

PROYECTO FIN DE GRADO

TÍTULO: ESTUDIO DE LA CAPACIDAD FORENSE DIGITAL DE SECURITY ONION

AUTOR/A: SERGIO IBÁÑEZ LORA

TITULACIÓN: GRADO EN INGENIERÍA TELEMÁTICA

TUTOR/A: CARLOS CARRILLO SÁNCHEZ

DEPARTAMENTO: INGENIERÍA TELEMÁTICA Y ELECTRÓNICA

VºBº TUTOR/A

Miembros del Tribunal Calificador:

PRESIDENTE/A: LUIS ARRIERO ENCINAS

TUTOR/A: CARLOS CARRILLO SÁNCHEZ

SECRETARIO/A: DAVID MELTZER CAMINO

Fecha de lectura: 8 DE JULIO DEL 2024

Calificación:

El Secretario/La Secretaria,

Resumen

El objetivo de este proyecto, titulado Estudio de la capacidad forense digital de Security Onion, es analizar la distribución Security Onion desde el punto de vista de su aplicación en temas de forensía digital. Para ello se ha comenzado por dar una visión general de lo que es el análisis forense digital, buscando diferentes distribuciones y sus herramientas para realizar una comparativa de lo que pueden hacer. En nuestro caso se ha decidido hacer una descripción general y algunas de sus herramientas, las distribuciones elegidas han sido Kali Linux, Parrot OS, Caine, Black Arch y principalmente Security Onion. Una vez descritas las distribuciones se ha realizado una comparativa hardware y de algunas de las herramientas que contienen en el ámbito de forensía digital de las mismas. Vista esta comparativa se observa que Security Onion está centrada en auditoría de red, por tanto para ver esa capacidad forense, que tenemos como objetivo, se decide crear simulaciones de casos que un analista forense podría tener. Para realizar estas simulaciones se va a utilizar un entorno virtualizado, por ello se hace una descripción general de qué es la virtualización. En este entorno virtualizado estarán interconectadas la suite Security Onion, una máquina víctima, que será Metasploitable3, y una máquina atacante que será Kali Linux. Todas estas máquinas estarán virtualizadas con VirtualBox. Los casos que se han elegido son el robo de archivos, un malware, el análisis de archivos y la recuperación de datos. Tras el análisis de estos casos, tenemos la conclusión que la distribución Security Onion únicamente se centra en temas de que tengan que ver con el tráfico de red pudiendo dar indicios de lo que ha ocurrido, pero en otros ámbitos de las forensía digital no dispone de herramientas para realizar ciertas tareas. Por tanto, se define que la distribución Security Onion no es la mejor opción para el análisis forense digital como lo pudieran ser otras distribuciones.

Abstract

The aim of this project, titled "Study of the digital forensic capacity of Security Onion," is to analyze the Security Onion distribution from the perspective of its application in digital forensics. To this end, the project begins with an overview of what digital forensic analysis entails, examining different distributions and their tools to compare their functionalities. In our case, we have decided to provide a general description and some of the tools of the selected distributions, which include Kali Linux, Parrot OS, Caine, Black Arch, and primarily Security Onion. After describing the distributions, a comparison of the hardware and some of the forensic tools they contain has been conducted. From this comparison, it is observed that Security Onion is focused on network auditing. Therefore, to assess its forensic capability, which is our goal, we decided to create simulations of cases that a forensic analyst might encounter. These simulations will be conducted using a virtualized environment, so a general description of virtualization is provided. In this virtualized environment, the Security Onion suite, a victim machine (Metasploitable3), and an attacker machine (Kali Linux) will be interconnected. All these machines will be virtualized using VirtualBox. The selected cases include file theft, malware, file analysis, and data recovery. After analyzing these cases, we concluded that the Security Onion distribution is only focused on network traffic-related issues, providing clues about what has occurred, but lacks the tools to perform certain tasks in other areas of digital forensics. Therefore, it is determined that the Security Onion distribution is not the best option for digital forensic analysis compared to other distributions.

Índice de ilustraciones

Ilustración 1 Fases del análisis forense digital	4
Ilustración 2 Logo Kali Linux	6
Ilustración 3 Logo Parrot OS.....	7
Ilustración 4 Logo Black Arch	8
Ilustración 5 Logo CAINE.....	9
Ilustración 6 Logo Security Onion.....	9
Tabla 1 Características hardware de las distribuciones	10
Tabla 2 Comparativa de las herramientas de las distribuciones.....	11
Ilustración 7 Diagrama lógico de la infraestructura.....	15
Ilustración 8 Escaneo de puertos	17
Ilustración 9 Ataque por fuerza bruta	18
Ilustración 10 Conexión con credenciales obtenidas tras ataque	18
Ilustración 11 Archivos en la máquina víctima	18
Ilustración 12 Robo de archivos	18
Ilustración 13 Datos obtenidos en la máquina atacante	19
Ilustración 14 Eventos generados en el robo de archivos	19
Ilustración 15 Evento de escaneo de puertos	19
Ilustración 16 Escaneo de puertos	20
Ilustración 17 Conexiones establecidas.....	20
Ilustración 18 Puertos abiertos.....	20
Ilustración 19 Eventos generados por ataque al servicio FTP	21
Ilustración 20 Acceso incorrecto al servicio FTP	21
Ilustración 21 Usuarios con interacción con el servicio FTP	21
Ilustración 22 Interacciones del usuario vagrant	22
Ilustración 23 Acceso del usuario vagrant	22
Ilustración 24 Obtención del archivo datos.txt	22
Ilustración 25 Filtro FTP DATA	22
Ilustración 26 Contenido archivo datos.txt.....	23
Ilustración 27 Importar archivo vía web.....	24
Ilustración 28 Inicio de importación	24
Ilustración 29 Importación completada	24
Ilustración 30 Rango de tiempo.....	24
Ilustración 31 Eventos ocurridos	25
Ilustración 32 Intento de acceso al servicio FTP.....	25
Ilustración 33 Filtro de eventos por IP origen	25
Ilustración 34 Filtro de posibles IPs de atacantes	26
Ilustración 35 Eventos con IP 120.78.199.189.....	26
Ilustración 36 Eventos agrupados por puerto destino	26
Ilustración 37 Eventos generados por la IP 3.109.209.43.....	26
Ilustración 38 Búsqueda de puertos abiertos.....	27

Ilustración 39 Puertos abiertos.....	27
Ilustración 40 Comienzo del ataque.....	28
Ilustración 41 Usuarios y contraseñas usados para el ataque.....	28
Ilustración 42 Usuarios FTP	28
Ilustración 43 Eventos FTP con IP origen la del atacante.....	29
Ilustración 44 Acceso confirmado	29
Ilustración 45 Archivos a los que se han tenido acceso	29
Ilustración 46 Datos en ese rango de tiempo	29
Ilustración 47 Archivo .backup.....	30
Ilustración 48 Puerto destino 29999	30
Ilustración 49 Secuencia de puertos	31
Ilustración 50 Evento de acceso	31
Ilustración 51 Acceso correcto al servidor FTP interno	31
Ilustración 52 Archivo .fetch.sh	31
Ilustración 53 Acceso por SSH del atacante	32
Ilustración 54 Posible descarga de malware.....	32
Ilustración 55 Descarga de Ransomware	32
Ilustración 56 Foto original.....	33
Ilustración 57 Imagen a añadir.....	33
Ilustración 58 Imagen con datos ocultos.....	34
Ilustración 59 Archivos de máquina víctima	34
Ilustración 60 Descarga archivo	34
Ilustración 61 Subida de archivo con datos ocultos.....	35
Ilustración 62 Alarmas que se han registrado durante la simulación	35
Ilustración 63 Eventos filtrados	36
Ilustración 64 Eventos de descarga y subida FTP	36
Ilustración 65 Datos FTP.....	36
Ilustración 66 Hash de Foto original.....	36
Ilustración 67 Hash de foto con datos ocultos	36
Ilustración 68 Dispositivo USB conectado	37
Tabla 3 Coste humano.....	39
Tabla 4 Coste materiales.....	39
Ilustración 69 Security Onion con la configuración Import.....	49
Ilustración 70 Security Onion con la configuración Evaluation	50
Ilustración 71 Security Onion con la configuración Standalone.....	51
Ilustración 72 Security Onion con la configuración Distributed.....	52
Ilustración 73 Diagrama con configuración NAT	53
Ilustración 74 Diagrama con configuración Bridge	53
Ilustración 75 Diagrama con configuración Red Interna	54
Ilustración 76 Diagrama con configuración Host-Only	54
Ilustración 77 Diagrama con configuración Red NAT	55
Ilustración 78 Instalación Security Onion mediante ISO	55
Ilustración 79 Elección de sistema operativo.....	56

Ilustración 80 Capacidad disco duro para Security Onion	56
Ilustración 81 Requisitos de memoria de la máquina virtual para Security Onion	56
Ilustración 82 Características hardware de la máquina virtual para Security Onion	57
Ilustración 83 Elección del tipo de infraestructura de Security Onion.....	57
Ilustración 84 Elección de la tarjeta de red para gestión.....	58
Ilustración 85 Utilización de DHCP en la tarjeta de red de gestión	58
Ilustración 86 Elección de la tarjeta de red para capturar tráfico	58
Ilustración 87 Cuenta de administrador	59
Ilustración 88 Acceso a Security Onion mediante IP	59
Ilustración 89 Opciones elegidas para Security Onion	60
Ilustración 90 Instalación completada correctamente y accesos	60
Ilustración 91 Comando so-ip-update	60
Ilustración 92 Comandos para descarga e instalación Metasploitable3	61
Ilustración 93 Configuración interfaz de red Host-Only estática	61
Ilustración 94 Cambio de IP estática a DHCP	62
Ilustración 95 IP asignada por VirtualBox para interfaz de red modo Host-Only	62
Ilustración 96 Paquete con Kali Linux	62
Ilustración 97 Características Hardware Kali Linux.....	63

Lista de acrónimos

CPU	Central Processing Unit
FTP	File Transfer Protocol
GB	Gigabyte
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
MB	Megabyte
NAT	Network Address Translation
NIDS	Network Intrusion Detection System
RAM	Random Access Memory
VM	Virtual Machine

Índice de contenidos

Resumen	i
Abstract	iii
Índice de ilustraciones	v
Lista de acrónimos.....	ix
1. Introducción.....	1
1.1 Marco y motivación del proyecto.....	1
1.2 Objetivos técnicos y académicos	1
1.3 Estructura del resto de la memoria	2
2. Marco tecnológico.....	3
2.1 Análisis forense digital	3
2.2 Distribuciones orientadas al análisis forense digital	4
2.2.1 Kali Linux	5
2.2.2 Parrot OS.....	6
2.2.3 Black Arch.....	7
2.2.4 CAINE	8
2.2.5 Security Onion	9
2.2.6 Comparativa entre las distintas distribuciones.....	10
2.3 Virtualización	11
3. Especificaciones y restricciones de diseño.....	13
3.1 Especificaciones	13
3.2 Restricciones de diseño	13
4. Descripción de la solución propuesta.....	15
5. Resultados	17
5.1 Robo de archivos	17
5.2 Malware	23
5.3 Análisis de archivos.....	33
5.4 Recuperación de datos	37
6. Presupuesto	39
7. Impacto del proyecto.....	41
8. Conclusiones	43
8.1 Conclusiones	43
8.2 Líneas futuras.....	43
9. Referencias	45
Anexo	49
A.1 Modos de configuración de Security Onion	49
A.2 Modos de configuración de los adaptadores de red de VirtualBox.....	52
A.3 Instalaciones de máquinas virtuales.....	55

A.3.1.	Montaje de Security Onion.....	55
A.3.2.	Montaje de Metasploitable3.....	61
A.3.3.	Montaje de Kali Linux	62

1. Introducción

1.1 Marco y motivación del proyecto

Hoy en día es muy importante la seguridad informática, y en especial la prevención de situaciones críticas debido a agentes tales como malware, virus, y usos indebidos del sistema informático. Pero es mucho más importante la detección de las huellas que estos agentes dejan, voluntaria e involuntariamente, y que nos permiten conocer, extraer y analizar la existencia de todas las evidencias electrónicas que forman parte de dichas huellas. Por ello, es fundamental encontrar la herramienta adecuada para poder extraer dichas evidencias, dotarlas de validez legal y poder demostrar que ha ocurrido en nuestra red o sistema.

Las amenazas informáticas afectan a todo sistema. Existen herramientas que analizan la existencia de posibles amenazas y otras que analizan huellas que han dejado dichas amenazas. Normalmente es bastante complicado y requiere mucho tiempo analizar detalladamente una red y poder asegurar que está libre de cualquier amenaza [1]. El nivel de seguridad de la red necesario depende mucho de la información que se maneja en esta. Por ello no está de más intentar mejorar los sistemas de seguridad, así como los sistemas de alertas de nuestra red. Existen bastantes distribuciones de suites Open Source como Security Onion, Kali Linux, Parrot OS, Caine, Black Arch que nos pueden ayudar a conocer que ha ocurrido en nuestros sistemas [2]. En este trabajo nos centraremos en analizar las posibles huellas o patrones de ataque que se producen.

Security Onion es la que más se centra en la auditoría de la seguridad a nivel de redes [3]. Por eso este proyecto se centrará en describir y analizar cada una de las herramientas que contiene esta distribución, además de ver el potencial de análisis que tienen frente a las situaciones críticas que hemos mencionado anteriormente.

1.2 Objetivos técnicos y académicos

El proyecto tendrá como objetivo analizar la distribución Security Onion desde el punto de vista de su aplicación en temas de análisis forense. Para ello realizaremos un estudio detallado de esta distribución comparándola con otras distribuciones como Kali Linux, Parrot, etc. Nos enfocaremos en las herramientas directamente relacionadas con el análisis forense incluidas en estas distribuciones.

Es necesario centrarnos en las herramientas que sean más idóneas para poder analizar las redes intentando demostrar que ha ocurrido en la red después de revisar la información que aportan dichas herramientas.

Por otro lado, intentar simular de alguna forma ya sean ataques, malware, intrusiones, corrupción o robo de archivos u otro tipo de situaciones, para que gracias a las herramientas de Security Onion podamos obtener información de lo que ha ocurrido en nuestra red.

Para el estudio de la capacidad de análisis forense digital de esta distribución se realizará el diseño de una pequeña infraestructura de red con varios equipos virtualizados. Esta infraestructura constará de al menos tres equipos siendo uno de ellos el que contenga la distribución.

Una vez montada esta infraestructura se realizarán pruebas contra o en los equipos, ya sea la simulación de ataques, simular que alguno de los equipos contenga algún malware o cualquier prueba que se pueda analizar gracias a esta distribución para evaluar su potencial.

1.3 Estructura del resto de la memoria

Para la realización del proyecto, siguiendo los objetivos y la metodología de trabajo propuesta se ha decidido seguir la siguiente estructura. En el capítulo 2, llamado Estado del arte, presentaremos los conceptos básicos respecto análisis forense, se hará un estudio comparativo entre diferentes Suites y la que se ha elegido para la realización del proyecto, Security Onion, así como las herramientas que contiene. A continuación, en el capítulo 3 se plantean los requisitos del proyecto y las restricciones impuestas para el desarrollo de este TFG. En el siguiente capítulo, desarrollaremos nuestro proyecto, donde diseñaremos la infraestructura del proyecto formada por varios equipos virtualizados. Además, se plantearán casos de uso para analizar la capacidad forense digital de Security Onion. De todos estos equipos, uno de ellos contendrá la propia suite que vamos a estudiar Security Onion, otro equipo será una máquina atacante y otro equipo será la máquina víctima. Todas ellas conectadas en la misma subred para que tengan conectividad. Finalmente, terminaremos con nuestras conclusiones y plantearemos las líneas futuras de este trabajo.

2. Marco tecnológico

En este capítulo vamos a detallar qué es el análisis forense digital y alguno de los conceptos básicos, así como realizar una breve descripción de varias distribuciones que pueden estar orientadas al análisis forense digital. También se realizará una comparativa de Security Onion con otras suites que también se utilizan actualmente.

2.1 Análisis forense digital

A lo largo de este apartado vamos a presentar diferentes conceptos relacionados con forensía digital, que es una rama de la ciencia forense que se concentra en la investigación y recuperación de información encontrada en dispositivos digitales relacionados con los delitos cibernéticos.

El término análisis forense digital se usó en primer lugar como un sinónimo de análisis forense informático. Desde entonces, se ha ampliado para cubrir la investigación de cualquier dispositivo que pueda almacenar datos digitales. Aunque, como indica el autor, el primer delito informático no se informó hasta el año 1978, no fue hasta la década de 1990 que se convirtió en un término reconocido. Fue sólo a principios del siglo XXI cuando surgieron políticas sobre análisis forense digital [4].

El análisis forense digital es el proceso de descubrir e interpretar datos electrónicos. El objetivo del proceso es preservar cualquier evidencia en su forma más original mientras se realiza una investigación estructurada recopilando, identificando y validando la información digital para reconstruir eventos pasados, como se describe en el artículo [5].

Para que la evidencia digital sea aceptada, el análisis forense digital normalmente suele definirse en cinco pasos [6], tal y como se ve en la ilustración 1 y que a continuación se detallan:

- **Identificación:** La primera de las fases es básicamente el proceso de determinar que ha ocurrido o qué evidencias se presentan.
- **Adquisición:** En esta fase se obtienen copias de la información que se sospecha que puede estar vinculada con algún incidente. De este modo, hay que evitar modificar cualquier tipo de dato realizando una copia bite a bite con las herramientas y dispositivos adecuados. La adquisición de las evidencias es muy importante seguir una norma fundamental que está ligada a la volatilidad, por lo que se deberán recolectar en el orden de la más volátil a la que tenga menor volatilidad. Un ejemplo sería recolectar primero datos relevantes a la memoria, contenidos del cache y por último datos contenidos en un soporte de almacenamiento.
- **Preservación:** Se debe garantizar que la información recopilada no se destruya o sea transformada. Es decir, que nunca debe realizarse un análisis sobre la muestra incautada, sino que deberá ser copiada y sobre esta se deberá realizar el peritaje. En muchos casos deberemos utilizar técnicas de Hash para identificar de forma inequívoca determinados archivos.

- **Análisis:** Una vez obtenida la información y preservada se pasa a la fase más técnica, donde se utiliza tanto hardware como software diseñado para este tipo de análisis. Es muy importante en esta fase la evaluación de criticidad del incidente.
- **Presentación:** En esta última fase se realiza un informe con el fin de exponer lo ocurrido con pruebas. Para ello previamente en las fases anteriores se ha debido documentar todo correctamente.



Ilustración 1 Fases del análisis forense digital

2.2 Distribuciones orientadas al análisis forense digital

En esta sección se explicará qué es una suite o distribución de software y exponer las características y herramientas que tienen algunas de ellas para realizar el análisis forense digital, así como una comparativa entre ellas.

Una suite o distribución consiste en un conjunto de aplicaciones software que aportan funciones o herramientas al usuario dentro de un mismo paquete [7]. Dentro de ese mismo paquete se pueden encontrar herramientas muy útiles para realizar auditorías de sistemas informáticos, como por ejemplo, captura y monitorización de tráfico, pruebas de intrusión o análisis de memorias, etc.

Actualmente existen muchas distribuciones que disponen de una gran cantidad de herramientas de informática forense, pero no existe una distribución perfecta ni validada por ningún estándar, lo cual lleva a depender de la necesidad de revisar que se necesita por parte del usuario para la elección de dichas distribuciones.

A continuación, vamos a describir y comparar varias de las distribuciones que pueden ser más conocidas.

2.2.1 Kali Linux

Kali Linux es una distribución de Linux basada en Debian desarrollada por Offensive Security. Esta distribución es muy conocida en la comunidad de hacking ético por contener un gran conjunto de herramienta de seguridad y auditoría de redes [8].

Las características más importantes de esta distribución son [9]:

- Contiene más de 600 herramientas preinstaladas para la realización de análisis forense digital, auditorías de red, pruebas de penetración, entre otro tipo de tareas.
- Utiliza una estructura modular lo que permite personalizar a los usuarios su entorno de trabajo. Pueden configurar el sistema operativo con las preferencias que necesiten, así como añadir o eliminar herramientas en función de las necesidades de uso.
- Se realizan actualizaciones frecuentemente que permiten que el sistema operativo y las herramientas tengan las últimas versiones y características. Lo más interesante es que estas actualizaciones se implementan a medida que están disponibles, utilizando así un modelo de lanzamiento continuo.
- Es compatible con varias arquitecturas de hardware, incluyendo ARM, ARM64, x86 y x86-64. Esta característica es muy importante, ya que, permite al usuario utilizar la distribución en distintos dispositivos, ya sea portátiles o computadores de escritorio incluso en dispositivos móviles.

Algunos de los requisitos de hardware son:

- Se necesita como mínimo 1 GB de RAM.
- Un mínimo de 20 GB de espacio en disco duro.
- Una CPU como mínimo de doble núcleo de 1 GHz.

Esta distribución es una de las más reconocidas en la comunidad de la ciberseguridad, además de una de las más completas.

Las funciones más importantes de esta distribución son [10]:

- **Recopilación de información:** Obtención de los máximos datos posibles del sistema objetivo.
- **Análisis de vulnerabilidades:** Identificación de fallos de seguridad que pueden ser utilizados para un ciberataque.
- **Análisis de aplicaciones web:** Escaneo de sitios web en busca de vulnerabilidades.
- **Evaluación de bases de datos:** Escaneo automatizado de vulnerabilidades de en bases de datos.
- **Ataques de contraseñas:** Realización de ataques de diccionario, tablas y fuerza bruta para descubrir contraseñas.
- **Ataques Wireless:** Realización de ciberataques por medio de redes inalámbricas.
- **Sniffing y Spoofing:** Robo de datos y suplantación de identidad.

Algunas de las herramientas forenses que contiene la distribución y se puede encontrar en su documentación son [11]:

- **RegRipper:** Es una herramienta que se utiliza para extraer información del registro de Windows.
- **Binwalk:** Se utiliza para buscar en una imagen binaria archivos embebidos y código ejecutable.
- **Xplico:** Su objetivo es extraer de una captura de tráfico de red los datos contenidos de las aplicaciones. Puede extraer cada correo electrónico o todo el contenido HTTP.
- **Bulk-extractor:** Es una herramienta que permite escanear imágenes de disco, archivos o directorios de archivo y extrae información como direcciones de correo, número de tarjetas de crédito, archivos JPEG u otro tipo de información sin tener que analizar el sistema de archivos.



Ilustración 2 Logo Kali Linux

2.2.2 Parrot OS

Parrot OS es una distribución de Linux basada en Debian creado y mantenido por Frozenbox. Es una herramienta muy parecida a Kali Linux, ya que incluye una gran cantidad de herramientas de seguridad y forense digital, pero esta distribución también incluye lo necesario para proteger la privacidad del usuario mientras navega por Internet así como desarrollar programas propios [12].

Las características más importantes de esta distribución son [13]:

- Contiene más de 600 herramientas preinstaladas para la realización de análisis forense digital, auditorías de red, pruebas de penetración, entre otro tipo de tareas. Algunas de las herramientas más reconocidas son TOR, Aircrack-ng o OpenVAS.
- Se realizan actualizaciones frecuentemente que permiten que el sistema operativo y las herramientas tengan las últimas versiones y características.
- Una de las grandes ventajas de esta distribución es que es muy liviana lo que permite su uso en dispositivos no tan potentes, ya que no requiere demasiados recursos en comparación de otras.
- Tiene una interfaz bastante más amigable que otras distribuciones lo que permite que usuarios más principiantes tengan un mejor manejo de la distribución.

Algunos de los requisitos de hardware son:

- Se necesita como mínimo 320 MB de RAM.
- Al menos 16 GB de espacio en disco duro.
- Una CPU como mínimo de doble núcleo de 1 GHz.

Dispone de herramientas forenses como [14]:

- **Autopsy:** Es una interfaz gráfica que contiene el conjunto de herramientas de Sleuth Kit, que permiten extraer datos de discos duros y otro tipo de almacenamientos.
- **Galleta:** Se centra en examinar el contenido de los archivos cookie.



Ilustración 3 Logo Parrot OS

2.2.3 Black Arch

Black Arch es una distribución basada en Arch Linux diseñada para pruebas de penetración e investigación de seguridad. Esta distribución está enfocada para usuarios con mayores conocimientos y experiencia en este ámbito, ya que es una de las distribuciones con más herramientas y mayor complejidad de uso [15].

Las características más importantes de esta distribución son [16]:

- Es una de las distribuciones con más herramientas, más de 2000, lo que conlleva la necesidad de que el usuario tenga mayor conocimiento para su uso.
- Tiene una organización única, ya que en comparación con las demás distribuciones, esta agrupa las herramientas por categorías o grupos.
- Otra de las características importantes es que si tenemos Arch Linux instalado podemos actualizarlo a Black Arch.
- Se realizan actualizaciones frecuentemente que permiten que el sistema operativo y las herramientas tengan las últimas versiones y características.
- Esta distribución está preparada para x64, ARMv6, ARMv7, Aarch64 y tiene imágenes listas para máquinas virtuales como VirtualBox o VMWare.

Algunos de los requisitos de hardware son [15]:

- Se necesita como mínimo 2 GB de RAM, recomendado 6 GB.
- Al menos 20 GB de espacio en disco duro, recomendado 40 GB.
- Requiere un procesador de 64 bits.
- Necesita un medio de arranque como un CD, DVD o una unidad flash USB.

Dispone de herramientas forenses como [17]:

- **Ddrescue:** Su objetivo es copiar datos de un fichero o un dispositivo a otro, intentando recuperar primero las partes que no están dañadas en caso de producirse errores de lectura.
- **Foremost:** Se utiliza para la recuperación de ficheros basado en sus cabeceras y estructura de datos.
- También tiene herramientas explicadas en anteriores distribuciones como **Galleta**, **Autopsy**, **Xplico** o **RegRipper**.



Ilustración 4 Logo Black Arch

2.2.4 CAINE

CAINE (Computer Aided Investigative Enviroment) es una distribución basada en Ubuntu creada como un proyecto de forense digital y actualmente dirigido por Nanni Bassetti (Italia). Esta distribución se centra en un entorno de investigación digital completo que está organizado para integrar las existentes herramientas software y proporcionar una interfaz gráfica amigable para el usuario [18].

Las características más importantes de esta distribución son [19]:

- Interfaz de usuario amigable.
- Fácil interoperabilidad durante todo el proceso de análisis digital (Preservación, Recolección, Análisis, Documentación)
- Una generación semiautomática de informes.
- Facilidad a la hora de la instalación. Por otro lado también se puede arrancar desde medios extraíbles como unidades flash o desde un disco.
- En modo Live, puede operar en unidades de almacenamiento de datos sin tener que iniciar el sistema operativo.

Algunos de los requisitos de hardware son [20]:

- Se necesita como mínimo 2 GB de RAM.
- Al menos 4 GB de espacio en disco duro.
- Una CPU como mínimo de doble núcleo de 2 GHz.

Contiene varias herramientas forenses [21], ya mencionadas anteriormente en otras distribuciones, como **Autopsy**, **Bulk-extractor** o **Galleta**.



Ilustración 5 Logo CAINE

2.2.5 Security Onion

Security Onion es una distribución Linux gratuita y de código abierto creada para detección de intrusos, monitoreo de seguridad de redes y administración de logs. Funciona como un sistema de gestión de eventos de seguridad (SIEM) y como un sistema de detección de intrusiones (IDS) en uno [22].

Las características más importantes de esta distribución son [23]:

- Sistema de gestión de eventos de seguridad (SIEM) y sistema de detección de intrusiones (IDS) todo en uno.
- Es una distribución de código abierto por lo tanto permite que exista una comunidad que contribuya a mejorar la distribución.
- Su interfaz es fácil de usar lo que permite que usuarios menos experimentados usen la distribución de manera efectiva. Respecto a esto la distribución viene con ajustes preconfigurados, lo que simplifica el proceso de configuración inicial.

Algunos de los requisitos mínimos de hardware son [24]:

- Se necesita como mínimo 12 GB de RAM.
- Al menos 200 GB de espacio en disco duro.
- Una CPU como mínimo de 4 núcleos.
- Sólo es compatible con la arquitectura x86-64 (procesadores estándar Intel o AMD de 64 bits).



Ilustración 6 Logo Security Onion

Las principales herramientas que se pueden utilizar en esta distribución son [25]:

- **NetworkMiner:** Esta herramienta de código abierto extrae archivos, imágenes, correo electrónicos y contraseñas del tráfico de red capturado en archivos PCAP,

también se utiliza para capturar el tráfico de la red en vivo mediante la detección de una interfaz de red.

- **Wireshark:** Es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones. Se basado en las librerías PCAP que permite filtrar según el protocolo para analizar los datos capturados de una red.
- **Zeek:** Es un framework de código abierto utilizado para analizar y encontrar amenazas de forma activa en una red. Sus herramientas de monitorización permiten que se use como un NIDS (Network-based Intrusion Detection System). Su funcionamiento se basa en filtrar el tráfico de una red y generar logs que describan lo que ocurre en ella.
- **Suricata:** Es una herramienta de código abierto que sirve para monitorizar el tráfico de red y capaz de detectar comportamientos maliciosos. Está basado en el análisis del tráfico de red según unas reglas predeterminadas diseñadas para la identificación de malware y actividades maliciosas.

2.2.6 Comparativa entre las distintas distribuciones

A continuación se muestra una tabla comparativa de las características hardware de las distribuciones que se han descrito anteriormente.

	Kali Linux	Parrot OS	BlackArch	CAINE	Security Onion
Distribución base	Debian	Debian	Arch Linux	Ubuntu	Oracle Linux 9
RAM	1 GB	320 MB	256 MB	2 GB	12 GB
Espacio en disco	20 GB	16 GB	2 GB	4 GB	200 GB
CPU	32y 64 bits	32 y 64 bits	32 y 64 bits	64 bits	64 bits

Tabla 1 Características hardware de las distribuciones

Tal como se ve en la tabla anterior Security Onion es la que más recursos necesita para funcionar.

También se añade una tabla comparativa de algunas de las herramientas de forensía digital que contienen las distribuciones.

	Kali Linux	Parrot OS	BlackArch	CAINE	Security Onion
RegRipper	✓	✓	✓	✓	✗
Binwalk	✓	✓	✓	✗	✗
Xplico	✓	✓	✓	✗	✗
Bulk-extractor	✓	✗	✓	✓	✗
Autopsy	✓	✓	✓	✓	✗
Galleta	✓	✓	✓	✓	✗
Ddrescue	✓	✓	✓	✓	✗
Foremost	✓	✓	✓	✓	✗
NetworkMiner	✗	✗	✓	✗	✓
Wireshark	✓	✓	✗	✓	✓
Zeek	✗	✗	✗	✗	✓
Suricata	✗	✗	✓	✗	✓

Tabla 2 Comparativa de las herramientas de las distribuciones

En la tabla comparativa anterior vemos que la mayoría de las distribuciones anteriormente examinadas disponen de diferentes herramientas de forensía digital. Security Onion es la que menos ámbitos abarca y se centra más en herramientas que tengan que ver con lo sucedido en el tráfico de red.

2.3 Virtualización

En este apartado se va a describir que es la virtualización y sus ventajas.

La virtualización es una tecnología que se usa para crear representaciones virtuales de servidores, almacenamiento, redes y otras máquinas físicas [26].

Algunas de las ventajas de la virtualización son:

- Reducción de la inversión en capital y los gastos operativos.
- Reducción o eliminación del tiempo de inactividad.
- Distribución más rápida de las aplicaciones y recursos.
- Una gestión más simplificada.
- Mejor recuperación ante desastres.
- Menor consumo energético.

Como nos hablan en el siguiente artículo [27], existen dos formas de empaquetar estas representaciones virtuales que tienen varios elementos y se aíslan del resto del sistema que son los contenedores y las máquinas virtuales. Su principal diferencia está en cuales de esas partes se aíslan, lo que conlleva que tengan una diferente portabilidad.

Por un lado los contenedores son paquetes de software que incluyen todo lo necesario para que una aplicación funcione sin depender de un hipervisor. Son ágiles y rápidos, ideales para aplicaciones modernas donde varios contenedores trabajan juntos. Son fáciles de mover entre diferentes entornos, permitiendo una gran portabilidad.

Docker es una plataforma open source clave para crear, implementar y administrar aplicaciones en contenedores.

Por otro lado, las máquinas virtuales han jugado un rol clave en el cloud computing al permitir la ejecución de sistemas operativos de manera independiente en un solo servidor. Con la ayuda de un hipervisor, que actúa como intermediario, se optimiza el uso de recursos y se brinda flexibilidad. Aunque surgieron en los años 2000 por motivos de consolidación y eficiencia, su utilidad se ha expandido para adaptarse a diversas necesidades empresariales, como la gestión dinámica de recursos y la optimización de recursos costosos como las GPU.

Además, al contar con su propio sistema operativo, las máquinas virtuales pueden realizar una variedad de tareas simultáneas, desde la gestión de bases de datos hasta la simulación de sistemas completos.

Algunos de los softwares más usados para la virtualización de estas máquinas virtuales son VMware, VirtualBox, QEMU o Hyper-V.

3. Especificaciones y restricciones de diseño

3.1 Especificaciones

Como se ha podido ver en la tabla comparativa de herramientas de las distintas distribuciones en el apartado 2.2.6, Security Onion se centra en utilizar dichas herramientas para ver lo sucedido en el tráfico de red y no dispone de otras que se centren en recuperación o extracción de datos.

Dado que el objetivo de este proyecto es el estudio de la capacidad forense digital de la suite Security Onion, se buscará realizar la simulación de diferentes casos posibles que pudiera encontrar un analista de este ámbito para ver que potencial tiene la suite.

El proyecto tendrá las siguientes especificaciones:

- Se usará el diseño de una estructura de red donde existan un atacante y múltiples máquinas víctimas.
- Se utilizará un entorno virtualizado de máquinas.
- Será independiente del Sistema operativo de las máquinas víctimas.
- Se utilizará una distribución que contenga herramientas de pentesting como máquina atacante.

3.2 Restricciones de diseño

De esas especificaciones se ha restringido el diseño del proyecto a las siguientes:

- El proyecto se realizará mediante software Open Source.
- Se utilizará el software de virtualización VirtualBox debido a una mejor instalación de la máquina Metasploitable3.
- Es necesario que la máquina que contiene la distribución Security Onion esté en la misma subred que las demás máquinas virtuales, atacante y víctima.
- Se utilizará VirtualBox con la versión 7.0.18 r162988 (Qt5.15.2) para la virtualización de las máquinas.
- La estructura de red contendrá tres máquinas virtuales, una atacante, otra víctima y la otra la máquina que contendrá Security Onion.
- Se utilizará la distribución Kali Linux como máquina atacante con la versión 2023.4.
- Security Onion sólo soporta arquitectura x86-64 (procesadores estándar Intel o AMD de 64 bits).
- Se va a utilizar la imagen ISO de Security Onion en la versión 2.4.10-20230821 ISO.
- Utilización de máquina víctima Metasploitable3 con Ubuntu 14.04.

4. Descripción de la solución propuesta

Proponemos un diseño de una pequeña red en la que estarán interconectados tres equipos virtualizados. Esta red estará formada por una máquina víctima que será Metasploitable3, que contiene diferentes vulnerabilidades, basada en Ubuntu 14.04, otra máquina virtual atacante que contendrá la suite de Kali Linux y una más en la que contenga la propia suite que vamos a estudiar, Security Onion.

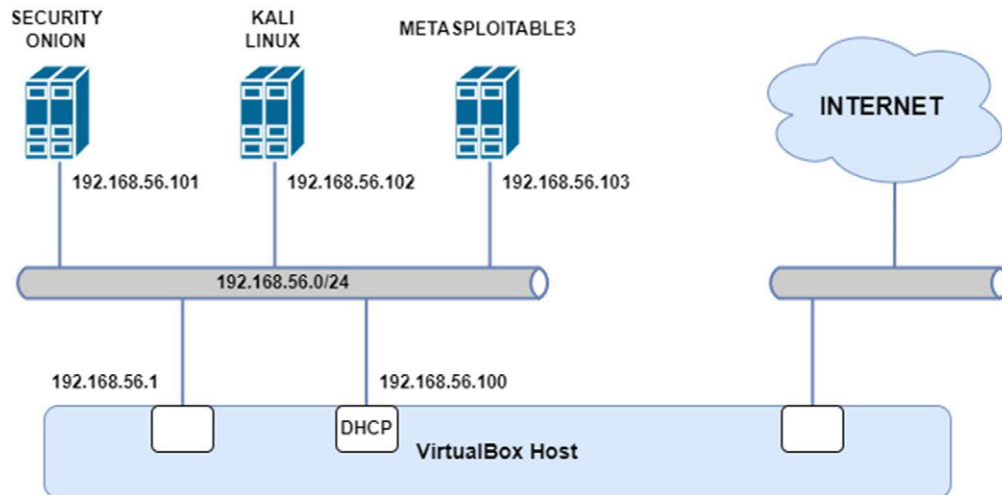


Ilustración 7 Diagrama lógico de la infraestructura

Hemos decidido utilizar el modo de configuración Standalone de Security Onion que es el recomendado para estos casos como explicamos en el Anexo I.

Dada la necesidad de la interconexión de las máquinas virtualizadas utilizaremos el modo del adaptador de red Host-Only de VirtualBox explicado en el Anexo II. Cuando sea necesario, cambiaremos a modo Bridge si se necesitara conexión a Internet para alguna actualización o descarga.

Utilizaremos Kali Linux debido a la necesidad de herramientas de pentesting para realizar la simulación de los casos a estudiar con la distribución Security Onion.

La descripción de la instalación de todas las máquinas virtualizadas queda descrita en el Anexo III.

Vamos a plantear casos prácticos de diferentes tipos de forensía digital. Esto nos permitirá demostrar que capacidad de análisis forense digital tiene la herramienta Security Onion.

Los casos que hemos decidido plantear son los siguientes.

El primer caso que vamos a simular se basa en el robo de archivos, buscaremos desde la máquina atacante realizar un escaneo de puertos ver cuál de ellos es vulnerable. Posteriormente realizaremos un ataque por fuerza bruta en uno de los servicios que usen alguno de los puertos vulnerables. Tras el ataque satisfactorio procederemos a robar alguno

de los archivos que haya en la máquina víctima. Con esto se busca que Security Onion pueda identificar información del atacante, de la víctima y qué archivos pudieran haberse extraído.

En el segundo caso utilizaremos un archivo PCAP que contiene los datos obtenidos del tráfico de red en el que un atacante ha introducido en un servidor un malware. Buscamos encontrar información del atacante, cómo y cuándo realizó el ataque, información de la víctima y qué tipo de malware es.

En el tercero es un caso de esteganografía dadas tres imágenes, la original, la que se va a ocultar y la imagen que contiene información oculta. Haremos una simulación parecida a la del robo de archivos pero buscando encontrar indicios e información de los archivos que extraen y luego se vuelven a colocar pero con información oculta. Esperamos que Security Onion nos ofrezca información sobre estos archivos.

Y en el último caso vamos a recuperar archivos borrados de un dispositivo de almacenamiento USB. Buscamos saber si Security Onion tiene alguna herramienta para poder realizar esta tarea.

5. Resultados

En este apartado vamos a simular ciertos casos prácticos que podría encontrarse un analista y ver la capacidad de Security Onion frente a ellos. Los casos prácticos planteados son :

- Robo de archivos.
- Malware.
- Análisis de archivos.
- Recuperación de datos.

5.1 Robo de archivos

La simulación que hemos creado se basa en el ataque por fuerza bruta a un servidor y el robo de datos al mismo. Por tanto, usaremos la máquina virtual con IP **192.168.56.106** con Kali Linux como atacante y la máquina virtual con IP **192.168.56.103** con Metasploitable3 como víctima.

Para comenzar vamos a realizar un escaneo de los puertos que puedan estar abiertos en la máquina Metasploitable3. Para ello usaremos la suite Kali Linux con la herramienta **nmap** con el siguiente comando:

nmap -v -sV 192.168.56.103

```
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.5
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux;
protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.7
445/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp    open  ipp          CUPS 1.7
3000/tcp   closed ppp
3306/tcp   open  mysql        MySQL (unauthorized)
8080/tcp   open  http         Jetty 8.1.7.v20120910
8181/tcp   closed intermapper
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE
: cpe:/o:linux:linux_kernel
```

Ilustración 8 Escaneo de puertos

Como vemos en la imagen los puertos abiertos son **21 (FTP)**, **22 (SSH)**, **80 (HTTP)**, **445 (NetBIOS)**, **631 (IPP)**, **3306 (MySQL)** y **8080 (HTTP)**. Vamos a utilizar el puerto 21 que tiene el servicio FTP para hacer el ataque por fuerza bruta.

Para realizar dicho ataque al servicio FTP, hemos creado dos archivos con usuarios y contraseñas aleatorios, en el cuál una pareja de ellos es el correcto para simular que el ataque ha sido satisfactorio. Usamos la herramienta **hydra** que tiene Kali Linux para iniciar el ataque con el siguiente comando:

hydra -L users -P passwords ftp://192.168.56.103

```
(kali㉿kali)-[~/Desktop]
$ hydra -L users -P passwords ftp://192.168.56.103
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-02-22 05:
18:57
[DATA] max 16 tasks per 1 server, overall 16 tasks, 188 login tries (l:4/p:47
), ~12 tries per task
[DATA] attacking ftp://192.168.56.103:21/
[21][ftp] host: 192.168.56.103 login: vagrant password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-02-22 05:
19:03
```

Ilustración 9 Ataque por fuerza bruta

Observamos que tras el ataque conseguimos descubrir un usuario **vagrant** y su contraseña **vagrant** que podremos usar para acceder al servidor FTP.

```
(kali㉿kali)-[~/Desktop]
$ ftp 192.168.56.103
Connected to 192.168.56.103.
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [192.168.56.103]
Name (192.168.56.103:kali): vagrant
331 Password required for vagrant
Password:
230 User vagrant logged in
```

Ilustración 10 Conexión con credenciales obtenidas tras ataque

Utilizamos las credenciales obtenidas para conectarnos al servicio FTP y una vez que entremos vemos que archivos hay.

```
ftp> ls
200 EPRT command successful
150 Opening ASCII mode data connection for file list
-rw-r--r-- 1 vagrant vagrant 86562816 Oct 29 2020 VBoxGuestAdditions.iso
drwxrwxr-x 2 vagrant vagrant 4096 Feb 20 22:53 backup
-rw-rw-r-- 1 vagrant vagrant 30 Feb 21 22:11 datos.txt
```

Ilustración 11 Archivos en la máquina víctima

Elegimos el archivo **datos.txt** para obtenerlo de forma fraudulenta que podría ser un archivo que contenga datos importantes.

```
ftp> get datos.txt
local: datos.txt remote: datos.txt
200 EPRT command successful
150 Opening BINARY mode data connection for datos.txt (30 bytes)
100% |*****| 30 171.32 KiB/s 00:00 ETA
226 Transfer complete
30 bytes received in 00:00 (9.39 KiB/s)
```

Ilustración 12 Robo de archivos

Como vemos en las imágenes descargamos el archivo y vemos su contenido.



Ilustración 13 Datos obtenidos en la máquina atacante

Una vez descrita la simulación del robo del archivo vamos a realizar el estudio de la capacidad de análisis forense que ofrece la herramienta Security Onion.

En la pestaña Alertas en los eventos que se han generado ya se nos muestra varias evidencias con gravedad alta relacionadas con el escaneo de puertos, el uso de Kali Linux y el ataque por fuerza bruta al servicio FTP.

rule.name	event.module	event.severity_label
ET POLICY Possible Kali Linux hostname in DHCP Request Packet	suricata	high
ET SCAN Possible Nmap User-Agent Observed	suricata	high
ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	suricata	high
ET SCAN Potential FTP Brute-Force attempt response	suricata	high

Ilustración 14 Eventos generados en el robo de archivos

Dado que hay varios eventos con gravedad alta vamos a comenzar por el evento de escaneo de puertos y ver qué información nos ofrece la herramienta.

ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)

rule.name	event.severity_label	source.ip	source.port	destination.ip	destination.port
ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	high	192.168.56.106	49884	192.168.56.103	80
ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	high	192.168.56.106	49894	192.168.56.103	80
ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	high	192.168.56.106	38350	192.168.56.103	631
ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	high	192.168.56.106	38358	192.168.56.103	631

Ilustración 15 Evento de escaneo de puertos

En la imagen anterior se muestra cierta información con la cual podemos deducir que la IP del atacante es la **192.168.56.106**, que es la que marca el evento como IP origen, la IP de la víctima que es la que se marca como IP destino **192.168.56.103** y a los puertos que se están atacando marcado por puerto destino como el **80** o el **631**.

Vamos a revisar los eventos que han sucedido filtrándolos y agrupándolos con IP origen **192.168.56.106** y puerto destino para confirmar si se trata de un escaneo de puertos. Nos iremos a la pestaña Hunt y aplicaremos el siguiente filtro:

**** AND source.ip: "192.168.56.106" | groupby "source.ip" "destination.port"***

192.168.56.106	20
192.168.56.106	21
192.168.56.106	22
192.168.56.106	23
192.168.56.106	24
192.168.56.106	25
192.168.56.106	26

Ilustración 16 Escaneo de puertos

En la ilustración anterior vemos como desde la IP **192.168.56.106** hay eventos hacia puertos de la víctima de manera secuencial lo que implica un escaneo de puertos. Vamos a buscar que tipo de conexión han tenido para averiguar que puertos abiertos ha encontrado el atacante. Para ello usamos el siguiente filtro:

*** AND source.ip: "192.168.56.106" | groupby "source.ip" "connection.state_description"**

source.ip	connection.state_description
192.168.56.106	Connection attempt rejected
192.168.56.106	Established, responder aborted
192.168.56.106	Connection established, originator aborted (sent a RST)
192.168.56.106	Normal SYN/FIN completion
192.168.56.106	Connection attempt seen, no reply

Ilustración 17 Conexiones establecidas

Para ver los puertos abiertos vamos a incluir en el filtro las conexiones que se han establecido pero el origen las ha abortado. Para ello usamos:

*** AND source.ip: "192.168.56.106" AND connection.state_description: "Connection established, originator aborted (sent a RST)" | groupby "source.ip" "connection.state_description" "destination.port"**

source.ip	connection.state_description	destination.port
192.168.56.106	Connection established, originator aborted (sent a RST)	21
192.168.56.106	Connection established, originator aborted (sent a RST)	22
192.168.56.106	Connection established, originator aborted (sent a RST)	80
192.168.56.106	Connection established, originator aborted (sent a RST)	445
192.168.56.106	Connection established, originator aborted (sent a RST)	631
192.168.56.106	Connection established, originator aborted (sent a RST)	3306
192.168.56.106	Connection established, originator aborted (sent a RST)	8080

Ilustración 18 Puertos abiertos

El resultado es que los puertos que da el filtro anterior son **21, 22, 80, 445, 631, 3306 y 8080**. Esta información sería la que habría conseguido el atacante.

Ahora vamos a revisar los eventos generados en los que indica que hay potencialmente un ataque por fuerza bruta al servicio FTP.

ET SCAN Potential FTP Brute-Force attempt response

source.ip	source.port	destination.ip	destination.port	rule.gid	rule.uuid	rule.category
192.168.56.103	21	192.168.56.106	44306	1	2002383	Unsuccessful User Privilege Gain
192.168.56.103	21	192.168.56.106	44354	1	2002383	Unsuccessful User Privilege Gain
192.168.56.103	21	192.168.56.106	44312	1	2002383	Unsuccessful User Privilege Gain

Ilustración 19 Eventos generados por ataque al servicio FTP

Los eventos generados en la herramienta informan de la respuesta que da la víctima al atacante por lo que podemos sacar que el origen, al ser una respuesta, es la víctima con IP **192.168.56.103** desde el puerto **21**, que es el que está siendo atacado, hacia la IP destino **192.168.56.106** que es la IP del atacante. En este caso la imagen muestra que no se ha conseguido realizar el acceso tanto como en la ilustración siguiente que muestra los usuarios y contraseñas que se han probado.

```

USER user
331 Password required for user
PASS 1234567
530 Login incorrect.
USER user
331 Password required for user
PASS buster
530 Login incorrect.
```

Ilustración 20 Acceso incorrecto al servicio FTP

En este punto deberíamos buscar si el atacante ha conseguido acceder al servicio FTP para ello podemos utilizar el siguiente filtro en la pestaña Hunt:

tags:ftp | groupby ftp.user ftp.argument

ftp.user	ftp.argument
vagrant	[1 192.168.56.106 49123
vagrant	[1 192.168.56.106 45493
vagrant	[1 192.168.56.106 40387
vagrant	ftp://192.168.56.103/./datos.txt

Ilustración 21 Usuarios con interacción con el servicio FTP

El usuario **vagrant** ha tenido acceso al servicio FTP y al archivo **datos.txt**. Viendo estos indicios nos vamos a revisar los eventos que se generan para tener información de cuándo y desde donde se producen estas conexiones con el servicio FTP.

Timestamp ▲	source.ip	source.port	destination.ip	destination.port	ftp.user	ftp.command	ftp.argument
2024-02-22 10:35:03.265 +00:00	192.168.56.106	42296	192.168.56.103	21	vagrant	EPRT	1 192.168.56.106 49123
2024-02-22 10:35:14.523 +00:00	192.168.56.106	42296	192.168.56.103	21	vagrant	EPRT	1 192.168.56.106 40387
2024-02-22 10:35:15.459 +00:00	192.168.56.106	42296	192.168.56.103	21	vagrant	EPRT	1 192.168.56.106 45493
2024-02-22 10:35:15.460 +00:00	192.168.56.106	42296	192.168.56.103	21	vagrant	RETR	ftp://192.168.56.103/./datos.txt

Ilustración 22 Interacciones del usuario vagrant

A las 10:35:03 del 22-02-22 comienza dichos eventos desde la IP del atacante **192.168.56.106**. Nos vamos a la pestaña PCAP para poder ver que ha sucedido.

```
USER vagrant
331 Password required for vagrant
PASS vagrant
230 User vagrant logged in
```

Ilustración 23 Acceso del usuario vagrant

Como muestra la imagen anterior vemos un acceso del usuario **vagrant** con la contraseña **vagrant**, por tanto, deducimos que el atacante ha conseguido obtener las credenciales de acceso. A las 10:35:15 del 22-02-22 extrae el archivo **datos.txt**.

```
RETR datos.txt
150 Opening BINARY mode data connection for datos.txt (30 bytes)
226 Transfer complete
```

Ilustración 24 Obtención del archivo datos.txt

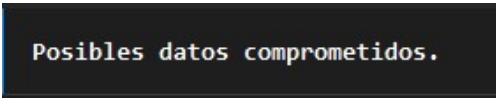
Sabiendo la hora podemos buscar los datos del servicio FTP y ver si está en texto plano para poder ver su contenido mediante el siguiente filtro:

tags:file AND file.source:"FTP_DATA" | groupby file.source source.ip

Timestamp ▼	source.ip	destination.ip	file.name	file.mime_type	file.source
2024-02-22 10:35:15.463 +00:00	192.168.56.103	192.168.56.106		text/plain	FTP_DATA
2024-02-22 10:35:14.529 +00:00	192.168.56.103	192.168.56.106		text/plain	FTP_DATA
2024-02-22 10:35:03.265 +00:00	192.168.56.103	192.168.56.106		text/plain	FTP_DATA

Ilustración 25 Filtro FTP DATA

Encontramos un evento con la misma marca de tiempo vamos a la pestaña PCAP y al ser en texto plano podemos ver su contenido. Sabiendo que el archivo que se ha extraído de manera fraudulenta contenía lo siguiente:



Posibles datos comprometidos.

Ilustración 26 Contenido archivo datos.txt

En resumen, tras toda esta búsqueda de evidencias y de análisis forense digital se ha obtenido que la IP del atacante es la **192.168.56.106**, la IP de la víctima es la **192.168.56.103**, los puertos abiertos en la máquina víctima son **21, 22, 80, 445, 631, 3306 y 8080**, que se ha realizado un ataque por fuerza bruta al servicio FTP en el puerto 21, el usuario utilizado para acceso al servicio FTP es **vagrant** y la contraseña para este usuario **vagrant**, el fichero extraído ha sido **datos.txt** y la fecha **10:35:15 del 22-02-22**, además del contenido del mismo.

Teniendo en cuenta este resumen, comparamos las evidencias mostradas con los que se pretendían simular y la herramienta ha sido capaz de mostrar toda la información necesaria para saber que se ha producido la extracción de un archivo de manera fraudulenta, sabiendo la fecha, el modo de obtención, con qué usuario y el archivo que se ha extraído.

5.2 Malware

Dado que analizar un caso con cualquier tipo de malware en las máquinas virtuales que tenemos en el entorno supondría un riesgo, se ha descargado un archivo¹ con el tráfico de datos que debería analizar un forense digital en un caso en el que han atacado un servidor de una empresa.

Con estos datos se va a analizar qué capacidad tiene la herramienta Security Onion para obtener la información útil para demostrar qué pasó.

Para ello en primer lugar vamos a realizar la importación del archivo, pasamos el archivo con los datos a la máquina virtual que contiene Security Onion.

Security Onion dispone de dos formas de importar el archivo, mediante consola o por vía web, siendo por consola más estable y seguro.

¹ Archivo descargado de <https://app.hackthebox.com>, esta página web se utiliza para realizar aprendizajes de ciberseguridad.

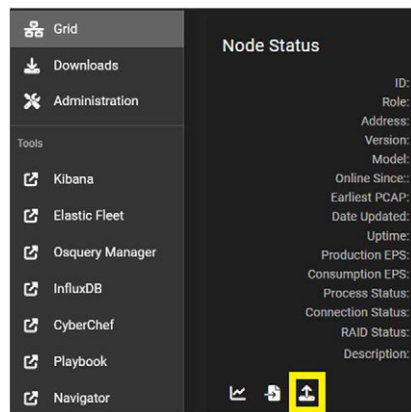


Ilustración 27 Importar archivo vía web

Dado el tamaño del archivo lo importaremos mediante consola mediante el comando **so-import-pcap**.

```
[so@securityonion ~]$ ll
total 270580
-rw-r--r--. 1 so so 277065872 Jul  5  2023 Capture.pcap
drwxr-xr-x. 10 so so      4096 Nov 20 22:21 SecurityOnion
[so@securityonion ~]$ sudo so-import-pcap Capture.pcap
[sudo] password for so:
Processing Import: /home/so/Capture.pcap
- verifying file
```

Ilustración 28 Inicio de importación

Una vez se completa la importación aparece una URL que nos permite abrir el dashboard de Security Onion con los datos importados.

```
Import complete!

Use the following hyperlink to view the imported data. Triple-click to quickly highlight the entire
hyperlink and then copy it into a browser:
https://192.168.56.101/#/dashboards?q=import.id:85d7dcbc8764a8422aelflc3e069ff87%20%7C%20groupby%20
-sankey%20event.dataset%20event.category%2a%20%7C%20groupby%20-pie%20event.category%20%7C%20groupby
%20-bar%20event.module%20%7C%20groupby%20event.dataset%20%7C%20groupby%20event.module%20%7C%20group
by%20event.category%20%7C%20groupby%20observer.name%20%7C%20groupby%20source.ip%20%7C%20groupby%20d
estination.ip%20%7C%20groupby%20destination.port&t=2023%2F03%2F20%2000%3A00%3A00%20AM%20-%202023%2F
03%2F22%2000%3A00%3A00%20AM&z=UTC

or, manually set the Time Range to be (in UTC):
From: 2023-03-20    To: 2023-03-22

Note: It can take 30 seconds or more for events to appear in Security Onion Console.
```

Ilustración 29 Importación completada

Como se muestra en la ilustración anterior se nos indica el rango de tiempo que debemos utilizar para ver los datos, en este caso de 2023-03-20 a 2023-03-22.

Cambiamos el rango de tiempo en el dashboard para ver los eventos que muestra la herramienta.

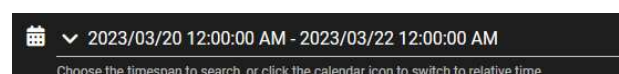


Ilustración 30 Rango de tiempo

Como vamos a ver en la siguiente ilustración podemos ver los eventos que han ocurrido en ese rango de tiempo, en los que podemos ver ataques como el escaneo de SSH o intentos de acceso por fuerza bruta a FTP.

	Count	rule.name
 	1126	ET DROP Dshield Block Listed Source group 1
 	298	ET SCAN Non-Allowed Host Tried to Connect to MySQL Server
 	212	ET SCAN Potential FTP Brute-Force attempt response
 	135	ET SCAN Potential SSH Scan
 	58	ET CINS Active Threat Intelligence Poor Reputation IP group 94
 	56	ET CINS Active Threat Intelligence Poor Reputation IP group 89
 	52	ET SCAN Suspicious inbound to mySQL port 3306
 	46	ET SCAN Multiple FTP Root Login Attempts from Single Source - Possible Brute Force Attempt
 	46	ET DROP Spamhaus DROP Listed Traffic Inbound group 3

Ilustración 31 Eventos ocurridos

Una vez tenemos esta información vamos a utilizar Security Onion para responder preguntas como por ejemplo cuando atacaron el servidor, que puertos encontraron abiertos, etc.

Para empezar vamos a buscar cuales son las IPs atacantes y atacadas para ello buscamos los eventos en los que la herramienta los marca con peligrosidad alta, en nuestro caso vemos un evento en el que hay muchas respuestas de intentos de acceso al servicio FTP.

 	212	ET SCAN Potential FTP Brute-Force attempt response
---	-----	--

Ilustración 32 Intento de acceso al servicio FTP

Vemos los eventos que se han categorizado con **ET SCAN Potential FTP Brute-Force attempt response**. Como vemos en la imagen anterior hay más de 200 intentos, por tanto vamos a agruparlos por IP destino, ya que serían las IPs a las que estaría respondiendo el atacado y así de paso también averiguamos que IP es la que están atacando.

Para ello realizamos el filtro para agrupar los eventos por IP origen, que en este caso sería la IP del atacado, ya que, el evento es una respuesta de intento de acceso, vamos a la pestaña Hunt con el siguiente comando:

"ET SCAN Potential FTP Brute-Force attempt response" | groupby "source.ip"

Count	  	source.ip
212		172.31.39.46

Ilustración 33 Filtro de eventos por IP origen

Como vemos sólo aparece una IP, por tanto, deducimos que la IP del atacado será la **172.31.39.46**. Ahora se va a añadir al filtro un agrupamiento por IPs destino para ver la IP de los posibles atacantes con el siguiente comando:

**"ET SCAN Potential FTP Brute-Force attempt response" | groupby "source.ip"
"destination.ip"**

Count	source.ip	destination.ip
28	172.31.39.46	3.109.209.43
184	172.31.39.46	120.78.199.189

Ilustración 34 Filtro de posibles IPs de atacantes

En esta última imagen vemos las posibles IPs de los atacantes que son la **3.109.209.43** y **120.78.199.189**. Vamos a investigar primero **120.78.199.189** que tiene más eventos. Agrupamos para que nos muestre todos los eventos con IP origen **120.78.199.189** de la siguiente manera:

*** AND source.ip:"120.78.199.189" | groupby "source.ip"**

Count	source.ip
1001	120.78.199.189

Ilustración 35 Eventos con IP 120.78.199.189

Vamos a agrupar los eventos por puerto destino para ver si desde esta IP se ha producido un escaneo de puertos.

*** AND source.ip: "120.78.199.189" | groupby "source.ip" "destination.port"**

Count	source.ip	destination.port
1000	120.78.199.189	21
1	120.78.199.189	0

Ilustración 36 Eventos agrupados por puerto destino

Como observamos desde esta IP sólo se centran en el puerto 21 por lo que no hay indicios de escaneo de puertos.

Ahora vamos a realizar el mismo proceso con la IP **3.109.209.43**.

*** AND source.ip: "3.109.209.43" | groupby "source.ip" "destination.port" "@timestamp"**

3.109.209.43	0	2023-03-21T10:42:26.644Z
3.109.209.43	1	2023-03-21T10:42:23.708Z
3.109.209.43	2	2023-03-21T10:42:23.709Z
3.109.209.43	3	2023-03-21T10:42:23.709Z
3.109.209.43	4	2023-03-21T10:42:23.709Z

Ilustración 37 Eventos generados por la IP 3.109.209.43

Como se ve en la imagen hay eventos con puerto destino de manera secuencial lo que lleva a pensar que desde la IP **3.109.209.43** se ha realizado un escaneo de puertos en la fecha **21-03-2023 a las 10:42:25**, por tanto deducimos que el ataque podría realizarse desde esta IP.

Una vez hemos averiguado la IP del atacante, vamos a buscar que puertos estaban abiertos, para ello iremos a la pestaña Hunt en el dashboard de Security Onion e iremos añadiendo filtros para agrupar los eventos. Para esta búsqueda usaremos la IP origen, la IP destino y la descripción del estado de la conexión.

**** AND source.ip: "3.109.209.43" | groupby "source.ip" "destination.ip" "connection.state_description"***

source.ip	destination.ip	connection.state_description
3.109.209.43	172.31.39.46	Established, responder aborted
3.109.209.43	172.31.39.46	Responder sent a SYN ACK followed by a RST, we never saw a SYN from the (purported) originator
3.109.209.43	172.31.39.46	No SYN seen, just midstream traffic (a 'partial connection' that was not later closed)
3.109.209.43	172.31.39.46	Connection established, originator aborted (sent a RST)
3.109.209.43	172.31.39.46	Normal SYN/FIN completion
3.109.209.43	172.31.39.46	Connection attempt rejected

Ilustración 38 Búsqueda de puertos abiertos

Para el estado de conexión buscaremos el valor en el que aparezca que se ha establecido la conexión pero el origen la ha cancelado y lo agruparemos por puerto destino quedando el filtro de la siguiente manera:

**** AND source.ip: "3.109.209.43" AND connection.state_description: "Connection established, originator aborted (sent a RST)" | groupby "source.ip" "destination.ip" "destination.port" "connection.state_description"***

Count	source.ip	destination.ip	destination.port	connection.state_description
2	3.109.209.43	172.31.39.46	21	Connection established, originator aborted (sent a RST)
2	3.109.209.43	172.31.39.46	22	Connection established, originator aborted (sent a RST)
1	3.109.209.43	172.31.39.46	3306	Connection established, originator aborted (sent a RST)
1	3.109.209.43	172.31.39.46	6379	Connection established, originator aborted (sent a RST)
1	3.109.209.43	172.31.39.46	8086	Connection established, originator aborted (sent a RST)
1	3.109.209.43	172.31.39.46	24456	Connection established, originator aborted (sent a RST)

Ilustración 39 Puertos abiertos

Una vez se muestra el resultado podemos ver que los puertos abiertos son el **21,22,3306,6379,8086 y 24456**.

Una vez tenemos esta información podemos centrarnos en cuando empezó a producirse el ataque al servicio FTP por parte de la IP **3.109.209.43**, para ello usamos:

**** AND source.ip: "3.109.209.43" AND destination.port:"21" | groupby "source.ip" "destination.ip" "destination.port"***

2023-03-21 10:49:43.923 +00:00	3.109.209.43	57392	172.31.39.46	21	tcp	ftp
2023-03-21 10:49:43.923 +00:00	3.109.209.43	57380	172.31.39.46	21	tcp	ftp
2023-03-21 10:49:43.923 +00:00	3.109.209.43	57366	172.31.39.46	21	tcp	ftp
2023-03-21 10:49:43.923 +00:00	3.109.209.43	57414	172.31.39.46	21	tcp	ftp

Ilustración 40 Comienzo del ataque

Este filtrado nos muestra, ordenándolo por la marca de tiempo, que el ataque comenzó el día 21-03-2023 a las 10:49:43 UTC. Abriendo los eventos generados con la pestaña PCAP podemos ver que usuarios y contraseñas se están utilizando.

USER alonzo.spire
331 Please specify the password.
PASS Winter2016!
530 Login incorrect.
USER alonzo.spire
331 Please specify the password.
PASS Summer2021!
530 Login incorrect.
USER tony.shephard
331 Please specify the password.
PASS Winter2017!
530 Login incorrect.

Ilustración 41 Usuarios y contraseñas usados para el ataque

Teniendo en cuenta la fecha del ataque vamos a buscar si en ese rango de tiempo ha habido alguna transferencia de archivos y con qué usuario y contraseña. Para ello usaremos el siguiente comando:

tags:ftp | groupby ftp.user ftp.argument

ftp.user 	ftp.argument 
tony.shephard	ftp://172.31.39.46/./fetch.sh
tony.shephard	ftp://172.31.39.46/./backup
abdullah.yasin	ftp://172.31.39.46/etc/shadow
abdullah.yasin	ftp://172.31.39.46/etc/passwd
abdullah.yasin	ftp://172.31.39.46/bin/whoami

Ilustración 42 Usuarios FTP

Una vez que hemos visto que con el filtro anterior hay usuarios, nos vamos a los eventos producidos y los ordenamos por el tiempo en el que han sucedido para ver cuando han conseguido el acceso.

Timestamp ▲	source.ip	source.port	destination.ip	destination.port	ftp.user	ftp.command	ftp.argument
2023-03-21 10:51:11.903 +00:00	3.109.209.43	44880	172.31.39.46	21	tony.shephard	EPSV	
2023-03-21 10:52:01.652 +00:00	3.109.209.43	44880	172.31.39.46	21	tony.shephard	EPSV	
2023-03-21 10:52:03.445 +00:00	3.109.209.43	44880	172.31.39.46	21	tony.shephard	EPSV	
2023-03-21 10:52:03.447 +00:00	3.109.209.43	44880	172.31.39.46	21	tony.shephard	RETR	ftp://172.31.39.46/.backup

Ilustración 43 Eventos FTP con IP origen la del atacante

```

220 (vsFTPD 3.0.5)
USER tony.shephard
331 Please specify the password.
PASS Summer2023!
230 Login successful.

```

Ilustración 44 Acceso confirmado

Como se muestra en las imágenes anteriores vemos que desde la IP atacante se ha conseguido acceder al servicio FTP de la víctima con el usuario **tony.shephard** y la contraseña **Summer2023!**. También se observa que ha tenido acceso a varios archivos.

2023-03-21 10:52:03.447 +00:00	3.109.209.43	44880	172.31.39.46	21	tony.shephard	RETR	ftp://172.31.39.46/.backup
2023-03-21 10:55:13.286 +00:00	3.109.209.43	44880	172.31.39.46	21	tony.shephard	EPSV	
2023-03-21 10:55:13.287 +00:00	3.109.209.43	44880	172.31.39.46	21	tony.shephard	RETR	ftp://172.31.39.46/.fetch.sh

Ilustración 45 Archivos a los que se han tenido acceso

Teniendo en cuenta las fechas vamos a averiguar a qué datos ha accedido, a las 10:52:03 al archivo .backup y a las 10:55:13 al archivo .fetch.sh. Mediante el siguiente comando podremos ver los datos de esos archivos viendo los eventos en esos rangos de tiempo.

tags:file AND file.source:"FTP_DATA" | groupby file.source source.ip

2023-03-21 10:52:03.447 +00:00	172.31.39.46	3.109.209.43	text/ini
2023-03-21 10:55:13.288 +00:00	172.31.39.46	3.109.209.43	text/x-shellscript

Ilustración 46 Datos en ese rango de tiempo

La herramienta Security Onion nos permite que una vez localizados los eventos anteriores abrirlos con la pestaña PCAP y poder ver que contenían dichos archivos.

```
[options]
.UseSyslog

[FTP-INTERNAL]
.sequence      = 29999,50234,45087
.seq_timeout   = 5
.command       = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 24456 -j ACCEPT
.tcpflags      = syn

# Creds for the other backup server abdullah.yasin:XhIhGame_90HJLDASxfd&hoooad
```

Ilustración 47 Archivo .backup

Al observar este archivo parece un archivo de configuración para el acceso a un servidor FTP interno, en el que si alguien usa los puertos 29999, 50234, 45087 de manera secuencial se añade al iptables que desde el puerto 24456 se pueda acceder a dicho servidor. También se filtran las credenciales de acceso las cuales son usuario **abdullah.yasin** y contraseña **XhIhGame_90HJLDASxfd&hoooad**.

En pasos anteriores se ha visto que dicho puerto, 24456, estaba abierto, por tanto vamos a comprobar que desde la IP del atacante se ha realizado la llamada a los puertos para verificar si el atacante ha abierto dicho puerto. Para ello usaremos el siguiente filtro en la pestaña Hunt:

*** AND source.ip: "3.109.209.43" AND destination.port: "29999" | groupby observer.name**

Mediante este filtro buscamos los eventos que tenga de origen la IP del atacante y puerto destino el primer puerto que se necesita para abrir el puerto 24456.

Timestamp ▲	source.ip	source.port	destination.ip	destination.port
2023-03-21 10:42:25.035 +00:00	3.109.209.43	58442	172.31.39.46	29999
2023-03-21 10:58:50.287 +00:00	3.109.209.43	56260	172.31.39.46	29999

Ilustración 48 Puerto destino 29999

El filtro nos da dos eventos siendo el primero a las 10:42:25, lo descartamos ya que previamente hablamos del escaneo de puertos en ese margen de tiempo, por tanto nos centramos en el evento fechado en 21-03-2023 a las 10:58:58. Usamos esta marca de tiempo para filtrar de nuevo:

*** AND source.ip: "3.109.209.43" AND @timestamp:"2023-03-21T10:58:50" | groupby observer.name**

Timestamp ▲	source.ip	source.port	destination.ip	destination.port
2023-03-21 10:58:50.287 +00:00	3.109.209.43	56260	172.31.39.46	29999
2023-03-21 10:58:50.287 +00:00	3.109.209.43	40650	172.31.39.46	50234
2023-03-21 10:58:50.288 +00:00	3.109.209.43	45018	172.31.39.46	45087

Ilustración 49 Secuencia de puertos

Como deducíamos anteriormente, desde la IP del atacante se ha realizado la secuencia de llamada a los puertos para abrir el puerto en el que se puede acceder al servidor FTP interno.

Vamos a comprobar si el atacante ha utilizado las credenciales mencionadas anteriormente y ha accedido al servidor FTP interno usando el puerto 24456. Para ello usaremos el siguiente filtro en la pestaña Hunt:

tags:ftp | groupby ftp.user ftp.argument

El primer evento que aparece con el usuario **abdullah.yasin** muestra que a las 11:00:06 el 21-03-2023 desde la IP del atacante y por el puerto 24456 hay un acceso correcto al servicio FTP.

2023-03-21 11:00:06.851 +00:00	3.109.209.43	38032	172.31.39.46	24456	abdullah.yasin
--------------------------------	--------------	-------	--------------	-------	----------------

Ilustración 50 Evento de acceso

```
USER abdullah.yasin
331 Please specify the password.
PASS Xh1hGame_90HJLDASxfd&hoooad
230 Login successful.
```

Ilustración 51 Acceso correcto al servidor FTP interno

Con esto se confirma que el atacante ha tenido acceso al servidor FTP interno de la empresa.

```
#!/bin/bash

# Define variables
DB_HOST="3.13.65.234"
DB_PORT="3306"
DB_USER="tony.shephard"
DB_PASSWORD="GameOfthronesRocks7865!"
DB_NAME="Internal_Tasks"
QUERY="SELECT * FROM Tasks;"

# Execute query and store result in a variable
RESULT=$(mysql -h $DB_HOST -P $DB_PORT -u $DB_USER -p$DB_PASSWORD $DB_NAME -e "$QUERY")

# Print the result
echo "$RESULT"
```

Ilustración 52 Archivo .fetch.sh

Otro de los archivos a los que accedió el atacante fue `fetch.sh` que contiene una posible automatización para una consulta a una base de datos. En la que hay datos y credenciales de un servidor de base de datos credenciales de acceso.

Ya que el atacante ha entrado en servidores con información crítica vamos a buscar si ha podido entrar por SSH al servidor.

Para ello vamos a utilizar un filtro en la pestaña Hunt que tenga la IP del atacante como origen y ver si se autenticado.

tags:ssh AND source.ip: "3.109.209.43" | groupby ssh.version ssh.client "source.ip" "ssh.authentication.success"

ssh.client	source.ip	destination.ip	ssh.authentication.success	@timestamp
SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.1	3.109.209.43	172.31.39.46	true	2023-03-21T11:25:42.572Z

Ilustración 53 Acceso por SSH del atacante

Confirmamos que el atacante ha accedido el 21-03-2023 a las 11:25:42 por SSH a la IP 172.31.39.46. Dado este acceso vamos a buscar que ha podido hacer el atacante en el servidor.

Una de las alarmas que nos muestra Security Onion es la siguiente:

ET INFO Dotted Quad Host ZIP Request

Timestamp	rule.name	event.severity_label	source.ip	source.port	destination.ip	destination.port
2023-03-21 11:42:34.413 +00:00	ET INFO Dotted Quad Host ZIP Request	medium	172.31.39.46	37808	13.233.179.35	80

Ilustración 54 Posible descarga de malware

Esta alarma muestra que tras el acceso del atacante por SSH hay una petición http, puerto 80, a la IP 13.233.179.35. Abrimos con la pestaña PCAP esta alarma y vemos que se utiliza Wget, versión 1.21.1, para descargar un archivo .zip con lo que parece un Ransomware.

```
GET /PKCampaign/Targets/Forela/Ransomware2_server.zip HTTP/1.1
Host: 13.233.179.35
User-Agent: Wget/1.21.2
Accept: */*
Accept-Encoding: identity
Connection: Keep-Alive
```

Ilustración 55 Descarga de Ransomware

Tras este análisis vamos a realizar un resumen de la información que hemos obtenido con Security Onion. Obtenemos la IP atacante **3.109.209.43**, la IP víctima **172.31.39.46** y un escaneo de puertos el **21-03-2023 a las 10:42:35 UTC**. Los puertos abiertos obtenidos por el atacante han sido **21, 22, 3306, 6379, 8086 y 24456**. Ataque por fuerza bruta el **21-03-2023 a las 10:49:43 UTC** obteniendo el usuario **tony.shepard** con la contraseña **Summer2023!**. Acceso a las **10:52:03 UTC** al archivo **.backup** y a las **10:55:13 UTC** al archivo **.fetch.sh** el 21-

03-2023. En el archivo .backup se filtran la técnica de portknocking con los puertos **29999, 50234 y 45087** en ese orden abre el puerto **24456** que permite el acceso a un servidor FTP interno, además del usuario **abdullah.yasin** y contraseña **XhIhGame_90HJLDASxfd&hoooad** para su acceso. Confirmamos el acceso al FTP interno con las credenciales filtradas. Vemos que el atacante ha accedido el **21-03-2023 a las 11:25:42** por SSH a la IP **172.31.39.46**. Por último obtenemos la información que el atacante utiliza **Wget** con la versión **1.21.1** para descargar un malware tipo Ransomware de la IP **13.233.179.35**.

Tras toda esta información podemos concluir que la distribución Security Onion tiene mucha capacidad de obtener información de lo que ocurre a nivel de red.

5.3 Análisis de archivos

Otro de los casos planteados sería el análisis de archivos que hayan podido ser modificados.

Dicho esto vamos a realizar la simulación de un caso de esteganografía que consiste en introducir datos ocultos en una imagen. Disponemos de tres imágenes.



Ilustración 56 Foto original

Está será la imagen principal que es un archivo JPG con un tamaño de 184 KB.

La siguiente imagen será la que introduciremos en la imagen principal que es un archivo JPG con tamaño de 49 KB.

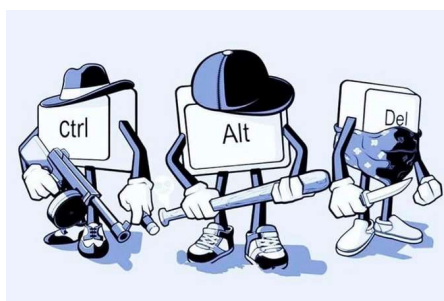


Ilustración 57 Imagen a añadir

Y por último tenemos la imagen con las dos juntas que podremos observar la imagen introducida no se ve a simple vista siendo un archivo JPG con tamaño 232 KB.



Ilustración 58 Imagen con datos ocultos

Con esta disposición se realiza la misma simulación que en el caso anteriormente explicado en el punto 5.1. Por tanto haríamos un escaneo de puertos y un ataque por fuerza bruta en el servicio FTP consiguiendo las credenciales correspondientes por parte del atacante.

Partiendo del punto anterior simularemos la extracción de un archivo y posteriormente devolveremos ese archivo con más datos incorporados para poder ver que nos muestra la herramienta Security Onion y si es capaz de encontrar algún indicio de lo que ha ocurrido.

```
ftp> dir
200 EPRT command successful
150 Opening ASCII mode data connection for file list
-rw-rw-r-- 1 vagrant vagrant 187611 Mar 11 11:07 FotoOriginal.jpg
-rw-r--r-- 1 vagrant vagrant 86562816 Oct 29 2020 VBoxGuestAdditions.iso
drwxrwxr-x 2 vagrant vagrant 4096 Feb 20 22:53 backup
-rw-rw-r-- 1 vagrant vagrant 30 Feb 21 22:11 datos.txt
drwxrwxr-x 3 vagrant vagrant 4096 Nov 29 17:26 elastic-agent-8.8.2-li
nux-x86_64
-rw-rw-r-- 1 vagrant vagrant 538004783 Nov 29 17:23 elastic-agent-8.8.2-l
inux-x86_64.tar.gz
226 Transfer complete
ftp> █
```

Ilustración 59 Archivos de máquina víctima

Como vemos en la imagen anterior tenemos acceso al FTP de la víctima y vemos el archivo FotoOriginal.jpg que es el que vamos a extraer y devolver con el mismo nombre y formato.

```
ftp> get FotoOriginal.jpg
local: FotoOriginal.jpg remote: FotoOriginal.jpg
200 EPRT command successful
150 Opening BINARY mode data connection for FotoOriginal.jpg (187611 bytes)
100% |*****| 183 KiB 65.68 MiB/s 00:00 ETA
226 Transfer complete
187611 bytes received in 00:00 (51.92 MiB/s)
ftp> quit
221 Goodbye.
```

Ilustración 60 Descarga archivo

```

ftp> put FotoOriginal.jpg
local: FotoOriginal.jpg remote: FotoOriginal.jpg
200 EPRT command successful
150 Opening BINARY mode data connection for FotoOriginal.jpg
100% |*****| 231 KiB 104.28 MiB/s 00:00 ETA
226 Transfer complete
237289 bytes sent in 00:00 (57.69 MiB/s)
ftp> quit
221 Goodbye.

```

Ilustración 61 Subida de archivo con datos ocultos

Como vemos en las anteriores imágenes es el mismo nombre de archivo pero con un tamaño mayor, ya que, contiene dentro de la imagen original la que se ha añadido. Una vez realizada la simulación vamos a ver que puede ofrecer Security Onion para este tipo de caso.

	Count	rule.name
 	12	ET SCAN Possible Nmap User-Agent Observed
 	12	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)
 	15	ET POLICY Possible Kali Linux hostname in DHCP Request Packet
 	37	ET SCAN Potential FTP Brute-Force attempt response
 	1	ET SCAN Potential VNC Scan 5900-5920
 	1	ET SCAN Potential VNC Scan 5800-5820
 	2	ET SCAN Suspicious inbound to MySQL port 3306
 	2	ET SCAN Suspicious inbound to PostgreSQL port 5432
 	2	ET SCAN Suspicious inbound to Oracle SQL port 1521
 	2	ET SCAN Suspicious inbound to MSSQL port 1433
 	2	ET SCAN Non-Allowed Host Tried to Connect to MySQL Server
 	15	ET SCAN Multiple FTP Root Login Attempts from Single Source - Possible Brute Force Attempt

Ilustración 62 Alarmas que se han registrado durante la simulación

Vista la imagen anterior no se ve ninguna alarma en la que de manera predefinida nos indique que ha habido transferencia de archivos, pero si nos indica una alarma que hay intentos de inicio de sesión desde un mismo punto y que puede ser un intento de fuerza bruta.

Una vez determinado esto vamos a seguir el mismo procedimiento que en los casos anteriores. Determinamos como anteriores casos de donde proceden los intentos de conexión y conseguimos la IP **192.168.56.106** (Kali Linux) como ya hicimos en el apartado 5.1.

En este punto deberíamos buscar si el atacante ha conseguido acceder al servicio FTP para ello podemos utilizar el siguiente filtro en la pestaña Hunt:

tags:ftp AND source.ip:"192.168.56.106" | groupby ftp.user ftp.argument

Timestamp ▲	source.ip	source.port	destination.ip	destination.port	ftp.user	ftp.command	ftp.argument
2024-04-23 11:23:26.632 +00:00	192.168.56.106	44562	192.168.56.103	21	vagrant	EPRT	1 192.168.56.106 49165
2024-04-23 11:28:08.753 +00:00	192.168.56.106	44562	192.168.56.103	21	vagrant	EPRT	1 192.168.56.106 43555
2024-04-23 11:28:09.872 +00:00	192.168.56.106	44562	192.168.56.103	21	vagrant	EPRT	1 192.168.56.106 58213
2024-04-23 11:28:09.873 +00:00	192.168.56.106	44562	192.168.56.103	21	vagrant	RETR	ftp://192.168.56.103//FotoOriginal.jpg
2024-04-23 11:29:29.095 +00:00	192.168.56.106	45466	192.168.56.103	21	vagrant	EPRT	1 192.168.56.106 60657
2024-04-23 11:29:29.096 +00:00	192.168.56.106	45466	192.168.56.103	21	vagrant	STOR	ftp://192.168.56.103//FotoOriginal.jpg

Ilustración 63 Eventos filtrados

En la siguiente ilustración se muestran dos eventos desde la IP **192.168.56.106** con puerto destino 21 (FTP) en los que se descarga el archivo FotoOriginal.jpg y sube otro archivo con el mismo nombre un minuto más tarde.

2024-04-23 11:28:09.873 +00:00	192.168.56.106	44562	192.168.56.103	21	vagrant	RETR	ftp://192.168.56.103//FotoOriginal.jpg
2024-04-23 11:29:29.095 +00:00	192.168.56.106	45466	192.168.56.103	21	vagrant	EPRT	1 192.168.56.106 60657
2024-04-23 11:29:29.096 +00:00	192.168.56.106	45466	192.168.56.103	21	vagrant	STOR	ftp://192.168.56.103//FotoOriginal.jpg

Ilustración 64 Eventos de descarga y subida FTP

Visto esto vamos a buscar si podemos encontrar algún indicio de si ha ocurrido algo con estos archivos utilizando el siguiente comando en la pestaña Hunt:

tags:file | groupby file.source source.ip

2024-04-23 11:28:09.875 +00:00	192.168.56.103	192.168.56.106	image/jpeg	FTP_DATA
2024-04-23 11:29:29.098 +00:00	192.168.56.106	192.168.56.103	image/jpeg	FTP_DATA

Ilustración 65 Datos FTP

Encontramos los siguientes eventos que corresponden en el tiempo con la descarga y subida del archivo. Algo útil que encontramos en la descripción del evento es el hash en MD5 y SHA1 que en caso de la descarga corresponde con:

hash.md5	b746ba9c65cf3244d19c6e79074f49f4
hash.sha1	6d95bf0dcf3df70d28843e6eb183d7335a560265

Ilustración 66 Hash de Foto original

Y en el caso de la subida del archivo corresponde con:

hash.md5	67d4ff38091d209ab3d860e888d30aa4
hash.sha1	e3152d6734794f68648e3f0fbdba79d7d7fbb2b

Ilustración 67 Hash de foto con datos ocultos

Comparando las dos imágenes muestra que nos son el mismo archivo. Se ha realizado la búsqueda de más indicios en los que podemos ver si se ha modificado el archivo, como por ejemplo, abrimos el PCAP de los eventos y no muestra nada concluyente. Por otro lado, nos ofrece una descripción en la que también vemos que el archivo tiene diferente tamaño comparando los dos eventos. Tampoco tiene la suite ninguna herramienta para revisar el archivo para ver si contiene algo oculto.

Visto esto podemos decir que en este caso deberíamos utilizar otras herramientas externas para analizar los archivos, ya que el único indicio que nos da es que el archivo se ha modificado tanto por ver que tiene diferente tamaño como hash.

5.4 Recuperación de datos

Hemos conectado un dispositivo de almacenamiento USB con distintos archivos. También hemos hecho las configuraciones necesarias en VirtualBox para que la distribución Security Onion detecte este USB.

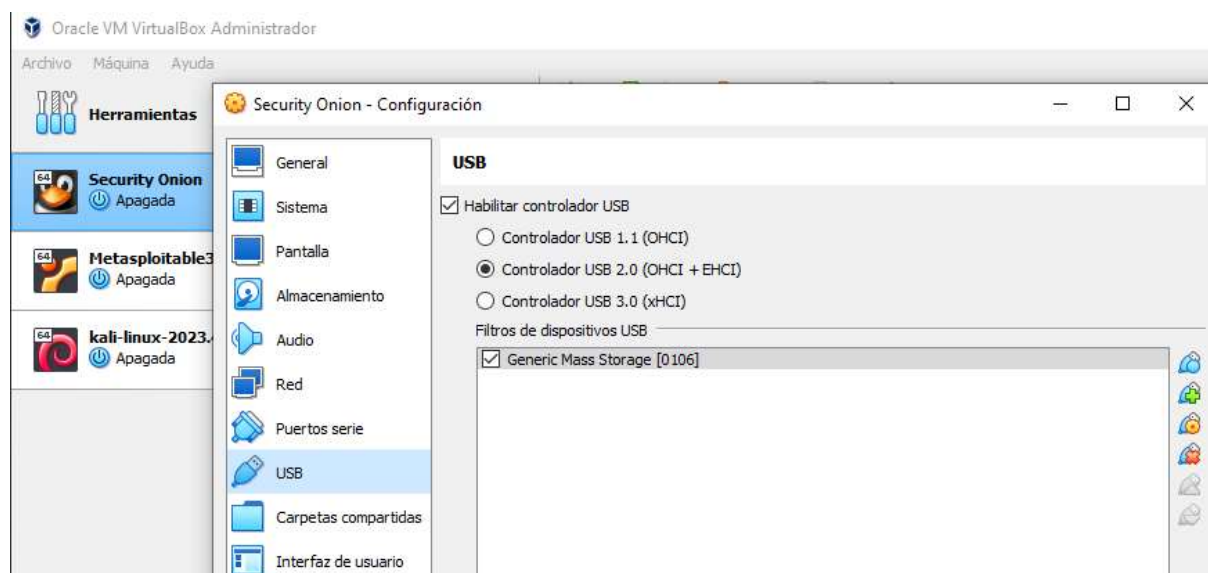


Ilustración 68 Dispositivo USB conectado

No hemos encontrado ninguna herramienta o capacidad por parte de la distribución Security Onion que pueda realizar esta tarea. Otras distribuciones como Kali Linux tienen herramientas como Foremost que si pueden realizarla.

6. Presupuesto

El presupuesto de este proyecto lo dividiremos en dos partes. La parte del coste de la mano de obra del alumno que se ha encargado de la realización del proyecto y la parte de los costes materiales.

Se ha supuesto una media de 10 horas semanales a lo largo de 8 meses por parte de alumno.

Empleado	Horas	Precio/Hora	Total
Alumno	320	15 €	4800 €

Tabla 3 Coste humano

Concepto	Unidades	Precio/Unidad	Total
HP OMEN	1	2000 €	2000 €
SO Security Onion	1	0 €	0 €
SO Kali Linux	1	0 €	0 €
SO Metasploitable3	1	0 €	0 €
SO VirtualBox	1	0 €	0 €

Tabla 4 Coste materiales

El total de los costes materiales incluyendo el 21% de IVA equivale a 2420 €.

Por tanto el coste final del proyecto asciende 7220 €.

7. Impacto del proyecto

Respecto al ámbito tecnológico, hablamos de que este estudio permite definir qué capacidad forense digital tiene la distribución Security Onion, tanto de manera positiva como negativa, pudiendo dar una valoración de dicha distribución en el ámbito forense. Otro de los puntos es que en los tiempos actuales, se están produciendo ciberataques de manera muy habitual, respecto a este ámbito social este estudio va a aportar conclusiones de si Security Onion puede ser útil como herramienta para encontrar las pruebas necesarias para realizar informes sobre estos ciberataques. Dadas estas explicaciones anteriores cabe destacar que en el ámbito económico también afecta, por un lado, ya que al ser una distribución gratuita puede usarla cualquiera y también hablamos de que estos ciberataques podrían ser a entidades bancarias con el efecto que supondría.

8. Conclusiones

En este apartado vamos a desarrollar las conclusiones obtenidas del análisis realizado en los casos anteriores y proponer ciertas ideas para continuar con el trabajo que se ha desarrollado.

8.1 Conclusiones

Una vez vistos los diferentes tipos de casos posibles en los que puede intervenir el análisis forense digital, la suite Security Onion está claramente orientada a auditar redes. En este ámbito la suite es capaz de obtener mucha información como se ha mostrado en los casos en los que implicaba un análisis del tráfico de red, demostrando que un analista podría realizar un informe pericial. Por otro lado la herramienta sólo muestra los eventos que es capaz de relacionar mediante las reglas predefinidas. Es responsabilidad del analista interpretar los datos obtenidos, ya que no tiene ninguna herramienta para realizar informes.

También se debe hablar de la carencia de herramientas fuera del ámbito de auditoría de red, ya que no es capaz de realizar ni el análisis ni la recuperación de archivos, ni es capaz de realizar ningún tipo de acción para revisar las vulnerabilidades de seguridad de diferentes tipos de dispositivos o sistemas.

En cuestión de otras suites, claramente hay otras suites que son capaces de realizar muchas más acciones a nivel de análisis forense digital, pero también hay que decir que la suite Security Onion a nivel de auditoría de red frente a otras suites, es capaz de dar mucha más información de una manera más sencilla.

Dado que el objetivo es centrarse en la capacidad de análisis forense digital de la suite Security Onion podemos decir que si el caso está centrado en temas de tráfico de red tiene muchísimo potencial, pero fuera de ese ámbito no dispone de herramientas para poder realizar acciones de análisis.

8.2 Líneas futuras

Dado que Security Onion no es la mejor opción para análisis forense digital, se podría realizar un estudio centrado en la auditorías de red en el que se utilice la suite en este ámbito en un entorno de trabajo más real, como por ejemplo, en algún departamento de alguna empresa, en el que existan diferentes tipos de sistemas operativos e integrar agentes en dichos sistemas operativos para que aporten más información.

También se podría intentar automatizar la información que analiza la herramienta para que se puedan realizar informes periciales.

9. Referencias

- [1] R. Velasco, «Redes Zone,» 26 Septiembre 2016. [En línea]. Available: <https://www.redeszone.net/2016/03/26/security-onion-una-distribucion-linux-auditar-la-seguridad-una-red/>. [Último acceso: Septiembre 2023].
- [2] Isaac, «Linux Adictos,» [En línea]. Available: <https://www.linuxadictos.com/security-onion-tu-distro-ideal-para-auditar-redes.html>. [Último acceso: Septiembre 2023].
- [3] S. Onion, «Security Onion Documentation,» 2022. [En línea]. Available: <https://securityonion.readthedocs.io/en/latest/>. [Último acceso: Septiembre 2023].
- [4] EC-Council, «EC-Council,» [En línea]. Available: <https://www.eccouncil.org/cybersecurity/what-is-digital-forensics>. [Último acceso: Septiembre 2023].
- [5] N. d. I. Web, «Ninjas de la Web,» [En línea]. Available: <https://ninjasdelaweb.com/metodologia-de-analisis-forense/>. [Último acceso: Septiembre 2023].
- [6] L. Paus, «welivesecurity,» 15 Abril 2015. [En línea]. Available: <https://www.welivesecurity.com/la-es/2015/04/15/5-fases-analisis-forense-digital/>. [Último acceso: Septiembre 2023].
- [7] K. Black, «EasyTechJunkie,» 7 Noviembre 2023. [En línea]. Available: <https://www.easytechjunkie.com/what-are-software-suites.htm>. [Último acceso: Noviembre 2023].
- [8] OffSec Services Limited, «kali.org,» [En línea]. Available: <https://www.kali.org/>. [Último acceso: Septiembre 2023].
- [9] derechodelared, «Derechodelared,» 19 Marzo 2023. [En línea]. Available: <https://derechodelared.com/kali-linux-que-es/>. [Último acceso: Septiembre 2023].
- [10] R. KeepCoding, «KeepCoding,» 6 Enero 2023. [En línea]. Available: <https://keepcoding.io/blog/que-es-kali-linux/>. [Último acceso: Septiembre 2023].
- [11] O. S. L. 2024. [En línea]. Available: <https://www.kali.org/tools/>. [Último acceso: Abril 2024].
- [12] R. Saive, «Tecmint,» 8 Junio 2022. [En línea]. Available: <https://www.tecmint.com/parrot-security-os/>. [Último acceso: Septiembre 2023].

- [13] J. Maury, «eSecurityPlanet,» 23 Agosto 2022. [En línea]. Available: <https://www.esecurityplanet.com/products/open-source-distros-for-pentesting-and-forensics/>. [Último acceso: Septiembre 2023].
- [14] P. Team. [En línea]. Available: <https://github.com/ParrotSec/parrot-tools/blob/master/debian/control>. [Último acceso: Abril 2024].
- [15] Travis, «Systran Box,» 4 Marzo 2023. [En línea]. Available: <https://www.systranbox.com/getting-started-with-black-arch-linux-a-guide-to-installing-and-using-the-open-source-security-distro/>. [Último acceso: Septiembre 2023].
- [16] «elhacker.net,» 14 Noviembre 2021. [En línea]. Available: <https://blog.elhacker.net/2021/11/blackarch-distro-de-hacking-etico-para-pentesting.html>. [Último acceso: Septiembre 2023].
- [17] B. Linux, «BlackArch Linux,» [En línea]. Available: <https://blackarch.org/forensic.html>. [Último acceso: Abril 2024].
- [18] CAINE, «CAINE,» [En línea]. Available: <https://www.caine-live.net/>. [Último acceso: Septiembre 2023].
- [19] DragonJar, «DragonJar,» [En línea]. Available: <https://www.dragonjar.org/caine-livecd-linux-para-informatica-forense.xhtml>. [Último acceso: Septiembre 2023].
- [20] «Geeks for Geeks,» [En línea]. Available: <https://www.geeksforgeeks.org/caine-forensic-environment/>. [Último acceso: Septiembre 2023].
- [21] «Caine Live,» [En línea]. Available: <https://www.caine-live.net/page11/page11.html>. [Último acceso: Abril 2024].
- [22] V. Garcia, «Servicios Vartech,» 24 Febrero 2023. [En línea]. Available: <https://www.serviciosvartech.com/security-onion-que-es-y-como-funciona>. [Último acceso: Septiembre 2023].
- [23] L. Fernández, «Redes zone,» 18 Agosto 2023. [En línea]. Available: <https://www.redeszone.net/tutoriales/seguridad/security-onion-linux-auditorias-redes/>. [Último acceso: Septiembre 2023].
- [24] Security Onion, «Security Onion Hardware,» [En línea]. Available: <https://docs.securityonion.net/en/2.3/hardware.html#hardware>. [Último acceso: Septiembre 2023].
- [25] «Security Onion,» [En línea]. Available: <https://docs.securityonion.net/en/2.4/index.html>. [Último acceso: Abril 2024].

-
- [26] E. Limones, «OpenWebinars,» 5 Mayo 2021. [En línea]. Available: <https://openwebinars.net/blog/virtualizacion-que-es-para-que-sirve-y-ventajas/>. [Último acceso: Septiembre 2023].
- [27] Redhat, «Diferencias entre los contenedores y las máquinas virtuales,» 2020. [En línea]. Available: <https://www.redhat.com/es/topics/containers/containers-vs-vms#%C2%BFqu%C3%A9-es-un-contenedor>. [Último acceso: Abril 2024].
- [28] Security Onion, «Security Onion Architecture,» [En línea]. [Último acceso: Septiembre 2023].
- [29] S. D. Luz, «Tipos de red Virtual Box,» 17 Julio 2023. [En línea]. Available: <https://www.redeszone.net/tutoriales/redes-cable/configuracion-red-maquina-virtual-virtualbox/>. [Último acceso: Noviembre 2023].
- [30] N. Team, «<https://www.nakivo.com/blog/virtualbox-network-setting-guide/>,» 1 Junio 2023. [En línea]. Available: <https://www.nakivo.com/blog/virtualbox-network-setting-guide/>. [Último acceso: Noviembre 2023].
- [31] J. M. Ousmane Diallo, «Real-time data management on wireless sensor networks: A survey,» *Journal of Network and Computer Applications*, 2011.
- [32] Ó. Ortiz, A. B. Garcia, R. Capilla, J. Bosch y M. Hinchey, *Runtime Variability for Dynamic Reconfiguration in Wireless Sensor Network Product Lines*.
- [33] Oracle Corporation, «SunSpotWorld,» [En línea]. Available: <http://www.sunspotworld.com/docs/Yellow/SunSPOT-Programmers-Manual.pdf>.
- [34] Ubuntu, «<http://ubuntu-manual.org/>,» [En línea].
- [35] E. C. -. I. S. a. M. DG, «Internet of Things. Strategic Research Roadmap,» 2009.
- [36] «Mono Documentation,» [En línea]. Available: <http://docs.go-mono.com/>.
- [37] IEEE, «IEEE 802.15.4g™-2012,» [En línea]. Available: <http://standards.ieee.org/getieee802/download/802.15.4g-2012.pdf>.
- [38] R. M. Castejón, «Interconexión de redes de sensores inalámbricos 802.15.4 en localizaciones remotas,» *PFC*, pp. 8,9,10, 2011.
- [39] F. B. J. S. A. C. Diego Martínez, «Redes de Sensores y Actuadores Inalámbricas: Una Caracterización y Caso de Estudio para Aplicaciones Médicas en Espacios Cerrados,» *Universidad Autónoma de Occidente, Universidad Politécnica de Valencia*.

- [40] E. International, «ECMAScript Language Specification. Standar 262,» [En línea]. Available: <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>.
- [41] Apache.org, «Manual Apache Ant,» [En línea]. Available: <http://ant.apache.org/manual/intro.html>.

Anexo

A.1 Modos de configuración de Security Onion

La distribución tiene cuatro tipos de configuración de trabajo [28]:

- **Import:** Esta es la configuración más simple. Se compone de las suficientes herramientas para ser capaz de importar archivos pcap o archivos evtx y después poder ver los logs en Security Onion Console (SOC). No permite añadir Elastic Agents o nodos adicionales.

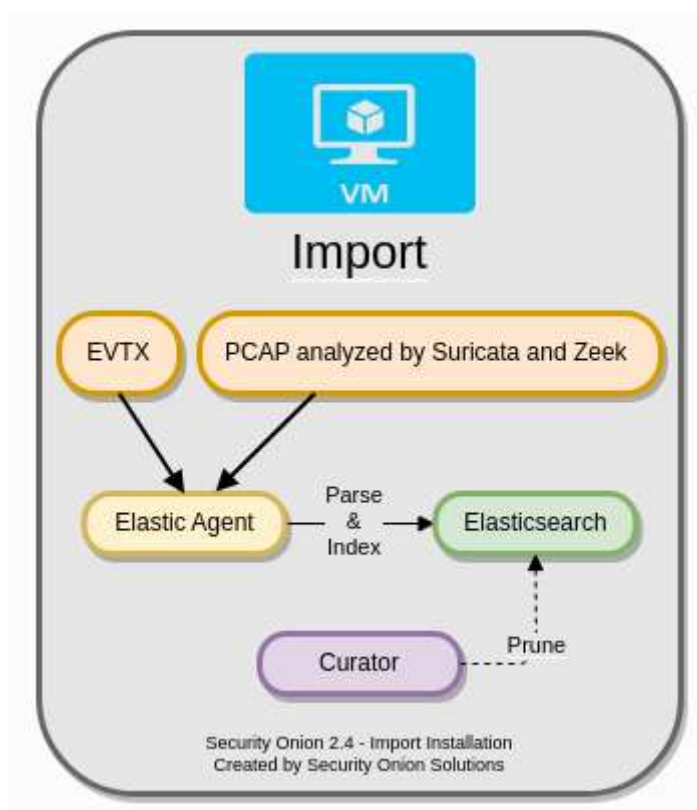


Ilustración 69 Security Onion con la configuración Import

- **Evaluation:** Esta configuración es algo más compleja que la anterior ya que usa una interfaz de red dedicada a capturar tráfico en vivo desde un puerto span o un TAP. Las herramientas que componen esta distribución monitorizan el tráfico de red en esa interfaz de captura y genera logs. Elastic Agent recolecta estos logs y los envía directamente a Elasticsearch. Esta arquitectura está diseñada para una instalación rápida y temporal. No permite añadir Elastic Agents o nodos adicionales.

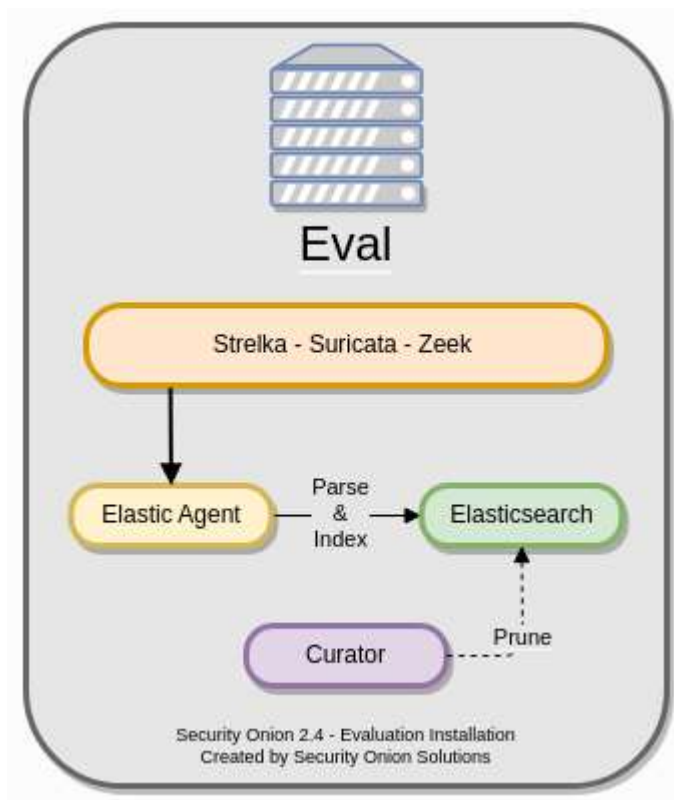


Ilustración 70 Security Onion con la configuración Evaluation

- **Standalone:** Es similar a la anterior respecto a todos los componentes que corren en la distribución. La diferencia está en que los datos capturados en vez de Elastic Agent enviar los logs directamente a Elasticsearch, este los manda a Logstash para que los mande a Redis para ponerlos en cola. Otra parte de Logstash extrae los datos de Redis y los manda a Elasticsearch, donde se analizan e indexan. Este tipo de arquitectura se utiliza para laboratorios de testeo, POCs o entornos en los que haya muy poco tráfico.

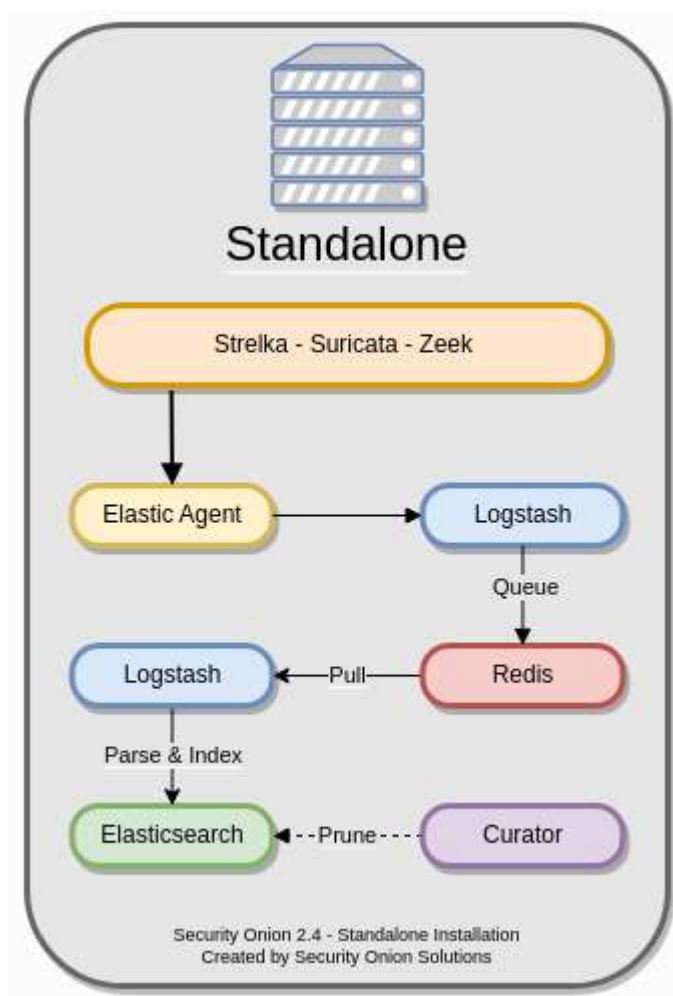


Ilustración 71 Security Onion con la configuración Standalone

- **Distributed:** La más recomendada para un entorno complejo es la configuración distribuida que incluye un nodo administrador, uno o más nodos de avance en los que están los sensores de red y uno o más nodos de búsqueda en los que está Elasticsearch. Este tipo de arquitectura es la más costosa, pero la que da más escalabilidad y rendimiento pudiendo añadir más nodos si es necesario.

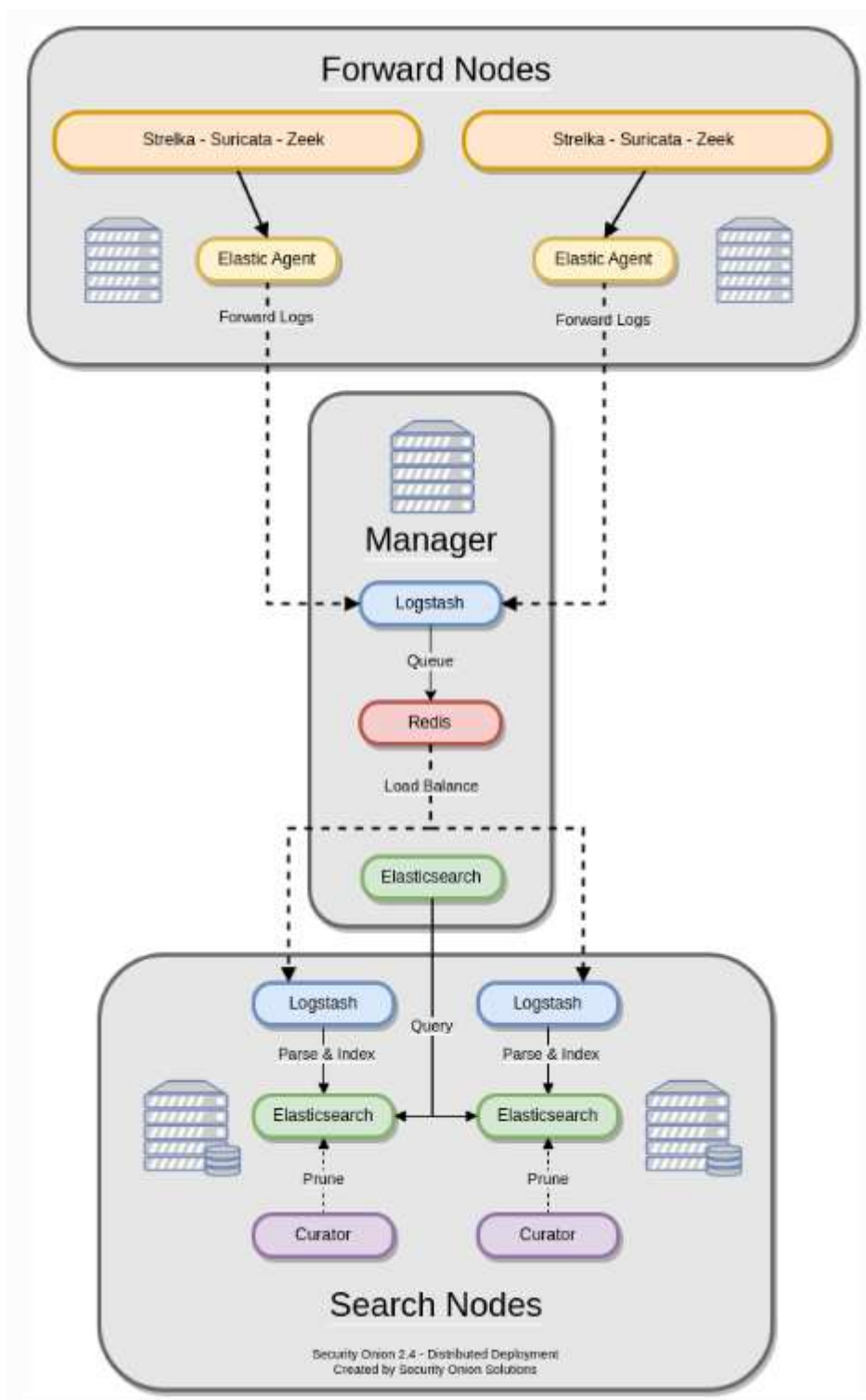


Ilustración 72 Security Onion con la configuración Distributed

A.2 Modos de configuración de los adaptadores de red de VirtualBox

En este apartado se describe algunas de las opciones de configuración de los adaptadores de red que dispone VirtualBox [29] junto a sus diagramas [30]:

- **NAT:** Es la opción predeterminada en los adaptadores de red de las máquinas virtuales en VirtualBox. Con este modo la máquina virtual obtiene una dirección IP privada establecida por VirtualBox la que permitirá salir a la red local doméstica y a

Internet utilizando la dirección privada del equipo real. Si un equipo de la red local quisiera comunicarse con la máquina virtual no podría hacerlo directamente y se tendría que utilizar la configuración de reenvío de puerto que tiene VirtualBox.

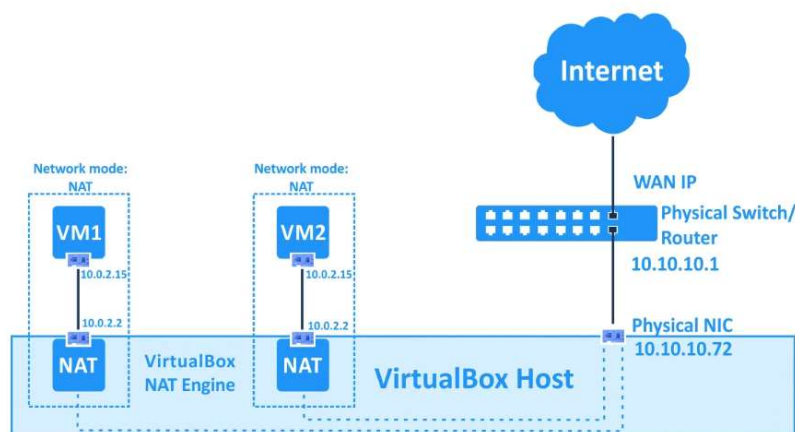


Ilustración 73 Diagrama con configuración NAT

- **Bridge:** Con esta opción la máquina virtual pasa a ser un dispositivo más dentro de la red local obteniendo un direccionamiento IP desde el servidor DHCP que tengamos en la red local doméstica o profesional. Mediante esta opción la máquina virtual tendrá acceso al resto de equipos de la red local y a Internet, el resto de los equipos de la red local también tendrá conectividad con la máquina virtual.

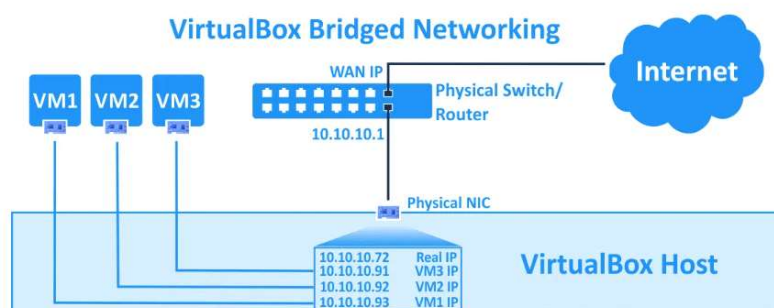


Ilustración 74 Diagrama con configuración Bridge

- **Red Interna:** Con este modo de configuración permite crear una red local de dos o más máquinas virtuales que se pueden comunicar entre sí, pero no tendrán acceso a la red local doméstica o profesional ni acceso a Internet.

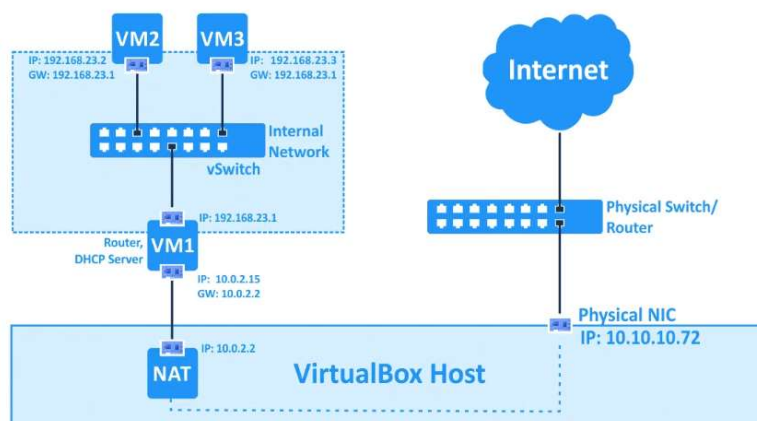


Ilustración 75 Diagrama con configuración Red Interna

- **Adaptador Host-Only:** Mediante este modo la máquina virtual se crea una red local en la que sólo puede comunicarse con otras máquinas virtuales con el adaptador de red en este modo y el equipo anfitrión. De esta forma las máquinas virtuales con esta configuración no tendrán acceso a la red local doméstica o profesional ni a Internet.

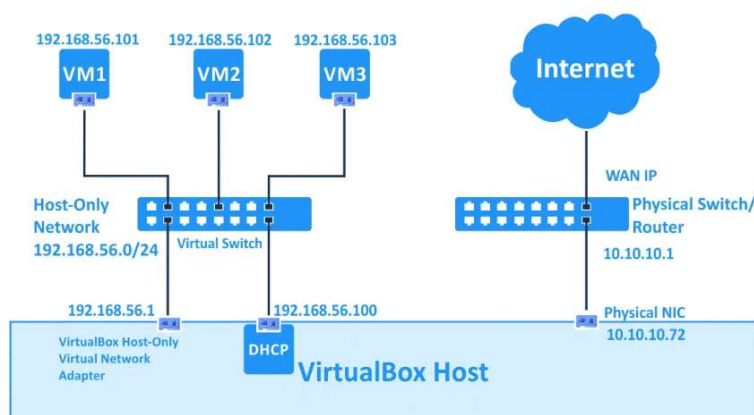


Ilustración 76 Diagrama con configuración Host-Only

- **Red NAT:** Esta opción combina la opción Red Interna y NAT permitiendo crear una red NAT con el direccionamiento a elegir por el usuario a la que podemos añadir un servidor DHCP que puede soportar IPv4 e IPv6. Esto permitirá que las máquinas virtuales estarán en una misma subred NAT y podrán comunicarse entre sí, con la red local doméstica o profesional y salir a Internet.

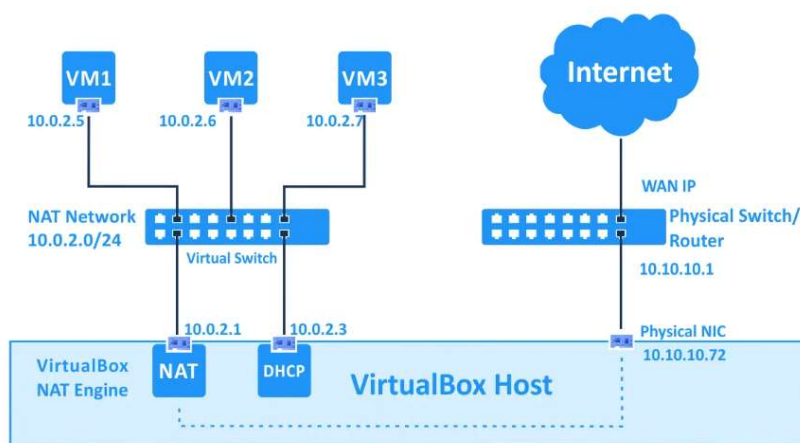


Ilustración 77 Diagrama con configuración Red NAT

A.3 Instalaciones de máquinas virtuales

A.3.1. Montaje de Security Onion

Se va a describir a continuación los pasos a seguir para la instalación de la suite Security Onion.

Se descarga la ISO de la página web oficial de Security Onion y se verifica la ISO tal y como se describe en la documentación².

Se crea una nueva máquina virtual utilizando la ISO correspondiente.

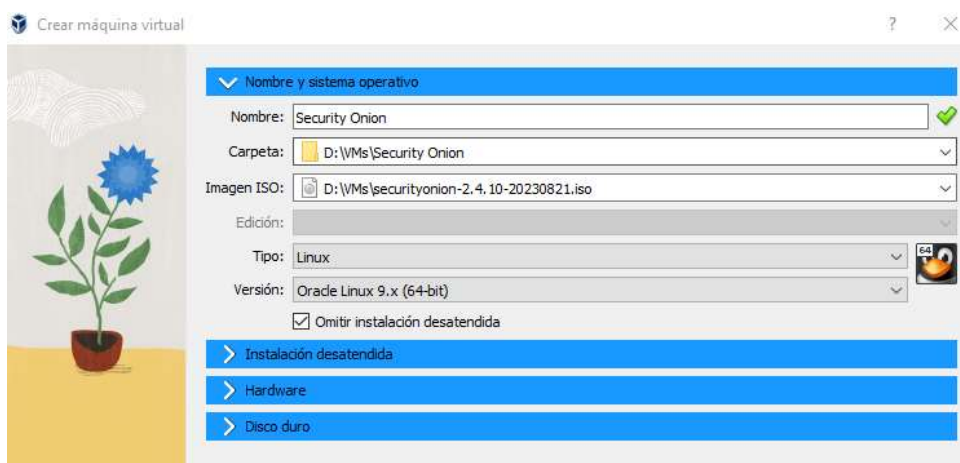


Ilustración 78 Instalación Security Onion mediante ISO

Como se indica se utiliza el sistema operativo Oracle Linux 9.x 64 bit y se elige el nombre para la máquina virtual y el directorio del host.

²https://github.com/Security-Onion-Solutions/securityonion/blob/2.4/main/DOWNLOAD_AND_VERIFY_ISO.md

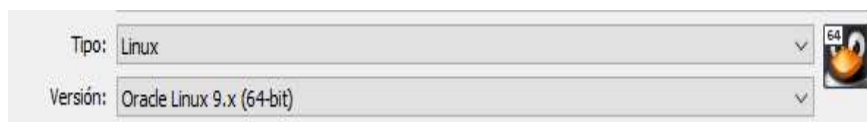


Ilustración 79 Elección de sistema operativo

Tal y como se recomienda en la documentación para la infraestructura Standalone la máquina virtual tendrá 200 GB de capacidad de disco duro, 4 procesadores y 16 GB de memoria RAM.

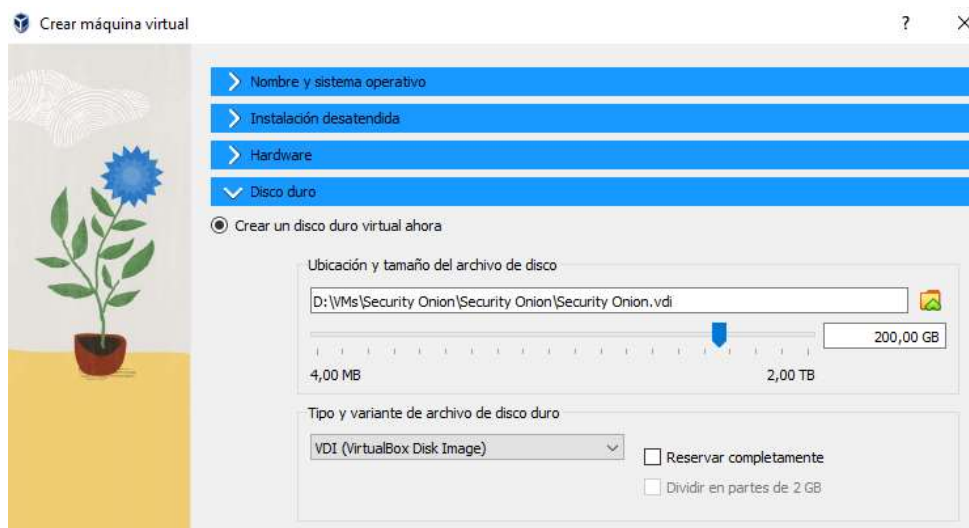


Ilustración 80 Capacidad disco duro para Security Onion

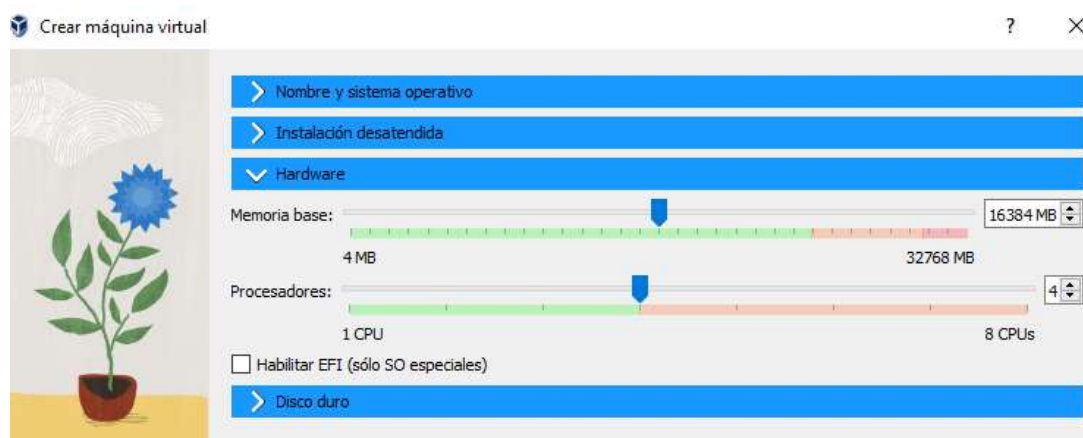


Ilustración 81 Requisitos de memoria de la máquina virtual para Security Onion

Para aislar las máquinas virtuales, tanto la de Security Onion Host como la máquina virtual atacante y la víctima se utilizarán los adaptadores de red en modo Host-Only explicados en el apartado 2.4 anteriormente. Aunque en algunos momentos puntuales se cambie la configuración de los adaptadores de red a modo NAT debido a instalaciones que necesiten salida a Internet.

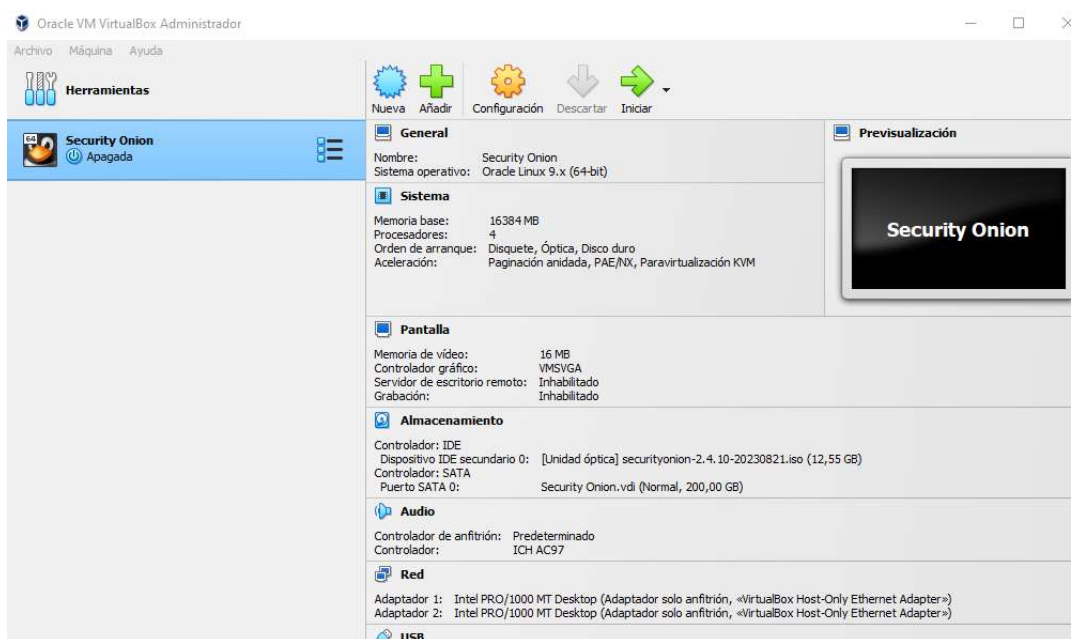


Ilustración 82 Características hardware de la máquina virtual para Security Onion

Una vez se arranca la máquina virtual con la suite de Security Onion comienza la instalación, una de las primeras opciones a decidir es el tipo de infraestructura que utilizar, que como ya se ha dicho anteriormente se toma la decisión de usar el tipo Standalone.

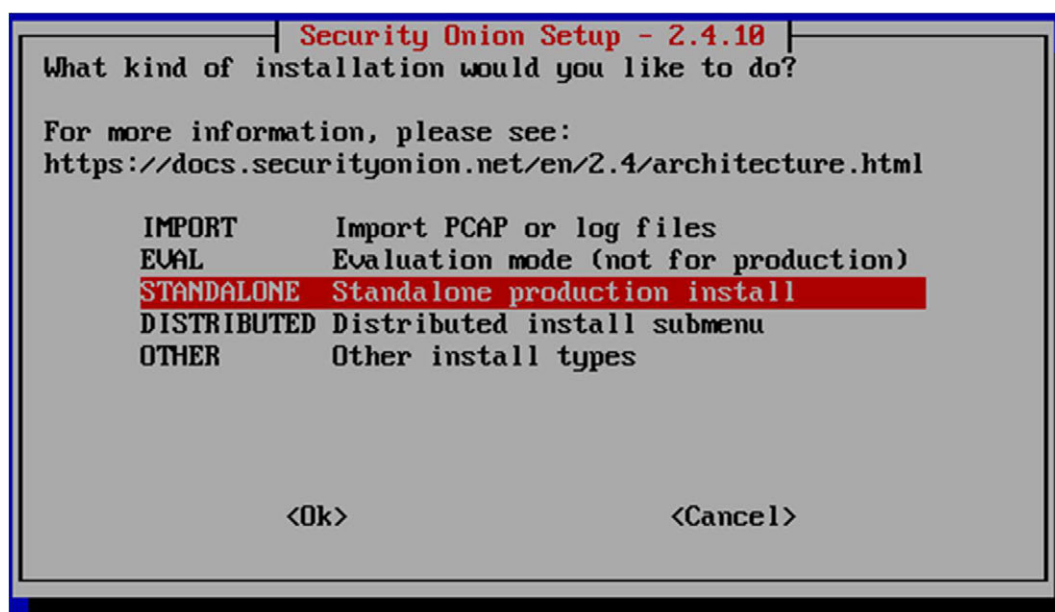


Ilustración 83 Elección del tipo de infraestructura de Security Onion

Se elige el adaptador de red que utilizara para la configuración de gestión.

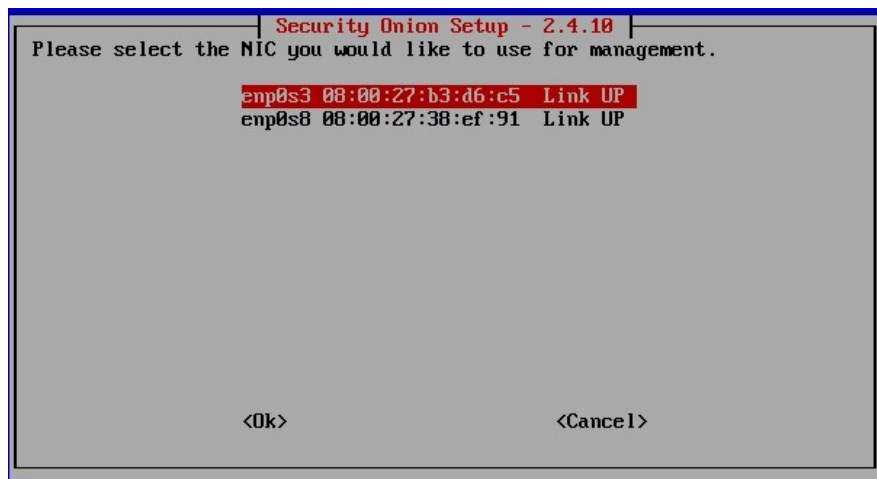


Ilustración 84 Elección de la tarjeta de red para gestión

Se utiliza la configuración del adaptador de red principal en modo NAT para la instalación, ya que requiere salida a Internet y en el modo Host-Only no tendría conectividad posteriormente a la instalación se configurará este adaptador en modo Host-Only.

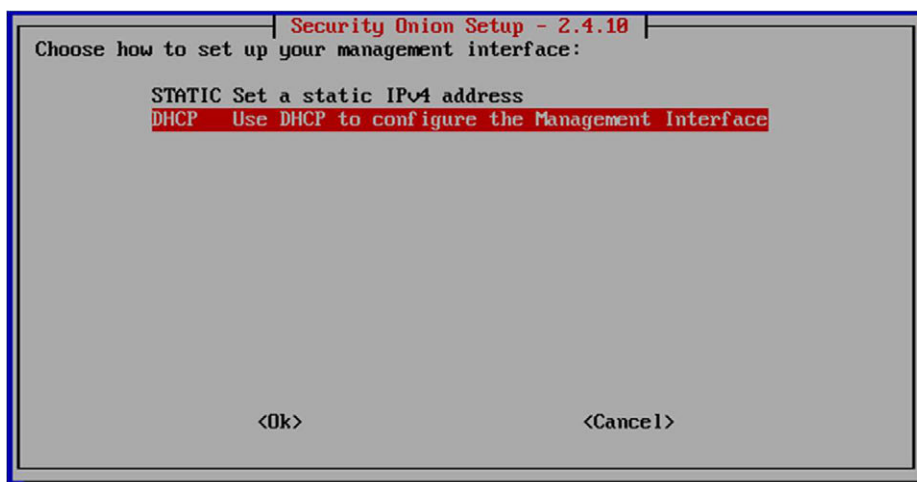


Ilustración 85 Utilización de DHCP en la tarjeta de red de gestión

Se utiliza el otro adaptador de red para la captura de tráfico que se utilizará para el estudio de lo que está sucediendo en la red.

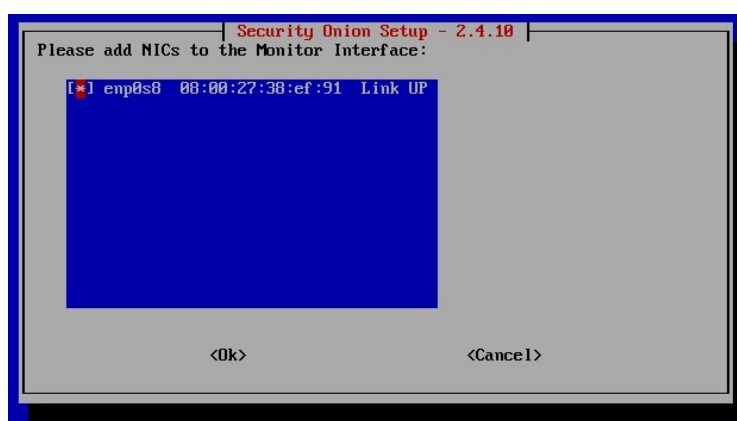


Ilustración 86 Elección de la tarjeta de red para capturar tráfico

Se añade un correo para el acceso a la consola de Security Onion y que se usará para recibir las alertas que se generen.

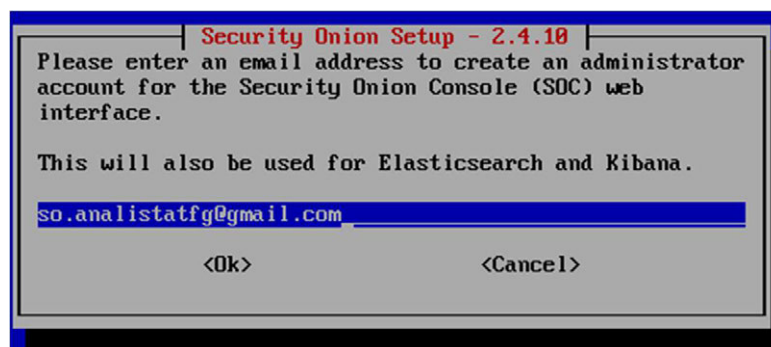


Ilustración 87 Cuenta de administrador

Para el acceso a la consola de gestión de la suite mediante interfaz web se va a realizar mediante IP.

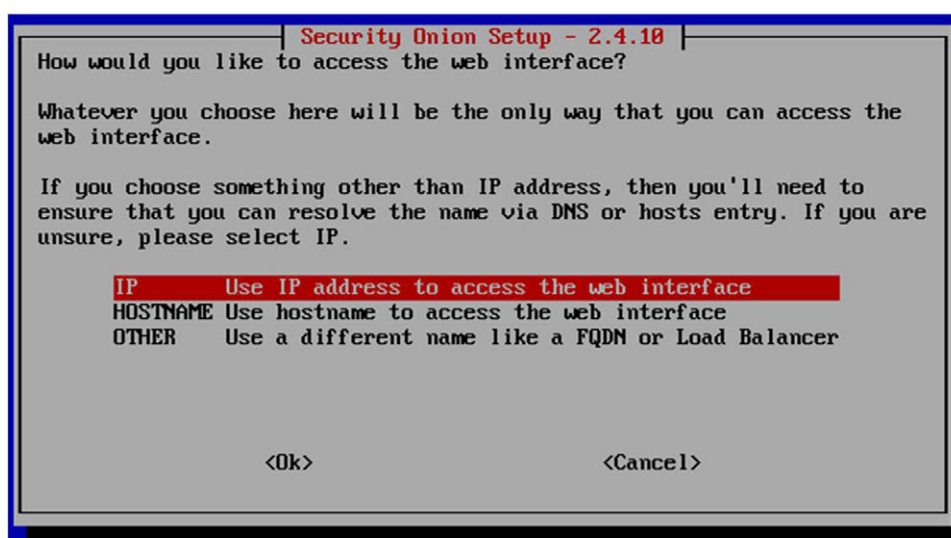


Ilustración 88 Acceso a Security Onion mediante IP

Una vez completados todos los pasos de selección de los parámetros que se necesitan para instalación aparece un resumen con todas estas elecciones para revisar y confirmar el procedimiento de instalación.

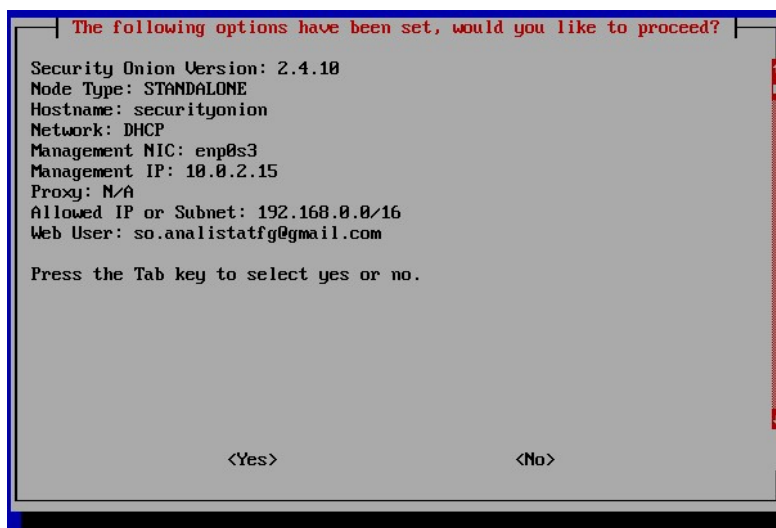


Ilustración 89 Opciones elegidas para Security Onion

Una vez terminada la instalación aparecerá un mensaje en la que podremos ver la IP para el acceso mediante interfaz web y el nombre de usuario de acceso.

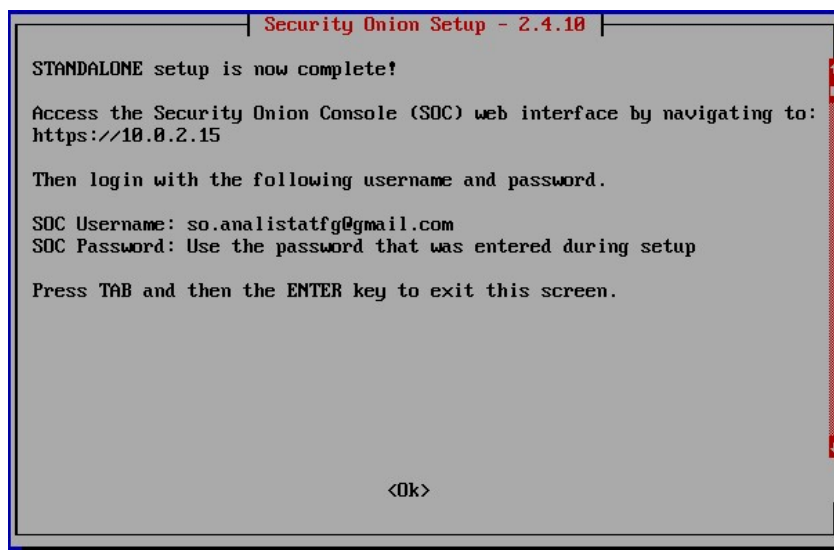


Ilustración 90 Instalación completada correctamente y accesos

Una vez se termina la instalación apagamos la máquina virtual. Se cambia el adaptador de red principal a modo Host-Only y una vez arrancada la máquina virtual utilizamos el comando **so-ip-update** para que cambie la IP de gestión a la subred correspondiente.

```

[so@securityonion ~]$ sudo so-ip-update
[sudo] password for so:
This tool will update a manager's IP address to the new IP assigned to the management network interface.

WARNING: This tool is still undergoing testing, use at your own risk!
  
```

Ilustración 91 Comando so-ip-update

A.3.2. Montaje de Metasploitable3

Para la máquina víctima utilizaremos Metasploitable3, que es una máquina virtual con una gran cantidad de vulnerabilidades de seguridad. Para su instalación y descarga seguiremos la documentación³ y los comandos correspondientes, en nuestro caso para Windows. Con los siguientes comandos se creará una máquina virtual con la versión de Ubuntu 14.04.

```
PS D:\VMs> cd .\metasploitable3-workspace\
PS D:\VMs\metasploitable3-workspace> Invoke-WebRequest -Uri "https://raw.githubusercontent.com/rapid7/metasploitable3/master/Vagrantfile" -OutFile "Vagrantfile"
PS D:\VMs\metasploitable3-workspace> vagrant up
```

Ilustración 92 Comandos para descarga e instalación Metasploitable3

Una vez creada la máquina virtual tendrá dos interfaces de red una en modo NAT necesaria para instalaciones, que se mantendrá desactivada exceptuando los casos de instalación, y otra en modo Host-Only, esta interfaz irá automáticamente al crear la máquina virtual con una configuración de IP estática que cambiaremos para que tenga la configuración en DHCP.

```
vagrant@metasploitable3-ub1404:/etc/network$ ls
if-down.d if-post-down.d if-pre-up.d if-up.d interfaces interfaces.d run
vagrant@metasploitable3-ub1404:/etc/network$ cd interfaces.d/
vagrant@metasploitable3-ub1404:/etc/network/interfaces.d$ ls
vagrant@metasploitable3-ub1404:/etc/network/interfaces.d$ cd ..
vagrant@metasploitable3-ub1404:/etc/network$ cat interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
auto eth1
iface eth1 inet static
    address 172.28.128.3
    netmask 255.255.255.0
#VAGRANT-END
```

Ilustración 93 Configuración interfaz de red Host-Only estática

³ <https://github.com/rapid7/metasploitable3>

```
vagrant@metasploitable3-ub1404:~$ cd /etc/network/
vagrant@metasploitable3-ub1404:/etc/network$ ls
if-down.d if-post-down.d if-pre-up.d if-up.d interfaces interfaces.d run
vagrant@metasploitable3-ub1404:/etc/network$ cat interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
auto eth1
iface eth1 inet dhcp
#VAGRANT-END
vagrant@metasploitable3-ub1404:/etc/network$
```

Ilustración 94 Cambio de IP estática a DHCP

Una vez realizados los cambios en la configuración ya tendríamos conectividad en la subred creada por VirtualBox para las interfaces de red con modo Host-Only.

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:c6:d4:59
          inet addr:192.168.56.103  Bcast:192.168.56.255  Mask:255.255.255.0
```

Ilustración 95 IP asignada por VirtualBox para interfaz de red modo Host-Only

A.3.3. Montaje de Kali Linux

Realizamos la instalación de la máquina atacante que en nuestro caso utilizaremos Kali Linux, descargamos el paquete que recomiendan para VirtualBox en su página oficial⁴.

Añadimos una nueva máquina virtual en VirtualBox y se cambia el adaptador de red a modo Host-Only.



Ilustración 96 Paquete con Kali Linux

⁴ <https://www.kali.org/get-kali/#kali-virtual-machines>



Ilustración 97 Características Hardware Kali Linux