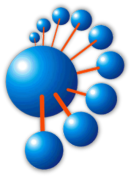
UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA
Y CIENCIAS DE LA COMPUTACIÓN



Sistema de Monitoreo de Indicadores de Compromiso con Panel Táctico para Plataforma TI de Organizaciones de la Defensa

POR

Sergio Esteban Cifuentes Torres

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Informático

Profesores Patrocinantes
Pedro Pinacho Davidson
Sergio Sobarzo Guzmán

Profesores Comisión
Javier Vidal Valenzuela
Julio Godoy del Campo

Concepción, 21 de septiembre 2023

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han sido parte fundamental para mí durante mis años universitarios y también durante mi crecimiento y desarrollo como persona.

Primero que todo quiero darle las gracias a Felipe Henríquez, compañero con quien desarrollé este trabajo y varios más durante nuestros estudios, y a quien considero un gran amigo. También le doy las gracias a su familia, a quienes les tengo mucho cariño, por acogerme en su hogar siempre con una actitud de respeto, apoyo, simpatía y generosidad.

Agradezco a los profesores Pedro Pinacho y Sergio Sobarzo por guiarnos durante este trabajo siempre dispuestos a ayudar a resolver cualquier problema que encontramos durante el desarrollo y por ofrecer sus conocimientos a lo largo de nuestra carrera universitaria.

Doy las gracias a todos los docentes y personal administrativo de la Facultad de Ingeniería y de la carrera de Ingeniería Civil Informática, quienes aportaron a mi desarrollo profesional y personal. Agradezco especialmente a Mauricio Echavarría, encargado de los laboratorios de Sistemas y Redes del Edificio de Ingeniería de Sistemas quien me ayudó a realizar pruebas del sistema desarrollado aportando conocimiento y experiencia de manera comprometida y alegre.

Quiero expresar mi enorme gratitud hacia mi padre Sergio Mariano Cifuentes Moraga, mi madre María Dolores Torres Saez, mi hermana María Pamela Cifuentes Torres y mi tía Guillermina Campos. Sin ellos no podría estar donde estoy hoy en día, siempre apoyándome y motivándome a ser mejor de lo que soy, acompañándome en cada instante, creyendo en que si puedo lograr lo que me propongo y dándome amor incondicional. Agradezco también al resto de mi familia, abuelos, tíos y tías, primos y primas por su gran apoyo y por compartir conmigo momentos de alegría que seguramente me ayudaron a lo largo de mi vida.

Agradezco también a mis compañeros de carrera y amigos Joaquín Vásquez y Leonardo Aravena por ayudarme y haber trabajado conmigo durante la carrera. También agradezco a mis amigos y compañeros de la Universidad de Concepción Felipe Castañeda, Marcos Cárcamo y Gonzalo Cartes por haber compartido momentos de estudio y entretenimiento conmigo.

Agradezco a mis amigos de la infancia y adolescencia David Zapata, Felipe Morrison, Adolfo Alcayaga, Vicente Ortega, Mauricio Hinojosa, Sebastian Ruíz, Gonzalo Soto, Felipe Toledo, Eric Muñoz, Roberto Palma, Nicolás Barahona, Francisco Navarro, Felipe Rodriguez y Lucas Fritz. Desde siempre los he querido y considerado muy buenas personas, muchas gracias por ser parte de mi vida y haberme ayudado en todo lo que han podido.

Resumen

Desde hace ya varios años la ciberseguridad ha sido un importante aspecto a considerar para las distintas empresas y organizaciones de todo el mundo, los ataques cibernéticos y las técnicas y métodos para protegerse de ellos están en constante evolución y a medida que nuevas tecnologías se vuelven disponibles, surgen nuevas formas de atacar. Una opción frente a este peligro es aprender de un ataque cibernético o incidente sospechoso, para poder identificar una amenaza y saber que medidas tomar para mitigarla. Para poder lograr esto son necesarios los Indicadores de Compromiso, esto es toda la información que evidencia actividad maliciosa o sospechosa dentro de un sistema y que a futuro ayuda a entender un ataque o comportamiento extraño dentro de una red.

En este trabajo se desarrolló una primera arquitectura y prototipo para un sistema de recopilación de Indicadores de Compromiso de una red local para sistemas Linux a través de un canal de comunicación seguro y un panel táctico para poder visualizar toda la información obtenida.

Se logró una plataforma capaz de recopilar Indicadores de Compromiso y reportes asociados al estado de estaciones de trabajo y servidores de una red local. Toda esta información está disponible para ser visualizada en un *dashboard*, el cual entrega el último estado registrado de las estaciones de trabajo y servidores de la red y la información recopilada.

Índice

Resumen **5**
Índice **6**
Lista de Tablas **8**
Lista de Figuras **9**
Glosario **11**
1. Introducción **14**
1.1 Objetivos 15
Objetivo General 15
Objetivo Específicos 15
1.2 Limitaciones 16
1.3 Metodología 16
2. Marco Teórico **17**
2.1 Importancia de la ciberseguridad 17
2.2 Indicadores de Compromiso 18
2.3 Herramientas utilizadas 20
3. Desarrollo **24**
3.1 Arquitectura propuesta 24
Estación de Trabajo / Servidor 25
Estación de Monitoreo 26
Comunicaciones 26
3.2 Prototipo implementado 27
Base de datos 28
Comunicación 30
Servidor gRPC 30
Mensajes enviados desde el servidor gRPC 31
Cliente gRPC 31
Mensajes enviados desde cliente gRPC 31
Monitoreo 32
Panel Táctico 38
4. Pruebas **45**
4.1 Canal Seguro 45
Análisis de vulnerabilidades con Nessus 46
Escaneo de puertos con Nmap 48
4.2 Comunicación gRPC 49
Conexión de un cliente al servidor gRPC 49
Reporte Global Periodico 49

Reporte por Compromiso....50
4.3 NAGIOS....51
Servicios....52
5. Conclusiones....56
5.1 Trabajo futuro....57
Redundancia en el hardware....57
Mejoramiento del panel táctico....58
Monitoreo y recopilación en Windows....58
7. Referencias....59

Lista de Tablas

Tabla 1. Resumen de pruebas realizadas al prototipo..... 55

Lista de Figuras

Figura 1. Esquema del funcionamiento de Nagios Remote Plugin Executor..... 22
Figura 2. Ejemplo regla de YARA usando strings y una lógica.....22
Figura 3. Arquitectura general del sistema de recopilación de Indicadores de Compromiso... 25
Figura 4. Elementos de un host.....25
Figura 5. Elementos de la estación de Monitoreo.....26
Figura 6. Placa Raspberry Pi 4 modelo B.....27
Figura 7. Escritorio Raspbian OS.....28
Figura 8. Diagrama MER de la base de datos implementada.....28
Figura 9. Servicios de gRPC definidos en archivo .proto.....30
Figura 10. Definición host umbral.....33
Figura 11. Template host NAGIOS.....33
Figura 12. Template servicio NAGIOS.....34
Figura 13. Definición de comandos de event handlers NAGIOS.....35
Figura 14. Event handler de NAGIOS escrito en Bash.....36
Figura 15. Dashboard Monitoreo y los paneles que contiene.....39
Figura 16. Panel Red Local.....39
Figura 17. Panel Detalle host umbral.....40
Figura 18. Panel Indicador.....40
Figura 19. Inspección del valor de un Indicador de Compromiso.....40
Figura 20. Detalle de un Indicador de Compromiso.....41
Figura 21. Texto copiado directamente de un Indicador de Compromiso.....41
Figura 22. Opciones para descargar un Indicador de Compromiso como CSV.....42
Figura 23. Panel Reporte.....42
Figura 24. Detalle de un Reporte.....43
Figura 25. Texto copiado directamente de un Reporte.....43
Figura 26. Opciones para descargar un Indicador de Compromiso como CSV.....44
Figura 27. Análisis de paquete encriptado utilizando Wireshark.....45
Figura 28. Paquetes con destino/origen el servidor gRPC en Wireshark.....45
Figura 29. Interfaz Nessus Essentials.....46
Figura 30. Lista de vulnerabilidades del servidor gRPC.....46
Figura 31. Puertos 22 y 80 abiertos.....47
Figura 32. Información sobre SSH.....47
Figura 33. Información sobre mDNS.....47
Figura 34. Configuración del firewall en servidor gRPC.....48
Figura 35. Vulnerabilidades del servidor después de configurar el firewall.....48

Figura 36. Comando para escanear puertos del servidor gRPC.....49
Figura 37. Resultados de Nmap.....49
Figura 38. Ingreso de contraseña de parte del cliente para conectarse al servidor gRPC.....49
Figura 39. Solicitud de reporte periodico Servidor gRPC.....50
Figura 40. Solicitud de reporte periodico Cliente gRPC.....50
Figura 41. Mensajes de Cliente gRPC al detectar compromiso con LOKI.....50
Figura 42. Mensajes de Servidor gRPC al recibir indicadores.....51
Figura 43. Servidor umbral de Ingeniería de Sistemas.....51
Figura 44. Estado OK de servicios del servidor umbral en NAGIOS.....52
Figura 45. Estado OK del servidor umbral y sus servicios en Panel Táctico.....52
Figura 46. Configuración de comando check_load en archivo de configuración de NRPE.52
Figura 47. Carga del CPU entrando a estado WARNING en NAGIOS.....53
Figura 48. Registro de cambio de estado de la carga del CPU en umbral y activación del event handler en logs de NAGIOS.....53
Figura 49. Cambio de estado a WARNING en Panel Táctico.....53
Figura 50. Carga del CPU entrando a estado CRITICAL en NAGIOS.....53
Figura 51. Cambio de estado a CRITICAL en logs de NAGIOS.....53
Figura 52. Cambio de estado a CRITICAL en Panel Táctico.....54
Figura 53. Servicio DHCP de umbral en estado CRITICAL en NAGIOS.....54
Figura 54. Cambio de estado a CRITICAL en logs de NAGIOS.....54
Figura 55. Cambio de estado a CRITICAL en Panel Táctico.....54
Figura 56. Ejemplo de caída y reconexión a una estación de monitoreo.....57

Glosario

Bash: Intérprete de comandos y un lenguaje de scripting utilizado principalmente en sistemas operativos basados en Unix y Linux. Es una de las shells más populares y ampliamente utilizadas en estos sistemas.

Certificado x.509: Estándar de seguridad ampliamente utilizado para establecer la autenticación y la seguridad en redes de comunicación, como Internet. Define el formato de certificados digitales, que son utilizados para verificar la identidad de entidades en línea, como sitios web, servidores de correo electrónico y usuarios.

Cron jobs: Los cron jobs son una función de programación en sistemas operativos tipo Unix (como Linux) que permite a los usuarios programar la ejecución automática de tareas o comandos en momentos específicos o a intervalos regulares.

CSV(Comma-Separated Values): Es un formato de archivo utilizado para almacenar y representar datos tabulares de manera sencilla y estructurada. En un archivo CSV, cada línea generalmente representa una fila de datos, y los valores de cada columna se separan por comas.

Dashboard: Representación visual y resumida de datos, métricas o información clave que permite a los usuarios obtener una vista rápida y fácil de entender sobre el estado de un sistema, proceso o conjunto de datos.

DHCP(Dynamic Host Configuration Protocol): Es un protocolo de red utilizado para asignar automáticamente direcciones IP y otros parámetros de configuración de red a dispositivos en una red, como computadoras, teléfonos, tablets y otros dispositivos conectados.

Framework: Conjunto de herramientas, librerías, estándares y pautas organizadas de manera coherente que proporcionan una estructura y base para desarrollar software de manera más rápida y eficiente.

Hash: Función matemática que toma una entrada y produce una cadena de caracteres de longitud fija, generalmente de longitud mucho menor que la entrada original. El resultado de esta función se llama "hash value" o "hash code". Un hash es como una especie de huella digital única para los datos de entrada.

HTTP/2: Versión actualizada y mejorada del protocolo de transferencia de hipertexto (HTTP), que es el protocolo utilizado para la transferencia de datos en la World Wide Web.

ICMP (Internet Control Message Protocol): Protocolo de la capa de red del conjunto de protocolos de Internet (TCP/IP). Es utilizado para enviar mensajes de control, diagnóstico y notificaciones de error entre dispositivos de red. ICMP proporciona un mecanismo para que los dispositivos informen y comuniquen errores y estados de red a otros dispositivos.

IPS(Intrusion Prevention System): Es una solución de seguridad informática diseñada para identificar y bloquear activamente intentos de intrusión y ataques en una red o sistema. A diferencia de los sistemas de detección de intrusiones (IDS), que solo alertan sobre posibles intrusiones, los sistemas de prevención de intrusiones toman medidas activas para bloquear o mitigar los ataques en tiempo real.

ISO/IEC 27001: Estándar internacional para la gestión de la seguridad de la información. El estándar proporciona un marco sistemático y amplio para establecer, implementar, mantener y mejorar un Sistema de Gestión de Seguridad de la Información (SGSI) dentro de una organización.

JSON(JavaScript Object Notation): JSON es un formato ligero y ampliamente utilizado para el intercambio de datos en aplicaciones informáticas.

Logs: Registros o registros de eventos que se generan por un sistema informático, aplicación o dispositivo para rastrear y registrar actividades y cambios significativos.

MAC(Media Access Control): Es un identificador único asignado a cada dispositivo de red, como una tarjeta de red o un adaptador de red inalámbrica. Esta dirección se utiliza en la capa de enlace de datos de un protocolo de red y se define en la capa física del modelo OSI.

MD5(Message Digest Algorithm 5): MD5 es un algoritmo de hash criptográfico ampliamente conocido y utilizado para convertir datos en un valor hash de 128 bits (16 bytes).

mDNS(Multicast Domain Name System): Protocolo de red que permite a los dispositivos en una red local (LAN) descubrir y comunicarse entre sí utilizando nombres de host o nombres de dominio sin la necesidad de un servidor de nombres de dominio (DNS) centralizado.

Phishing: ciberataque en el que los atacantes se hacen pasar por entidades legítimas, como organizaciones, empresas o individuos de confianza, para engañar a las víctimas y obtener información confidencial, como contraseñas, información financiera, datos personales o credenciales de acceso.

Plugin: Pieza de software que se puede agregar a una aplicación o sistema existente para agregar características, funcionalidades o capacidades adicionales. Son diseñados para extender o modificar la funcionalidad de un software sin necesidad de cambiar su código fuente principal.

Rasomware: El ransomware es un tipo de malware (software malicioso) que cifra los archivos de una computadora o sistema y luego exige un rescate (ransom) a cambio de proporcionar la clave de descifrado necesaria para restaurar el acceso a los archivos.

SHA1: Algoritmo de hash criptográfico que produce un valor hash de 160 bits (20 bytes).

SHA256: SHA-256 (Secure Hash Algorithm 256-bit) es un algoritmo de hash criptográfico que pertenece a la familia de algoritmos SHA-2. Al igual que otros algoritmos de hash, SHA-256 toma una entrada de datos y produce un valor hash único de 256 bits (32 bytes).

SNMP (Simple Network Management Protocol): Protocolo utilizado para la gestión y supervisión de dispositivos de red. Se basa en una arquitectura cliente-servidor, donde los dispositivos de red actúan como agentes SNMP y envían información a un sistema de gestión de red (NMS, por sus siglas en inglés) a través de mensajes SNMP.

SSH (Secure Shell): protocolo de red seguro que permite a los usuarios acceder y administrar de forma remota sistemas informáticos y transferir datos de manera segura.

TCP: Uno de los protocolos fundamentales en el conjunto de protocolos de Internet. Se basa en un modelo cliente-servidor, donde un cliente establece una conexión con un servidor para el intercambio de datos. Proporciona una comunicación confiable y ordenada, garantizando que los datos enviados sean recibidos correctamente y en el orden correcto.

TLS/SSL: Protocolos de seguridad diseñados para cifrar y asegurar la comunicación entre dos sistemas en una red, como un cliente y un servidor en Internet.

UDP(User Datagram Protocol): Protocolo de transporte de datos utilizado en redes de computadoras para transmitir información de manera rápida y eficiente, pero sin garantizar la entrega confiable ni el orden de los datos. UDP es uno de los dos protocolos principales de transporte en Internet, junto con TCP (Transmission Control Protocol).

1. Introducción

En el contexto actual, la ciberseguridad es un tema de gran relevancia y preocupación a nivel mundial. La creciente dependencia de la tecnología y la digitalización de procesos y sistemas en diferentes ámbitos, como el gubernamental, empresarial, financiero y personal, han llevado a un aumento en el número y la sofisticación de los ciberataques [1].

Además, la pandemia de COVID-19 ha llevado a un aumento en la digitalización de muchos procesos [2]. El aumento del trabajo remoto y el uso de herramientas de colaboración en línea también han generado nuevos desafíos de seguridad cibernética, como la exposición de información sensible a través de conexiones inseguras y el aumento de la vulnerabilidad a los ataques de *phishing* [3].

Ante este panorama, las organizaciones y los individuos están cada vez más conscientes de la importancia de la ciberseguridad y la necesidad de implementar medidas efectivas para protegerse contra los ciberataques. La industria de la ciberseguridad está en constante evolución, con nuevos avances y tecnologías emergentes para proteger los sistemas y datos contra las amenazas cibernéticas [4].

Es por esto que se han desarrollado sistemas de monitoreo de redes enfocados en la prevención de desastres informáticos como Zabbix[1] o Nagios[2] , este último es un sistema de monitoreo de redes el cual puede registrar el estado de servicios de red, servidores y hardware de red teniendo soporte para monitorear vía protocolos SNMP, TCP, ICMP y SSH.

Sin embargo, para afirmar que un ataque está ocurriendo, el estado de la red por sí solo no es información suficiente. Es necesario otro elemento que, de acuerdo al contexto de la red, nos pueda acercar más a la identificación de un posible ataque. Aquí entran en juego los Indicadores de Compromiso (IoC) como elemento adicional que nos ayuda a confirmar una intrusión o ataque dentro de la red. Estos indicadores son toda información relevante que, mediante el análisis de sus patrones, describe una actividad maliciosa de ciberseguridad. Estos indicadores pueden incluir una variedad de pistas, como actividades inusuales en los registros de eventos, cambios en la configuración de seguridad, tráfico de red inusual o desconocido, archivos o procesos desconocidos en el sistema, entre otras [5].

Organizaciones de todo tipo y tamaño pueden utilizar Indicadores de Compromiso como parte de su estrategia de seguridad, sin embargo las agencias que manejan información confidencial son particularmente sensibles a las amenazas cibernéticas como son agencias

[1] https://www.zabbix.com/
[2] https://www.nagios.org/

gubernamentales y de la defensa. Se vuelve necesaria entonces la incorporación de Indicadores de Compromiso en los sistemas de seguridad cibernética de estas agencias como evidencia adicional de un posible ataque o actividad sospechosa.

En un principio y como colaboración entre la Armada de Chile y la Universidad de Concepción, se estableció la necesidad de una plataforma de monitoreo del estado de seguridad de la plataforma TI de la institución naval.

Se propuso entonces una plataforma donde se incorporen Indicadores de Compromiso y sistemas de monitoreo de redes con la cual se pueda recolectar y almacenar toda la actividad dentro de la red al momento de detectar una anomalía.

Adicionalmente se propone la implementación de un panel táctico donde se pueda ver el estado actual de la red, disponibilidad de estaciones de trabajo, servidores y servicios y la información recolectada por el sistema de monitoreo, la cual posteriormente pudiese ser utilizada para afirmar que la causa de lo ocurrido fue un ataque cibernético.

1.1 Objetivos

Objetivo General

Desarrollar un sistema de recopilación periódica de Indicadores de Compromiso desde una red local, generación de una alerta de compromiso y despliegue de esta información a través de un Panel Táctico.

Objetivo Específicos

1. Levantamiento de requerimientos del sistema de monitoreo que cumpla con estándares de ciberseguridad para una red local para el sector de defensa.

2. Diseñar una arquitectura para el sistema de recopilación de Indicadores de Compromiso y monitoreo de una red local acorde a los requerimientos establecidos.

3. Implementar un prototipo como prueba de concepto que demuestre el funcionamiento del sistema principal de recopilación y almacenamiento de la información.

4. Implementar un Panel Táctico que donde pueda ser visualizada la información recopilada y el estado actual de la red local.

1.2 Limitaciones

Este trabajo contempla una primera versión de la arquitectura, un prototipo acorde a la arquitectura propuesta y un panel táctico donde se pueda visualizar la información recolectada.

1.3 Metodología

Este trabajo fue llevado a cabo por dos alumnos memoristas, Felipe Henriquez Ricart y Sergio Cifuentes Torres, guiados por los profesores Pedro Pinacho Davidson y Sergio Sobarzo Guzman.

Etapas

- Investigación
  Durante la primera etapa se investigó sobre Indicadores de Compromiso, que son y para qué sirven, y alternativas para poder identificarlos y recolectarlos dentro de una red local.

- Diseño
  Se propuso una arquitectura que estuviera acorde a los requerimientos levantados y lo investigado sobre Indicadores de Compromiso.

- Implementación
  Se implementó un primer prototipo como prueba de concepto que se asemejara a la arquitectura propuesta.

- Testing
  Para validar el funcionamiento del sistema se realizaron pruebas de conexión segura, correcta recolección y almacenamiento de la información y visualización en el Panel Táctico.

2. Marco Teórico

2.1 Importancia de la ciberseguridad

La ciberseguridad es de suma importancia en la actualidad debido al aumento de las amenazas cibernéticas y su impacto potencialmente devastador en organizaciones, individuos y la sociedad en general [6,7,8]. Aquí se presentan algunas razones clave que resaltan la importancia de la ciberseguridad:

- Protección de datos sensibles: La ciberseguridad garantiza la protección de datos confidenciales, como información financiera, datos personales, secretos comerciales y propiedad intelectual. Estos activos son valiosos tanto para las organizaciones como para los individuos, y su pérdida, robo o exposición puede tener consecuencias graves, como el robo de identidad, el fraude financiero o el daño a la reputación.

- Continuidad del negocio: Las brechas de seguridad y los ataques cibernéticos pueden interrumpir las operaciones comerciales normales, lo que puede resultar en pérdida de ingresos, daño a la reputación y costos significativos de recuperación. La ciberseguridad ayuda a minimizar estos riesgos y garantiza la continuidad del negocio al proteger los sistemas y datos críticos.

- Protección contra amenazas emergentes: Las amenazas cibernéticas evolucionan constantemente, con ataques cada vez más sofisticados y dirigidos. La ciberseguridad se centra en estar al tanto de las últimas tendencias y vulnerabilidades, y en implementar medidas de protección adecuadas para hacer frente a estas amenazas en constante evolución.

- Protección de la privacidad: La ciberseguridad contribuye a proteger la privacidad de los usuarios al prevenir el acceso no autorizado a sus datos personales, evitar el seguimiento no deseado y garantizar que sus interacciones sean seguras y confidenciales.

La ciberseguridad es un campo en constante evolución debido a la rápida evolución de las amenazas y técnicas maliciosas. Uno de los enfoques es el análisis de comportamiento y detección de anomalías. Esto implica monitorear constantemente el comportamiento normal de los sistemas y usuarios, y alertar sobre comportamientos sospechosos o anómalos que puedan indicar un ataque en curso.

2.2 Indicadores de Compromiso

Los Indicadores de Compromiso en ciberseguridad son señales que indican que un sistema o red ha sido comprometido por un atacante. Estos indicadores pueden incluir una variedad de pistas, como actividades inusuales en los registros de eventos, cambios en la configuración de seguridad, tráfico de red inusual o desconocido, archivos o procesos desconocidos en el sistema, actividad de procesos sospechosos o intentos de conexión a servidores maliciosos.

Estos indicadores son importantes porque pueden ayudar a los profesionales de ciberseguridad a detectar y responder a los ataques de manera temprana. Los equipos de seguridad pueden utilizar los Indicadores de Compromiso para analizar y mitigar los riesgos de seguridad, y tomar medidas preventivas para evitar futuros ataques.

Una manera de clasificar Indicadores de Compromiso según Cisco Systems es basada en la tríada CIA [9], separándolos en indicadores que comprometen confidencialidad, integridad o disponibilidad [10].

- **Confidencialidad**
  Se refiere a la protección de la información sensible contra el acceso no autorizado. Garantizar la confidencialidad significa que solo las personas o sistemas autorizados tienen permiso para acceder a la información confidencial. Esto se logra a través de medidas como la autenticación, autorización y cifrado de datos.

  Los Indicadores de Compromiso relacionados con la confidencialidad son evidencias en el sistema o red que demuestran que información confidencial o privada pueda ser filtrada o expuesta de manera promiscua. Ejemplos de Indicadores de Compromiso de la confidencialidad son:
  - Cambios en la telemetría del tráfico de la red (IP/dominios maliciosos conocidos)
  - Tráfico desconocido que se origina o termina en el dispositivo (SSH, Telnet, HTTP/HTTPS, entre otros)
  - Cambios de permisos en archivos del sistema
  - Creación de usuarios nuevos con privilegios de superusuario

- **Integridad**
  La integridad se centra en la precisión y la exactitud de la información. Significa que la información no debe ser modificada de manera no autorizada y que cualquier modificación legítima debe ser rastreable y verificable. La integridad se mantiene

mediante la implementación de mecanismos de control, como firmas digitales y registros de auditoría.

Los Indicadores de Compromiso relacionados con la integridad son evidencias de que la información del sistema o red pudo haber sido modificada y, por lo tanto, deja de ser confiable. Ejemplos de Indicadores de Compromiso de la integridad son:

- Detenciones frecuentes del software durante el funcionamiento normal del dispositivo
- Comportamiento extraño de una plataforma
- Anomalías en el sistema operativo o valores hash de paquetes
- Anomalías en las características de firma de certificados
- Binarios desconocidos instalados
- Procesos desconocidos activos

- **Disponibilidad**
  La disponibilidad se refiere a asegurarse de que la información y los sistemas estén disponibles y accesibles cuando sea necesario por los usuarios autorizados. Esto implica la implementación de medidas de prevención y recuperación de desastres para evitar interrupciones no planificadas.

  Los Indicadores de Compromiso relacionados con la disponibilidad son evidencias de que se ha perdido el acceso o hay riesgo de perderlo. Ejemplos de Indicadores de Compromiso de la disponibilidad son:
  - Uso anormalmente alto de CPU
  - Reinicio constante del sistema o de algún módulo
  - Altos volúmenes de tráfico
  - Cambios en permisos de super usuarios legítimos

Los Indicadores de Compromiso pueden ser recolectados a través de la monitorización continua de la red y los sistemas, así como a través del análisis de incidentes de seguridad previos. También pueden ser recopilados por herramientas de detección de amenazas, tales como sistemas de prevención de intrusiones (IPS), antivirus y sistemas de detección y respuesta de seguridad (EDR).

Al momento de detectar compromiso, específicamente en sistemas Linux, el CSIRT[3] recomienda recolectar la siguiente información asociada a ransomware [11]:

- Registro de logs del sistema
- Usuarios del sistema

[3] https://www.csirt.gob.cl/

- Cron Jobs o tareas programadas
- Historial de los comandos en la bash

Adicionalmente para sistemas Linux la siguiente información es relevante para en contrar comportamiento inusual en la red [12]:

- Conexiones de red activas
- Puertos abiertos
- Últimos archivos modificados

Todos estos datos se deben recopilar y ser analizados para detectar actividad maliciosa dentro de la red y tomar las medidas de prevención y respuesta necesarias.

2.3 Herramientas utilizadas

Para el desarrollo de este trabajo se necesitaron varias herramientas de distinto tipo. A continuación se describe cada una:

Python [13]
Lenguaje de programación interpretado de alto nivel, diseñado para ser fácil de leer y escribir. Se destaca por su sintaxis clara y concisa, lo que lo hace muy legible y accesible. Se utilizó como lenguaje principal para el funcionamiento del prototipo ya que a través del uso de distintas librerías relativas a otras herramientas logra conectar los distintos componentes del sistema.

SQLite3 [14]
SQLite3 es una biblioteca de software que implementa un sistema de gestión de bases de datos relacionales basado en archivos. A diferencia de otros sistemas de gestión de bases de datos como MySQL o PostgreSQL, SQLite3 se enfoca en ser una biblioteca ligera y autónoma.

gRPC [15]
Es un sistema de comunicación de código abierto desarrollado por Google. Es un framework de alto rendimiento que permite la comunicación y la llamada a procedimientos remotos. Las razones por las cuales se decidió usar gRPC son las siguientes:

- Sencillo, ya que son simples llamadas a funciones, y no requiere tanto código. En nuestro caso, a través de la inclusión de bibliotecas de gRPC en Python.

- Flexible, se puede trabajar con diversos lenguajes de programación, permitiendo la comunicación independiente del lenguaje de programación y el sistema operativo.

- Rendimiento, gRPC utiliza HTTP/2 para el envió de datos a través de la capa de transporte, reduciendo la latencia de red. Adicionalmente permite mantener conexiones persistentes entre el cliente y el servidor, lo que elimina la necesidad de abrir y cerrar una nueva conexión para cada solicitud.

- Seguridad, cuenta con autenticación integrada mediante el uso de SSL/TLS.

- Escalabilidad, gRPC está diseñado para que sea escalable y tolerante a fallos, permitiendo su ejecución en entornos de alta disponibilidad.

CFSSL [16]
Herramienta de código abierto desarrollada por Cloudflare[4] utilizada para la creación y autenticación de certificados x.509 y SSL/TLS necesarios para una comunicación de cliente-servidor segura.

NAGIOS Core [17]
Nagios Core es una herramienta de monitoreo de código abierto basado en el uso de plugins para supervisar la infraestructura de TI, incluidos servidores, redes, servicios y aplicaciones. Proporciona una plataforma flexible y extensible para monitorear y alertar sobre el estado y el rendimiento de los componentes de una red. Adicionalmente ofrece la capacidad de configurar respuestas a ciertos estados de hosts y servicios para intentar volver al estado normal de la red.

NRPE [18]
Nagios Remote Plugin Executor es una extensión de NAGIOS la cual permite ejecutar plugins en hosts de manera remota para poder obtener información sobre parámetros del sistema y servicios.

En la figura 1 podemos ver como funciona NRPE ejecutando los comandos check_disk y check_load:

[4] https://www.cloudflare.com/

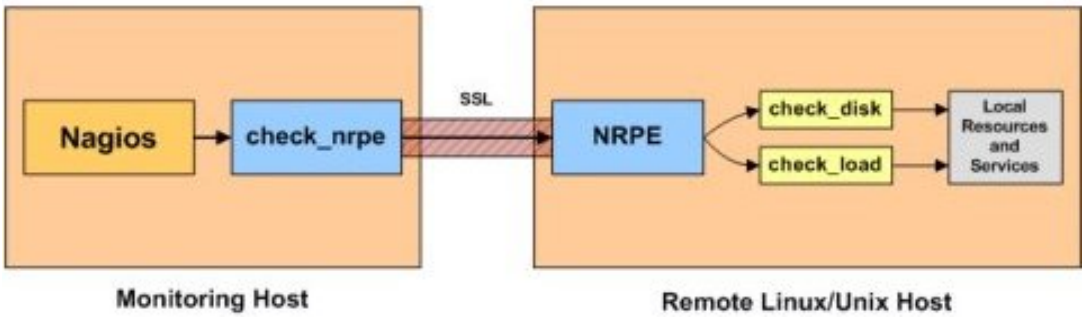


Figura 1. Esquema del funcionamiento de Nagios Remote Plugin Executor.

YARA

Yet Another Recursive Acronym es una herramienta de código abierto usada para la creación de reglas para la identificación de malware o patrones en archivos y procesos. Una de las principales características de YARA es el escaneo de Indicadores de Compromiso usando reglas de YARA para filtrar hashes de archivos, claves de registro o strings específicos asociados con archivos maliciosos.

En la figura 2 podemos ver un ejemplo de regla de YARA [19] usando strings a buscar dentro de archivos y una lógica.

```
rule Example
{
    strings:
        $a = "text1"
        $b = "text2"
        $c = "text3"
        $d = "text4"

    condition:
        ($a or $b) and ($c or $d)
}
```

Figura 2. Ejemplo regla de YARA usando strings y una lógica.

LOKI [20]
Es una herramienta de análisis de Indicadores de Compromiso, que funciona bajo cuatro métodos de detección, que se detallan a continuación:

- Nombre del archivo IoC (Coincidencia de expresiones regulares en la ruta o nombre completo del archivo).
- Comprobación de las reglas con YARA.
- Comparación de hashes maliciosos conocidos (MD5, SHA1 y SHA256), mediante el escaneo de archivos.
- Compara los puntos finales de conexión de los procesos con las C2 de IoC.

Grafana [21]
Grafana es una plataforma de visualización y análisis de datos de código abierto. Proporciona herramientas para crear paneles interactivos y gráficos personalizados que permiten visualizar y analizar datos de diferentes fuentes en tiempo real.

Wireshark [22]
Wireshark es una herramienta de código abierto que se utiliza para capturar y examinar datos que se transmiten a través de una red de manera promiscua, lo que incluye el análisis de paquetes de datos, mensajes, protocolos y flujos de información.

Nessus Essentials [23]
Versión gratuita del escáner de vulnerabilidades Nessus. Es una herramienta ampliamente utilizada para identificar vulnerabilidades de seguridad en redes, sistemas y aplicaciones. Ayuda a detectar posibles debilidades que los atacantes podrían aprovechar.

3. Desarrollo

Para lograr el sistema de recopilación de Indicadores de Compromiso y monitoreo propuesto se necesita garantizar la confidencialidad e integridad de la información al momento de ser recopilada, esto se logra a través de un canal seguro construido con la herramienta gRPC gracias a que comunica la estación de monitoreo y los distintos hosts de la red local simultáneamente a través del protocolo HTTP/2, el cual permite la multiplexación de los mensajes recibidos y enviados en las distintas conexiones que se generen entre los hosts de la red y el servidor gRPC. Adicionalmente se garantiza la confidencialidad de la información ya que gRPC cuenta con autenticación y cifrado a través de certificados SSL/TLS.

Para el monitoreo de la disponibilidad y respuesta de los hosts, servidores y servicios se utilizó NAGIOS Core, el cual es un software de monitorización y diagnóstico de redes de código abierto que proporciona gran versatilidad para consultar prácticamente cualquier parámetro de un sistema a través de plugins hechos por la comunidad.

Toda la información recopilada es almacenada en una base de datos local para la cual se utilizó el sistema de gestión SQLite3. Por el hecho de no necesitar conexión a internet y funcionar directamente en memoria del dispositivo, las transacciones a la base de datos ocurren en tiempo real, por lo que se garantiza la disponibilidad de la información y en caso de ser necesario cuenta con la capacidad de crear copias de seguridad y recuperación de los datos.

Adicionalmente debe poder ser visualizado el estado actual de la red y la información recopilada para su posterior análisis. Para esto se utilizó la herramienta Grafana con la cual se creó un dashboard y paneles en los cuales se pueden ver los detalles de hosts y servidores, y detalles de los reportes e indicadores de compromiso recolectados.

A continuación se presenta una primera propuesta de la arquitectura y la implementación de un prototipo correspondiente a esta.

3.1 Arquitectura propuesta

Como podemos ver en la figura 3, se tiene una estación de monitoreo dentro de una red local. Esta estación se comunica constantemente con las distintas estaciones de trabajo y servidores correspondientes a la red para solicitar información relevante al momento de ser necesario.

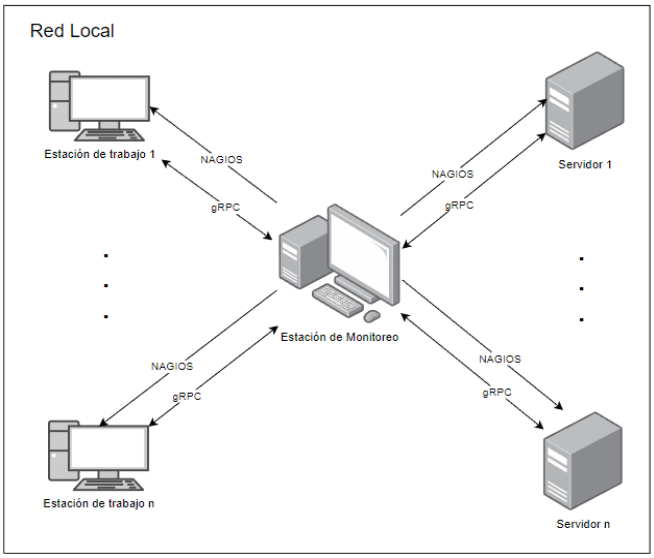


Figura 3. Arquitectura general del sistema de recopilación de Indicadores de Compromiso.

Estación de Trabajo / Servidor

Dentro de cada estación de trabajo y servidor se encuentran dos agentes distintos.

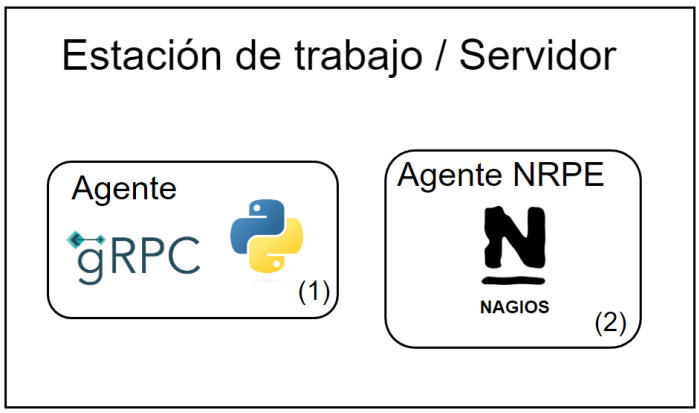


Figura 4. Elementos de un host.

- El primero es un agente gRPC (1) que se comunica como cliente al servidor gRPC en la estación de monitoreo. Este recibe solicitudes de reporte por parte del servidor gRPC y envía la información correspondiente.

- El segundo es un agente Nagios Remote Plugin Executor (NRPE) (2) el cual recibe la orden de chequear distintas características del sistema e informar al servidor de NAGIOS sobre su estado actual y, en caso de ser un servidor, el estado de sus servicios.

Estación de Monitoreo

La estación de monitoreo que se encuentra en la Raspberry Pi contiene varios elementos:

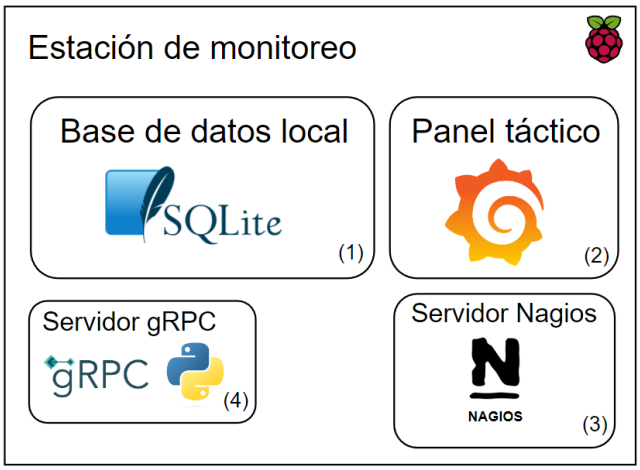


Figura 5. Elementos de la estación de Monitoreo.

- Base de datos local (1) donde se almacena toda la información recolectada de la red: Reportes, Indicadores, Cambios de estado de hosts y servicios.

- Panel táctico (2) donde se visualiza el estado actual de la red, reportes e indicadores de compromiso.

- Servidor de NAGIOS (3) el cual se comunica con el agente NRPE para ejecutar comandos remotamente y obtener el estado del sistema de un host y sus servicios.

- Servidor gRPC (4) que se comunica con los agentes gRPC para pedir reportes a los hosts de la red.

Comunicaciones

Existen dos comunicaciones distintas entre las Estaciones de trabajo / Servidores y la Estación de Monitoreo, esto es porque las herramientas NAGIOS y gRPC funcionan en base a una relación Servidor-Cliente, teniendo el servidor de cada herramienta en la estación de monitoreo y los agentes/clientes dentro de cada estación de trabajo y servidores de la red.

Primero está el servidor gRPC dentro de la estación de monitoreo que se comunica a través de una conexión TCP que utiliza el puerto 50051 con los agentes gRPC en cada estación de

trabajo y servidor. Esta es la comunicación principal entre la estación de monitoreo y las estaciones de trabajo y servidores, a través de la cual se solicitan reportes y se recopila el estado actual de la red.

Después está la comunicación entre el servidor de NAGIOS, que se encuentra dentro de la estación de monitoreo, y los agentes NRPE que se encuentran en las estaciones de trabajo y servidores. Este utiliza el puerto 5666 a través del protocolo de comunicación TCP.

3.2 Prototipo implementado

Para la implementación de los distintos componentes de la estación de monitoreo, donde se encuentra el servidor gRPC, servidor NAGIOS y panel táctico, se utilizó una Raspberry Pi 4 modelo B, la cual pertenece a la serie de computadoras Raspberry Pi, de placa única de bajo costo y tamaño reducido.

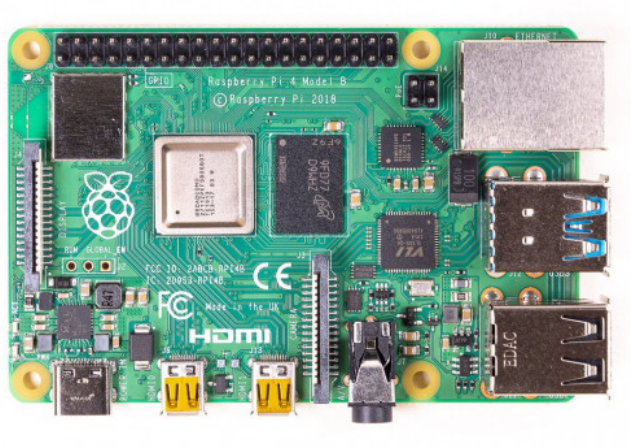


Figura 6. Placa Raspberry Pi 4 modelo B.

Este modelo cuenta con las siguientes especificaciones:

- Procesador Cortex-A72 quad-core 64-bit
- Memoria RAM LPDDR4 4GB
- Puerto Ethernet Gigabit
- LAN inalámbrica 802.11 b/g/n/ac
- Bluetooth 5.0
- 2 puertos micro-HDMI
- 2 puertos USB 2.0
- 2 puertos USB 3.0

Adicionalmente se utilizó una memoria SanDisk Ultra de 64GB donde se decidió instalar el sistema operativo Raspbian OS, diseñado específicamente para las placas Raspberry Pi, debido a que está basado en linux y es relativamente sencillo de usar y navegar al contar con interfaz de escritorio.

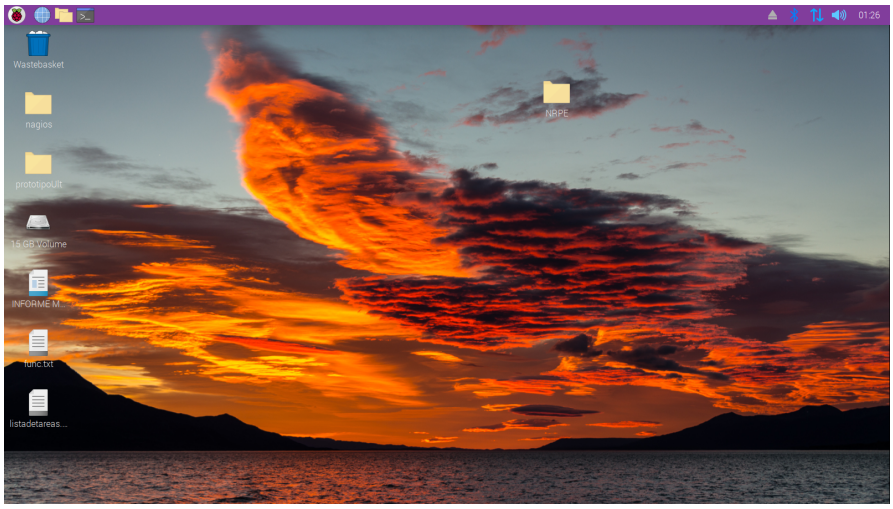


Figura 7. Escritorio Raspbian OS.

Base de datos

Para la base de datos se decidió utilizar SQLite, un motor de base de datos relacional que funciona de manera totalmente local dentro del sistema y basándose en el lenguaje SQL.

En la figura 8 se muestra el modelo de la base de datos implementada.

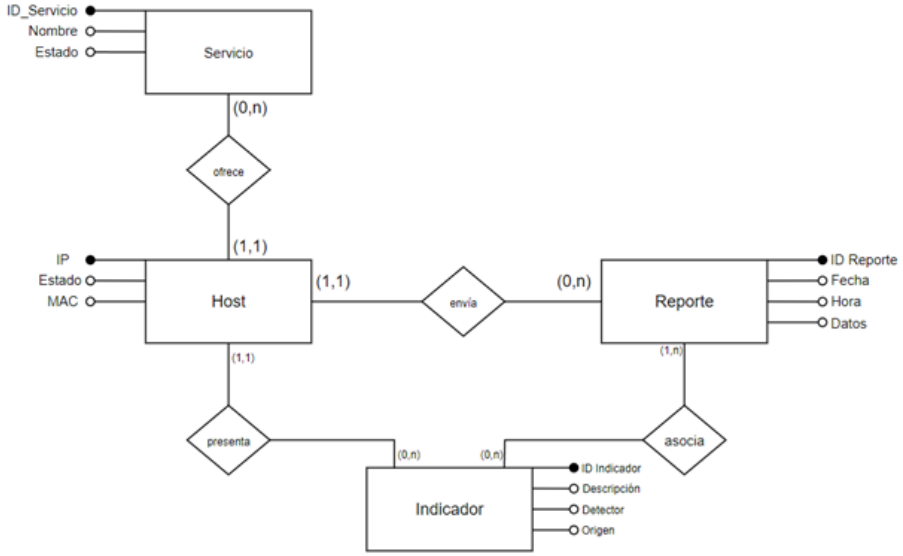


Figura 8. Diagrama MER de la base de datos implementada.

Entidades:

- **Host**: Son todas las estaciones de trabajo y servidores dentro de la red local. Contienen: IP, Estado y MAC.

- **Servicio**: Corresponde a un servicio que ofrezca un host que se esté monitoreando a través de NAGIOS.
  Contienen: ID_Servicio, Nombre y Estado.

- **Reporte**: Son los reportes recolectados al momento de encontrar algún indicador de compromiso o periódicamente dentro de la red con un tiempo establecido.
  Contienen: ID_Reporte, Fecha, Hora y Datos, este último corresponde al reporte en sí.

- **Indicador**: Indicadores de Compromiso encontrados en los hosts durante el monitoreo ya sea con LOKI o NAGIOS.
  Contienen: ID_Indicador, Descripción, Detector y Origen.

Relaciones:

- **Ofrece:** Un Host puede ofrecer una cantidad de 0 a n Servicios. En caso de ser igual a 0 se considera una estación de trabajo, si es distinto de 0 entonces es un servidor con n cantidad de servicios. Un Servicio puede ser ofrecido por solamente un Host.

- **Envía:** Un Host puede haber enviado una cantidad de 0 a n Reportes durante el monitoreo. Un Reporte puede ser enviado por solamente un Host.

- **Presenta:** Un Host puede presentar una cantidad de 0 a n Indicadores durante el monitoreo. Un Indicador se puede presentar en solamente un Host.

- **Asocia:** Un Reporte puede estar asociado a una cantidad de 0 a n Indicadores, esto es porque hay solicitudes de Reporte periódicas sin presencia de indicadores. Un Indicador puede estar asociado a una cantidad de 1 a n Reportes.

Comunicación

Lo que comunica y conecta los distintos componentes del sistema es gRPC (google Remote Procedure Call), lo cual a través del uso de HTTP/2 y autenticación de certificados SSL/TLS creados con la herramienta CFSSL crea un canal de comunicación seguro y que tiene la capacidad de mantener múltiples conexiones simultáneamente.

Existe una relación de Servidor-Cliente con una comunicación bidireccional donde cada uno puede enviar y responder mensajes de una manera predeterminada en archivos de configuración, los cuales definen la estructura de los mensajes y la información que contiene cada uno de ellos. Se separan en mensajes y servicios con los cuales el servidor gRPC puede comunicarse con los hosts. Los servicios definidos en un archivo .proto, que se encuentra en el servidor gRPC y el cliente gRPC, toman los mensajes del servidor o cliente y generan la respuesta correspondiente.

En la figura 9 podemos ver como está definido cada servicio, el mensaje que recibe y el que retorna como respuesta.

```
service Communication {

        rpc SubmitReport (ReportMessage) returns (ServerMessage); // Envia un reporte al servi
        rpc BidirectionalCommunication (stream ClientMessage) returns (stream ServerMessage);
  rpc IndicatorReport (IndicatorMessage) returns (ServerMessage); // Envia un reporte de indic
  rpc SaveIndicatorReport (ReportXIndicator) returns (ServerMessage); // Envia ids para relaci
  rpc ServerComprobationMD5 (ComprobationMD5) returns (ServerMessage); // Envia un md5 para co
```

Figura 9. Servicios de gRPC definidos en archivo .proto.

Servidor gRPC

El servidor gRPC se encuentra en la Raspberry Pi, la cual se conecta a la red local y se hace disponible para todos los hosts y servidores.

Sus funciones son:

- Recibir peticiones o alertas desde los componentes de monitoreo.
- Solicitar reportes a los clientes conectados donde se encuentre compromiso.
- Solicitar reportes a los clientes periódicamente.
- Almacenar la información recolectada en la base de datos.

Mensajes enviados desde el servidor gRPC

ServerMessage: Mensaje enviado desde el servidor gRPC como respuesta al ClientMessage recibido del cliente gRPC.

Contiene:

- message: Puede ser “Ok” en caso de que no pase nada, o “Enviame tu reporte” en caso de que se detecte uno o más indicadores de compromiso.

ReportXIndicator: Mensaje por parte del servidor gRPC que confirma la asociación de reportes con indicadores detectados.

Contiene:

- idReport: ID del reporte en la base de datos local.
- idIndicator: ID del indicador en la base de datos local.

Cliente gRPC

Los clientes gRPC son agentes en los hosts dentro de la red que serán monitoreados. Para poder conectarse al servidor gRPC, al cliente se le solicita una contraseña la cual se usa para desencriptar los certificados asociados al host y finalmente crear la conexión.

Cada un minuto ocurre un intercambio de mensajes entre el servidor y los clientes. Cada cliente tiene una de las siguientes opciones para enviar al servidor:

- Comunicar que no hay problemas.
- Alertar que LOKI encontró compromiso.
- Enviar Reporte e Indicadores asociados.

Mensajes enviados desde cliente gRPC

ComprobationMD5: Mensaje enviado desde un host al momento de crear la conexión para verificar si el hash del archivo contenido es distinto al original.

Contiene:

- ip: IP del host que envió el mensaje al servidor.
- md5: MD5 del archivo a comprobar.
- file: archivo.

ClientMessage: Mensaje enviado desde un host al servidor gRPC cada minuto que alerta si se encontró algún problema en el sistema.

Contiene:

- ip: IP del host que envió el mensaje al servidor.
- message: Sólo puede ser una de dos opciones: “Tengo un problema” o “No pasa nada”.

ReportMessage: Mensaje enviado desde el host hacia el servidor que contiene el reporte solicitado.

Contiene:

- ip: IP del host que envió el mensaje al servidor.
- json: String con formato json que contiene la información del reporte enviado por el host.

IndicatorMessage: Mensaje enviado desde el host hacia el servidor que contiene un indicador encontrado.

Contiene:

- ip: IP del host que envió el mensaje al servidor.
- timestamp: Fecha y hora cuando se encontró el indicador.
- indicator: Descripción del indicador encontrado entregada por LOKI.
- detector: Indica si fue un indicador detectado por LOKI o si falló la comprobación del hash MD5.

Monitoreo

Con la herramienta NAGIOS los hosts y servicios a monitorear se definen y configuran principalmente a través de archivos de configuración con la extensión .cfg, NAGIOS trae algunos por defecto que son esenciales para su funcionamiento y se pueden crear archivos adicionales de configuración simplemente añadiendo su ruta al archivo nagios.cfg.

```
58  # UMBRAL ----------------------
59
60
61  # Host
62
63  define host {
64
65      use                 host-prueba
66      host_name           umbral
67      alias               umbral
68      address             152.74.52.8
69      contacts            Sergio-Cifuentes
70      contact_groups      memoristas
71      event_handler       actualiza_estado_host
72
73  }
74
75  # Servicios
76
77  define service{
78
79      use                 servicio-prueba
80      host_name           umbral
81      service_description CPU_load
82      check_command       check_nrpe!check_load
83      event_handler       actualiza_estado_servicio
84  }
85
86  define service{
87
88      use                 servicio-prueba
89      host_name           umbral
90      service_description DHCP
91      check_command       check_dhcp
92      event_handler       actualiza_estado_servicio
93
94  }
95
```

Figura 10. Definición host umbral.

Se pueden definir también plantillas de host y servicios para facilitar la definición de un host nuevo.

```
define host {

    name                            host-prueba             ;
    check_period                    24x7                    ;
    check_interval                  2                       ;
    retry_interval                  1                       ;
    max_check_attempts              3                       ;
    check_command                   check-host-alive        ;
    notification_interval           120                     ;
    notification_options            d,u,r                   ;
    ;contact_groups                                        ;
    notifications_enabled           1                       ;
    event_handler_enabled           1                       ;
    flap_detection_enabled          1                       ;
    process_perf_data               1                       ;
    retain_status_information       1                       ;
    retain_nonstatus_information    1                       ;
    notification_period             24x7                    ;
    register                        0                       ;

}
```

Figura 11. Template host NAGIOS.

```
define service{

    name                            servicio-prueba         ;
    active_checks_enabled           1                       ;
    passive_checks_enabled          1                       ;
    parallelize_check               1                       ;
    obsess_over_service             1                       ;
    check_freshness                 0                       ;
    notifications_enabled           1                       ;
    event_handler_enabled           1                       ;
    flap_detection_enabled          1                       ;
    process_perf_data               1                       ;
    retain_status_information       1                       ;
    retain_nonstatus_information    1                       ;
    is_volatile                     1                       ;
    check_period                    24x7                    ;
    max_check_attempts              3                       ;
    check_interval                  2                       ;
    retry_interval                  1                       ;
    contact_groups                  memoristas              ;
    notification_options            w,c,r                   ;
    notification_interval           0                       ;
    notification_period             24x7                    ;
    flap_detection_options          c                       ;
    register                        0                       ;

}
```

Figura 12. Template servicio NAGIOS.

Hay que tener en cuenta que para cada host y servicio hay diferentes estados y tipos de estados a los cuales se llega cuando se cumplen diferentes condiciones.

Para los hosts, la variable $HOSTSTATE$ puede tomar los estados: UP, cuando un host responde al PING que realiza NAGIOS para encontrarlo dentro de la red, DOWN es cuando este no responde al PING y finalmente UNREACHABLE, lo que significa que NAGIOS pierde conexión a la red a la que pertenecen los hosts, es decir, son inalcanzables. Los servicios usan la variable $SERVICESTATE$ que toma los estados OK, WARNING y CRITICAL, estos son definidos por medio de rangos de acuerdo con el parámetro que se esté monitoreando.

Adicionalmente tenemos los tipos de estado que se registran en la variable $STATETYPE$ y pueden ser SOFT o HARD. Se puede llegar a ambos tipos de estado de varias maneras:

Estado SOFT

Se llega a un estado de tipo SOFT cuando:

- El chequeo de un servicio o host resulta en no-UP para host o no-OK para servicio y no se ha llegado todavía al número máximo de intentos, esto se considera un error SOFT.
- Un host o servicio se recupera de un error SOFT, es decir, que no se llegó al número máximo de intentos del chequeo y se logró el UP o OK, entonces se llega a un estado de tipo SOFT.

Cuando se llega a un estado de tipo SOFT ocurre lo siguiente:

- El estado SOFT se registra en el log de NAGIOS.
- Se disparan los event handlers del host o servicio.

Estado HARD

Se llega a un estado de tipo HARD cuando:

- Se termina el último intento de chequeo de un host o servicio y se mantiene el estado no-UP o no-OK, entonces se pasa a un estado de tipo HARD.
- El estado de un servicio pasa de un estado WARNING a CRITICAL o viceversa.
- El chequeo de un servicio resulta en no-OK y el host se encuentra en estado DOWN o UNREACHABLE.
- Un host se recupera de un estado de tipo HARD.
- Se recibe un chequeo pasivo a un host.

Cuando se llega a un estado de tipo HARD ocurre lo siguiente:

- El estado HARD se registra en el log de NAGIOS.
- Se disparan los event handlers del host o servicio.
- Los contactos asociados al host son notificados.

Para cada host y servicio NAGIOS cuenta con manejadores de eventos(event handlers), a los cuales se les puede pasar límites dentro de los parámetros que monitorean para que estos se disparen y ejecuten una acción dentro del sistema.

```
#Comandos

define command{

    command_name    actualiza_estado_host
    command_line    /usr/local/nagios/libexec/eventhandlers/actualiza_estado_host.sh $HOSTADDRESS$ $HOSTSTATE$

}


define command{

    command_name    actualiza_estado_servicio
    command_line    /usr/local/nagios/libexec/eventhandlers/actualiza_estado_servicio.sh $HOSTADDRESS$ $SERVICEDESC$ $SERVICESTATE$

}
```

Figura 13. Definición de comandos de event handlers NAGIOS.

```
case "$3" in

    OK)

        python3 /usr/local/nagios/libexec/eventhandlers/cambiaEstadoServicio.py "$1" "$2" 1

        #esac
        ;;


    WARNING)

        python3 /usr/local/nagios/libexec/eventhandlers/cambiaEstadoServicio.py "$1" "$2" 2

        python3 /home/pi1/Desktop/prototipoUlt/MonitoreoIoC-main/communicationNagios.py "$1" "$2" 2
        ;;

    CRITICAL)

        python3 /usr/local/nagios/libexec/eventhandlers/cambiaEstadoServicio.py "$1" "$2" 3

        python3 /home/pi1/Desktop/prototipoUlt/MonitoreoIoC-main/communicationNagios.py "$1" "$2" 3

        #   ;;
        #esac
        ;;
```

Figura 14. Event handler de NAGIOS escrito en Bash.

Lo importante aquí es que, el llegar o recuperarse de los estados de tipo SOFT y HARD disparan los event handlers. Es decir, podemos tomar acciones al momento de encontrarnos con problemas en la red.

De esta manera, cuando un parámetro de un servicio que se monitoree pase a un estado de OK, WARNING o CRITICAL, se refleja en la base de datos en tiempo real haciendo uso de los event handlers mencionados anteriormente. Estos toman el IP del host al que pertenecen ($HOSTADDRESS$), la descripción del servicio ($SERVICEDESC$) o parámetro y su estado ($SERVICESTATE$) para evaluar qué hacer en cada caso. En caso de pasar a un estado OK o WARNING, simplemente se actualizará el estado en la base de datos. Si por cualquier motivo se llegó a un estado CRITICAL, entonces adicionalmente a la actualización del estado se le pedirá un reporte al host respectivo.

Información recolectada de los hosts

Los reportes recolectados durante el monitoreo contienen información del sistema la cual se obtiene mediante comandos de Linux, se detallan a continuación

- **export LC_ALL=C**: Salida de la consola en inglés.
- **netstat**: Lista de todas las conexiones activas en el sistema.
    - **netstat -antup | grep 'ESTABLISHED'**
      Conexiones TCP y UDP activas en el estado ESTABLISHED.
    - **netstat -antup | grep 'LISTEN'**
      Conexiones TCP y UDP activas en el estado LISTEN.
    - **netstat -antup | grep 'TIME_WAIT'**
      Conexiones TCP y UDP activas en el estado TIME_WAIT.
    - **netstat -tulpna**
      Lista de todos los puertos abiertos en el sistema con los procesos que los están utilizando.
- **ss**: Información detallada de las conexiones de red activas.
    - **ss | grep ssh**
      Conexiones relacionadas con SSH.
- **/etc/passwd**: Archivo que contiene la información de los usuarios registrados en el sistema.
    - **cut -d: -f1 /etc/passwd**
      Listado de los nombres de usuario registrados en el sistema.
- **/var/log/auth.log**: Archivo que registra todas las actividades de autenticación, con esto se pueden monitorear posibles intentos de intrusión.
    - **cat /var/log/auth.log**
      Archivo /var/log/auth.log en salida estándar.
- **history:** Listado de los últimos comandos utilizados.
- **last:** Listado de los últimos inicios de sesión.
- **ps:** Lista de procesos que se están ejecutando.
    - **ps auxf**
      Listado jerárquico de todos los procesos en ejecución, con información detallada de cada uno, con su relación padre-hijo.
- **crontab**: Programar tareas para su ejecución automática en un momento específico.
    - **crontab -l**
      Lista todas las tareas programadas del usuario actual.
- **/var/log/syslog**: Archivo que almacena mensajes y eventos del sistema, con él se pueden llegar a identificar actividades sospechosas o maliciosas, ya que contiene mucha información sobre el equipo y lo que sucede en él.
    - **cat /var/log/syslog**
      Archivo /var/log/syslog en salida estándar.

Toda la información recolectada se estructura como un diccionario desde el agente gRPC en el host, para posteriormente ser enviada al servidor gRPC y ser guardado en la base de datos.

La solicitud de reportes de parte del servidor gRPC a los agentes en los hosts en la red se establece en los siguientes casos:

- Detección de Indicador(es) de Compromiso en un host por LOKI.
- Anomalía en el comportamiento de un servicio de un host monitoreado por NAGIOS.
- Periódicamente con un tiempo establecido.

En los primeros 2 casos solo se le solicita reporte en el host donde se detectó compromiso o comportamiento anómalo, mientras que en el último es de forma global a todos los hosts en la red.

Panel Táctico

Para el desarrollo del Panel Táctico se utilizó Grafana, plataforma de análisis y monitoreo de código abierto basada en plugins que permite la creación de dashboards con paneles personalizados para visualizar la información integrada de la base de datos. La navegación se basa en links a la ubicación de un panel localmente, estos links pueden ser definidos como hipertexto o simples botones dentro de la interfaz y se puede elegir entre abrir una pestaña nueva para visualizar el siguiente panel o simplemente cargarlo en la pestaña actual.

Se tuvo que añadir un plugin adicional para poder enlazar y realizar las consultas a la Base de Datos creada con SQLite3 [24].

Como primera versión se crearon paneles para poder ver el estado actual de la red y la información almacenada en la base de datos.

Dashboard Monitoreo

El dashboard con nombre Monitoreo es la cual contiene los paneles donde se inicia la navegación. Los Paneles contenidos en ella están conectados a través de links que llevan a una ubicación en el sistema local de archivos donde se encuentra el siguiente panel.

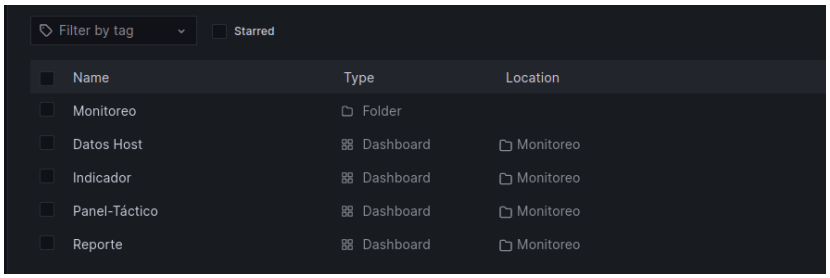


Figura 15. Dashboard Monitoreo y los paneles que contiene.

Panel red local

Se visualiza la disponibilidad de cada host de la red. Tenemos la lista de hosts donde se muestra su IP, nombre y estado actual, el cual puede ser OK o No se encuentra. Se puede hacer click en un IP para seleccionar el host e ir a sus detalles.

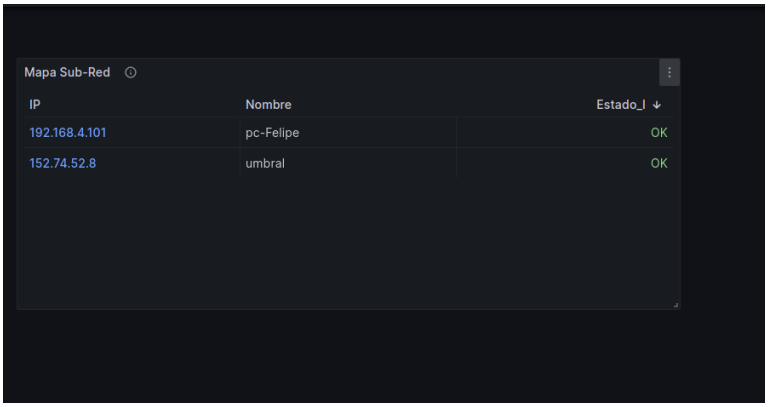


Figura 16. Panel Red Local.

Panel detalle host

Se visualiza el estado del host, los servicios monitoreados del host y una lista de Reportes recibidos e Indicadores de Compromiso encontrados en el host. Se puede seleccionar un Reporte o Indicador de Compromiso de las listas para ver su detalle.

Figura 17. Panel Detalle host umbral.

Panel indicador

Se visualiza el detalle de un Indicador de Compromiso, Origen y Detector. Se puede inspeccionar el valor de Datos para ver el detalle con formato JSON y poder exportarlo como CSV.

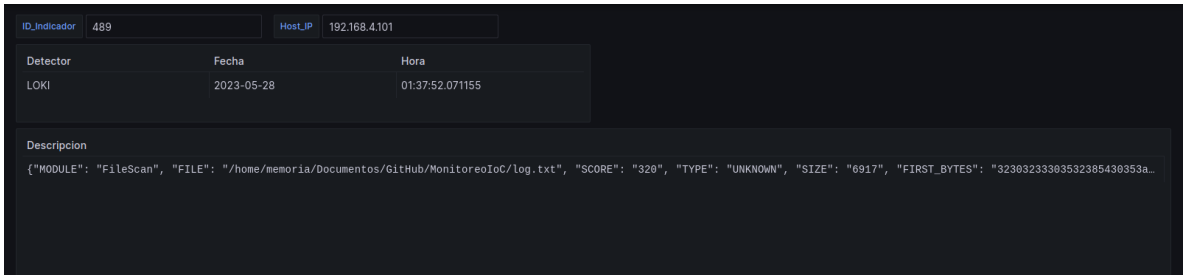


Figura 18. Panel Indicador.

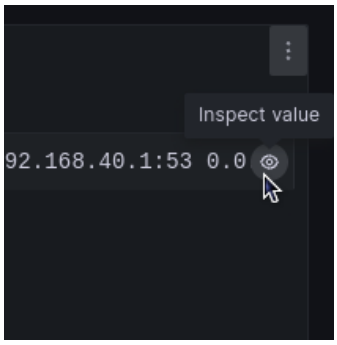


Figura 19. Inspección del valor de un Indicador de Compromiso.

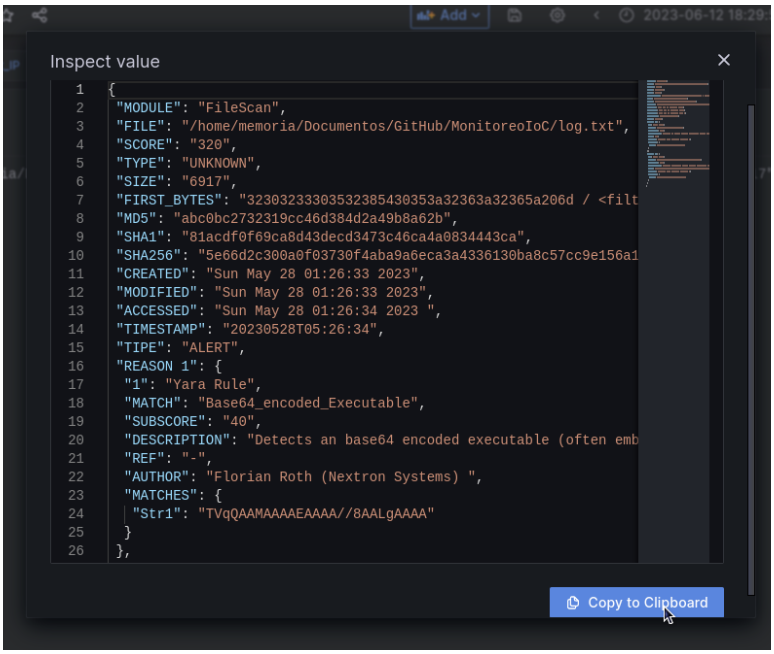


Figura 20. Detalle de un Indicador de Compromiso.

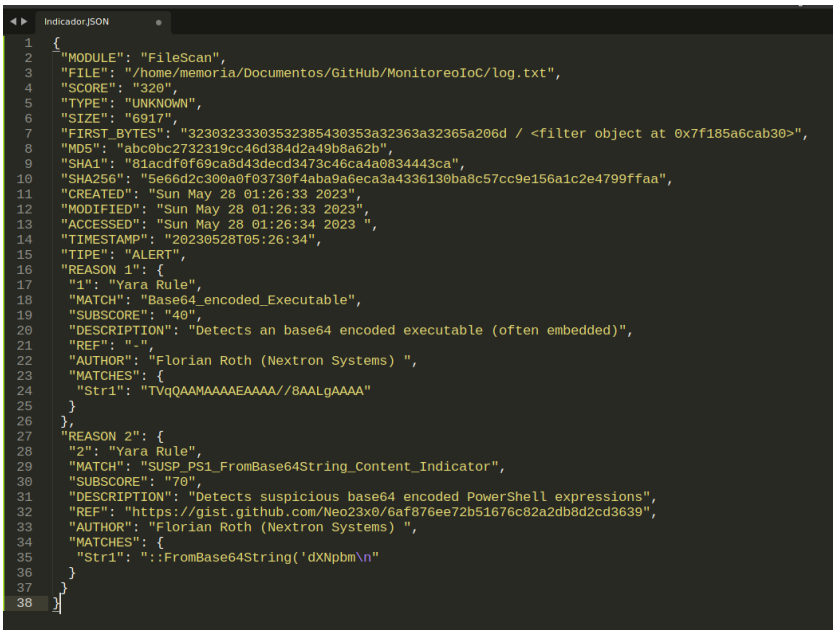


Figura 21. Texto copiado directamente de un Indicador de Compromiso.

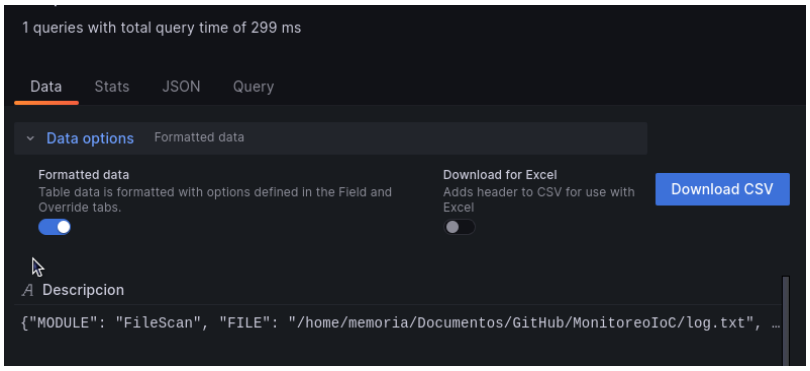


Figura 22. Opciones para descargar un Indicador de Compromiso como CSV.

Panel reporte

Se visualiza el detalle de un Reporte, Fecha, Hora y la información extraída de un host. Al igual que con los Indicadores se puede inspeccionar el valor de Datos para ver el detalle con formato JSON y poder exportarlo como CSV.

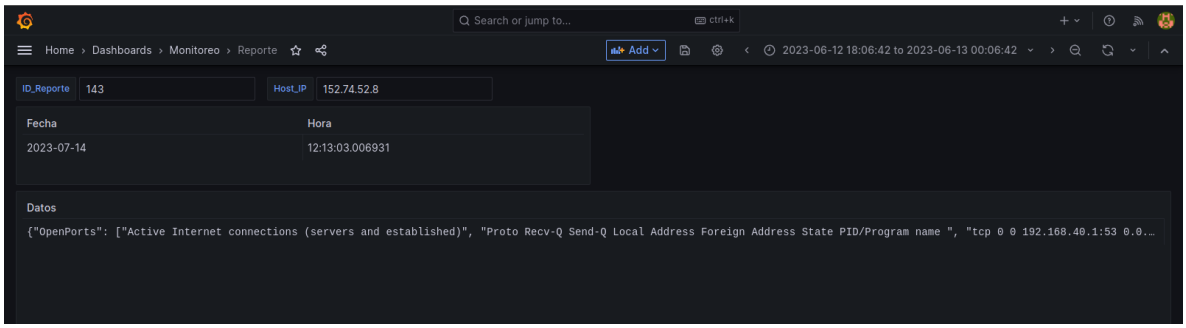


Figura 23. Panel Reporte.

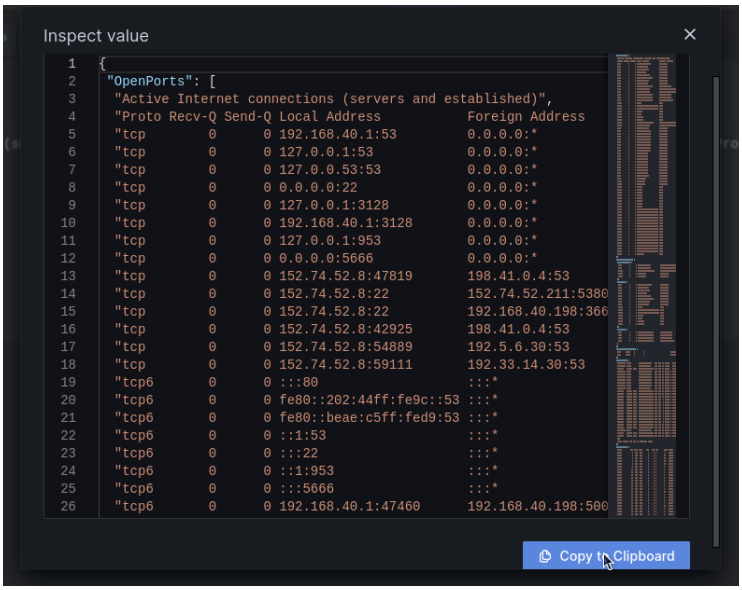


Figura 24. Detalle de un Reporte.

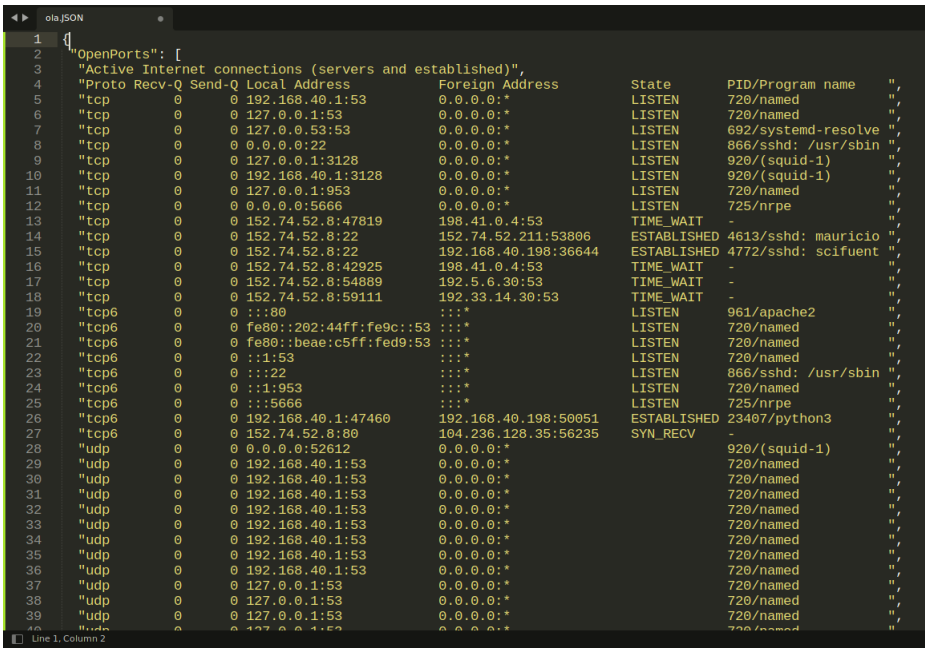


Figura 25. Texto copiado directamente de un Reporte.

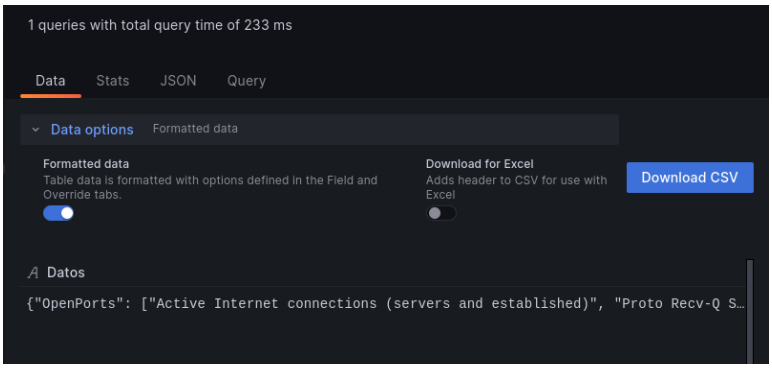


Figura 26. Opciones para descargar un Indicador de Compromiso como CSV.

4. Pruebas

4.1 Canal Seguro

Se utilizó la herramienta Wireshark para capturar el tráfico de la red y verificar que la comunicación de la plataforma se realizaba a través de un canal seguro.

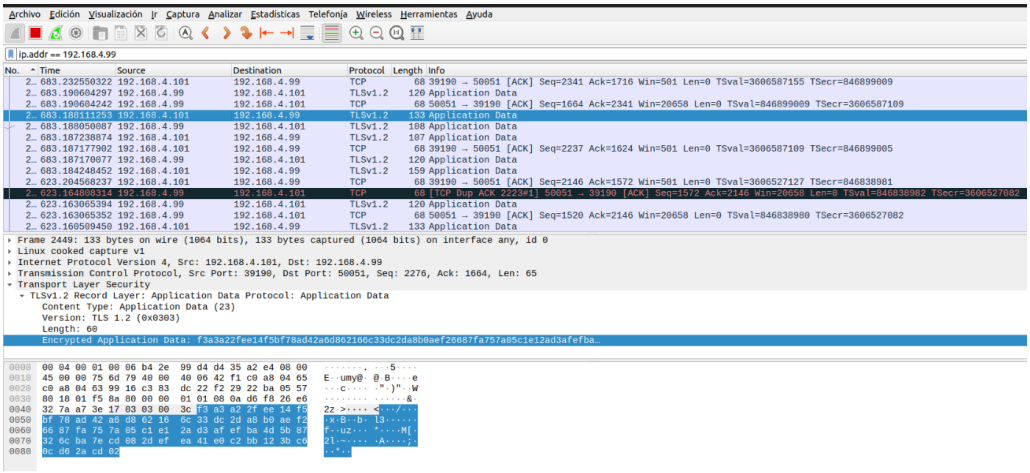


Figura 27. Análisis de paquete encriptado utilizando Wireshark.

Se filtraron los paquetes enviados y recibidos por el equipo con dirección IP 192.168.4.99 que corresponde al servidor gRPC, en la figura 27 se puede observar que los protocolos utilizados son TCP y TLSv1.2, siendo el último el utilizado para enviar la información dentro de la plataforma.

Source	Destination	Protocol	Length	Info
192.168.4.101	192.168.4.99	TCP	68	39190 → 50051 [ACK] Seq=2341 Ack=1716 Win=501 Len=0 TSval=3606587155 TSecr=846899009
192.168.4.99	192.168.4.101	TLSv1.2	120	Application Data
192.168.4.99	192.168.4.101	TCP	68	50051 → 39190 [ACK] Seq=1664 Ack=2341 Win=20658 Len=0 TSval=846899009 TSecr=3606587109
192.168.4.101	192.168.4.99	TLSv1.2	133	Application Data
192.168.4.99	192.168.4.101	TLSv1.2	108	Application Data
192.168.4.101	192.168.4.99	TLSv1.2	107	Application Data
192.168.4.101	192.168.4.99	TCP	68	39190 → 50051 [ACK] Seq=2237 Ack=1624 Win=501 Len=0 TSval=3606587109 TSecr=846899005
192.168.4.99	192.168.4.101	TLSv1.2	120	Application Data
192.168.4.101	192.168.4.99	TLSv1.2	159	Application Data
192.168.4.101	192.168.4.99	TCP	68	39190 → 50051 [ACK] Seq=2146 Ack=1572 Win=501 Len=0 TSval=3606527127 TSecr=846838981

Figura 28. Paquetes con destino/origen el servidor gRPC en Wireshark.

Al seleccionar alguno de estos paquetes que utilizan el protocolo TLSv1.2, nos encontramos con que la información de la aplicación se encuentra encriptada, tal cual como se puede apreciar en , probando de esta manera que la información que se transmite dentro de la plataforma no es visible y no puede ser modificada, confirmando confidencialidad e integridad de la comunicación.

Análisis de vulnerabilidades con Nessus

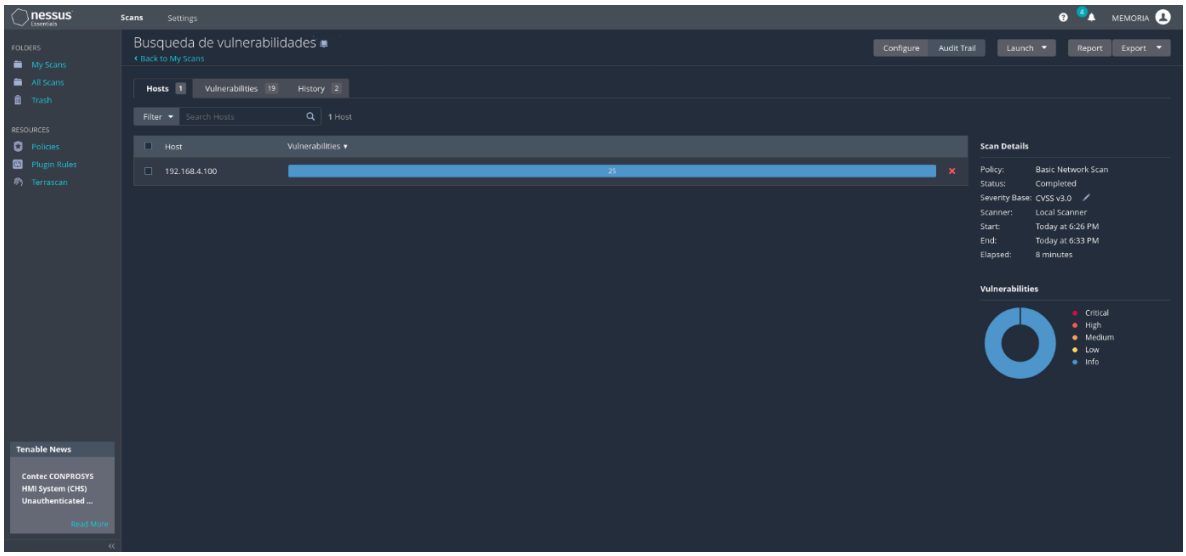


Figura 29. Interfaz Nessus Essentials.

Al servidor gRPC con la dirección IP *192.168.4.100* (se hizo un cambio de Raspberry con respecto al anterior experimento, debido a una falla) se le realizó un análisis para encontrar vulnerabilidades utilizando la herramienta Nessus Essentials (figura 29), en la cual solo se encontraron algunos puertos abiertos.

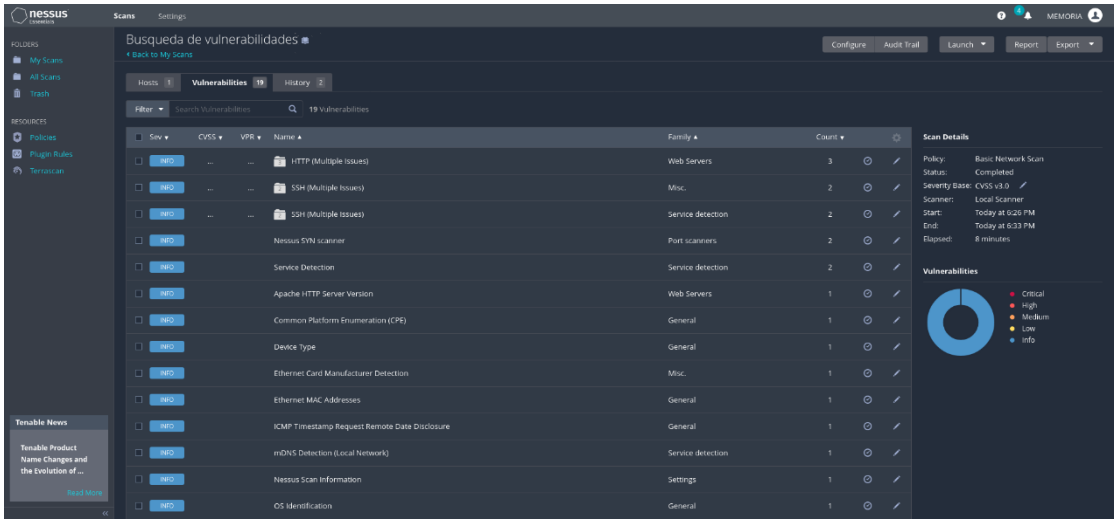


Figura 30. Lista de vulnerabilidades del servidor gRPC.

Dentro de lo que se ve listado en la figura 30, lo relevante es la información relacionada SSH, mDNS y HTTP.

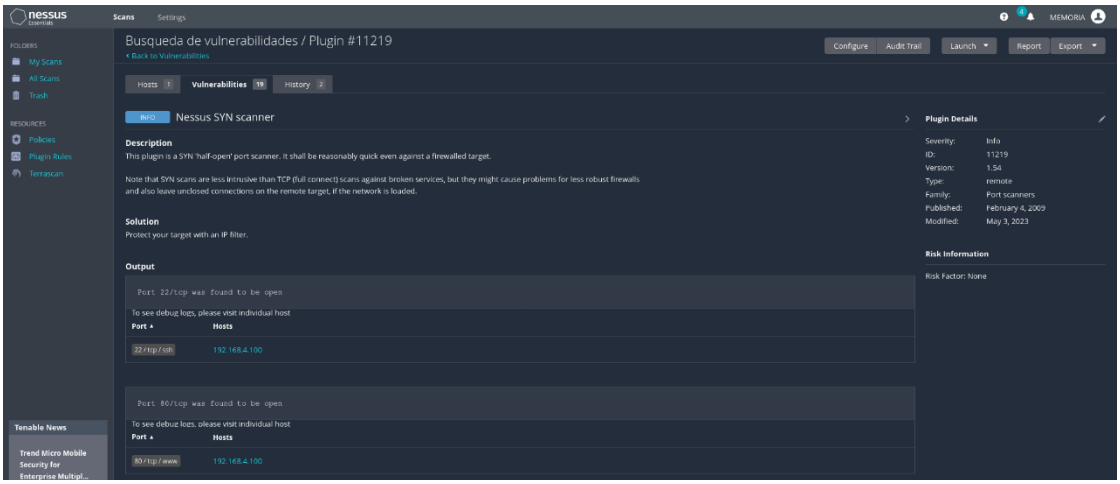


Figura 31. Puertos 22 y 80 abiertos.

Se encontraba abierto el puerto 22, utilizado para conexiones SSH, y el puerto 80, conexiones HTTP (figura 31).

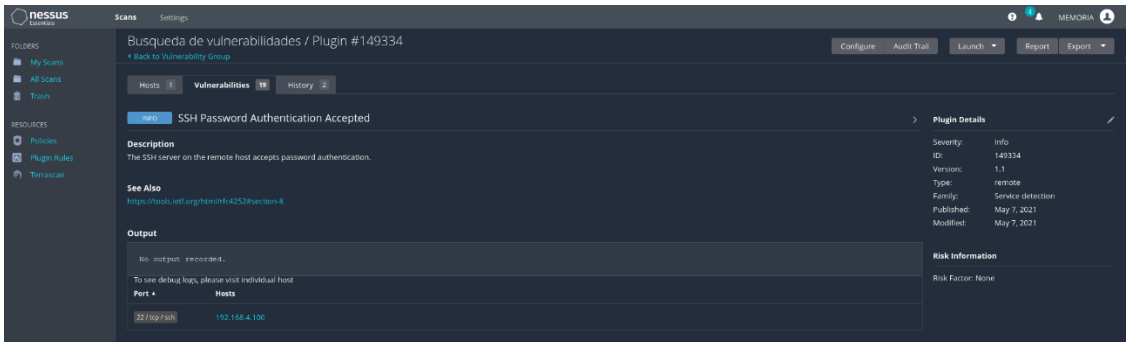


Figura 32. Información sobre SSH.

También, se encontraba activa la autenticación por contraseña para el servidor SSH (figura 32).

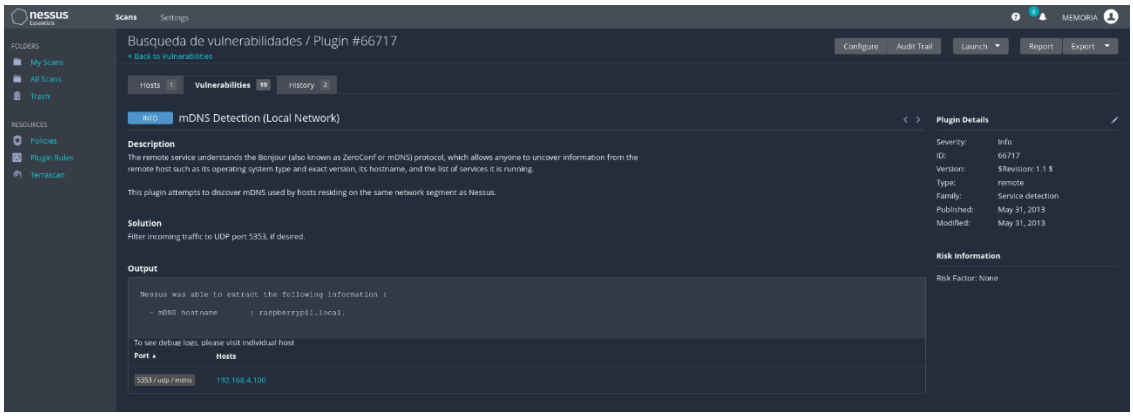


Figura 33. Información sobre mDNS.

Por último, como se ve en la figura 33, se detectó que el puerto 5353 estaba abierto, el cual es utilizado para mDNS, y puede ser vulnerable a ataques [25].

Para corregir todo lo anteriormente detallado, se configuró el firewall en el servidor gRPC, utilizando *ufw* (herramienta en consola para la configuración de firewall en Linux), para dejar solamente disponibles los puertos utilizados por NAGIOS con NRPE (puerto 5666) y gRPC (50051). En la figura 34 se puede ver esta configuración al consultar el estado del firewall.

```
pi1@raspberrypi1:~/Downloads/MonitoreoIoC-main $ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
50051                      ALLOW       Anywhere
5666                       ALLOW       Anywhere
50051 (v6)                 ALLOW       Anywhere (v6)
5666 (v6)                  ALLOW       Anywhere (v6)
```

Figura 34. Configuración del firewall en servidor gRPC.

En la figura 35 se puede ver que, al realizar nuevamente un análisis de vulnerabilidades con Nessus Essentials, se realizó correctamente el proceso de hardening en el sistema del servidor gRPC.

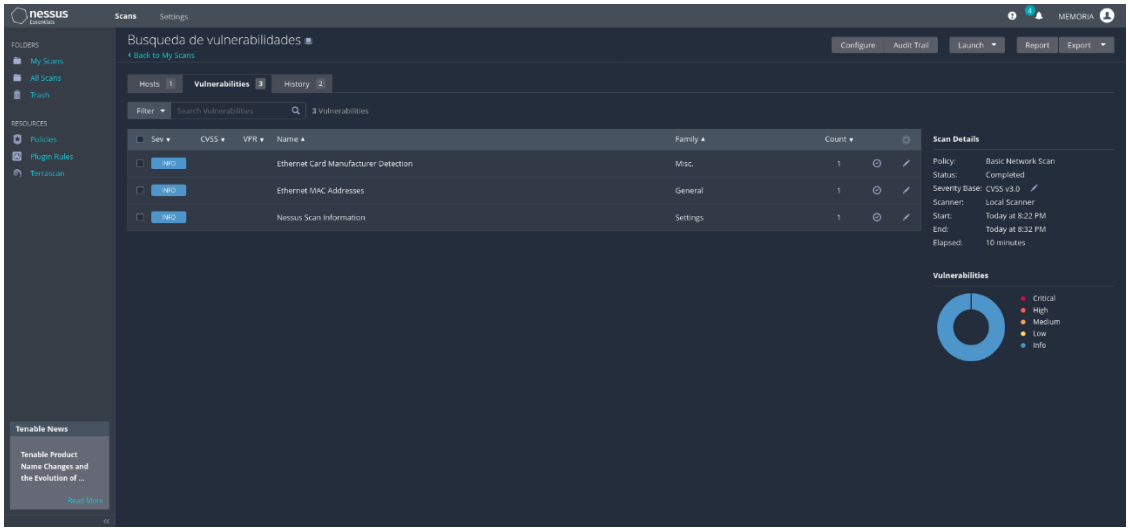


Figura 35. Vulnerabilidades del servidor después de configurar el firewall.

Escaneo de puertos con Nmap

Utilizando la herramienta *Nmap* con el comando de la figura 36 y con el objetivo de escanear todos los puertos del servidor gRPC, se obtuvo que solamente están disponibles los puertos 5666 (cerrado ya que al momento de ejecutar el análisis no se estaba utilizando) y el 50051, reafirmando que la configuración del firewall fue exitosa, tal como se ve en la figura 36.

```
memoria@memoria:~$ sudo nmap -p 1-65535 192.168.4.100
```

Figura 36. Comando para escanear puertos del servidor gRPC.

```
Nmap scan report for 192.168.4.100
Host is up (0.012s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT      STATE  SERVICE
5666/tcp  closed nrpe
50051/tcp open   unknown
MAC Address: E4:5F:01:37:50:E2 (Raspberry Pi Trading)

Nmap done: 1 IP address (1 host up) scanned in 958.26 seconds
```

Figura 37. Resultados de Nmap.

4.2 Comunicación gRPC

Conexión de un cliente al servidor gRPC

Se confirma que al cliente se le solicita una contraseña para poder verificar la integridad del propio archivo del cliente gRPC usando hash MD5 y realizar la conexión. Esto se puede ver en la figura 38.

```
scifuentes2017@umbral:~/gRPC$ sudo python3 communicationClient.py

Password:
M008zY9aU6HoKx9BHWWa03sFXgRtUNo7scny1HTYJjc=
message: "El archivo no ha sido modificado"

Se ejecutara un analisis
nohup: appending output to 'nohup.out'
Respuesta: Ok
```

Figura 38. Ingreso de contraseña de parte del cliente para conectarse al servidor gRPC.

Reporte Global Periodico

Para el prototipo implementado se decidió definir un tiempo de cinco minutos como intervalo para solicitar un reporte a cada cliente gRPC.

```
Server Started
Se comprobara el hash del archivo client en el host 152.74.52.8
Solicitud de 152.74.52.8: No pasa nada
Solicitud de 152.74.52.8: No pasa nada
Solicitud de 152.74.52.8: No pasa nada
Solicitud de 152.74.52.8: No pasa nada
Solicitud de 152.74.52.8: No pasa nada
Solicitud de 152.74.52.8: No pasa nada
Se recibio el reporte de 152.74.52.8
```

Figura 39. Solicitud de reporte periodico Servidor gRPC.

```
Se ejecutara un analisis
nohup: appending output to 'nohup.out'
Respuesta: Ok
Respuesta: Ok
Respuesta: Ok
Respuesta: Ok
Respuesta: Ok
Respuesta: Solicitud de reporte global
```

Figura 40. Solicitud de reporte periodico Cliente gRPC.

Reporte por Compromiso

Cada 20 minutos desde que se inicia la conexión se revisa el análisis en busca de Indicadores de Compromiso realizado por LOKI, y en caso de detectar uno o más indicadores, se notifica al servidor gRPC, el cual responde con una solicitud de reporte al host.

Se utilizaron archivos de un ransomware llamado Pay2Kitten [26], los cuales fueron exitosamente detectados por LOKI. Posterior a eso, los Indicadores de Compromiso fueron enviados al servidor gRPC, el cual respondió con una solicitud de reporte (figura 41).

```
Respuesta: Ok
Respuesta: Ok
Respuesta: Ok
Respuesta: Ok
Respuesta: Ok
Respuesta del servidor al mandar indicador: 462
Respuesta del servidor al mandar indicador: 463
Respuesta del servidor al mandar indicador: 464
Respuesta del servidor al mandar indicador: 465
terminamos de mandar todos los indicadores
Respuesta: Dame tu reporte
indicador a guardar: 465 Tipo: <class 'str'>
indicador a guardar: 464 Tipo: <class 'str'>
indicador a guardar: 463 Tipo: <class 'str'>
indicador a guardar: 462 Tipo: <class 'str'>
Respuesta: Ok
Respuesta: Ok
Respuesta: Ok
Respuesta: Ok
```

Figura 41. Mensajes de Cliente gRPC al detectar compromiso con LOKI.

```
Solicitud de 192.168.4.101: No pasa nada
Solicitud de 192.168.4.101: No pasa nada
Se recibio el indicador de 192.168.4.101
Se recibio el indicador de 192.168.4.101
Se recibio el indicador de 192.168.4.101
Se recibio el indicador de 192.168.4.101
Solicitud de 192.168.4.101: Tengo un problema
Se recibio el reporte de 192.168.4.101
```

Figura 42. Mensajes de Servidor gRPC al recibir indicadores.

4.3 NAGIOS

Para verificar que el monitoreo con NAGIOS se realizaba de manera correcta se probaron los cambios de estado tanto en hosts como servicios y se chequeó que se activaran los event handlers cuando fuese necesario, los cuales actualizan el estado de hosts y servicios en la base de datos y pueden solicitar reporte al servidor. Esto se puede ver reflejado tanto en los logs de eventos de NAGIOS como en el Panel Táctico, el cual extrae la información directamente de la base de datos.

Para esto se probó monitoreando el servidor umbral del Edificio de Ingeniería de Sistemas, chequeando el estado del servicio DHCP del servidor y la carga en el su CPU. El servidor umbral cuenta con las siguientes especificaciones:

- Procesador i7 870
- Memoria RAM 16GB
- Almacenamiento HDD 250GB
- LAN 2x 1Gbps

Figura 43. Servidor umbral de Ingeniería de Sistemas.

En las figuras 44 y 45 se puede ver el estado normal de umbral y sus servicios tanto en NAGIOS como en el panel táctico respectivamente.

umbral	CPU_load	OK	07-13-2023 14:23:58	0d 23h 36m 19s	1/3	OK - load average: 0.01, 0.08, 0.93
	DHCP	OK	07-13-2023 14:24:43	0d 20h 34m 27s	1/3	OK: Received 1 DHCPOFFER(s), max lease time = 600 sec.

Figura 44. Estado OK de servicios del servidor umbral en NAGIOS.

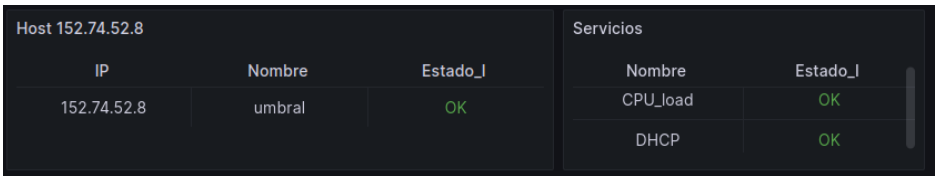


Figura 45. Estado OK del servidor umbral y sus servicios en Panel Táctico.

Servicios

El estado de un servicio cambia de OK a WARNING y de este a CRITICAL cuando un parámetro que está siendo monitoreado sobrepasa ciertos rangos. Para el caso de umbral se estableció el monitoreo de la carga del CPU y el servicio DHCP que ofrece a los laboratorios de redes y sistemas del edificio.

Carga del CPU

Para la carga del CPU se utilizó el plugin check_load de NAGIOS y se definieron los siguientes parámetros en la configuración del archivo nrpe.cfg en el servidor umbral.

```
command[check_load]=/usr/local/nagios/libexec/check_load -w 5.6,4.4,3.2 -c 7.2,5.6,4.0
```

Figura 46. Configuración de comando check_load en archivo de configuración de NRPE.

Para entender mejor la configuración, **-w** y **-c** definen que los siguientes tres números representan límites para cambiar el estado del servicio respecto al total de carga promedio del CPU en los últimos 1, 5 y 15 minutos respectivamente. Esto es el promedio entre todos los núcleos del CPU y para representar el 100% de carga para solamente un núcleo se utiliza la unidad 1, es decir que un valor de 2.7 significa que en promedio se realiza una carga del 270% de un núcleo repartida entre todos los núcleos disponibles.

Por ejemplo, el rango para el estado WARNING: "-w 5.6, 4.4, 3.2" significa que si existen los promedios: 5.6 para el último minuto, 4.4 para los últimos cinco minutos o 3.2 para los últimos quince minutos, entonces el estado de la carga del CPU cambiará a WARNING. Y de la misma manera se definen los siguientes tres números para cambiar el estado a CRITICAL.

Para aumentar la carga del CPU en el servidor umbral se utilizó el comando *stress* de linux con el cual se especifica la cantidad de núcleos y el tiempo que durará la prueba de la siguiente manera:

stress -c 8 -t 300

Desde la figura 47 hasta la figura 49 podemos ver cuando se sobrepasan los límites definidos para el estado WARNING para la carga del CPU del servidor umbral.

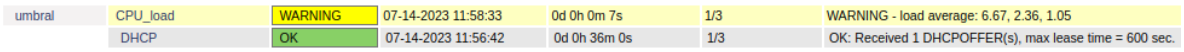


Figura 47. Carga del CPU entrando a estado WARNING en NAGIOS.

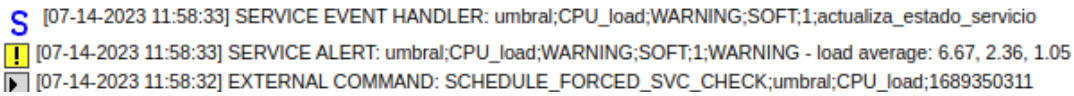


Figura 48. Registro de cambio de estado de la carga del CPU en umbral y activación del event handler en logs de NAGIOS.

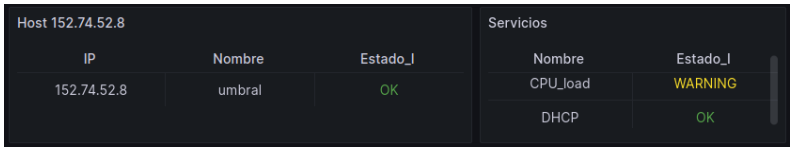


Figura 49. Cambio de estado a WARNING en Panel Táctico.

Y de la misma manera, desde la figura 50 hasta la figura 52 podemos ver cuando se sobrepasan los límites definidos para el estado CRITICAL para la carga del CPU del servidor umbral.

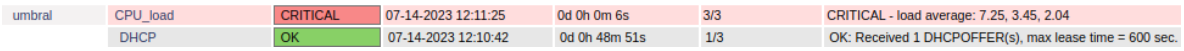


Figura 50. Carga del CPU entrando a estado CRITICAL en NAGIOS.

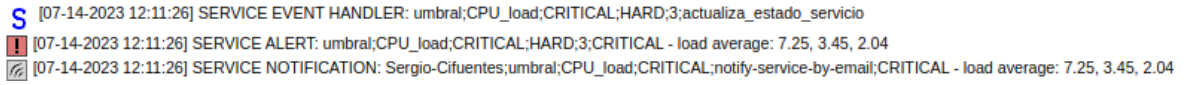


Figura 51. Cambio de estado a CRITICAL en logs de NAGIOS.

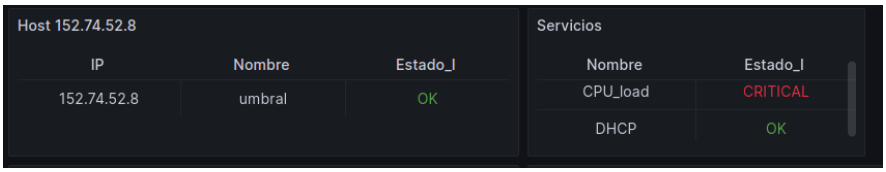


Figura 52. Cambio de estado a CRITICAL en Panel Táctico.

Servicio DHCP

Para probar el servicio DHCP de umbral primero se conectó la Raspberry Pi al servidor umbral mediante una conexión ethernet y se usó el comando check_dhcp de NAGIOS, el cual testea la correcta asignación de una IP dentro de la red recibiendo ofertas al momento de ejecutar el comando.

En las figuras 53, 54 y 55 se puede ver cuando NAGIOS detecta que no se recibieron ofertas DHCP y cambia el estado a CRITICAL, lo cual se refleja en el detalle del servidor en el panel táctico.

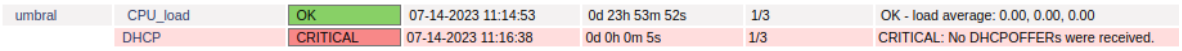


Figura 53. Servicio DHCP de umbral en estado CRITICAL en NAGIOS.

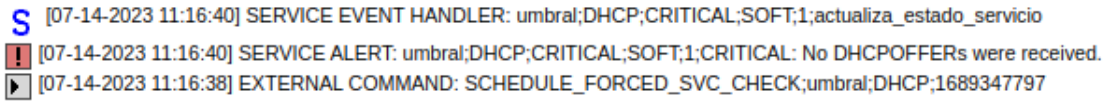


Figura 54. Cambio de estado a CRITICAL en logs de NAGIOS.

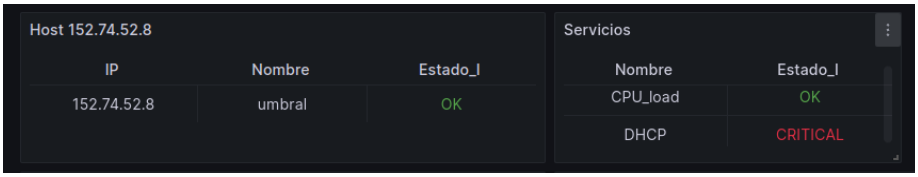


Figura 55. Cambio de estado a CRITICAL en Panel Táctico.

Todas las pruebas realizadas demuestran el funcionamiento deseado para este sistema de recopilación y despliegue de la información. A continuación se muestra una tabla resumen de las pruebas:

Tipo de prueba	**Prueba realizada**	**Resultado esperado**	**¿Logrado?**
Tríada CIA	Análisis de tráfico en red de la plataforma con herramienta Wireshark	Comunicación cifrada con certificados SSL/TLS	**SI**
	Análisis de vulnerabilidades con herramienta Nessus	Disminución de vulnerabilidades encontradas	**SI**
Funcionamiento del prototipo	Verificación de integridad del archivo	Conexión exitosa luego de la comprobación de la contraseña y comparación del hash	**SI**
	Solicitud de reporte periódico	Cada 5 minutos solicitar y recibir un reporte del cliente gRPC	**SI**
	Solicitud de reporte por detección de indicador de compromiso	Al momento de detectar un indicador de compromiso, solicitar y recibir un reporte del cliente gRPC	**SI**
	Aumentar carga en CPU	NAGIOS detecta el cambio en el estado del CPU, actualiza el estado en base de datos y solicita reporte del host afectado. Panel despliega correctamente la información obtenida de la base de datos	**SI**
	Caída de servicio DHCP	NAGIOS detecta el cambio en el estado del servicio DHCP, actualiza el estado en base de datos y solicita reporte del servidor afectado. Panel despliega correctamente la información obtenida de la base de datos	**SI**

Tabla 1. Resumen de pruebas realizadas al prototipo.

5. Conclusiones

Toda actividad maliciosa deja rastros o evidencias de haber ocurrido. Asimismo ocurre en ciberseguridad con los ciberataques al encontrarse indicadores de compromiso en sistemas dentro de una red. Es necesario entonces recopilar estos indicadores para poder ser analizados y lograr desarrollar medidas preventivas respecto a posibles vulnerabilidades encontradas.

En este trabajo se presentó una primera versión de un sistema de recopilación de indicadores de compromiso enfocado en la seguridad de la información y su disponibilidad. Se logró una comunicación segura y estable gracias a gRPC y, a pesar de que solamente fueron monitoreados el servicio DHCP del servidor umbral y su carga en CPU, Nagios y Grafana ofrecen una alta personalización al estar basados en el uso de plugins que permiten monitorear y desplegar la información de varios tipos de hosts y servicios [27], lo que brinda escalabilidad al proyecto.

Lo logrado fue un prototipo que sirve como prueba de concepto. El haber sido implementado en una placa Raspberry Pi demuestra el bajo costo requerido para lograr un sistema de este tipo, por lo que sería accesible para cualquier organización que necesite ampliar sus sistemas de ciberseguridad para infraestructuras de TI.

Todo lo desarrollado se encuentra en un repositorio en GitHub [28].

5.1 Trabajo futuro

Si bien se logró un sistema de recopilación de indicadores de compromiso básicos relacionados a malware conocido y una interfaz para poder visualizar la información de la red recopilada, se pueden realizar varias mejoras.

Redundancia en el hardware

Para la disponibilidad de los servidores y el panel táctico se propone aplicar redundancia en cuanto al hardware, replicando los servidores NAGIOS y gRPC, la Base de Datos local y el Panel Táctico. En caso de fallar la estación principal, la conexión se redirige a la siguiente estación disponible. Esto no afectaría al funcionamiento de los servidores gRPC y NAGIOS ya que para todas las estaciones, la configuración respecto al monitoreo y la comunicación es la misma, siendo diferenciadas solamente por el IP de cada una. La única dependencia que se genera respecto de la Estación de Monitoreo 1 es la información almacenada en la base de datos, para lo cual es necesario comunicar las bases de datos para copiar la información y así garantizar la disponibilidad de los datos.

Esto se ejemplifica en la figura 56.

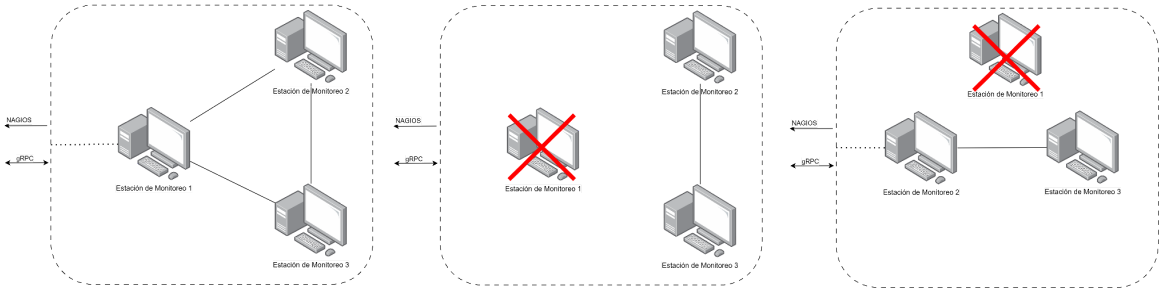


Figura 56. Ejemplo de caída y reconexión a una estación de monitoreo

Mejoramiento del panel táctico

En esta primera versión las funcionalidades del panel táctico son muy limitadas, siendo posible visualizar el último estado registrado de la red en la base de datos local y detalles de reportes e indicadores. Sin embargo, utilizando la herramienta Grafana es posible crear gráficas de distinto tipo, como por ejemplo una gráfica que despliegue las anomalías que se han detectado a lo largo del tiempo y poder filtrar por dirección ip o incluso por tipos de indicadores de compromiso. De esta manera dentro de una red local se podría tener un mejor acercamiento o indicio de alguna posible vulnerabilidad no considerada. Sería necesario entonces rediseñar la base de datos y elegir un sistema de gestión de bases de datos para el cual existan plugins compatibles [29], además de añadir información a lo recopilado en los reportes para poder hacer las comparaciones y análisis necesarios posteriormente.

Monitoreo y recopilación en Windows

Este trabajo se centró principalmente en dispositivos Linux, sin embargo todas las herramientas que fueron utilizadas tienen compatibilidad con los sistemas Windows [30]. Esto amplía las posibilidades para el sistema recopilando muchas más evidencias de actividad maliciosa dentro de la red.

7. Referencias

[1] Statista. (2021). Cyber Crime Statistics - Reported Cybercrime Offenses 2020. https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

[2] McKinsey & Company. (2020). The COVID-19 recovery will be digital: A plan for the first 90 days. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-covid-19-recovery-will-be-digital-a-plan-for-the-first-90-days

[3] Los ataques de phishing se incrementaron a nivel mundial casi un 50% en 2022. Obtenido de itDigitalSecurity. https://www.itdigitalsecurity.es/actualidad/2023/04/los-ataques-de-phishing-se-incrementaron-a-nivel-mundial-casi-un-50-en-2022#:~:text=Los%20ataques%20de%20phishing%20se,2022%20%7C%20Actualidad%20%7C%20IT%20Digital%20Security

[4] Las cinco principales tendencias de ciberseguridad para 2023. Obtenido de Forbes. https://forbes.es/empresas/194234/las-cinco-principales-tendencias-de-ciberseguridad-para-2023/

[5] Qué son los Indicadores de Compromiso: la evidencia de que puedes haber sido víctima de malware. Obtenido de Welivesecurity https://www.welivesecurity.com/la-es/2021/02/22/que-son-indicadores-compromiso-evidencia-puedes-haber-sido-victima-malware/

[6] ¿Que es la ciberseguridad?. Obtenido de IBM. https://www.ibm.com/es-es/topics/cybersecurity

[7]La importancia de la ciberseguridad: ¿Por qué y cómo protegernos?- Obtenido de Ibermática. https://ibermaticaindustria.com/blog/la-importancia-de-la-ciberseguridad-por-que-y-como-protegernos/

[8] Mike Chapple, James Michael Stewart & Darril Gibson. (2021). (ISC)2 CISSP Certified Information Systems Security Professional Official Study Guide. https://drive.google.com/drive/folders/17lSyaE8EAhYjFWkkdMBAyyn9YwoO5kAD?usp=sharing

[9]¿Que es la tríada de la CIA?. Obtenido de ComputerWeekly
https://www.computerweekly.com/es/opinion/Que-es-la-triada-de-la-CIA

[10]Cisco Security Indicators of Compromise Reference Guide. Obtenido de Cisco.
https://sec.cloudapps.cisco.com/security/center/resources/iocs.html#conf

[11]CSIRT. (2020). AN2-2020-12: Vulnerabilidades en VPNs SSL/TLS y cómo mitigarlas.
https://www.csirt.gob.cl/media/2020/07/AN2-2020-12.pdf

[12] Brock, B. (2021, 27 de enero). How to Determine if Linux is Compromised. Linux Hint.
https://linuxhint.com/determine_if_linux_is_compromised/

[13] Python Software Foundation. Python.
https://www.python.org/

[14] SQLite. SQLite Home Page.
https://sqlite.org/index.html

[15] Noelia Martín. (2022). gRPC, ¿qué es y cómo funciona? Paradigmadigital.com.
https://www.paradigmadigital.com/dev/grpc-que-es-como-funciona/

[16] Cloudflare. (2021). cfssl: CFSSL - Cloudflare's PKI toolkit. GitHub.
https://github.com/cloudflare/cfssl

[17] Nagios Enterprises LLC. Nagios.
https://www.nagios.org/

[18] Galstad, E. NRPE - Nagios remote plugin executor - Nagios Exchange.
Nagios.org.
https://exchange.nagios.org/directory/Addons/MonitoringAgents/NRPE--2D-Nagios-Remote-Plugin-Executor/details

[19] KeepCoding. (2023). ¿Qué son las reglas YARA? KeepCoding Blog.
https://keepcoding.io/blog/que-son-las-reglas-yara/

[20] Neo23x0. (2021). Loki. GitHub.
https://github.com/Neo23x0/Loki

[21] Grafana Labs. Grafana
https://grafana.com/

[22] Wireshark. Wireshark · Go Deep.
https://www.wireshark.org/

[23] Tenable. Nessus Essentials.
https://es-la.tenable.com/products/nessus/nessusessentials

[24] Plugin SQLite para Grafana.
https://grafana.com/grafana/plugins/frser-sqlite-datasource/

[25] HackTricks. UDP Multicast DNS (mDNS).
https://book.hacktricks.xyz/networkservices-pentesting/5353-udp-multicast-dns-mdns

[26] MalwareSamples. (2021). Malware-Feed. Github.
https://github.com/MalwareSamples/Malware-Feed

[27] Nagios Exchange Community Plugins. Nagios
https://exchange.nagios.org/

[28] Felipe Henriquez y Sergio Cifuentes. (2023). MonitoreoIoC. GitHub.
https://github.com/FelipeHR/MonitoreoIoC

[29] Plugins Data-Sources para Grafana
https://grafana.com/grafana/plugins/data-source-plugins/

[30] Windows monitoring with Nagios.
https://www.nagios.com/solutions/windows-monitoring/