

**Análisis de vulnerabilidades a un servidor usando un kubernetes clúster de computadoras  
Raspberry pi y software libre**

Juan Camilo Rodríguez Padilla

Universitaria Agustiniana  
Facultad de Ingeniería  
Programa Ingeniería en Telecomunicaciones  
Bogotá, D.C.  
2022

**Análisis de vulnerabilidades a un servidor usando un kubernetes clúster de computadoras  
Raspberry pi y software libre**

Juan Camilo Rodríguez Padilla

Director  
Francisco Valle

Trabajo de grado para optar al título de Ingeniero en Telecomunicaciones

Universitaria Agustiniana  
Facultad de Ingeniería  
Programa Ingeniería en Telecomunicaciones  
Bogotá, D.C.  
2022

## **Resumen**

En el siguiente documento se plasma el desarrollo de un proyecto que se enfoca en comprobar la eficiencia del cómputo en paralelo de alto rendimiento mediante análisis de vulnerabilidades usando como máquina de escáner un kubernetes clúster conformado por cuatro computadoras Raspberry PI implementadas con Kali Linux y varios programas de Software Libre que analizan un servidor con servicios activos en una red de laboratorio que simula las redes encontradas en el entorno real y empresarial. En este documento se recopila información sobre la configuración y el procedimiento necesario para poner en marcha un clúster de computadoras que trabaje en paralelo y como usar este para dirigir análisis de vulnerabilidades, esto en busca de comparar los análisis de vulnerabilidades ejecutados en la misma red de laboratorio con una máquina de uso personal con similitudes en hardware y costo al clúster de Raspberrys usando el mismo software de análisis con el fin de comprobar la utilidad y efectividad del cómputo en paralelo de alto rendimiento. Palabras clave: Cómputo en Paralelo de Alto Rendimiento, Kubernetes Cluster, Raspberry Pi, Red de Laboratorio, Análisis de Vulnerabilidades.

## Tabla de contenido

Introducción	9
Problemática	9
Pregunta de investigación	9
Idea de proyecto	9
Objetivos	11
Objetivo general	11
Objetivos específicos	11
Marco referencial	12
Estado del arte	12
Implementación de un laboratorio de seguridad de informática para la realización de técnicas de ataque y defensa (Pentesting) en un ambiente real controlado, utilizando una distribución de Kali Linux	12
Pentesting sobre aplicaciones web basado en la metodología OWASP utilizando un clúster conformado por dispositivos SBC de bajo costo.	13
Marco teórico	16
Clúster de computadoras	16
Kubernetes clúster	17
Raspberry pi	17
Máquina virtual	18
Máquina intencionalmente vulnerable	18
Análisis de vulnerabilidades o Pentesting	18
Kali Linux ARM	18
Marco legal	19
Ley 1273 de 2009 Ley 1273 de 2009	19
Artículo 269C. Interceptación de datos informáticos	19
Artículo 269E: Uso de software malicioso	20
Artículo 269B: Obstaculización ilegítima de sistema informático o red de telecomunicación.	20
Artículo 269d: Daño informático	20
Norma (ISO/IEC 27032.2)	21

Ley 1341 de 2009 .....	21
Metodología	22
Administración del proyecto	24
Cronograma .....	24
Presupuesto .....	25
Conclusiones.....	49
Referencias.....	51

## Lista de tablas

Tabla		1.
	Cronograma.....	1E
<b>rror! Marcador no definido.</b>		
Tabla		2.
	<i>Presupuesto</i> .....	2E
<b>rror! Marcador no definido.</b>		
Tabla 3.	Comparación de hardware.....	47
Tabla 4.	Comparación de resultados.....	48

## Lista de figuras

<b>Figura 1.</b>	Resultados de cada aplicación (Muniz&Lakhani,2015).....	15
<b>Figura 2.</b>	Comparación de tiempo en diferentes aplicaciones (Muniz&Lakhani,2015).....	15
<b>Figura 3.</b>	Hardware reunido para conformar el clúster de computadoras Raspberry pi.....	26
<b>Figura 4.</b>	Rufus listo para formatear memoria con sistema operativo KALI ARM.....	27
<b>Figura 5.</b>	Asignación de nombre a la primer Raspberry .....	28
<b>Figura 6.</b>	Mpich funcionando correctamente en una maquina PI de forma individual.....	29
<b>Figura 7.</b>	Mpi4py funcionando correctamente en maestro.....	30
<b>Figura 8.</b>	Ejemplo como debe quedar el direccionamiento de la maquina maestro .....	30
<b>Figura 9.</b>	Archivo de configuración Ssh.....	31
<b>Figura 10.</b>	Archivo de configuración para las maquinas que componen el clúster.....	33
<b>Figura 11.</b>	Procesos de scribed de prueba distribuido en los Equipos del clúster.....	34
<b>Figura 12.</b>	Servidor virtual en marcha.....	35
<b>Figura 13.</b>	Red de máquina virtual configurada como adaptador puente .....	35
<b>Figura 14.</b>	servidor conectado a red de clúster de Raspberrys.....	36
<b>Figura 15.</b>	Nmap reconoce el servidor virtual conectado a la Red.....	37
<b>Figura 16.</b>	K3s en nodo maestro.....	38
<b>Figura 17.</b>	Token acceso al clúster.....	38
<b>Figura 18.</b>	clúster k3s funcionando.....	39
<b>Figura 19.</b>	Helm funcionando.....	39
<b>Figura 20.</b>	Buscador de helm en función.....	40
<b>Figura 21.</b>	secureCodeBox corriendo en máster.....	40
<b>Figura 22.</b>	nmap para kubernetes corriendo.....	41
<b>Figura 23.</b>	Archivo de configuración nmap.....	41
<b>Figura 24.</b>	Ejecutando análisis de vulnerabilidad .....	42
<b>Figura 25.</b>	primer análisis ejecutado y finalizado.....	42
<b>Figura 26.</b>	resultado scanner con nmap.....	42
<b>Figura 27.</b>	Instalación de Nikto.....	43
<b>Figura 28.</b>	archivo ejecución nikto.....	43
<b>Figura 29.</b>	ejecución de análisis nikto.....	44
<b>Figura 30.</b>	1correcta instalacion whatweb.....	44

**Figura 31.** Archivo de configuración whatweb.....45  
**Figura 32.** ejecución de análisis de whatweb en paralelo.....45  
**Figura 33.** computador HP a comparar.....46  
**Figura 34.** Ejecución de nmap.....46  
**Figura 35.** Resultado de usar nikto.....47  
**Figura 36.** whatweb en ejecución.....47



## **Introducción**

### **Problemática**

Es posible destacar que en la actualidad debido al desarrollo tecnológico la necesidad computacional cada vez es mayor, conforme pasan los años los servicios prestados requieren cada vez una mayor exigencia computacional, para esto, se resaltan posibles soluciones como el desarrollo de nuevas tecnologías en procesadores aumentando su capacidad en caché o disminuyendo el tamaño de sus transistores y con ello su cantidad según Amor (2012). Estas actualizaciones conllevan un incremento económico y generan otros problemas indirectos como el aumento de residuos debido al desuso de componentes de generaciones pasadas. Sin embargo, la computación paralela permite a las organizaciones aumentar la capacidad de procesamiento con unos requerimientos de hardware y software que son en comparación mucho más económicos pero que necesitan de otros requerimientos como el conocimiento y la investigación exhaustiva para poner en marcha una solución viable.

Por otra parte, es notable el incremento de ataques cibernéticos y las consecuencias que estos tienen, tanto en empresas como en individuos particulares se puede resaltar que los ataques van en incremento en América latina debido a la falta de conocimiento, y al retraso tecnológico por el que pasan los países dice Lavinder (2016), lo que ha provocado como centro de estos ataques las entidades financieras y gubernamentales.

Debido a lo anterior, este proyecto se expone el diseño, construcción y configuración de un esquema de computadoras agrupadas en clúster implementadas para el escaneo de vulnerabilidades a un servidor de laboratorio, con el fin de comparar sus resultados con una máquina de uso personal y así comprobar la efectividad del cómputo en paralelo aplicado en el hacking y testeo de vulnerabilidades.

### **Pregunta de investigación**

¿Qué ventajas podemos obtener al usar un clúster computadoras en el testeo de vulnerabilidades a un servidor?

### **Idea de proyecto**

El propósito de este proyecto es realizar análisis de vulnerabilidades desde un sistema de hardware con cualidades distintas a las utilizadas normalmente por los grupos que se dedican a mitigar los ataques cibernéticos. Esta configuración, se enfoca en el procesamiento paralelo de alto rendimiento de varias computadoras de bajo costo, que proporcionan tiempos en los testeos de

vulnerabilidades completamente diferentes a los que muestran computadoras normales con procesadores de varios núcleos como lo sería un Intel Core i9 (8 núcleos) el mejor de los procesadores presentados por Intel o el procesador Threadripper de RYZEN (hasta 64 núcleos).

El kubernetes clúster presentado en este proyecto utiliza menos recursos tanto energéticos como económicos, ya que en comparación, el computador a usar para el testeo de puertos es un Laptop HP 15-dw0083wm con un procesador Intel Pentium Silver N5000 de 4 núcleos a 1.1GHz y 4gb de ram ,cuyo costo es de aproximadamente 1'650.000 COP, y con el clúster de equipos, (Raspberry pi 4) es posible lograr un total de 16 núcleos y 4 GB de ram uniendo tan solo 4 equipos, y esto generaría un costo de 1'300.000 COP aproximadamente, costo que incluye todos sus componentes, como fuentes, cables, switches o memorias para el almacenamiento del sistema operativo. En este proyecto se busca hacer una comparación en los tiempos de ejecución de los análisis de vulnerabilidades con equipos conectados en un clúster de alto rendimiento y una computadora común con un único procesador de varios núcleos como se usan rutinariamente en la oficina o para trabajos escolares.

## **Objetivos**

### **Objetivo general**

Analizar las vulnerabilidades de un servidor usando software libre en un clúster de computadoras Raspberry PI.

### **Objetivos específicos**

- Implementar Hardware y software necesaria para el funcionamiento de las computadoras Raspberry PI en clúster paralelo.
- Implementar una red que permita establecer conexión entre el clúster de Raspberrys y el servidor con servicios funcionales al cual se ejecutarán análisis de vulnerabilidades.
- Ejecutar análisis de vulnerabilidades al servidor desde dos escenarios siendo primero el clúster de Raspberrys con software libre y el segundo desde una maquina común con hardware y costo similar.
- Comparar los resultados de los análisis de vulnerabilidades en computadoras de hardware común, con un clúster de computadoras Raspberry pi.

## Marco referencial

### Estado del arte

#### **Implementación de un laboratorio de seguridad de informática para la realización de técnicas de ataque y defensa (Pentesting) en un ambiente real controlado, utilizando una distribución de Kali Linux**

La principal funcionalidad de este proyecto fue entregar una herramienta en un ambiente virtualizado que servirá para detectar fallas y vulnerabilidades a los diferentes servicios y recursos de red de la infraestructura tecnológica de la empresa ANDESTEC en la cual pertenecían los productores de este texto. Para lograr esto, en primera estancia se planteó hacer una auditoria web, ataques de fuerza bruta, ejecución de exploits y escaneo de puertos a una plataforma virtualizada que cuente con características similares a la de la empresa.

El objetivo de este proyecto fue la entregar un informe detallado del testeo de puertos que se realizó en el laboratorio controlado, junto con un manual de instrucciones donde se especifica las posibles soluciones a las vulnerabilidades presentadas.

Para el montaje del laboratorio fue necesario un servidor Kali Linux, un servidor Windows server estándar 2012, una licencia VNC para las páginas web. Una vez montado el laboratorio con los servicios y la infraestructura simulada, mediante aplicativos de uso libre en kali Linux se empieza el rastreo de vulnerabilidades, en el trabajo investigativo se usan los siguientes aplicativos o algoritmos:

Nmap, que es un software de código abierto, cuya función principal es efectuar rastreo de puertos, para obtener información de los servicios, en esta práctica, todos los ataques van dirigidos al servidor Windows 2012 donde se encuentran los servicios de DHCP, active directory, DNS apache y demás.

Una vez ejecutados los comandos de rastreo de puertos con la herramienta Nmap, se corresponde a hacer uso una herramienta llamada Metasploit este es un framework que sirve para proveer información acerca de las vulnerabilidades que pudiera tener algún dispositivo, esta herramienta de igual forma es de código abierto y utilizada comúnmente en seguridad informática. Para este caso, con otra herramienta se planea crear un virus, que ingrese a la red, afectando las vulnerabilidades anteriormente mostradas por la herramienta Nmap. Los aplicativos anteriores mostraron el método de ataque a los servidores de la empresa, ya que, una vez ingresado el virus,

se procede a enviar información capturada por medio de un puerto que se encontraba abierto y en escucha.

De igual forma se especifican procedimientos para vulnerar la red inalámbrica de la empresa usando la terminal del servidor kali, una vez capturado el mensaje handshake, se procede a usar cruch para descifrar la contraseña de la red, para esto se usa la aplicación hydra, que les ayudó a buscar credenciales que coincidían con la contraseña.

Como resultado, los investigadores Cesar & Jasson (2018) dejan un panel de recomendaciones para la empresa en el actual se resaltan algunas de las mencionadas:

- Tomar en consideración los puertos abiertos o que escuchan algún tipo de servicio, es importante habilitar solo los puertos necesarios y de igual forma filtrarlos con algún tipo de herramienta, como un firewall perimetral.
- En cuanto a las redes Wireless, su autenticación y tráfico generado, se recomienda que al menos al SSID exclusivo del departamento de T.I. sea blindado con algún otro tipo de seguridad como puede ser un filtrado MAC, si bien es cierto las contraseñas es demostrado que pueden ser descifradas un filtrado de direcciones MAC seria de muy buena ayuda.

### **Pentesting sobre aplicaciones web basado en la metodología OWASP utilizando un clúster conformado por dispositivos SBC de bajo costo.**

En este documento los investigadores Cesar & Jasson (2018) tuvieron la intención de optimizar los escaneos de pentesting, utilizando software y hardware especializado en el escaneo de redes y páginas webs, el objetivo de este proyecto de investigación es identificar de manera eficiente las vulnerabilidades más comunes en las aplicaciones web, siguiendo la metodología OWASP, que no es más que otro aplicativo o proyecto de código abierto dedicado a combatir y verificar cuales son las vulnerabilidades de un software o programa.

Los investigadores Cesar & Jasson, 2018 notan como problema, que los aplicativos que actualmente se usan para pentesting tienen limitaciones respecto a la identificación de vulnerabilidades sobre aplicaciones web, ya que, en ocasiones, estas pueden crear falsos negativos (detección de falsas vulnerabilidades) o los falsos positivos (omisión de vulnerabilidades existentes). Dada la necesidad anterior, en esta investigación se presenta la implementación de un prototipo hardware y software que pueda automatizar el pentesting sobre aplicaciones, donde se compara la eficiencia y eficacia a la hora de encontrar vulnerabilidades con respecto a 4 herramientas por medio de un análisis mediante curvas ROC.

Actualmente existen trabajos relacionados acerca del pentesting con dispositivos SBC. Un dispositivo SBC es la Raspberry Pi, con el cual se ha desarrollado plataformas portátiles para realizar pentesting sobre redes inalámbricas con el fin de lograr un consumo de recursos computacionales menor a los que utiliza un computador convencional de última generación Muniz&Lakhani(2015).

Como procedimiento del proyecto, los investigadores proponen hacer un clúster de 7 Raspberry Pi conectadas entre sí, con el fin de ejecutar varios softwares de pentesting en coordinación, apoyados de una base de datos que automatiza la presentación de los reportes generados por los aplicativos, y de esta forma comparar que software tiene más precisión y genera menos reportes erróneos.

Como inicio del trabajo, se tomaron 4 aplicaciones de testeo que son: Netsparker, owasp zap acunetix y Nessus, y la aplicación que correrán las Raspberry Pi fue Scanlynx que permite usar el hardware en stock.

Teniendo los resultados cuantitativos, se realizó un análisis de curvas ROC, esto con el fin de hacer una demostración haciendo uso de la estadística, confrontando los resultados y mostrándolos en una tabla que facilite su lectura:

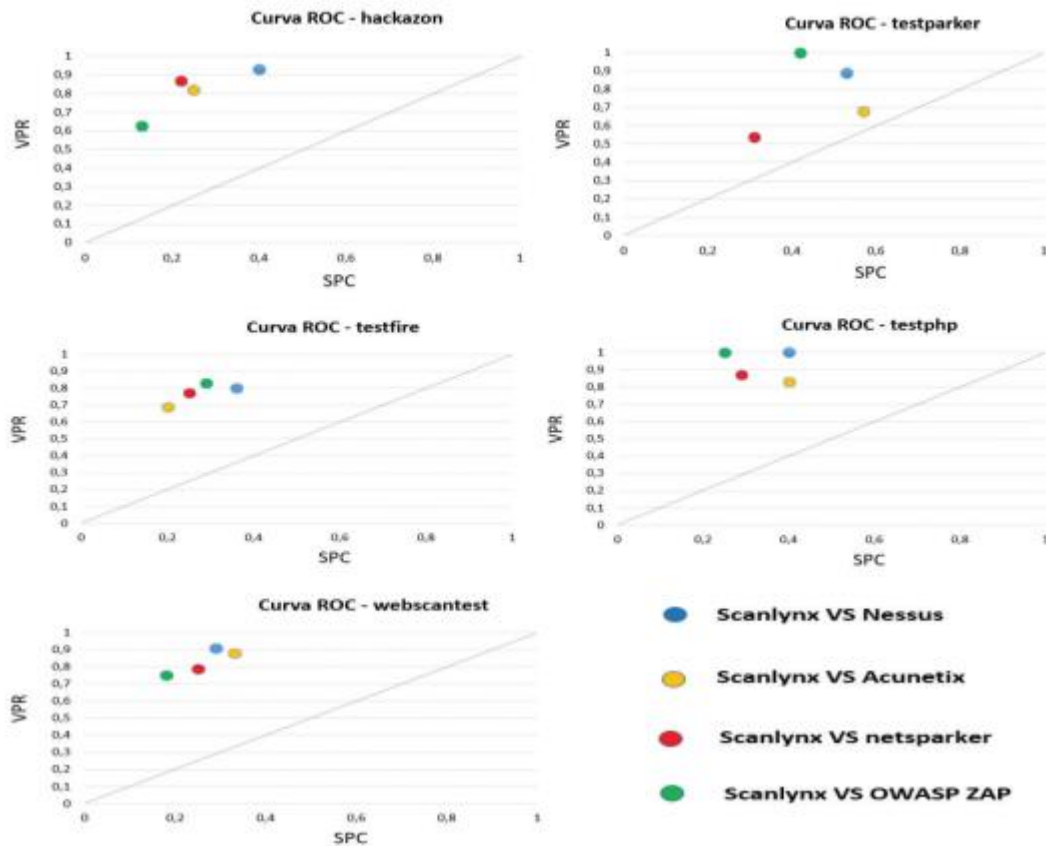


Figura 1. Resultados de cada aplicación (Muniz&Lakhani,2015).

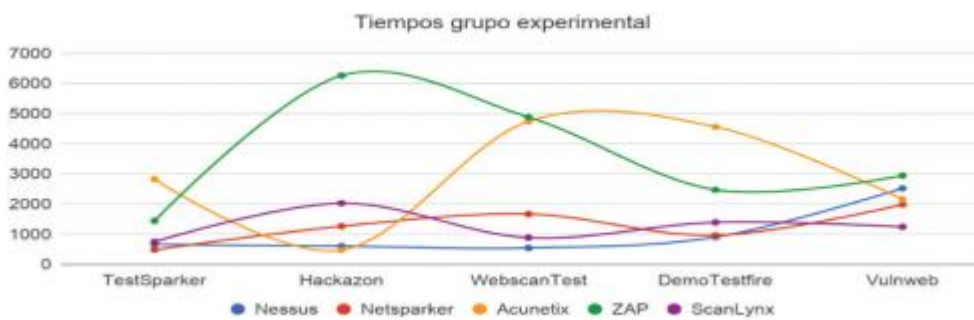


Figura 2. Comparación de tiempo en diferentes aplicaciones (Muniz&Lakhani,2015).

El comportamiento del experimento controlado de ScanLynx contra las herramientas seleccionadas fue excelente, ya que, los resultados de verdaderos positivos y la razón para los verdaderos negativos estuvo por encima de la línea de no-discriminación, esto significa que las

otras herramientas no lograron detectar vulnerabilidades adicionales a las que identificó el prototipo hardware y software construido.

## **Marco teórico**

### **Clúster de computadoras.**

Aunque no es fácil dar una definición perfecta para lo que son es un clúster de computadoras, es posible estar de acuerdo que un clúster de computadoras se puede clasificar como un sistema distribuido de cómputo paralelo. Por ejemplo, un autor dice: “Es un tipo de sistema distribuido o paralelo conformado por una colección de computadoras interconectadas usado como un único recurso de computación unificado” Pfister (1998)

Entonces es posible afirmar que un clúster se entiende como un conjunto de computadoras conectadas a una red de alta velocidad que unen sus características y componentes para trabajar a un solo tiempo en un servicio informático, con fin de ser una computadora con potenciamiento en sus recursos.

En un sistema de clústeres es común encontrar una configuración tal que: uno de los dispositivos que se conectan a la red funciona como maestro, se encarga de administrar, controlar y monitorear los recursos del sistema, por otra parte, el resto de los dispositivos que se encuentran tienen como objetivo el procesamiento de datos, o la ejecución de aplicaciones, a estos se les conoce como nodos-esclavos como es el caso de los Kubernetes clúster.

De acuerdo con el investigador Gerson (2012) es posible clasificar los tipos de clústeres en 3 partes:

- Clústeres de alto rendimiento o High Performance Computing Clúster (HPC): Donde lo más importante es que entre los dispositivos del sistema podemos compartir los recursos más importantes como lo serían la capacidad de procesamiento y de almacenamiento. Es este tipo de clúster el que se planea usar en este documento, debido a que su uso se enfoca en potenciar el procesamiento y mejorar la efectividad a la hora de solucionar problemas matemáticos.
- Clúster de alta disponibilidad o High Availability Computing Clúster (HAC): Tiene como característica principal, tener la mayor cantidad de tiempo posible, ya que si en dado caso falla alguno de los dispositivos, es posible que otro llegue a reemplazarlo, el enfoque principal de este tipo es que si se afecta un componente del clúster, el resto de



los procesos o datos se ejecuten o configuren en otra de las máquinas que componen el clúster.

- Clúster de balanceo de cargas o Load Balancing (LBC): El objetivo de un clúster de Balanceo de Carga es el de distribuir el trabajo entre todos los nodos del clúster, esto se consigue asignando el trabajo al nodo que posee más recursos disponibles, Este tipo de clúster se usan comúnmente en servicios de alta cantidad de carga como lo serían los servidores WEB.

### **Kubernetes clúster**

Un Kubernetes clúster se comprende como un conjunto de computadoras unidas a una red local que usaran sus recursos físicos para mejorar la efectividad de los procesos y resolución de problemas según los autores Andrés, Santiago, & Siler (2018) dicen:

“La estructura básica de Kubernetes clúster se compone de un dispositivo principal llamado maestro, el cual se encarga de distribuir mantener y organizar los procesos que serán enviados a los demás dispositivos denominados esclavos, los cuales ejecutan las tareas de manera distribuida y/o paralela”.

### **Raspberry pi.**

La Raspberry pi es una computadora completa de un solo circuito, de muy bajo costo y del tamaño de la palma de una mano, nace con fines didácticos en la universidad de Cambridge, fue diseñada principalmente para disminuir el costo en las instalaciones educativas y con el fin de que personas de bajos recursos pudieran tener acceso a la Tecnología informática dice Rodríguez (2018).

Esta computadora tiene todos los componentes principales que se requieren para correr un sistema operativo. Como nos cuenta un autor, esta placa usa el controlador Broadcom, que es un SOC (System on Chip). Este SOC tiene un poderoso procesador ARM, (según su versión se va actualizando el procesador). Tiene un puerto Ethernet que permite conectarlo a redes, se pueden cargar sistemas operativos desde Mac, Windows y Linux a una memoria SD que funcionara como almacenamiento interno, cuenta con varios chips de memoria RAM incrustados a la placa, esta cambia según su versión y también cuenta con 4 conectores usb según Aldea (2017), lo impresionante de esta máquina es el desarrollo que tiene con respecto al poco tiempo que lleva en el mercado, es posible resaltar su evolución en los procesadores, con un aumento sustancial en la

frecuencia reloj del procesador pasando de 1.1GHz a 1.5GHz , y un aumento en su ram, pasando de 1 a 4 GB,Esto al comparar las versiones de Raspberry pi 3 y 4.

### **Máquina virtual**

#### **Maquina intencionalmente vulnerable**

#### **Análisis de vulnerabilidades o Pentesting.**

El pentesting es una abreviatura formada por dos palabras, penetration y testing, consiste en prácticas o técnicas que evalúan entornos o sistemas de cómputo, con la finalidad de encontrar y prevenir o explotar según se desee los estos fallos encontrados. Dice Serrano (2021)

Las pruebas de intrusión siempre se ejecutan desde la posición de un potencial atacante, el propósito es determinar la viabilidad de un ataque, saber si es posible vulnerar un error en la red o los servicios, y conocer qué impacto tendrá el ataque al sistema Gonzales, Sánchez, & Soriano (2013).

Para Facilitar este tipo de analisis existen varios tipos de aplicativos que unen funciones en algoritmos previamente utilizados, y que facilitan en gran medida su uso. Gran parte de estos aplicativos se usan en software libre como podrían ser los sistemas operativos basados en kernel, todo esto con el fin de evitar problemas legales, como es bien sabido, gran parte de estos ataques son ejecutados por personas sin escrúpulos, cuya finalidad principal es el bien propio. Entre estos aplicativos podemos encontrar:

- ZEd Attack Proxy: Esta herramienta actúa entre la página web y el navegador utilizado, se enfoca en capturar todo el tráfico posible, lo inspecciona y analiza para poder identificar todos los fallos posibles, este software es gratuito y abierto.

#### **Kali Linux ARM.**

Es un sistema operativo de código abierto orientado a servicios de seguridad informática de todo tipo, como pruebas de penetración, investigación de seguridad, informática forense, ingeniería inversa entre otras, hablando de Kali dice Altube (2021): “Kali Linux es una distribución de Linux basada en Debian, específicamente diseñada para temas de seguridad muy variados, como análisis de redes, ataques inalámbricos, análisis forenses y otros que más adelante citaremos”

La versión Kali ARM de linux trae consigo en su instalación, aproximadamente 600 programas dedicados a la seguridad informática, estos incluyen NMap, Wireshark, Aircrack-n, kali tiene una característica muy particular, y es que es muy versátil ya que es posible instalar su sistema operativo en una amplia variedad de dispositivos, como computadoras de una sola placa, dispositivos móviles

como tabletas o celulares, en dispositivos de Amazon web services, computadoras de todo tipo, ya que usa una cantidad muy baja de recursos, lo que equivale a una amplia gama de computadoras dice el autor Hertzog (2017).

### **Marco legal**

#### **Ley 1273 de 2009 Ley 1273 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.

Esta ley modifica el código penal, creando un nuevo bien jurídico tutelado, denominado: “de la protección de la información y de los datos”. Con el fin de preservar los sistemas de tecnologías de la información y las comunicaciones, los sistemas de almacenamiento de información, y la prestación de servicios informáticos CONGRESO DE COLOMBIA (2009). Esta ley es clave para sustentar el proyecto, ya que en sus artículos se especifican los delitos o las malas acciones, que no se deben tomar en los sistemas informáticos, a continuación, se expondrán algunos de estos artículos que se relacionan con esta investigación:

#### **Artículo 269C. Interceptación de datos informáticos.**

“El que, sin orden judicial previa intercepte datos informáticos en su origen, destino o en el interior de un sistema informático, o las emisiones electromagnéticas provenientes de un sistema informático que los transporte incurrirá en pena de prisión de treinta y seis (36) a setenta y dos (72) meses”.

Como se menciona anteriormente en este documento, los testeos de vulnerabilidades, se planea hacer desde la vista de un usuario del común, que no tiene conocimiento respecto a los esquemas y configuraciones de red de una empresa, con el fin de simular un entorno real en el que los ciberdelincuentes intenten escanear o vulnerar la red de una empresa.

Como es de notar, estos escaneos a la red se planean hacer desde un entorno en el que los administradores de red de no sean capaces de detectar los escaneos o la presencia de intrusos, es por esto que de cierta manera se intenta acceder a información de la empresa sin el debido consentimiento con el fin de crear un entorno más fiel a la realidad.

No obstante, se deja claro que las redes a las que se le harán los escaneos son simuladas y creadas con el este único fin, de forma que no afecte los sistemas de ninguna compañía o particular.

### **Artículo 269E: Uso de software malicioso**

“El que, sin estar facultado para ello, produzca, trafique, adquiera, distribuya, venda, envíe, introduzca o extraiga del territorio nacional software malicioso u otros programas de computación de efectos dañinos, incurrirá en pena de prisión”.

Para el desarrollo de esta práctica, es necesario hacer uso de algunas aplicaciones que tocan de forma explícita los apartados de esta ley, El uso de software para escaneo de vulnerabilidades pueden clasificarse como software malicioso, ya que escanean datos, puertos y equipos de una red sin consentimiento, en ocasiones, algunos tipos de escaneos pueden alterar y comprometer la disponibilidad e integridad de estos equipos y servicios afectando de forma directa a la víctima.

El uso de este tipo de software, no incumplirá en ninguna ley a la hora de su adquisición y uso, debido a que la ejecución y las metodologías que se aplican, pues estas se realizan en un escenario controlado, la red a la que se planea hacer el escaneo de vulnerabilidades es una red simulada, con características parecidas a las de un entorno empresarial, pero sin afectar a ningún individuo o empresa, sabiendo que el objetivo de este estudio es el de evitar y prevenir ataques reales, ejecutados por verdaderos delincuentes, a este tipo de prácticas se les conoce comúnmente como Ethical hacking.

### **Artículo 269B: Obstaculización ilegítima de sistema informático o red de telecomunicación.**

“El que, sin estar facultado para ello, impida u obstaculice el funcionamiento o el acceso normal a un sistema informático, a los datos informáticos allí contenidos, o a una red de telecomunicaciones, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes, siempre que la conducta no constituya delito sancionado con una pena mayor”.

Es importante aclarar, este artículo se relaciona con las finalidades del proyecto de cierta forma, ya que, aunque en este documento no se especifiquen la explotación de las vulnerabilidades a la red, si se dan indicaciones para evitar este tipo de delitos, evitando los conocidos ataques DDOS (Distributed Denial of Service) el cual no es más que un ataque de varias computadoras, que reúnen y dirigen su ancho de banda hacia un servidor en específico, con el fin de saturar sus procesos, y denegar los servicios a los clientes que hacen uso de este. Este tipo de ataques normalmente se solucionan un sistema de detección y prevención de intrusiones (IDS/IPS). (INCIBE, 2020)

### **Artículo 269d: Daño informático**

Donde se especifica que, sin estar facultado para ello, destruya, dañe, borre, deteriore, altere o suprima datos informáticos, o un sistema de tratamiento de información o sus partes o componentes lógicos, incurrirá en pena de prisión.

Como el anterior punto, el artículo se relaciona con la explotación de las vulnerabilidades, tema que no toca este documento. Pero se relaciona en cierta medida, ya que en esta práctica se exponen las vulnerabilidades, pero no se ejecutan.

### **Norma (ISO/IEC 27032.2)**

Publicada en 2012 por La Organización Internacional de Normalización, la norma ISO/IEC se enfoca en generar prácticas que vayan en defensa de los ataques de ingeniería social, malware, spyware y demás tipos de software que se enfoquen en el hackeo de usuarios y su información. La norma facilita la colaboración segura y fiable para proteger la privacidad de las personas en todo el mundo. De esta manera, puede ayudar a prepararse, detectar, monitorizar y responder a los ataques.

Esta norma sirve como base, para generar el informe que se dirigirá a la empresa luego de hacer un análisis a las pruebas de vulnerabilidades. El informe, es con el fin de proponer posibles soluciones a las vulnerabilidades anteriormente plasmadas, mejorando las prácticas y los componentes de la red, con el objetivo de evitar hackeos por parte de agentes mal intencionados.

### **Ley 1341 de 2009**

Esta ley se enfoca en establecer los principios y conceptos sobre la información y la organización de las tecnologías de la información y las comunicaciones. Esta normativa determina el marco general para la formulación de políticas que rigen el sector de las telecomunicaciones, se enfoca en la protección del usuario, su ordenamiento general, la calidad de servicio, la promoción de la inversión en el sector y el desarrollo de las tecnologías.

De igual forma en esta ley se deja en claro los reglamentos y los cumplimientos que debe tener un sistema de tecnología o telecomunicaciones, también especifica la protección del uso de información de usuarios, y el protocolo a seguir para su debida seguridad e integridad. Respecto a la relación de esta ley con esta investigación, podemos resaltar los protocolos que fortalecemos, al mejorar las defensas de una red empresarial a estas prácticas se les conoce como ciberdefensa.

## Metodología

En esta investigación se planea estudiar el uso de un clúster de computadoras para el testeo de vulnerabilidades a servidor virtualizado, para llevar a cabo el proyecto, es necesario como primer punto hacer una investigación que se enfoque tanto en construcción de redes virtuales como en el desarrollo de un sistema de computadoras en clúster que trabajen con un software singular para el testeo de vulnerabilidades (MPI, Nmap y Kali Linux), este punto es clave, ya que en la disyuntiva de software se dará a conocer un resultado que facilite su lectura, debido a la falta de propiedades y características en algunas aplicaciones (sin compatibilidad para ejecución en clúster) es posible que, al momento de comparar los resultados, estos no favorezcan la comparación puesto que las computadoras no funcionarán de la forma adecuada.

Para proseguir, es necesario reunir el hardware para los montajes, tanto del servidor virtual como del clúster de computadoras Raspberry pi, puede ser tardío y aunque no represente mucho esfuerzo, se reconoce el tiempo y el costo que esto llega a exigir, también es menester tener en cuenta la cantidad de tiempo y esfuerzo necesario para hacer el montaje de un servidor con vulnerabilidades disponibles, y que sean vistas por el software de pentesting. Debido a esta dificultad, se procede a hacer el montaje de una máquina virtual intencionalmente hackeable llamada BLACKROSE, y encontrada en los repositorios de VulnHub, la máquina cuenta con una computadora independiente que dedicara la parte del hardware suficiente para el correcto funcionamiento en el laboratorio. Esta máquina virtual facilita todo el proceso nombrado anteriormente y permite comparar los tiempos de ejecución, es importante plasmar que la máquina virtual se debe conectar en modo adaptador puente a la maquina física, para que pueda ser conectada a la red y analizada por el clúster de Raspberrys y la computadora de uso común

Una vez con los componentes de hardware y software sean obtenidos y se encuentren trabajando en conjunción, se procede a ejecutar los testeos de vulnerabilidades al servidor virtualizado, dando uso de varias aplicaciones enfocadas al Pentesting, estas aplicaciones fueron ejecutadas en las mismas condiciones por el clúster de raspberrys con MPI y Kali linux y el computador con procesador Intel Pentium corriendo Kali Linux de igual manera.

Por ultimo y como foco de este proyecto, se comparó los tiempos y efectividad de los análisis de vulnerabilidades, tomando como variable principal el tiempo y como variable secundaria el éxito del análisis, factor que también se tuvo en cuenta debido a problemas de compatibilidad por parte del clúster de Raspberrys. De este apartado se plasmó un informe donde se especifican los

resultados de las pruebas y los tiempos de ejecución, dando así una conclusión a la efectividad del clúster de Raspberrys enfocado al análisis de vulnerabilidades.

## Administración del proyecto

### Cronograma

Actividad 1. Adquirir el Hardware necesario para implementar el clúster (Cables, Switch, fuente, Memorias, ETC)

Actividad 2 Construir el sistema y la configuración del clúster de Raspberrys.

Actividad 3 Evaluar y configurar el software más optimo que se pueda correr en el clúster de computadoras (aplicaciones de Pentesting y sistemas operativos).

Actividad 4 Adquirir el Hardware y software necesario para implementar un servidor virtual que simule un entorno real.

Actividad 5 Configurar los dispositivos y servicios necesarios que componen la red (Clúster de Raspberrys-Servidor Victima- computadora para escáner y comparación.

Actividad 6 Comprobar el funcionamiento de la red y sus servicios.

Actividad 7 Ejecutar test de vulnerabilidades al servidor virtualizado usando clúster de Raspberry pi y software libre, con computadora común.

Actividad 8 Comparar los resultados de las pruebas con varias aplicaciones de pentesting.

Actividad 9 Hacer un breve informe de los resultados encontrados.

Tabla 1 *Cronograma*

Semana \ Actividad	Mes 1				Mes 2				Mes 3				Mes 4					
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4		
Actividad 1	■	■	■	■	■	■	■	■										
Actividad 2									■	■								
Actividad 3											■	■	■	■				
Actividad 4	■	■																
Actividad 5													■					
Actividad 6													■					
Actividad 7														■	■	■		
Actividad 8																■		
Actividad 9																■		

Fuente: Elaboración propia



## Presupuesto

Tabla 2 *Presupuesto*

Concepto	Valor unitario	Cantidad	Valor
Pc servidor	\$1.700.000	1	\$1'700.000
Mano de obra	\$3.000.000	3	\$9'000.000
PC para comparar	\$1'700.000	1	\$1'700.000
Raspberry pi 3 modelo B con SD	\$ 280.000	4	\$1'120.000
Raspberry rack 5 Layer	\$56.000	1	\$ 56.000
Switch 8 puertos	\$30.000	1	\$30.000
Cable UTP CAT5E	\$1000	10	\$10.000
Conectores RJ-45	\$800	15	\$12.000
Total			<b>\$ 13'628.000</b>

Fuente: Elaboración propia

### **Investigación:**

Investigación de software necesaria para el funcionamiento de las computadoras Raspberry PI en clúster paralelo:

Se hizo contacto vía Gmail con un investigador de la universidad del cauca para consultar sobre un proyecto parecido donde se usaba un clúster de Raspberrys, el SR Andrés Felipe Muños responde dando a conocer el sistema operativo (Kali Linux ARM) y el software para el uso del procesamiento en paralelo de las computadoras Raspberry: MPI y un intérprete a Python llamado mpi4py para las aplicaciones de escaneo de vulnerabilidades.

Como primera acción se debe conseguir los componentes físicos primordiales para la construcción de la red que conforma el clúster de raspberrys, estos son:

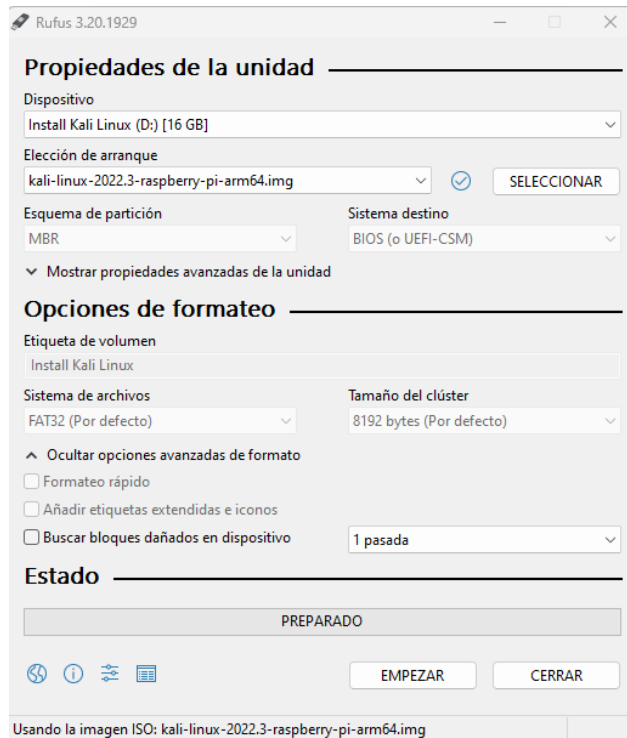
Cable cat 5E, conectores rj-45, ponchadora, switch cisco no administrable de 8 puertos, y las cuatro Raspberrys pi a usar.

En la siguiente imagen se puede notar los componentes reunidos y en uso:



**Figura 3.** Hardware reunido para conformar el clúster de computadoras Raspberry pi  
En la figura anterior no se nota el rack para raspberrys puesto que el componente no ha llegado debido a su envío puesto que viene desde china, se planea que para las siguientes semanas el componente se encuentre en el país listo para ser usado y enseñado en la siguiente entrega del documento

El primer parte del desarrollo del software comienza descargando la imagen ISO de Kali ARM, esta se puede descargar directamente de la página web de Kali en la sección de Download, seguido se debe usar un software capaz de cargar el sistema operativo en formato ISO a la memoria SD esto puede ser: Raspberry imager, Rufus, o AOMEI. En este proyecto se usó Rufus como se muestra en la siguiente imagen



**Figura 4.** Rufus listo para formatear memoria con sistema operativo KALI ARM

El paso siguiente es insertar la memoria a la minicomputadora y automáticamente se configura el sistema operativo.

Como todo debe llevar un orden, se debe iniciar cambiando el nombre de las Raspberrys, y asignando una de ellas como controladora de las otras. Esto quiere decir que el computador Máster asignara los procesos a las computadoras metidas en el clúster.

Los comandos para cambiar el nombre a la Raspberry son:

```
Sudo hostname Maestro
```

```
Sudo Nano/etc/hostname
```

Y cambiar el nombre en el archivo de configuración:

```
GNU nano 6.3 /etc/hostname
Maestro
```

**Figura 5.** Asignación de nombre a la primer Raspberry

Antes de continuar se recomienda desactivar la interfaz GUI para que la Raspberry pi libere procesos, y seguido ejecutar un reinicio para cargar los cambios realizados.

Los comandos son:

```
systemctl set-default multi-user.target
```

Para Volver a habilitar GUI

```
systemctl set-default graphical.target
```

```
sudo reboot
```

Para continuar se debe actualizar los repositorios y el sistema operativo con los siguientes comandos:

```
Sudo apt update && upgrade
```

Si el comando anterior devuelve un problema, este puede estar relacionado a la hora y fecha que tienen asignadas las Raspberrys debido a la versión del sistema operativo Kali ARM 2022.3

Se soluciona de la siguiente manera:

```
Sudo date -s "MM/DD/AAAA HH:mm" [Se debe Tener en cuenta que primero se especifica el mes antes que el día]
```

```
Sudo dpkg-reconfigure tzdata y seleccionar la zona horaria en la que se encuentra
```

Una vez solucionado el problema anterior, sigue la instalación y configuración de MPICH -Hydra Para clúster en paralelo

```
sudo apt install mpich
```

```
sudo mkdir /home/pi
```

```
sudo mkdir /home/pi/hydra4
```

```
cd /home/pi/hydra4
```

```
Sudo wget http://www.mpich.org/static/downloads/4.0.2/hydra-4.0.2.tar.gz [nota, tener en cuenta que se debe usar la última versión de mpich y hydra, esta se puede conseguir navegando en la página mpich.org]
```

```
Sudo tar xzf hydra.4.0.2.tar.gz
```

Se debe construir las carpetas para ubicar las utilidades del programa, estas las puede hacer como usted prefiera, en este proyecto se crearon así:

```

sudo mkdir /home/rpimpi/
sudo mkdir /home/rpimpi/mpi-install
sudo mkdir /home/pi/mpi-build
cd /home/pi/mpi-build
sudo /home/pi/hydra4/hydra-4.0.2/configure -prefix=/home/rpimpi/mpi-install [configurar la ruta
para la build de mpich]
sudo make [Construir los archivos de configuración de mpich]
sudo make install [Instalar los paquetes de configuración]
cd ~
sudo nano. bashrc

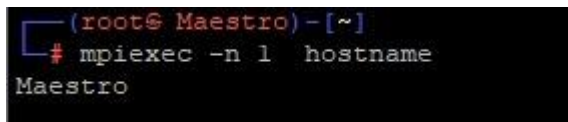
```

Para dejar la configuración de forma permanente y que no cambie de se debe configurar el archivo bashrc con la siguiente sentencia al final del archivo de texto:

```
PATH=$PATH:/home/rpimpi/mpi-install/bin
```

Ya realizados los pasos anteriores, al ejecutar la siguiente sentencia se debe mostrar el nombre anteriormente escogido para la Raspberry:

```
mpiexec -n 1 hostname
```



**Figura 6.** Mpich funcionando correctamente en una maquina PI de forma individual.

Naturalmente Mpich viene configurado para trabajar con software basado en c, c++ y fortran debido a esto el paso a seguir es instalar el intérprete a Python MPI4PY para que MPICH funcione con software desarrollado en Python y pueda correr los aplicativos de pentesting:

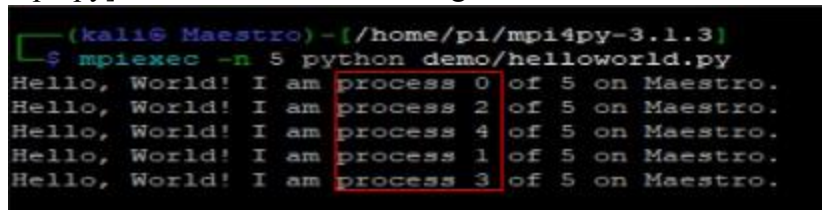
```

cd /home/pi
wget https://bitbucket.org/mpi4py/mpi4py/downloads/mpi4py-3.1.3.tar.gz
sudo tar -zxf mpi4py-3.1.3.tar.gz
cd mpi4py-3.1.3
sudo apt install python3
sudo apt install python2-dev
sudo python setup.py install [configurar los archivos de instalación]
export PYTHONPATH=/home/pi/mpi4py-3.1.3[Configurar el path de Mpi4py]

```

```
mpirun -n 5 python demo/helloworld.py
```

[Aquí se corre un ejemplo de Scribd en python para comprobar el correcto funcionamiento de mpi4py] El resultado debe ser el siguiente



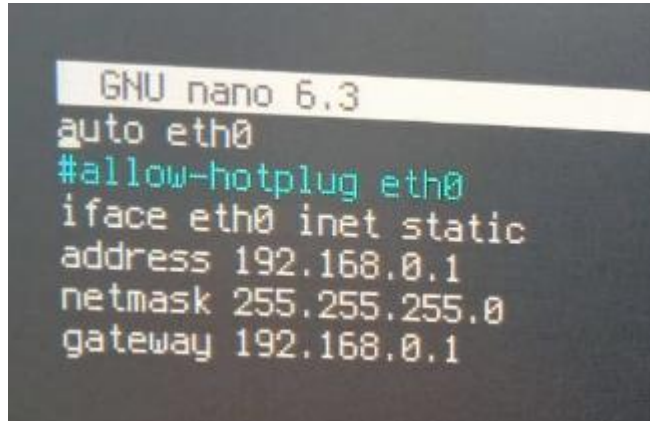
```
(kali@Maestro)~/home/pi/mpi4py-3.1.3$ mpirun -n 5 python demo/helloworld.py
Hello, World! I am process 0 of 5 on Maestro.
Hello, World! I am process 2 of 5 on Maestro.
Hello, World! I am process 4 of 5 on Maestro.
Hello, World! I am process 1 of 5 on Maestro.
Hello, World! I am process 3 of 5 on Maestro.
```

**Figura 7.** Mpi4py funcionando correctamente en maestro.

En la anterior imagen se especifica el número de procesos que se desea correr que son 5, y en el resultado se especifica en que núcleo del procesador de maestro corre cada proceso.

El paso por seguir es modificar la tarjeta de red especificando una dirección estática:

```
sudo nano /etc/network/interfaces.d/*
```



```
GNU nano 6.3
auto eth0
#allow-hotplug eth0
iface eth0 inet static
address 192.168.0.1
netmask 255.255.255.0
gateway 192.168.0.1
```

**Figura 8.** Ejemplo como debe quedar el direccionamiento de la maquina maestro

En este caso la dirección seleccionada es la 192.168.1.1 cambiando el segmento de red. Más adelante en este documento se especifica la topología de red del clúster de Raspberry pi indicando sus direcciones y hostname respectivamente.

El siguiente paso es activar ssh y permitir la conexión remota con el usuario root, esto con los siguientes comandos

```
update-rc.d ssh enable
```

```
nano /etc/ssh/sshd_config
```

```
#Port 22
#ListenAddress 0.0.0.0

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes_
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
#AuthorizedPrincipalsFile none
```

**Figura 9.** Archivo de configuración Ssh

sudo passwd root [Especificar una contraseña para el root, en este caso se escogió Kali]  
y listo, con esto es posible concluir la configuración necesaria para el archivo maestro.

Ahora se necesita repetir los pasos anteriores en los nodos esclavos que serán conectados al clúster. En proyectos similares, los investigadores deciden crear backups de los sistemas operativos con el fin de clonar estos y facilitar el proceso, pero debido al esfuerzo investigativo respecto a una herramienta que pueda hacer una copia del sistema operativo correctamente se desiste de esta opción y se procede a ejecutar los comandos anteriores en el resto de las Raspberrys.

Una vez montados y corriendo Mpich y Mpi4py en todos los nodos que conforman el clúster se deben conectar las Raspberrys en el switch y asegurarse de que exista conexión entre ellas, esto es indispensable para el paso a seguir.

Ahora es necesario generar e intercambiar las contraseñas rsa de ssh para que el protocolo MPI pueda usar los recursos de las maquinas conectadas en el clúster con total libertad, es decir con permisos de super usuario, con lo cual se debe ejecutar la siguiente secuencia de comandos:

desde #Maestro:

ssh-keygen

cd ~

cd .ssh

cp id\_rsa.pub Maestro [En este punto se crea un nuevo archivo de texto que copia la clave anteriormente generada].

```
ssh root@192.168.1.2
ssh-keygen
cd
cd .ssh
cp id_rsa.pub Esclavo01
scp 192.168.1.1:/root/.ssh/Maestro .
cat Maestro >> authorized_keys
exit
ssh root@192.168.1.3
ssh-keygen
cd .ssh
cp id_rsa.pub Esclavo02
scp 192.168.1.1:/root/.ssh/Maestro
ssh cat Maestro >> authorized_keys
exit
ssh root@192.168.1.3
ssh-keygen
cd
cd .ssh
cp id_rsa.pub Esclavo03
scp 192.168.1.1:/root/.ssh/Maestro
cat Maestro >> authorized_keys
exit
```

En este punto ya fueron configuradas las llaves rsa de los nodos en la computadora Maestro, para comprobar el correcto funcionamiento, es posible intentar una conexión ssh desde el equipo Maestro hacia todos los nodos esclavos, y estos permitirán la conexión ssh con root sin solicitar ninguna contraseña

El paso para seguir es ejecutar los siguientes comandos desde Maestro

```
cd ~
cd .ssh
scp 192.168.1.2:/root/.ssh/Esclavo01 .
cat Esclavo01 >> authorized_keys
```



```

scp 192.168.1.3:/root/.ssh/Esclavo02 .
cat Esclavo02 >> authorized_keys
scp 192.168.1.4:/root/.ssh/Esclavo03 .
cat Esclavo03 >> authorized_keys.
Ahora desde Esclavo 01 se debe ejecutar la misma sentencia para publicar las llaves rsa
cd ~
cd .ssh
scp 192.168.1.3:/root/.ssh/Esclavo02 .
cat Esclavo02 >> authorized_keys
scp 192.168.1.4:/root/.ssh/Esclavo03 .
cat Esclavo03 >> authorized_keys
Ahora desde Esclavo 02 se debe ejecutar la misma sentencia para publicar las llaves rsa
cd ~
cd .ssh
scp 192.168.1.2:/root/.ssh/Esclavo01 .
cat Esclavo01 >> authorized_keys
scp 192.168.1.4:/root/.ssh/Esclavo03 .
cat Esclavo03 >> authorized_keys
Ahora desde Esclavo 03 se debe ejecutar la misma sentencia para publicar las llaves rsa
cd ~
cd .ssh
scp 192.168.1.2:/root/.ssh/Esclavo01 .
cat Esclavo01 >> authorized_keys
scp 192.168.1.3:/root/.ssh/Esclavo02 .
cat Esclavo02 >> authorized_keys
Con todos los pasos anteriores correctamente ejecutados se debe crear la carpeta machine file en la
raíz del sistema #MAESTRO y especificar la dirección IP de todos los dispositivos en el clúster.
La sentencia es la siguiente
#Maestro cd ~
sudo nano machinefile

```

The screenshot shows a terminal window with the prompt 'root@Maestro: ~'. The user has opened the 'machinefile' file in 'GNU nano 6.3'. The file contains four lines of IP addresses: 192.168.1.1, 192.168.1.2, 192.168.1.3, and 192.168.1.4. The cursor is positioned at the beginning of the first line.

**Figura 10.** Archivo de configuración para las maquinas que componen el clúster

En este punto es posible comprobar el funcionamiento del clúster con la siguiente sentencia:

```

sudo mpiexec -f machinefile -n 4 python /home/pi/mpi4py-3.1.3/demo/helloworld.py [El
resultado debe ser el siguiente]

```

```
(root@ Maestro) - [~]
└─# mpiexec -f machinefile -n 4 python /home/pi/mpi4py-3.1.3/demo/helloworld.py
Hello, World! I am process 0 of 4 on Maestro.
Hello, World! I am process 1 of 4 on Esclavo01.
Hello, World! I am process 2 of 4 on Esclavo02.
Hello, World! I am process 3 of 4 on Esclavo3.
```

**Figura 11.** Procesos de scribd de prueba distribuido en los Equipos del clúster

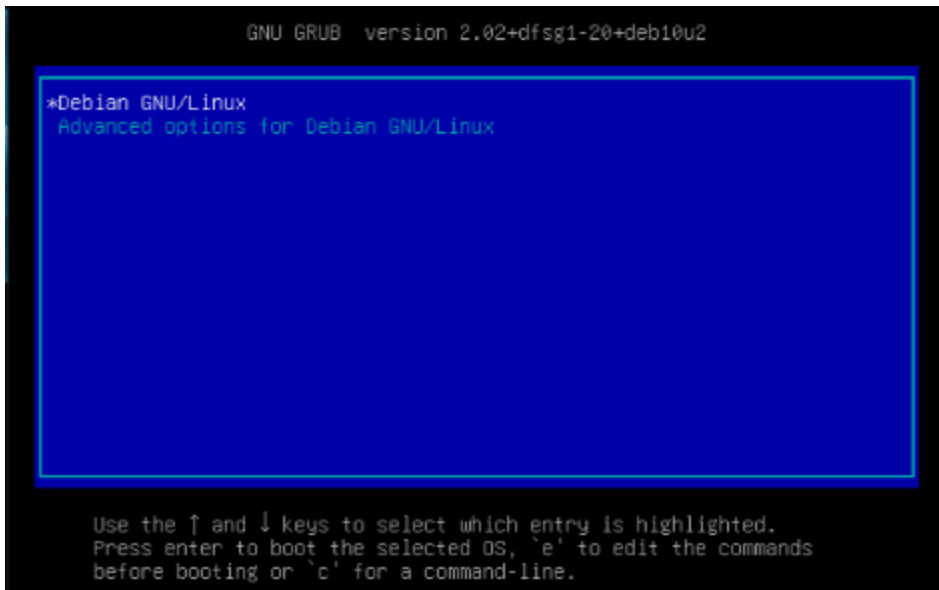
En la figura anterior se resalta que los procesos en la ejecución del Scribd “helloworld” desarrollado en python se está distribuyendo entre los procesadores de las raspberrys que conforman el clúster.

Con esto se daría por terminado el primer objetivo.

Como segundo objetivo se debe dedicar una maquina y todo su hardware para que funcione como víctima a los análisis de vulnerabilidades.

El primer paso es descargar virtual box o vmware para poder instalar una maquina intencionalmente Hackeable.

El servidor se decide montar virtual, ya que no es posible encontrar maquinas actuales en formato ISO para ser montadas en una maquina real, esto debido a protocolos de seguridad propuestos por la comunidad de aprendizaje, y así limitar la distribución de las maquinas con vulnerabilidades hackeables. Tampoco existe un método eficaz para poder convertir una máquina virtual. OVA en una imagen .ISO y que este se pueda pasar a una maquina real. Debido a esta dificultad se monta el servidor de forma virtual en virtual box, y se configura la red virtual en adaptador puente para que la maquina sea reconocida por el clúster de Raspberrys.

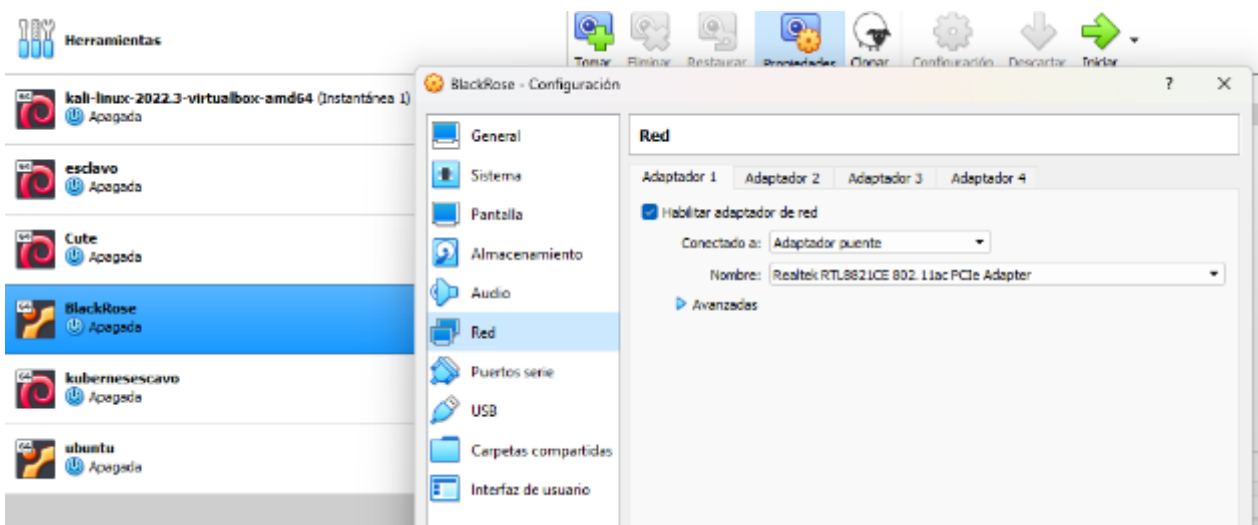


**Figura 12.** Servidor virtual en marcha

Una vez instalado y configurado Virtual Box o VMware en la maquina dedicada, se procede a descargar una maquina con servicios y vulnerabilidades activas que permitan simular un entorno que se pueda encontrar en la vida real y en el entorno empresarial.

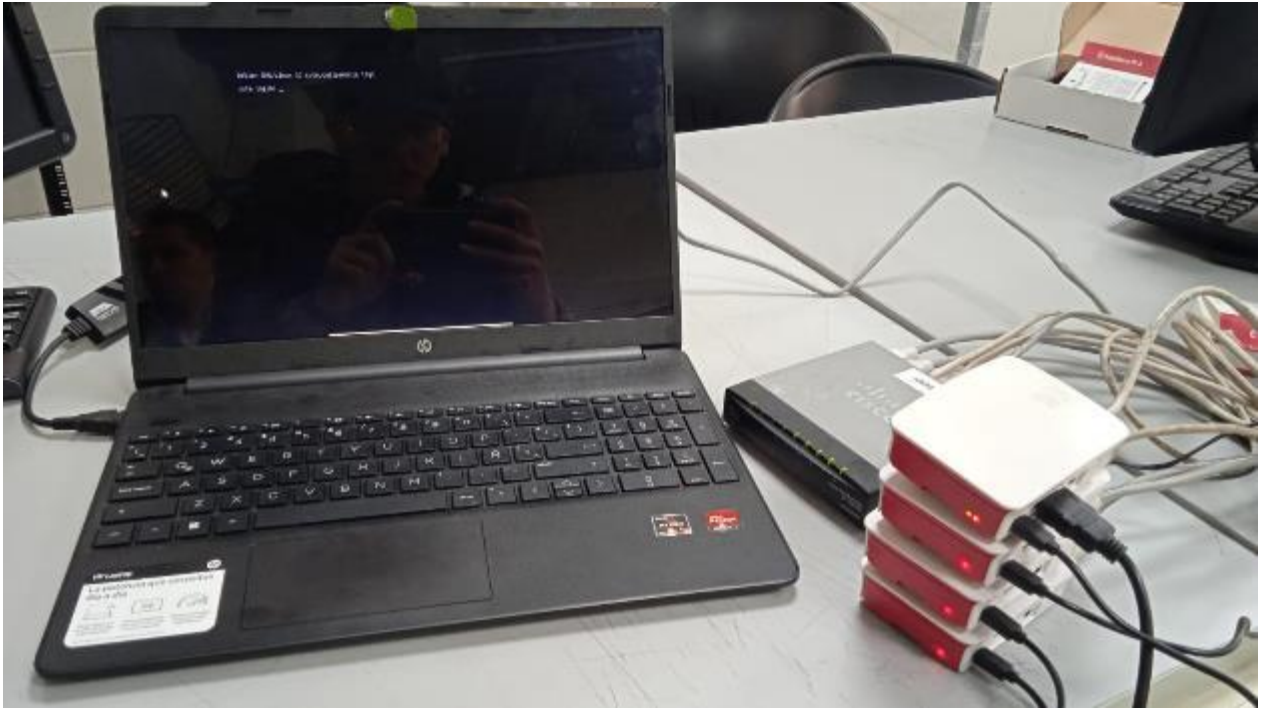
La máquina escogida lleva el nombre de BLACK ROSE y fue tomada de los repositorios de VulnHub, esta máquina cuenta con varios puertos abiertos y muchas vulnerabilidades para ser encontradas por el software de análisis.

El paso para seguir es poner la configuración de la red de la máquina virtual en modo adaptador puente para que su direccion ip sea reconocida por cualquier dispositivo de la red.



**Figura 13.** Red de máquina virtual configurada como adaptador puente

Ahora solo queda conectar la maquina real al switch donde se encuentra el clúster de computadoras utilizando un cable de red como se muestra en la siguiente imagen



**Figura 14.** servidor conectado a red de clúster de Raspberrys

Por último, validar que el clúster de Raspberrys es capaz de reconocer el dispositivo en la red

```
(root@Maestro)-[~]
# nmap -n 192.168.1.1-255
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-12 02:48 -05
Nmap scan report for 192.168.1.1
Host is up (0.000058s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap scan report for 192.168.1.2
Host is up (0.00056s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: B8:27:EB:65:09:F8 (Raspberry Pi Foundation)

Nmap scan report for 192.168.1.3
Host is up (0.00057s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: B8:27:EB:A7:EF:C5 (Raspberry Pi Foundation)

Nmap scan report for 192.168.1.4
Host is up (0.00056s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: B8:27:EB:B0:44:29 (Raspberry Pi Foundation)

Nmap scan report for 192.168.1.9
Host is up (0.0012s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2000/tcp  open  conf
MAC Address: 00:E0:4C:68:09:FD (Realtek Semiconductor)
```

**Figura 15.** Nmap reconoce el servidor virtual conectado a la Red

En la anterior imagen se resalta el reconocimiento del servidor en la red.

Una vez reconocido servidor en la red, se debe ejecutar los análisis de vulnerabilidades usando el clúster, para poder usar el clúster con este fin se hace efectiva la herramienta de uso libre, llamada secureCodeBox, cuya función principal es orquestar y automatizar herramientas usadas para el escáner de puertos y muchas otras actividades relacionadas con la seguridad informática de manera gratuita.

Comprobar la efectividad de esta herramienta es una de las actividades que impulsaron este proyecto, puesto que en investigaciones pasadas se ejecutaron análisis de vulnerabilidades con clúster de raspberrys, pero haciendo uso de metodologías y herramientas diferentes.

Para poder ejecutar secureCodeBox con éxito en el clúster hace falta unos prerrequisitos: Kubernetes y el gestor de archivos Helm, a continuación, se presenta como poner en marcha estos aplicativos:

Desde la maquina Maestro se debe ejecutar el siguiente comando con el fin de poner en marcha K3S una librería ligera de kubernetes que permite compartir recursos entre maquinas que componen el clúster, es esencial que las raspberrys cuenten con conexión a internet para ejecutar los siguientes comandos:

nano /boot/cmdline

y agregar lo siguiente al final de la línea:

cgroup\_memory=1 cgroup\_enable=memory [Necesario en para el funcionamiento de k3s]

curl -sfL https://get.k3s.io | sh - [Iniciamos K3s como maestro en el nodo maestro]

```
(root@ Maestro) - [~]
# curl -sfL https://get.k3s.io | sh -
[INFO] Finding release for channel stable
[INFO] Using vl.25.3+k3s1 as release
[INFO] Downloading hash https://github.com/k3s-io/k3s/releases/download/vl.25.3+k3s1/sha256sum-amd64.txt
[INFO] Skipping binary downloaded, installed k3s matches hash
[INFO] Skipping installation of SELinux RPM
[INFO] Skipping /usr/local/bin/kubectl symlink to k3s, already exists
[INFO] Skipping /usr/local/bin/crictl symlink to k3s, already exists
[INFO] Skipping /usr/local/bin/ctr symlink to k3s, already exists
[INFO] Creating killall script /usr/local/bin/k3s-killall.sh
[INFO] Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO] env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO] systemd: Creating service file /etc/systemd/system/k3s.service
[INFO] systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service -> /etc/systemd/system/k3s.service.
[INFO] No change detected so skipping service start
```

**Figura 16.** K3s en nodo maestro

Ahora para obtener el token y conectar las Raspberrys esclavos al cluster se debe copias el contenido del siguiente documento:

Cat /var/lib/rancher/k3s/server/node-token

```
(root@ Maestro) - [~]
# cat /var/lib/rancher/k3s/server/node-token
K10a9e640ea256401c082b90f2101169aceb792dca336177382640f3c347b294173::server:1221
eae6ea09e29056fc09cfaa388baf
```

**Figura 17.** Token acceso al clúster

Ahora desde las raspberrys esclavo se debe ejecutar lo siguiente:

nano /boot/cmdline

y agregar lo siguiente al final de la línea:

cgroup\_memory=1 cgroup\_enable=memory

ahora para añadir el esclavo al cluster:

curl -sfL https://get.k3s.io | K3S\_URL=https://myserver:6443 K3S\_TOKEN=mynodetoken sh

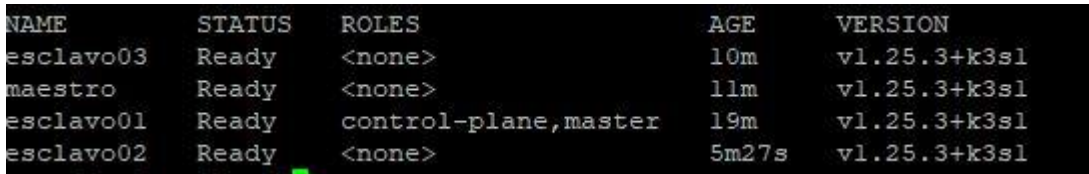
Tener en cuenta que en la sección señalada con rojo myserver se debe ubicar la dirección ip del nodo (192.168.1.1) Maestro, y en la sección mynodetoken se debe pegar el token anteriormente leído.

De esta forma se añaden los esclavos al clúster creado por el nodo Maestro.

Los dos comandos anteriores se deben ejecutar en todos los nodos esclavos.

Una vez añadidos todos los nodos esclavos al cluster k3s, en el nodo maestro se puede comprobar su conexión con la siguiente sentencia:

Kubectl get nodes:



NAME	STATUS	ROLES	AGE	VERSION
esclavo03	Ready	<none>	10m	v1.25.3+k3s1
maestro	Ready	<none>	11m	v1.25.3+k3s1
esclavo01	Ready	control-plane, master	19m	v1.25.3+k3s1
esclavo02	Ready	<none>	5m27s	v1.25.3+k3s1

**Figura 18.** clúster k3s funcionando

El paso para seguir es instalar el gestor de archivos Helm para kubernetes, este se ejecuta con los siguientes comandos en todas las raspberrys:

```
mkdir /home/pi/helm
```

```
cd /home/pi/helm
```

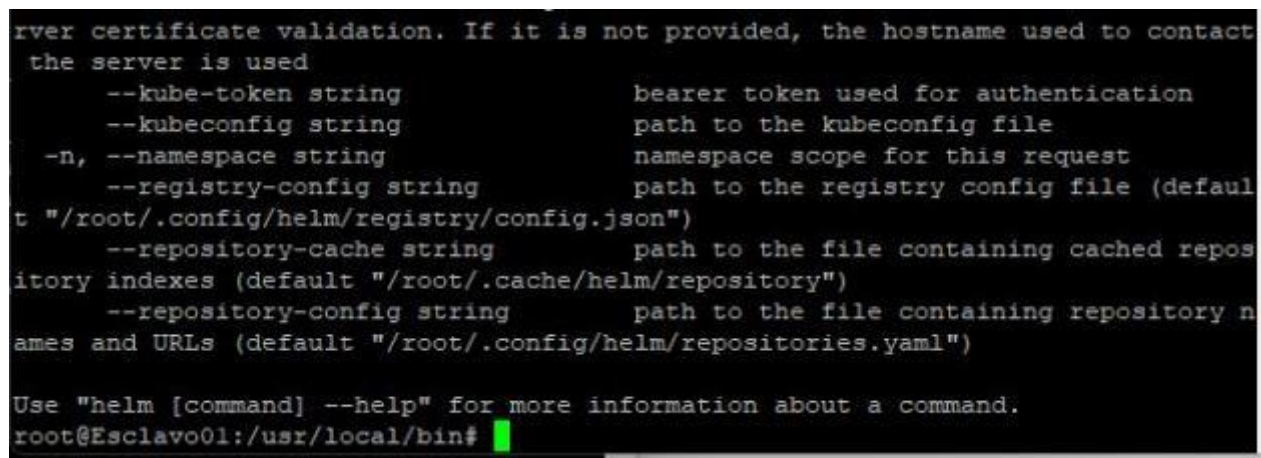
```
wget https://get.helm.sh/helm-v3.10.2-linux-arm64.tar.gz [helm para ARm64]
```

```
tar -zxvf helm-v3.10.2-linux-arm64.tar.gz
```

```
cp linux-arm64/helm /usr/local/bin/helm
```

para comprobar la correcta instalación de helm se debe ejecutar:

```
helm
```



```
server certificate validation. If it is not provided, the hostname used to contact
the server is used
  --kube-token string          bearer token used for authentication
  --kubeconfig string         path to the kubeconfig file
-n, --namespace string       namespace scope for this request
  --registry-config string    path to the registry config file (default
t "/root/.config/helm/registry/config.json")
  --repository-cache string   path to the file containing cached repos
itory indexes (default "/root/.cache/helm/repository")
  --repository-config string  path to the file containing repository n
ames and URLs (default "/root/.config/helm/repositories.yaml")

Use "helm [command] --help" for more information about a command.
root@Esclavo01:/usr/local/bin#
```

**Figura 19.** Helm funcionando

En este punto es donde comienza lo interesante del montaje del clúster de raspberrys, puesto que Helm cuenta con muchas herramientas y servicios, como wordpress, Redis, postgresql,kafka y demás, incluso es posible encontrar servidores de minecraft:

```
root@Esclavo01:/home/esclavo02/helm/linux-arm64# helm search hub minecraft
URL                                CHART VERSION  APP VERS
ION    DESCRIPTION
https://artifacthub.io/packages/helm/minecraft-...  4.4.0          SeeValue
s      Minecraft server
https://artifacthub.io/packages/helm/cloudnativ...  1.0.0          1.13.1
      Minecraft server
https://artifacthub.io/packages/helm/paull36597...  3.0.1          SeeValue
```

**Figura 20.** Buscador de helm en función

La instalación anterior de helm se debe ejecutar en todas las raspberrys que componen el clúster, se debe tener en cuenta la versión de procesador y sistema operativo que se está usando para descargarla en el cluster, para este caso de uso la versión ARM64.

Los pasos para instalar nmap son los siguientes:

```
helm repo add secureCodeBox https://charts.securecodebox.io [Añadir el repositorio de secure code box]
```

```
kubectl create namespace securecodebox-system
```

```
helm --namespace securecodebox-system upgrade --install securecodebox-operator secureCodeBox/operator
```

```
secureCodeBox Operator Deployed 📄
The operator can orchestrate the execution of various security scanning tools inside of your cluster.
You can find a list of all officially supported scanners here: https://www.securecodebox.io/
The website also lists other integrations, like persisting scan results to DefectDojo or Elasticsearch.

The operator send out regular telemetry pings to a central service.
This lets us, the secureCodeBox team, get a grasp on how much the secureCodeBox is used.
The submitted data is chosen to be as anonymous as possible.
You can find a complete report of the data submitted and links to the source-code at: https://www.securecodebox.io/docs/telemetry
The first ping is send one hour after the install, you can prevent this by upgrading the chart and setting 'telemetryEnabled' to 'false'.

~(root@Maestro) - [~/home/kali/nmap/nmap]
#
```

**Figura 21.** secureCodeBox corriendo en máster

Para este proyecto se encontró el siguiente error al instalar secure code box:

```
http://localhost:8080/version?timeout=32s": marcar tcp 127.0.0.1:8080: conectar: conexión rechazada.
```

La solución se dio con la siguiente sentencia:

```
export KUBECONFIG=/etc/rancher/k3s/k3s.yaml
```

```
kubectl config view --raw > ~/.kube/config
```

Con esto ya está funcionando SecurecodeBox, que es la herramienta encargada de orquestar los análisis de vulnerabilidades. A partir de aquí ya es posible descargar las aplicaciones encargadas de ejecutar los análisis de vulnerabilidades.



Para seleccionar los programas a usar, se realizó un breve estudio donde se ubicaron aplicaciones que pueden ser ejecutadas desde el clúster

de raspberrys y la maquina común, debido a que algunas aplicaciones de pentesting no pueden ser ejecutadas desde el cluster, y otras que están diseñadas específicamente para kubernetes tampoco pueden ponerse en marcha en la maquina HP con Kali Linux. como primera aplicación, se procede a descargar nmap:

```
helm install my-nmap securecodebox/nmap --version 3.15.0
```

```
└─# helm install nmap secureCodeBox/nmap
NAME: nmap
LAST DEPLOYED: Mon Nov 21 11:14:56 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

**Figura 22.** nmap para kubernetes corriendo

ahora se debe crear una carpeta para ubicar un archivo de configuración, en este archivo se especifican los target y características del escaneo que se planea hacer:

```
mkdir /home/pi/namp
```

```
cd /home/pi/namp
```

```
nano nmap-scan.yaml
```

y poner lo siguiente en el archivo de configuración: donde “name” será la dirección ip del host al que se desea testear. el archivo de configuración tiene las características de un escáner básico, pro este puede ser cambiado como indica el manual de la aplicación, para saber más puede ver la documentación en la página web

```
apiVersion: "execution.securecodebox.io/v1"
kind: Scan
metadata:
  name: "192.168.1.21"
spec:
  scanType: "nmap"
  parameters:
    - scanme.nmap.org
```

**Figura 23.** Archivo de configuración nmap

Una vez esté configurado el archivo, se puede ejecutar con la siguiente sentencia:

```
kubectl apply -f nmap-scan.yaml
```

y de esta manera se valida el proceso creado en kubernetes:

```
kubectl get scans
```

```
(root@ Maestro) - [~]
# kubectl get scans
NAME          TYPE    STATE    FINDINGS
192.168.1.21  nmap   Done     6
```

**Figura 24.** Ejecutando análisis de vulnerabilidad

Con la siguiente sentencia es posible ver los procesos ejecutados desde el clúster de kubernetes:

Kubectl get jobs

```
(root@ Maestro) - [~]
# kubectl get jobs
NAME                                COMPLETIONS  DURATION  AGE
scan-192.168.1.21-6n6qm             1/1           4s        160m
parse-192.168.1.21-gb4rt           1/1           21s       159m
```

**Figura 25.** primer análisis ejecutado y finalizado

Con la siguiente sentencia es posible ver los resultados de los análisis

kubectl logs job/scan-nmap-scanme.nmap.org-w66rp nmap

```
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-21 11:55 -05
Nmap scan report for 192.168.1.21
Host is up (0.0059s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
3306/tcp  open  mysql
MAC Address: 50:C2:E8:D4:74:ED (Cloud Network Technology Singapore PTE.)
```

**Figura 26.** resultado scanner con nmap

Para este proyecto no se tendrá en cuenta los puertos encontrados debido a que el análisis de estas vulnerabilidades sale del alcance de este proyecto, es debido resaltar que solo se planea evaluar la efectividad en tiempo de los análisis ejecutados.

De los resultados pasados se logra ver que el primer análisis ejecutado con nmap fue ejecutado con éxito, tuvo una duración de 4 segundos y encontró 6 puertos abiertos.

Ahora continua la ejecución del programa Nikto, también encontrado en secureCodeBox

El primer paso es la instalación con la siguiente sentencia:

```
helm install my-nikto securecodebox/nikto --version 3.15.0
```

```
(root@Maestro) - [~]
# helm install my-nikto securecodebox/nikto --version 3.15.0
NAME: my-nikto
LAST DEPLOYED: Mon Nov 21 14:39:43 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

**Figura 27.** Instalación de Nikto

Ahora crear el archivo de configuración donde se especifican las etiquetas de Nikto:

```
mkdir /home/pi/Nikto
```

```
cd /home/pi/nikto
```

```
nano /demo-docs.securecodebox.yaml
```

y agregar lo siguiente:

```
SPDX-FileCopyrightText: the secureCodeBox authors
#
# SPDX-License-Identifier: Apache-2.0

apiVersion: "execution.securecodebox.io/v1"
kind: Scan
metadata:
  name: "nikto-www.securecodebox.io"
  labels:
    organization: "secureCodeBox"
spec:
  scanType: "nikto"
  parameters:
    - "-h"
    - "192.168.1.21"
    - "-Tuning"
    # Only enable fast (ish) Scan Options, remove attack option like SQLi and R
    - "1,2,3,5,7,b"
```

**Figura 28.** archivo ejecución nikto

Por último, correr el programa

```
kubectl apply -f demo demo-docs.securecodebox.yaml
```

```
(root@Maestro) - [~/home/kali/Nikto]
# kubectl get jobs
NAME                                COMPLETIONS  DURATION  AGE
scan-nikto-www.securecodebox.io-vmsk8  1/1           55s       81s
parse-nikto-www.securecodebox.io-rtg6b  1/1           18s       26s

(root@Maestro) - [~/home/kali/Nikto]
# kubectl get scans
NAME                                TYPE  STATE  FINDINGS
nikto-www.securecodebox.io          nikto Done    7
```

**Figura 29.** ejecución de análisis nikto

La siguiente aplicación para probar es whatweb

La sentencia de instalación es la siguiente:

```
(root@Maestro) - [~/home/kali/whatweb]
# helm upgrade --install whatweb secureCodeBox/whatweb
Release "whatweb" does not exist. Installing it now.
NAME: whatweb
LAST DEPLOYED: Mon Nov 21 16:48:04 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

**Figura 30.** lcorrecta instalacion whatweb

Como en las aplicaciones anteriores se debe especificar el archivo de configuración:

`mkdir /home/pi/whatweb`

`cd /home/pi/whatweb`

`nano /whatweb.taml`

y en el archivo de configuración debe agregar lo siguiente:

```
GNU nano 6.4 whatweb.yaml
# SPDX-FileCopyrightText: the secureCodeBox authors
#
# SPDX-License-Identifier: Apache-2.0

apiVersion: "execution.securecodebox.io/v1"
kind: Scan
metadata:
  name: "whatweb-example"
spec:
  scanType: "whatweb"
  parameters:
    - 192.168.1.21
    -a 3
```

**Figura 31.** Archivo de configuración whatweb

Ahora se inicia el programa con:

```
kubectl apply -f whatweb.yaml
```

```
(root@ Maestro) - [~/home/kali/whatweb]
# kubectl get scans
NAME                                TYPE      STATE   FINDINGS
nikto-www.securecodebox.io         nikto     Done    7
whatweb-example                    whatweb   Done    2

(root@ Maestro) - [~/home/kali/whatweb]
# kubectl get jobs
NAME                                COMPLETIONS  DURATION  AGE
scan-nikto-www.securecodebox.io-vmsk8  1/1          55s       48m
parse-nikto-www.securecodebox.io-rtg6b  1/1          18s       48m
scan-whatweb-example-g29b7            1/1          56s       2m15s
parse-whatweb-example-jm6qj           1/1          21s       79s
```

**Figura 32.** ejecución de análisis de whatweb en paralelo

Con estas tres aplicaciones ya es posible hacer una comparación con la maquina HP.

Para el correcto despliegue de los análisis de vulnerabilidades con una maquina de un solo procesador de 4 núcleos, se instaló Kali linux como software principal, usando los mismos pasos necesarios para instalar el sistema operativo en las raspberrys, cambiando únicamente el sistema operativo por uno para procesadores x64/86.



**Figura 33.** computador HP a comparar

El primer programa seleccionado fue nmap, en este proyecto el software ya se encontraba instalado por defecto, así que no fue necesario realizar una instalación.

Por otra parte, se ejecutó el siguiente comando para realizar un análisis simple de la máquina virtual, y que tuviera las mismas etiquetas que la prueba ejecutada por el clúster:

```
nmap -p- -A -v 192.168.1.21
```

```
Completed Ping Scan at 17:09, 3.04s elapsed (1 total hosts)
Nmap scan report for 192.168.0.21 [host down]
NSE: Script Post-scanning.
Initiating NSE at 17:09
Completed NSE at 17:09, 0.00s elapsed
Initiating NSE at 17:09
Completed NSE at 17:09, 0.00s elapsed
Initiating NSE at 17:09
Completed NSE at 17:09, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 4.08 seconds
Raw packets sent: 8 (304B) | Rcvd: 0 (0B)
```

**Figura 34.** Ejecución de nmap

Como fue posible notar, la maquina arrojó un resultado en segundos de 4.08, dato que se tendrá en cuenta a la hora de comparar con las ejecuciones realizadas por el clúster de raspberrys-kubernetes.

La segunda aplicación que se instaló fue Nikto scanner, aplicación usada para encontrar vulnerabilidades y puertos abiertos en máquinas de todo tipo, para ejecutar los análisis fue necesario:

```
sudo apt install nikto
```

```
sudo nikto -h 192.168.1.21
```

```
+ Target Port:      80
+ Start Time:      2022-11-21 15:52:29 (GMT-5)
-----
+ Server: Apache/2.4.29 (Ubuntu)
+ Cookie PHPSESSID created without the httponly flag
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ OSVDB-3233: /icons/README: Apache default file found.

+ /login.php: Admin login page/section found.
+ 7915 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:        2022-11-21 15:54:14 (GMT-5) (105 seconds)
-----
+ 1 host(s) tested
```

**Figura 35.** Resultado de usar nikto

La aplicación que se escogió por último fue whatweb diseñada y enfocada para pentesting de aplicaciones web, funciona con el servidor del laboratorio, puesto que esta cuenta servicios web funcionales.

El paso que seguir para la ejecución del programa es:

```
sudo apt install whatweb
```

```
whatweb 192.168.1.21
```

```
(root@Prototipo)-[~]
# whatweb 192.168.1.21
http://192.168.1.21 [302 Found] Apache[2.4.29], Cookies[PHPSESSID], Country[RESERVED][22], HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[192.168.1.21], RedirectLocation[login.php]
http://192.168.1.21/login.php [200 OK] Apache[2.4.29], Bootstrap, Cookies[PHPSESSID], Country[RESERVED][22], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[192.168.1.21], JQuery, PasswordField, Script[text/javascript], Title[BlackRose]
```

**Figura 36.** whatweb en ejecución

whatweb no cuenta con una etiqueta que permita mostrar en pantalla el tiempo que le tomó su ejecución, así que este se tuvo que tomar con un cronometro por parte del investigador, debido a esto se debe tener en cuenta el error humano cuando se desee compara su efectividad.

Tabla 1. Comparación de hardware

Clúster de raspberry pi 3 modelo B+		Laptop HP 15-dw0083wm
Procesador	CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz4	Procesador Intel® Pentium® Silver N5000 1,10 GHz
Memoria RAM	1Gb x 4	4GB
Ethernet	10/100 /1000 Mbit/s	10/100 /1000 Mbit
Almacenamiento	SD 32GB	128GB SSD

Una vez ejecutados los análisis de vulnerabilidades, es posible presentar estos en una tabla, de la siguiente manera:

Tabla 2. Comparación de resultados

Clúster 4 raspberrys pi 3model b		Laptop HP 15-dw0083wm	
Programa	Tiempo (s)	Programa	Tiempo en (s)
Nmap	4	Nmap	4.8
Nikto	55	Nikto	105
WhatWeb	21	Whatweb	32

Como es posible resaltar si existe una mejora en los tiempos de ejecución de algunos programas, esto debido a la cantidad de núcleos de procesamiento que superan en un 400% a la maquina HP, pero que en términos matemáticos este x 4 en la cantidad de núcleos no refleja en los tiempos finales. aunque está claro que se deben hacer más estudios con más hardware incluido, este proyecto termina demostrando la efectividad del cómputo en paralelo en el ámbito del hacking ético e invita a otros investigadores a efectuar otras aplicaciones que den uso al clúster, pero llegando más afondo, y tal vez vulnerando alguna brecha aquí plasmada y aprovechando el computo en paralelo



## Conclusiones

La implementación de un clúster de computadoras se podría considerar como una solución de bajo costo, frente al valor de un ordenador moderno en escenarios donde se requiere procesar cierta cantidad información, como lo podrían ser las ciencias de la computación. Sin embargo, su despliegue conlleva cierto grado de dificultad técnica, en especial asociadas al montaje y a la búsqueda de información relacionada, esto puede significar un problema insalvable para una persona con pocos conocimientos técnicos en cuanto a sistemas operativos, virtualización, y redes se refiere.

Para la correcta configuración y funcionamiento del clúster de computadoras fue necesario leer y estudiar manual de instalación de MPICH redactado en inglés, puesto que en estudios previos se encontraron muchos errores de instalación debido a que estos no se construyeron con las últimas versiones de los softwares utilizados en el clúster (Mpich, Hydra, y Mpi4py). Es importante resaltar que en este documento queda plasmado el correcto uso de la herramienta Hydra-Mpich para conectar dispositivos en clúster de alto rendimiento (HPC) y que propone reemplazar los procesos y comandos mpiexec que se ejecutaban anteriormente con la aplicación de MPICH.

Se evidencio cuando se desea ejecutar múltiples pruebas pentesting en un ambiente controlado, es más práctico implementar un servidor intencionalmente hackeable en máquinas virtuales que en máquinas reales como sistema operativo principal, puesto que las maquinas encontradas en plataformas de aprendizaje como “try to hack me”, “Hack the Box” o vulhub, en su gran mayoría son máquinas virtuales en formato OVA. Aunque fue posible encontrar algunas imágenes de sistemas operativos hackéales y disponibles para uso del laboratorio, estas no fueron reconocidas por el computador cuando fueron cargadas en una memoria SD booteable, debido a una incompatibilidad en la lectura de la BIOS del ordenador víctima y el ISO del sistema operativo, estos errores se pueden generar debido a la fecha de publicación de las máquinas de laboratorio puesto que son de generaciones pasadas, en algunos casos como lo es la maquina “DE-ICE” la diferencia es de casi 15 años puesto que fue publicada en 2007.

Una posible solución en la cual se estuvo trabajando, fue generar copias de seguridad del sistema operativo de las maquinas hackéales mediante un disco externo y utilizando virtual box, pero esto requería más investigación para poder activar todos los servicios y recursos con los que no cuentan estas máquinas por defecto debido a los autores, que limitan el uso de los servicios de estas máquinas con fin de no ser usadas de forma indiscriminada por la comunidad.

Por lo tanto, se descartó esta opción y se decidió utilizar una máquina virtual con adaptador puente como configuración en virtual box, que le permite conectar a la red física y recibir los análisis de vulnerabilidades como si fuera una maquina real.

Por lo tanto, se descarta esta opción y se decide utilizar una máquina virtual con adaptador puente como configuración en virtual box, que le permite conectar a la red física y recibir los análisis de vulnerabilidades como una maquina real

Para ejecutar los análisis de vulnerabilidades fue preciso realizar una breve investigación sobre programas para realizar análisis de vulnerabilidades y seleccionar entre estos los que son compatibles entre kubernetes software que trabaja con el clúster de computadoras y Kali Linux amd64, el sistema operativo que se ejecuta en el dispositivo HP, esto con el fin de preparar de mejor manera el escenario de laboratorio con el fin de comparar la efectividad enfocándose en el hardware y no en el software, Dado que algunos programas están desarrollados netamente para el cluster como Kube Hunter o Kubeaudit , y que estos no pueden ser ejecutados de manera singular sin ninguna librería o extensión como es el caso del ordenador HP, de igual manera sucede con el cluster, algunas aplicaciones como Nessus o Netcat no están implementadas, o no se encuentran forma estable para el cluster de computadoras.

Como consecuencia de lo expuesto, es posible afirmar que el clúster de raspberry pi cumple una mejora significativa en el tiempo de ejecución de aplicaciones para el análisis de vulnerabilidades, aunque su ganancia dependa del software, puesto que en programas como Nikto se evidencia una mejora del 47.6% y en whatweb se denota una mejora del 34.5%, en programas como nmap solo se ve una pequeña mejora menor a un segundo equivalente al 2%. De manera global clúster de computadoras ha presentado una mejora en rendimiento del 28% en cuanto a la ejecución de análisis de vulnerabilidades utilizando 4 raspberry pi modelo 3B+ y un total de 16 núcleos de procesamiento. Se espera que, para otros ámbitos del hacking como inyección, cracking, o ataques de fuerza bruta se dé un mejor rendimiento debido a la naturalidad del cluster y del cómputo en paralelo.

## Referencias

- 1991, C. P. (2009). *Ley 1273*. bogota.
- Aldea, E. (2017). *Raspberry Pi Fundamentos y Aplicaciones*. RA-MA.
- Altube, R. (05 de 11 de 2021). *openwebinars.net*. Obtenido de <https://openwebinars.net/blog/kali-linux-que-es-y-caracteristicas-principales/>
- Amor, A. (2012). *Herramienta de Simulacion Remota en un Cluster de Computacion Cientifica*. Leganes.
- Andrés, M., Santiago, S., & Siler, D. (2018). *Pentesting sobre aplicaciones web basado en*. cauca : risti.
- Almeida, C., & Jasson, P. (2018). *implementacion de un laboratorio de seguridad de informatica para la realizacion de tecnicas de ataque y defensa*. guayaquil: Repositorio Universidad de Guayaquil .
- CONGRESO DE COLOMBIA. (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 comu*. Bogotá Dc.
- Fuentes, A. (2016). procesamiento estadístico en una empresa productora. *Eumed.net.*, 11.
- Coloma, G. (2012). *estrategia de implementación de un clúster de alta disponibilidad de n nodos sobre linux usando software libre*. quito: universidad san francisco de quito.
- Gonzales, P., Sánchez, G., & Soriano, M. (2013). *Pentesting con Khali*. Madrid: Oxword Computing .
- Hertzog, R. (2017). *Kali Linux*. Offsec Press.
- INCIBE. (2020). ¿Qué son y para qué sirven los SIEM, IDS e IPS? *INSTITUTO NACIONAL DE CIBER SEGURIDAD* , 12.
- Lavinder, K. (2016 ). *Ataques Cibernéticos ¿Está lista america latina?* Washington, D.C: lavinder.
- Pfister, F. G. (1998). *In Search of Clusters*. 2da Edición. Prentice Hall PTR.
- Rodríguez, E. (18 de 09 de 2018). *xataka.com*. Obtenido de <https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi#:~:text=La%20Raspberry%20Pi%20es%20un,igual%20que%20cualquier%20otra%20computadora.>
- SEGURIDAD, I. N. (2020). ¿Qué son y para qué sirven los SIEM, IDS e IPS? *INCIBE*, 12.

Serrano, J. (19 de 02 de 2021). *campusciberseguridad.com*. Obtenido de <https://www.campusciberseguridad.com/blog/item/139-que-es-el-pentesting#:~:text=El%20Pentesting%20es%20una%20abreviatura,posibles%20fallos%20en%20el%20mismo>.

Uniagustiniana. (22 de Septiembre de 2020). *Uniagustiniana*. Obtenido de <https://twitter.com/uniagustoficial>

Universitaria Agustiniana. (2018). *Estilo APA para la presentación de trabajos de grado*. Bogotá, Bogotá, Colombia.