

Figura 49. Preprocesador sfPortscan. Elaboración propia.

```
root@kali:~/logs# cat portscan.log
Time: 11/21-17:53:38.892343
event_ref: 0
10.0.2.6 → 10.0.2.4 (portscan) TCP Portscan
Priority Count: 10
Connection Count: 23
IP Count: 1
Scanner IP Range: 10.0.2.6:10.0.2.6
Port/Proto Count: 23
Port/Proto Range: 21:8888

Time: 11/21-17:56:34.343347
event_ref: 0
10.0.2.6 → 10.0.2.4 (portscan) TCP Portscan
Priority Count: 10
Connection Count: 29
IP Count: 1
Scanner IP Range: 10.0.2.6:10.0.2.6
Port/Proto Count: 29
Port/Proto Range: 21:8888
```

Figura 50. Visual log preprocesador sfPortscan. Elaboración propia.

Conclusiones

Durante la fase inicial se afianzaron concepto de las herramientas Nmap, Wireshark, Metasploit y Snort, adquiriendo conocimientos para el manejo de las mismas, entre ellas se incluyeron comandos básicos y necesarios para su uso respectivo. Se destacó en el proceso de desarrollo del trabajo la utilización del aplicativo Snort, como base de automatización de alertas para detectar anomalías en un entorno local.

Por otro lado, en la configuración la red se idéntico todo un proceso de funcionamiento de gestión para el análisis de tráfico del entorno creado para la actividad de escaneo y pruebas de vulnerabilidad a un host especifico, creando un entorno de comunicación de máquinas virtuales u físicas.

Al realizar la configuración de las reglas en el IDS Snort, además de seguir la estructura básica para su creación, fue necesario analizar muy bien el tráfico de entrada a la red, se tuvieron en cuenta comportamientos inusuales de escaneo de puertos y explotación de una vulnerabilidad

Referencias

Amador, S., Arboleada, A., & Bedon, C. (2006). Utilizando Inteligencia Artificial para la deteccion de Escaneos de Puertos. *Utilizando Inteligencia Artificial para la deteccion de Escaneos de Puertos*, 12. Cauca.

Brown, J., Anwar, M., & Dozier, G. (2017). Detection of Mobile Malware: An Artificial Immunity Approach. *EURASIP Journal on Information Security*, 1-10.

Cisco Systems. (1998). *Snort*. Obtenido de https://www.snort.org/

Congreso de la republica. (5 de enero de 2009). *Ley 1273 de 2009*. Obtenido de http://www.secretariasenado.gov.co/senado/basedoc/ley_1273_2009.html

Congreso de la republica. (18 de octubre de 2012). *Ley estatutaria 1581 de 2012*. Obtenido de http://www.secretariasenado.gov.co/senado/basedoc/ley_1581_2012.html.

Gomez Lopez, J. (2009). *Optimizacion de Sistemas de deteccion de intrusos en Red Utilizando Tecnicas Computacionales avanzadas.* Obtenido de book Google: https://books.google.com.co/books?Id=qtdbaqaaqbaj&printsec=frontcover&dq=sistema+de+detecci%C3%b3n+de+intrusos&hl=es&sa=X&ved=2ahukewjx6miyj9vsahvspvkkhd0ec78q6aewaxoecamqag#v=onepage&q=sistema%20de%20detecci%C3%b3n%20de%20intrusos&f=false

Gutierrez del Moral, L. (2014). Curso de Ciberseguridad y Hacking Ético 2013. En L. Gutierrez del Moral, *Curso de Ciberseguridad y Hacking Ético 2013* (págs. 1-560). Punto Rojo Libros, 2014.

kasperky. (2020). *Kasperky*. Obtenido de Kasperky: https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security

Lyon, G. (1 de 09 de 1997). NMAP. Obtenido de https://nmap.org/docs.html

Martinez Lopez, M. A. (13 de 8 de 2010). *Capitulo 2.* Obtenido de Capitulo 2: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/martinez_l_ma/capitulo2.pdf

Rivero Perez, J. L. (2014). Técnicas de aprendizaje automático para la detección de intrusos en redes de computadoras. *Rev cuba cienc informat, 8*(4), 52-73.

Romero castro, m. I., figueroa moran, g. L., vera navarrete, d. S., alava cruzatty, j. E., parrales anzules, g. R., alava mero, c. J., . . . Castillo merino, m. A. (2018). *Introducción a la seguridad informática y el análisis de vulnerabilidades.* Area de innovacion y desarrollo

si. Obtenido de https://books.google.com.co/books?Id=5z9ydwaaqbaj&printsec=frontcover&dq=Ataque+cibern%C3%a9tico+definicion&hl=es&sa=X&ved=2ahukewj47svjmnvsahvrmlkkhtvebdkq6aewbxoecacqag#v=onepage&q=Ataque&f=false

The Wireshark team. (1990). *Wireshark*. Obtenido de Wireshark: https://www.wireshark.org/

Universidad Nacional Autonoma de Mexico. (2018). *Universidad Nacional Autonoma de Mexico.* Obtenido de https://revista.seguridad.unam.mx/numero-19/pruebas-de-penetraci%C3%B3n-para-principiantes-explotando-una-vulnerabilidad-con-metasploit-fra

WonderHowTo Network. (2008). *WonderHowTo*. Obtenido de https://null-byte.wonderhowto.com/how-to/gather-information-postgresql-databases-with-metasploit-0218317/