IMPLEMENTACIÓN DE SECURITY ONION EN UN ENTORNO DE LABORATORIO

Miguel E. Pérez

1. Objetivo general

Diseñar y desplegar una infraestructura de monitoreo de seguridad de red (NSM) y detección de amenazas basada en Security Onion sobre un entorno virtualizado, validando sus capacidades de ingesta, correlación y visualización de eventos de seguridad en un escenario de laboratorio controlado

2. Objetivos específicos del Proyecto

- Implementar la arquitectura base de Security Onion sobre el hipervisor Proxmox VE, optimizando la asignación de recursos de hardware virtual para garantizar la estabilidad de los servicios de Elastic Stack
- Configurar una arquitectura de despliegue tipo Standalone, centralizando los roles de gestión, búsqueda y sensores de red para consolidar la ingesta de tráfico y logs en un único nodo unificado
- Simular un entorno de red corporativo mediante el despliegue de máquinas virtuales cliente (Windows/Linux) para generar tráfico de red y telemetría real que alimente el sistema de detección
- Desplegar y configurar agentes de punto final (Endpoint Agents) en los servidores clientes, aplicando políticas de recolección de logs personalizadas para asegurar la visibilidad completa de eventos del sistema operativo
- Diseñar tableros de control (Dashboards) y consultas de Threat Hunting personalizadas en Kibana/Security Onion Console, facilitando la interpretación de alertas y la reducción del tiempo de análisis de incidentes
- Elaborar documentación técnica detallada sobre el proceso de despliegue, configuración y resolución de problemas (troubleshooting), sirviendo como base de conocimiento transferible para futuras implementaciones

3. Recursos Utilizados

- Hardware: 1 servidores HP, 1 switch de red, 2 interfaces de red en el servidor.
- Software: ISO de Security Onion, Virtualizador de Proxmox,

4. Arquitectura de la solución

Para el cumplimiento de los objetivos del proyecto, se optó por un despliegue de Security Onion bajo la arquitectura Standalone (Nodo Único). Esta topología consolida todos los roles operativos de la plataforma —Gestión (Master), Búsqueda (Search Node) y Detección (Sensor)— en una única instancia virtualizada.

Justificación Técnica del Diseño: La elección de la arquitectura Standalone responde a tres criterios fundamentales para este entorno de laboratorio:

1. Optimización de Recursos: Al centralizar los servicios en un solo servidor virtual, se reduce la latencia interna entre componentes (como la ingesta de logs hacia la base de datos) y se minimiza la huella de hardware requerida en el hipervisor Proxmox.
2. Facilidad de Gestión: Permite la administración unificada de políticas de detección y actualizaciones sin la complejidad de mantener la sincronización entre clústeres distribuidos.
3. Idoneidad para pruebas de concepto: Es el estándar de la industria para Pruebas de Concepto (PoC) y entornos de formación, permitiendo validar la funcionalidad completa de la suite (NSM + EDR + SIEM) de manera aislada y controlada.

Componentes del Stack Tecnológico: La solución desplegada integra las siguientes tecnologías clave para la defensa en profundidad:

- Capa de Monitoreo de Red (NSM):
    - Suricata: Motor de detección de intrusos basado en firmas para identificar amenazas conocidas y exploits en tiempo real.
    - Zeek (anteriormente Bro): Analizador de tráfico de red que genera metadatos detallados (transacciones DNS, conexiones SSL/TLS, flujos HTTP) esenciales para el análisis forense y la caza de amenazas (Threat Hunting).
- Capa de Endpoint (HIDS/EDR):

    - Elastic Agent: Agentes desplegados en las máquinas víctima para recolectar telemetría del sistema operativo, eventos de seguridad y auditar la integridad de archivos.

- Capa de Almacenamiento y Análisis:
    - Elasticsearch (OpenSearch): Motor de búsqueda y analítica distribuido que indexa y correlaciona los logs provenientes de las capas de red y endpoint.
    - Security Onion Console (SOC) & Kibana: Interfaces web para la visualización de datos, gestión de alertas y ejecución de consultas de investigación.

5. Implementación

El despliegue de la solución se estructuró en tres fases consecutivas: instalación del nodo central, configuración de la interfaz de gestión y despliegue de agentes, y finalmente, la optimización mediante políticas personalizadas.

5.1. Instalación de Security Onion en entorno virtualizado (Proxmox)

El proceso inició con el aprovisionamiento de una Máquina Virtual (VM) en el hipervisor Proxmox, destinada a operar como nodo Standalone de Security Onion. Este nodo centraliza las funciones de recolección de logs, gestión de alertas y visualización de dashboards.

Configuración de Hardware Virtual: Para garantizar el rendimiento de los servicios de indexado (Elasticsearch), se configuró la VM con las siguientes especificaciones (Ver Anexo 1):

- CPU: Tipo Host (para permitir el paso de instrucciones AVX), con asignación de 16 núcleos.
- Memoria RAM: 32 GB.
- Almacenamiento: Disco virtual de 200 GB (mínimo recomendado).

- Red: Dos interfaces de red virtuales:

    1. Gestión (Management): Con acceso a la red corporativa e Internet.

    2. Monitoreo (Monitor): Configurada en modo promiscuo para la captura de tráfico (sniffing) de otras instancias.

Proceso de Instalación: Tras iniciar la VM con la imagen ISO oficial, se procedió con la instalación del sistema operativo base. Se configuraron las credenciales de administrador local y se ejecutó el asistente de instalación estándar (Standard Security Onion Installation).

Durante la configuración del aplicativo (Ver Anexos 2-6), se seleccionó la arquitectura STANDALONE, ideal para entornos de laboratorio y pruebas de concepto (PoC), aceptando los términos de licencia correspondientes.

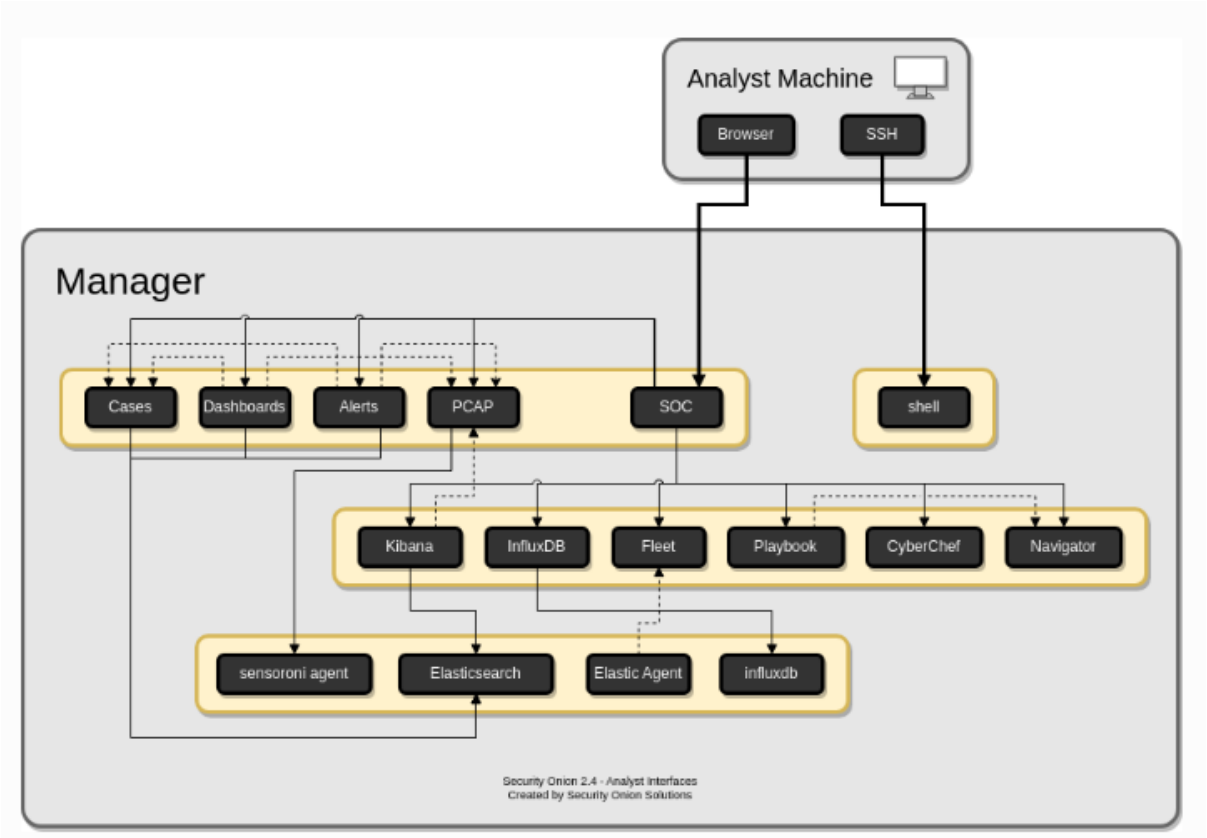
Configuración de Red: Se optó por el modo de instalación Standard (con conexión directa a Internet) para facilitar la descarga de actualizaciones y reglas de detección, descartando el modo Airgap (Ver Anexo 7). Se definió el hostname del nodo y se configuraron las interfaces de red de la siguiente manera:

- Interfaz de Gestión (ens18): Configurada con direccionamiento estático (IP: 10.10.100.3/24, Gateway: 10.10.100.1) y servidores DNS corporativos (Ver Anexos 9-13).

- Interfaz de Monitoreo (ens19): Asignada para la captura de paquetes sin dirección IP lógica.

Finalmente, se configuró el acceso a la interfaz web mediante dirección IP y se creó la cuenta de administrador principal. Tras finalizar el despliegue, se validó el estado de los servicios mediante el comando de consola so-status, confirmando la operatividad del sistema.

5.2. Gestión mediante Interfaz Web y Despliegue de Agentes



Una vez finalizada la instalación, se accedió a la consola de gestión (Security Onion Console) a través de un navegador web en la dirección https://10.10.100.3, aceptando los certificados de seguridad autofirmados (Ver Anexo 19).

La consola centralizada (Ver Anexo 20) proporciona acceso a módulos críticos como:

- Dashboards: Visualización gráfica de la telemetría recolectada.
- Alerts: Gestión de incidentes basados en reglas de detección.
- Cases: Sistema de gestión de tickets para la investigación de incidentes. (OTRS)
- PCAP: Repositorio de capturas de tráfico de red completo.
- Grid: Monitoreo de salud y estado de los nodos del clúster.

- Hunt: Herramienta para la búsqueda proactiva de amenazas (Threat Hunting).

5.2.1. Instalación y Enrolamiento de Elastic Agents

Para habilitar la recolección de logs desde los endpoints, se realizaron las siguientes configuraciones previas:

1. Configuración de Firewall: Se habilitó el tráfico entrante desde los agentes hacia el servidor. A través de la ruta Administration > Configuration > firewall > hostgroups > elastic_agent_endpoint, se definió el rango de IP de la red de laboratorio y se sincronizaron las reglas del grid (Ver Anexo 21).

2. Gestión en Elastic Fleet: Desde la consola SOC, se accedió al módulo Elastic Fleet (Ver Anexo 22), autenticándose en Kibana.

3. Despliegue del Agente:

    - Se utilizó la función "Add Agent", asignando inicialmente la política por defecto endpoints-initial (Ver Anexo 24).

    - Se seleccionó la plataforma Windows, generando el comando de instalación automatizado.

    - Este script se ejecutó en la terminal (PowerShell) del servidor objetivo, logrando una conexión exitosa visualizada posteriormente en Kibana (Ver Anexo 26).

El mismo procedimiento se replicó para servidores Linux, adaptando los comandos de instalación, resultando en un inventario de activos completamente enrolado y visible en la plataforma (Ver Anexos 27-30).

5.3. Optimización y Configuración Avanzada

Con el objetivo de mejorar el rendimiento del sistema y la calidad de los datos recolectados, se implementó una estrategia de segmentación de políticas.

5.3.1. Personalización de Políticas de Agentes para evitar la ingesta de datos innecesarios y asegurar la recolección de eventos críticos de seguridad, se crearon políticas dedicadas por sistema operativo:

- Política Windows: Se creó la política Policy-Windows-Server en Fleet > Agent Policies. A esta política se le añadieron integraciones específicas para enriquecer el análisis: Elastic Defend (para capacidades EDR), Windows Security (logs de eventos) y System (Ver Anexos 31-37).

- Política Linux: De manera análoga, se configuró una política dedicada (Policy-Linux-Servers) con integraciones nativas para dicho sistema (Auditd, System logs).

Reasignación de Agentes: Finalmente, se procedió a migrar los agentes existentes a sus nuevas políticas correspondientes. Desde el menú de Agentes en Fleet, se utilizó la opción Assign to new policy, asegurando que cada activo reciba la configuración óptima para su entorno operativo (Ver Anexos 38-39).

5.3.2 Validación y Afinamiento (Tuning)

Para certificar la operatividad del sistema antes del paso a producción, se ejecutó un protocolo de pruebas en dos niveles: validación de firmas básicas y simulación de amenazas avanzadas.

1. Nivel 1: Validación de Conectividad y Firmas (TestMyIDS)

    - Acción: Se ejecutó el comando estándar de prueba curl http://testmyids.com desde una máquina cliente dentro de la red monitoreada.

- Objetivo: Verificar que el sensor estuviera capturando paquetes y comparándolos contra la base de datos de firmas de Suricata.

- Resultado: La consola generó inmediatamente la alerta GPL ATTACK_RESPONSE id check returned root, confirmando que el flujo de tráfico desde el switch virtual hacia el sensor estaba funcionando correctamente (Ver anexo 40).

2. Nivel 2: Simulación de Amenaza Interna (HackTools)

    - Acción: Se procedió a la descarga de herramientas de post-explotación (paquete HackTool.Rubeus) en la máquina Kali Linux.

    - Resultado (Verdadero Positivo): El sistema detectó la firma del binario en la red, generando la alerta ET CURRENT_EVENTS Possible HackTool, validando la capacidad del NIDS para identificar software malicioso real.

3. Afinamiento Operativo (Supresión)

    - Acción: Tras confirmar el éxito de la detección en el punto 2, se creó una regla de supresión para la IP de la máquina de pentesting. Para ello, donde se nos muestra la alerta se debe abrir el panel de detalles, bajar hasta el apartado de tuning y ahí designar lo que se hará en caso de que la regla se active, por ejemplo, suprimirla dependiendo de la IP de origen (Ver anexos 41-42).

    - Justificación: Esto elimina el ruido de alertas benignas generadas por actividades de auditoría autorizadas, manteniendo la vigilancia activa para el resto de los activos de la red.

5.3.3 Queries y Dashboards en Security Onion.

En el panel principal de Security Onion se puede observar el apartado de Dashboards, los cuales SO trae por defecto y no pueden ser personalizados. Entre ellos, se puede encontrar los dashboards de Alertas, Alertas de NIDS,

Elastic Agent Overview, Host Overview y demás opciones que facilitan el análisis. Sin embargo, el no poder personalizarlo resta un poco de la capacidad total de la herramienta. Security Onion tiene su propio módulo de Kibana en la misma interfaz web al cual puedes ingresar con el mismo correo y contraseña que usamos para ingresar a SO. Se mostrarán diferentes dashboard prediseñados, aunque la herramienta permite crear dashboards propios. Esto lo convierte en una herramienta potente al momento de la detección de amenazas (Ver anexo 43).

Para optimizar los resultados de Kibana, es necesario aplicar filtros mediante KQL (Kibana Query Language) de tal manera que se vea solo lo que el analista está buscando y esconder el ruido momentáneamente (Ver anexo 44). Por último, se realizó la creación de un dashboard desde 0 el cual funciona para correlacionar eventos y ver las alertas más acontecidas en un lapso de tiempo (Ver anexo 45).

Optimización de Reglas de Endpoint (Sysmon) Adicionalmente a la supresión de red, se detectó ruido excesivo a nivel de host generado por la integración de Microsoft Office (procesos WINWORD.EXE interactuando con el registro).

Acción: Se modificó el manifiesto XML de Sysmon (sysmonconfig.xml) implementando una exclusión lógica <RegistryEvent onmatch="exclude">. Resultado: Esto redujo la carga de procesamiento del agente y limpió los dashboards de falsos positivos recurrentes, permitiendo enfocar la detección en anomalías reales (Ver anexo 46).

5.3.4 Hunt y escalado de casos

Las alertas que sean generadas se podrán visualizar en el panel Alert de Security Onion, además podemos ver los demás logs en el panel de Hunt. En caso de ver una alerta que se considere crítica, se puede asignar a un caso para que un analista la pueda investigar detenidamente al hacer clic derecho y seleccionar la opción de Add to Case(Ver anexo 47). En el mismo menú podemos ver distintas opciones entre las cuales se encuentra:

- Include, donde se incluye esta entrada en la Query que hayamos realizado anteriormente con el fin de buscar alertas repetidas
- Exclude, donde se excluye esta entrada de la Query que hayamos realizado para evitar ver resultados como este durante el análisis.
- Only, permite ver solo las alertas de ese tipo.
- Tune Detection, el cual permite al operador modificar la acción que se realiza al momento de que se dispare la alerta como por ejemplo modificarla, suprimirla o seleccionar un umbral de tolerancia.
- Group by, agrupar las alertas
- Clipboard, se podrá copiar los valores del log en el formato que sea requerido
- Actions, permite varias acciones como buscar en VirusTotal, Agregar al caso, Correlacionar, PCAP, CyberChef, Google.

En la pestaña de casos, podemos ver cada uno de los casos escalados con anterioridad (Ver anexo 48). Estos casos se van a investigar individualmente por los analistas, por lo cual se selecciona el caso y se abrirá una pestaña donde se puede comentar la alerta, adjuntar evidencia como .pdf o .jgp, agregar información adicional como si es un dominio, un hash o archivos; colocar los eventos relacionados con ese caso y el historial (Ver anexo 49-53).

6. Resultados Obtenidos

- Despliegue de la infraestructura virtual en Proxmox e instalación del sistema operativo Security Onion en modo Standalone.
- Configuración del Bridge de Proxmox en modo promiscuo y ajuste del ageing a 0 (Modo Hub) para replicar tráfico hacia el sensor.
- Verificación de visibilidad de tráfico en la interfaz de monitoreo mediante tcpdump.
- Deshabilitación de la validación de *checksum* (checksum-validation: no) en la configuración de Suricata para evitar el descarte de paquetes por offloading virtual.
- Corrección del mapeo de interfaces en el archivo Pillar de SaltStack, reasignando el sensor de bond0 a la interfaz física ens19.
- Validación funcional del NIDS mediante simulación de ataque (curl http://testmyids.com) y confirmación de alertas en la consola.

- Creación de consultas personalizadas (Queries) y visualizaciones gráficas (Dashboards) en Kibana.
- Implementación de reglas de supresión para mitigar falsos positivos de herramientas legítimas (Rubeus) en la máquina Kali Linux.
- Despliegue y enrolamiento de Elastic Agents en los endpoints.
- Configuración de políticas en Elastic Fleet para la ingesta de logs de Windows (Sysmon) y Linux (Auditd).
- Activación de la integración "Elastic Defend" y generación de actividad para la creación inicial de índices de EDR.
- Optimización de la visualización en Kibana mediante la corrección de métricas de Sysmon, reemplazando la clasificación por severidad (vacía por defecto) por la clasificación basada en Event ID, logrando una interpretación efectiva de la actividad del endpoint.

7. Desafíos y Soluciones

7.1. Visibilidad de Red en Entornos Virtuales (Proxmox)

- Desafío: La interfaz de monitoreo del sensor no recibía copia del tráfico de la red. El *Linux Bridge* (vmbr0) de Proxmox actuaba como un switch eficiente, descartando tráfico unicast no destinado al sensor, provocando "ceguera" en el NIDS.
- Solución: Se implementó una configuración de "Modo Hub" en el bridge virtual. Se habilitó el modo promiscuo y se configuró el parámetro ageing a 0, forzando al switch virtual a inundar (flood) el tráfico a todos los puertos para garantizar la captura total de paquetes.

7.2. Integridad de Paquetes y Checksum Offloading

- Desafío: Suricata descartaba silenciosamente los paquetes capturados antes de analizarlos. Esto se debía a que los drivers de red virtuales (VirtIO) delegan el cálculo de *checksum* al hipervisor (Offloading), entregando paquetes con sumas de verificación técnicamente inválidas al sistema operativo invitado.

- Solución: Se modificó la configuración del motor Suricata para deshabilitar la validación estricta de integridad (checksum-validation: no), permitiendo el análisis de paquetes con checksums delegados.

7.3. Persistencia de Configuración y Orquestación (SaltStack)

- Desafío: Las modificaciones manuales en los archivos de configuración (suricata.yaml) eran revertidas automáticamente tras cada reinicio del servicio. Además, existía una discrepancia entre la interfaz lógica por defecto (bond0) y la física real (ens19).
- Solución: Se identificó y editó el archivo maestro de configuración Pillar (.sls) de SaltStack. Esto permitió persistir los cambios (mapeo de interfaz correcta y variables del motor) y asegurar que el orquestador aplicara la configuración deseada en cada despliegue.

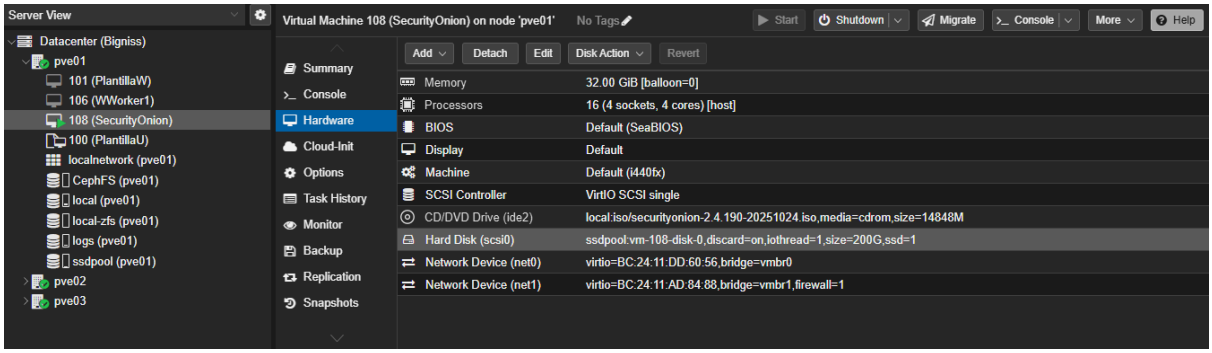
7.4. Puntos Ciegos en Movimientos Laterales

- Desafío: La configuración por defecto de Suricata define la red externa como !$HOME_NET (todo lo que no es local). Esto impedía la generación de alertas ante ataques de *Pivoting* o escaneos internos entre máquinas de la misma subred (10.10.100.0/24).
- Solución: Se corrigió la definición de la variable $EXTERNAL_NET a any. Esto amplió el alcance de las reglas de detección para evaluar tráfico origen/destino dentro de la misma red local, sacrificando rendimiento por mayor visibilidad de seguridad.

7.5. Gestión de Falsos Positivos (Ruido Operativo)

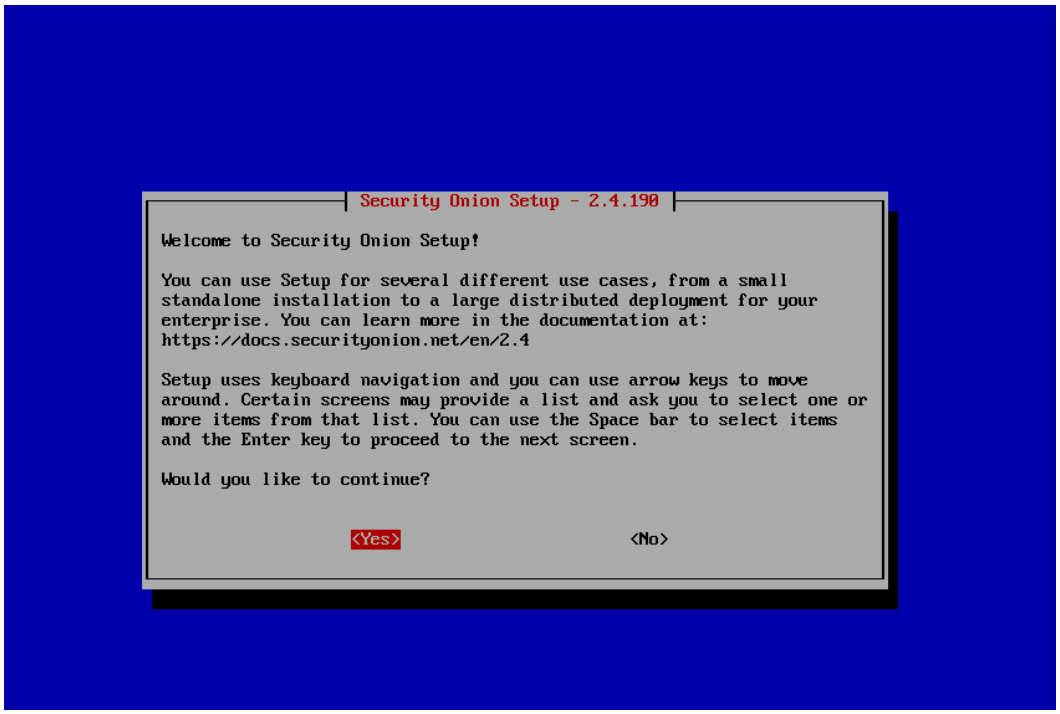
- Desafío: Herramientas legítimas de pentesting instaladas en la máquina atacante (Kali Linux) generaban alertas críticas de malware (ej. HackTool.Rubeus), saturando la consola de verdaderos positivos que
- Solución: Se aplicaron reglas de Supresión (Suppression) específicas basadas en la IP de destino. Esto permitió silenciar las alertas conocidas en la máquina de pruebas sin desactivar la regla globalmente, manteniendo la protección activa para el resto de la red.

8. Anexos

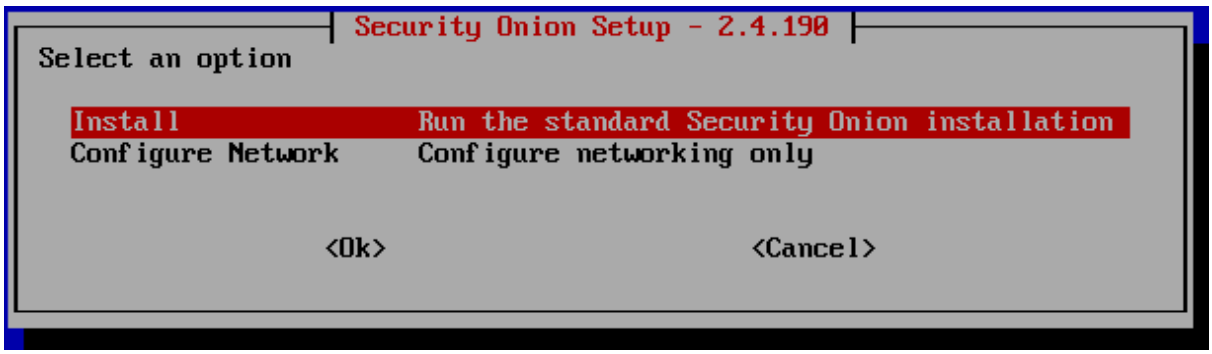


Anexo 1. Configuración inicial de Security Onion.

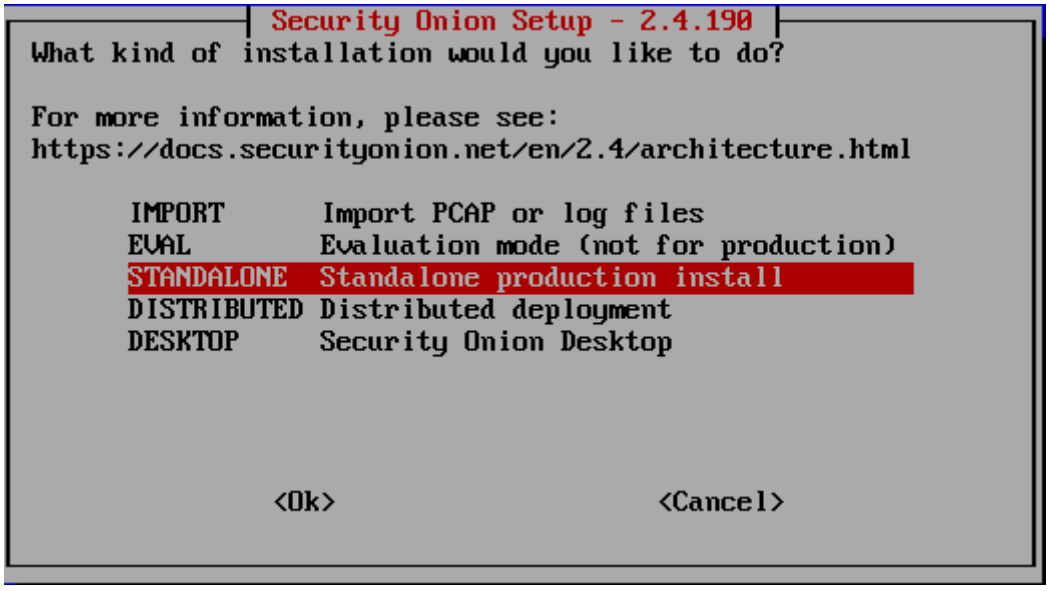
Anexo 2. Menú de instalación de Security Onion.



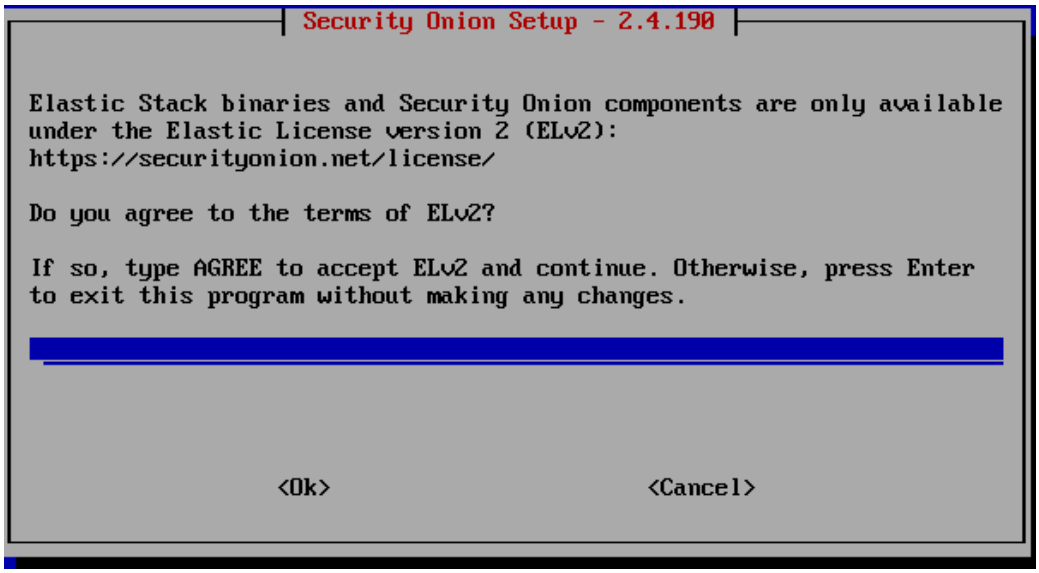
Anexo 3.



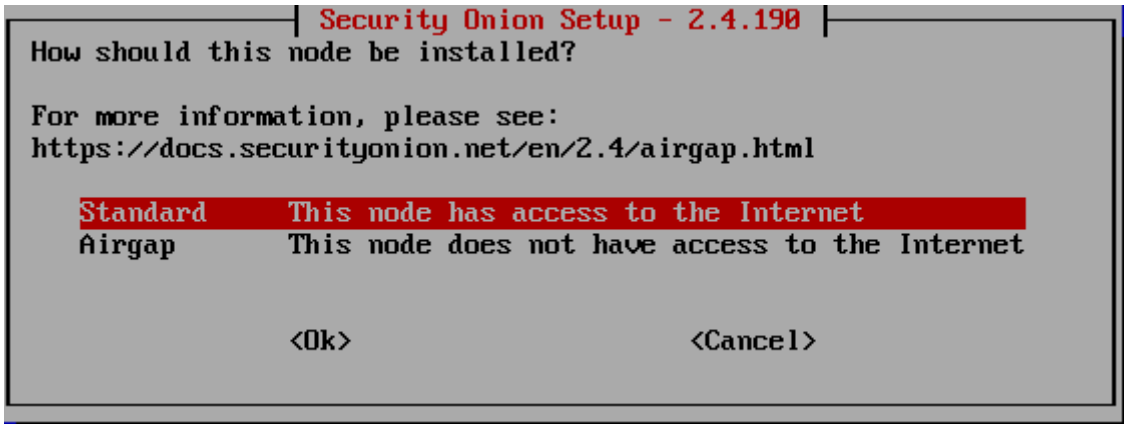
Anexo 4. Instalación de Security Onion



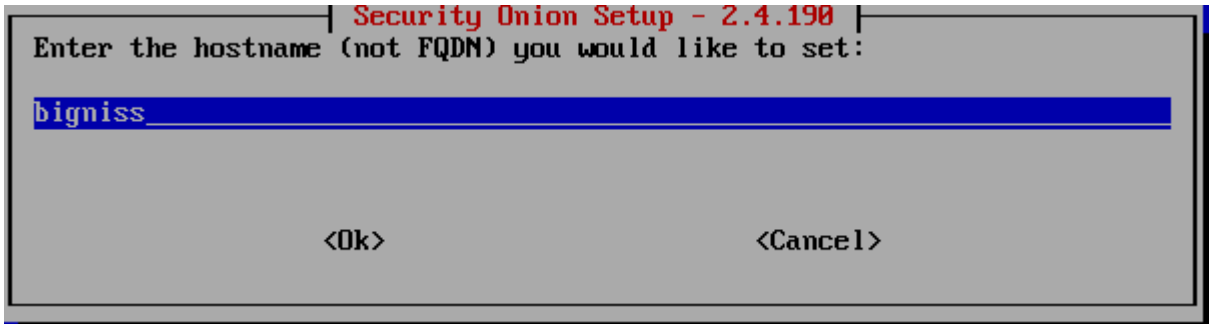
Anexo 5. Tipo de instalación a realizar.



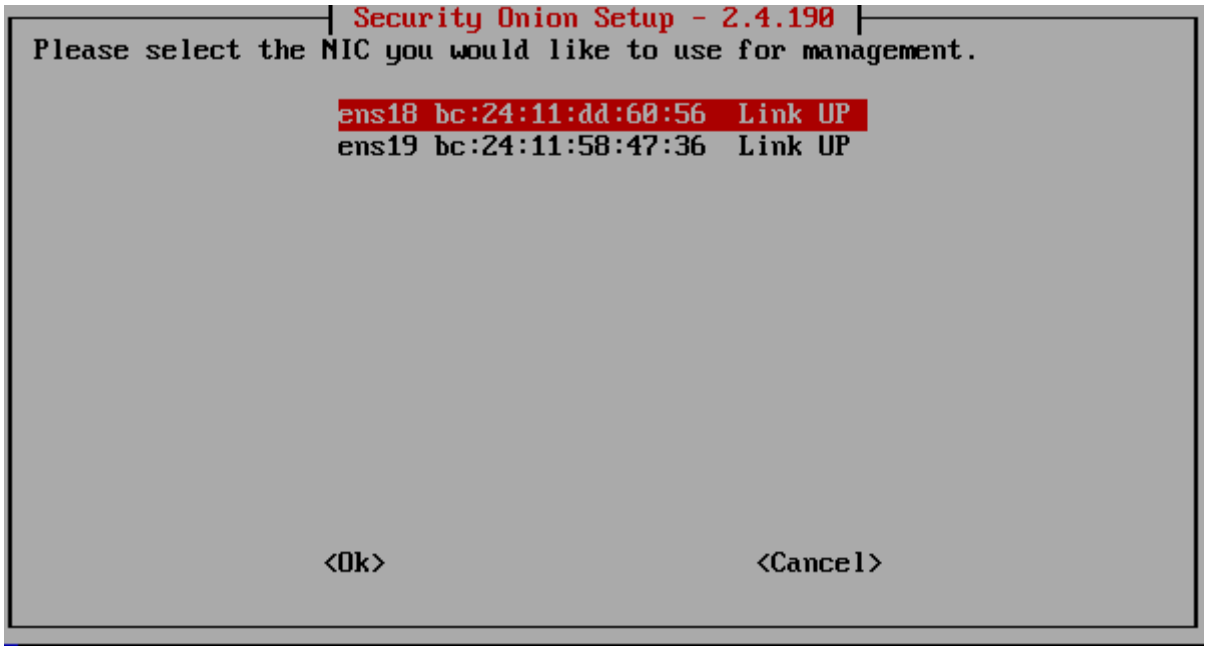
Anexo 6. Aceptación de términos de ELv2



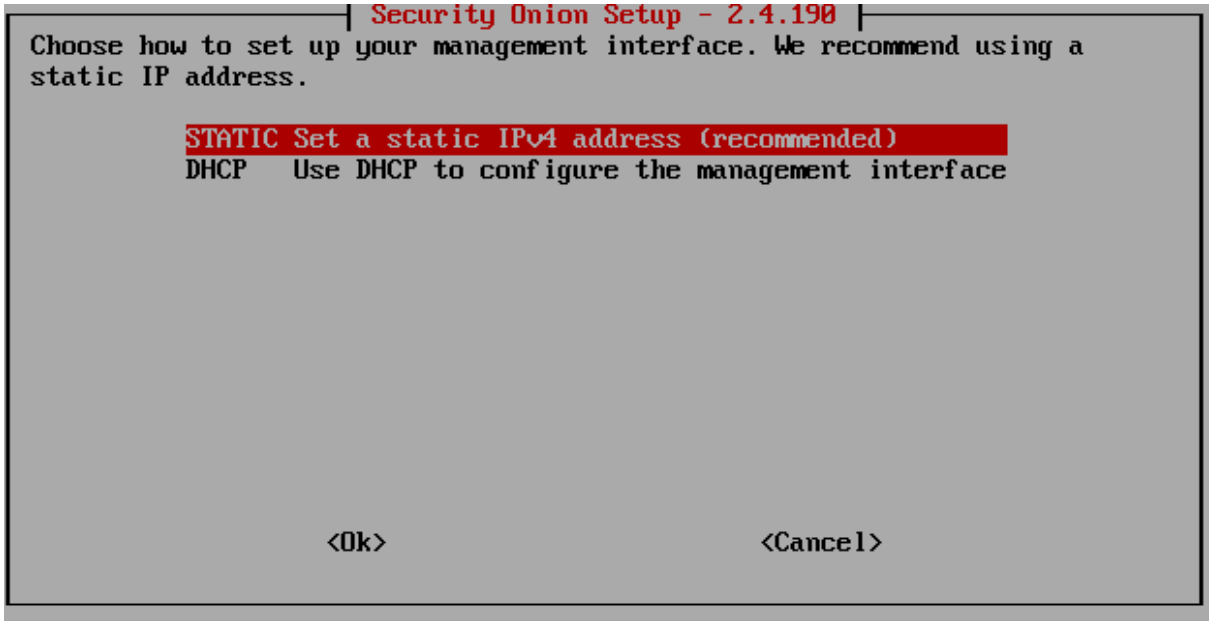
Anexo 7. El modo en el cual el nodo será instalado



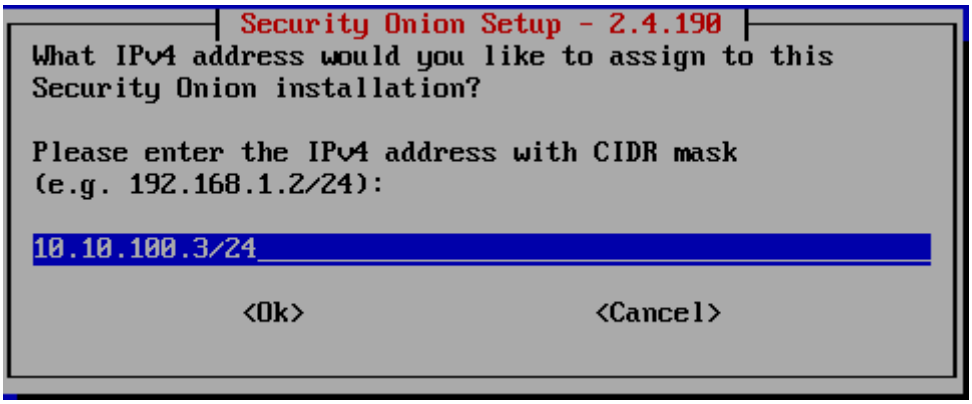
Anexo 8. Asignación de hostname



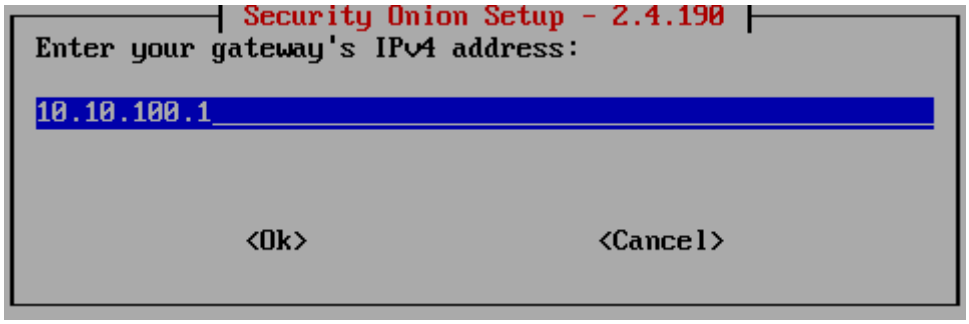
Anexo 9. Asignación de interfaz management.



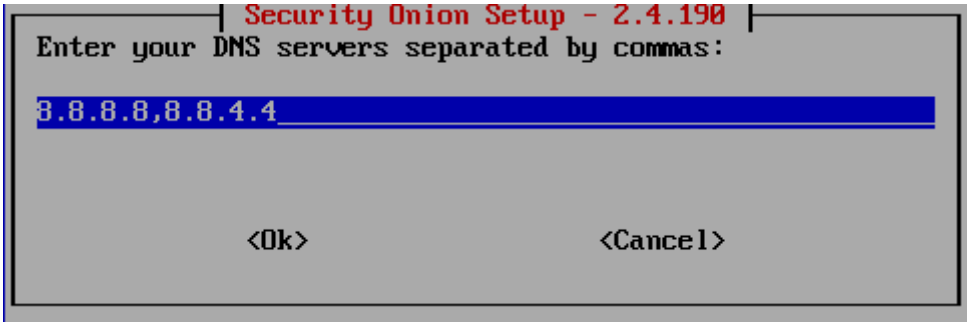
Anexo 10. Selección de dirección IP de la interfaz management.



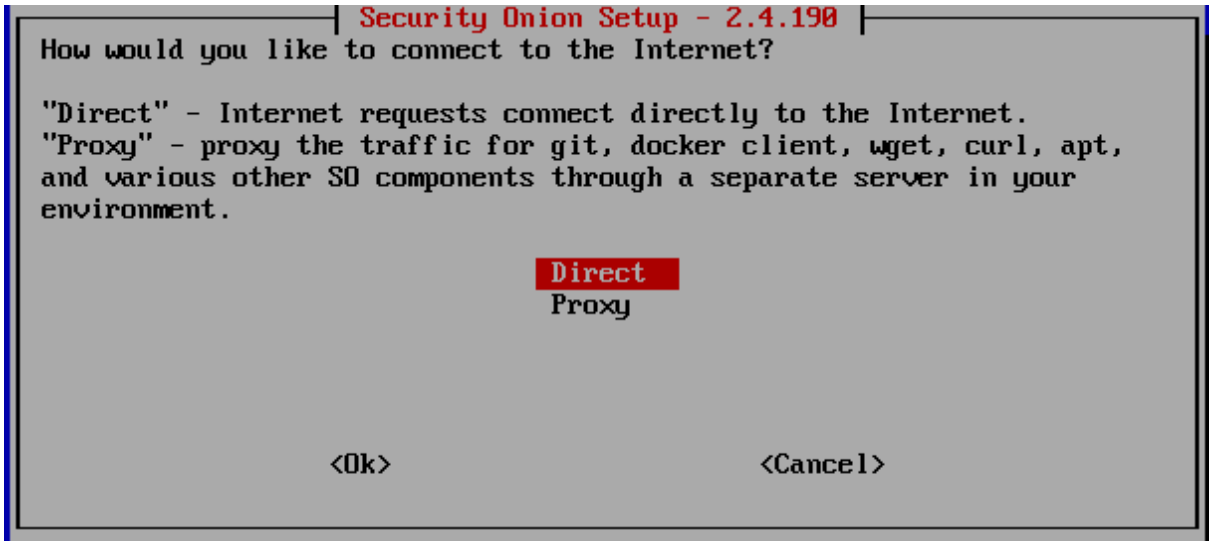
Anexo 11.



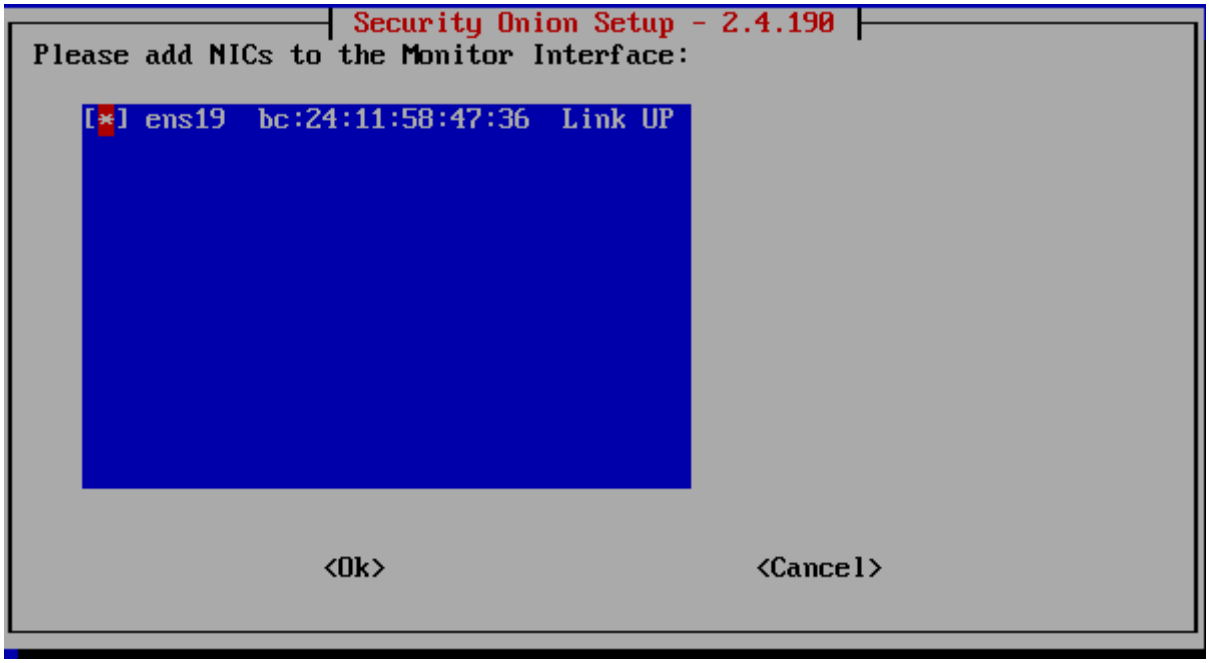
Anexo 12. Asignación de Gateway



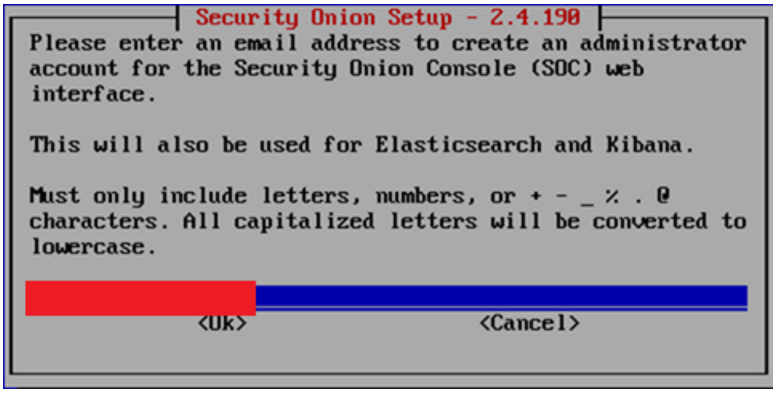
Anexo 13. Designación de servidores DNS



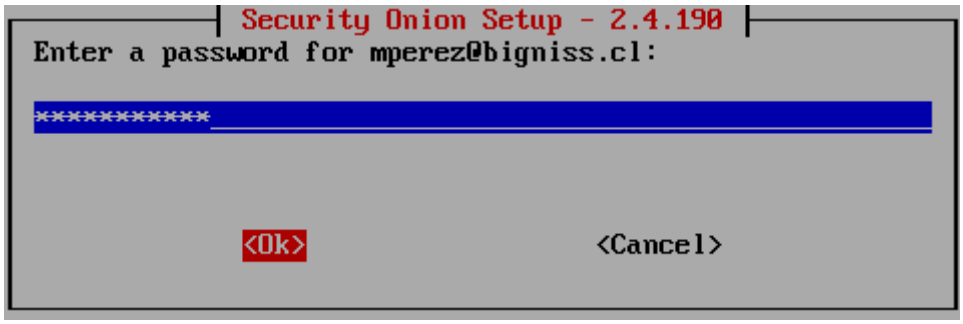
Anexo 14. Tipo de conexión a internet



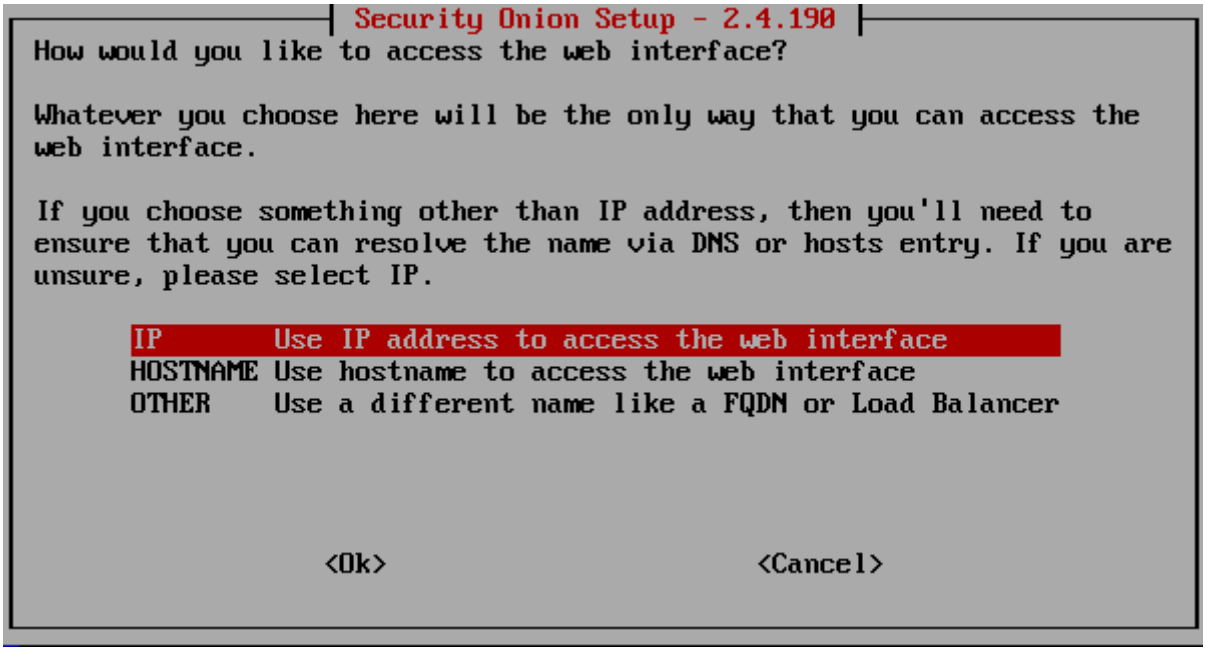
Anexo 15. Asignación de interfaz monitor.



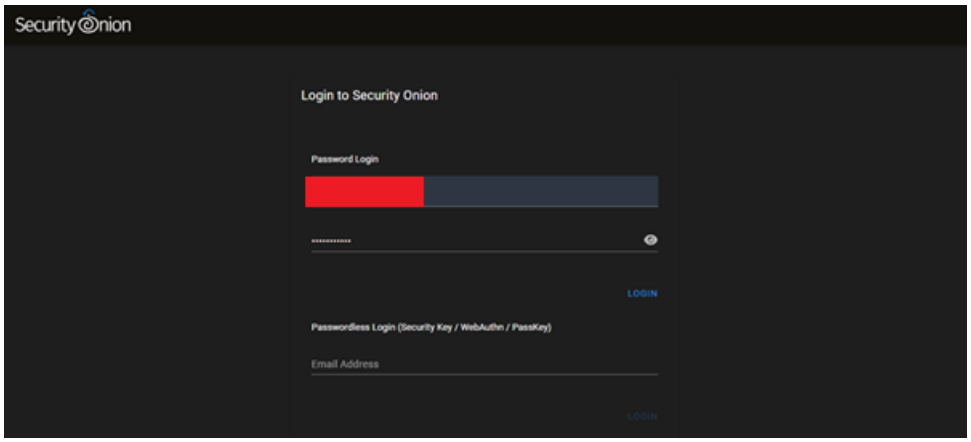
Anexo 16. Creación de cuenta de administrador para la interfaz web



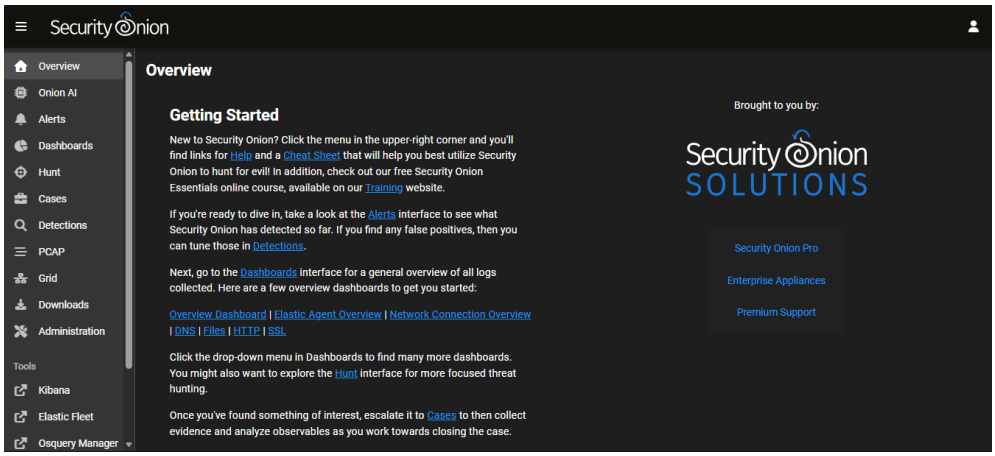
Anexo 17. Creación de contraseña para la cuenta administrador.



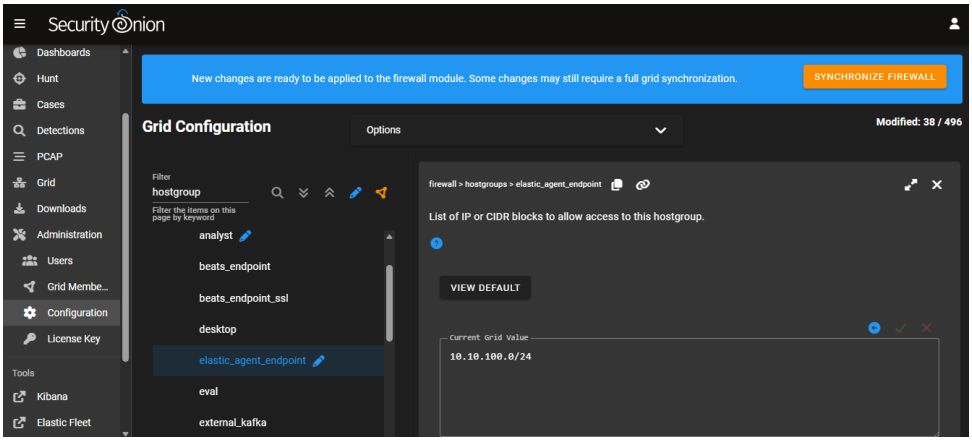
Anexo 18. Seleccionar método de acceso a la interfaz web.



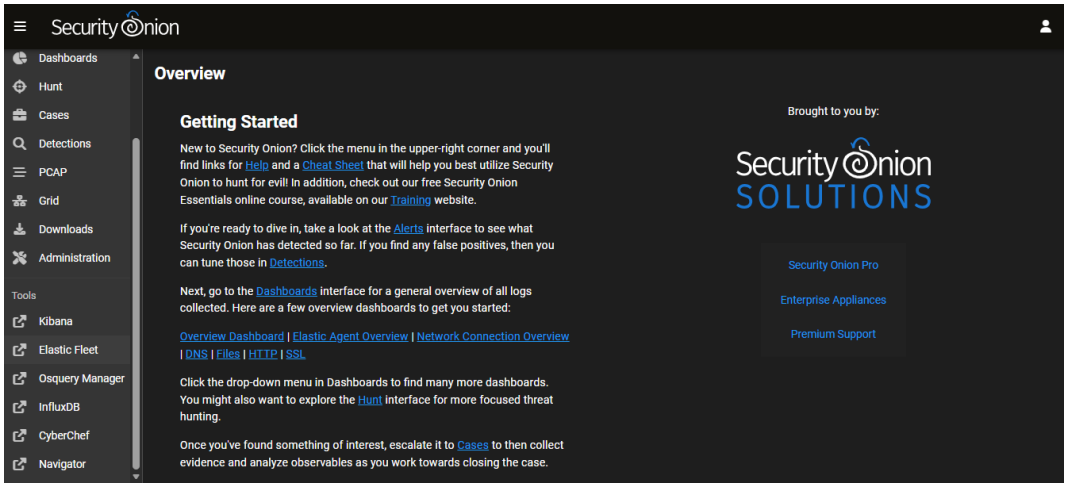
Anexo 19. Panel de inicio de sesión de SO



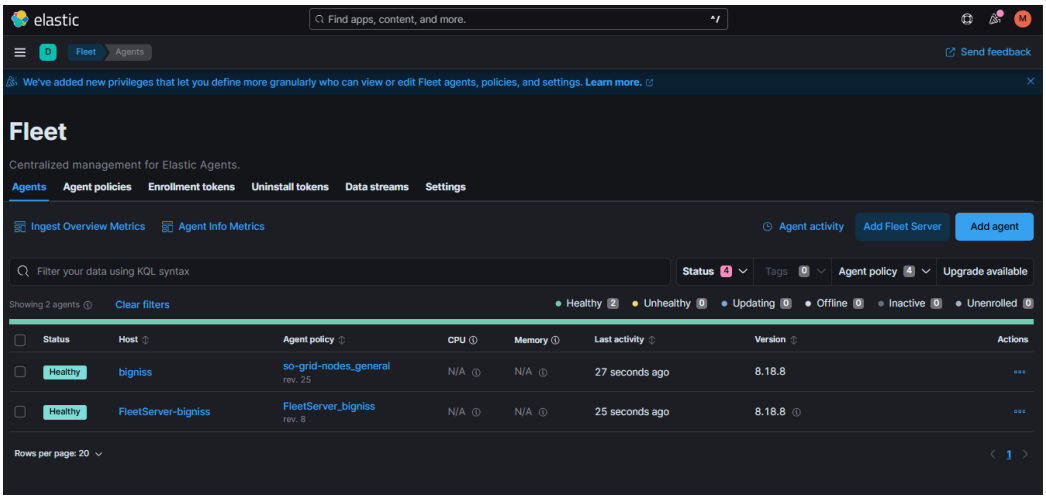
Anexo 20. Consola de Security Onion.



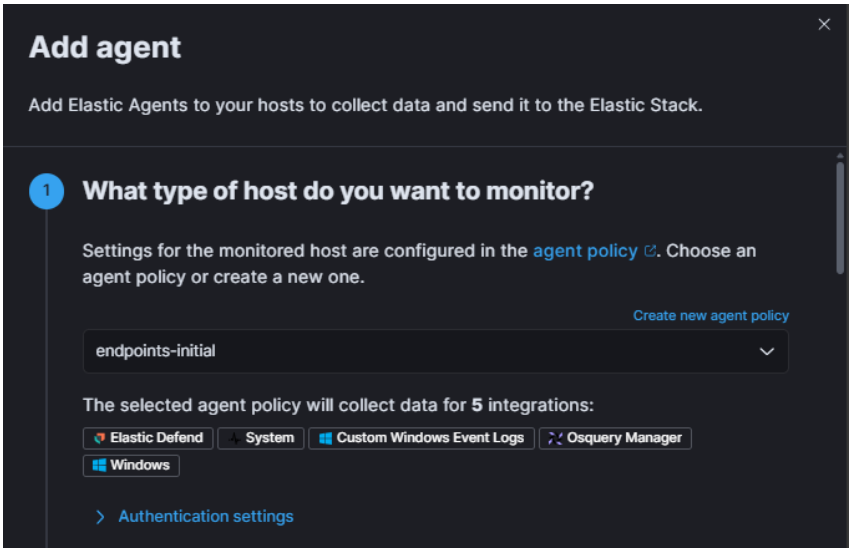
Anexo 22. Adición del agente a la lista blanca y sincronización de configuración



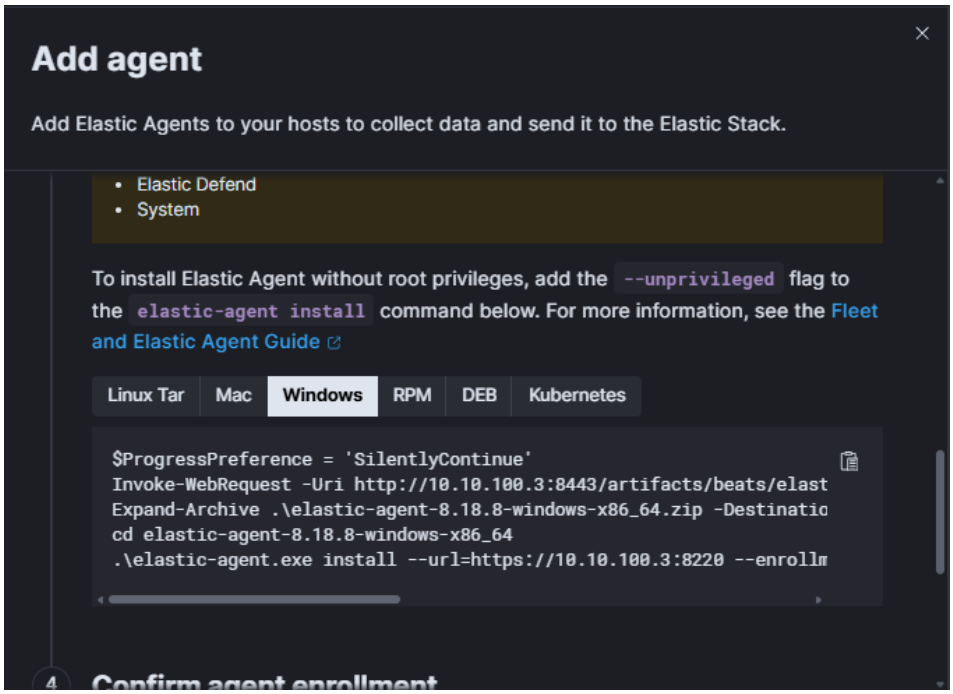
Anexo 21. Selección de Elastic Fleet para incorporación de agentes



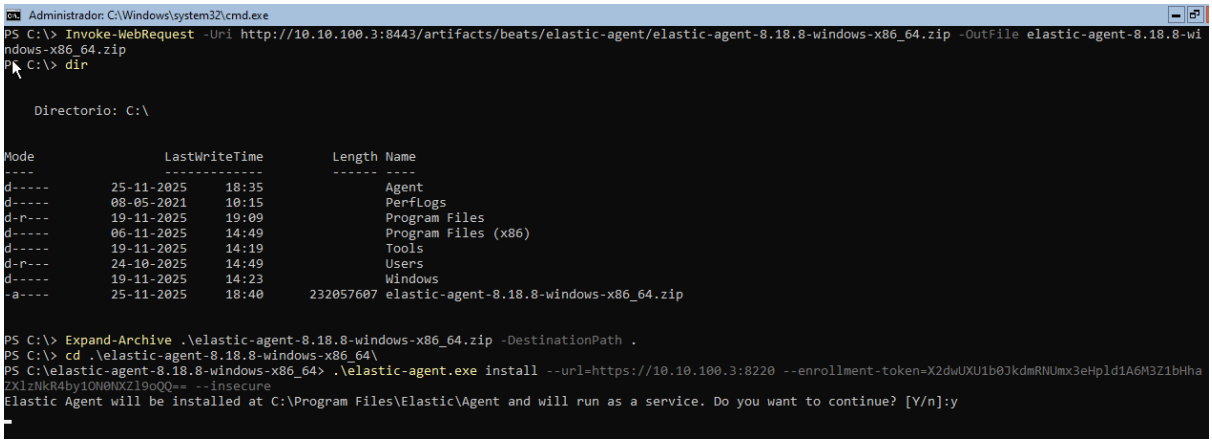
Anexo 23. Menú de configuración de agentes



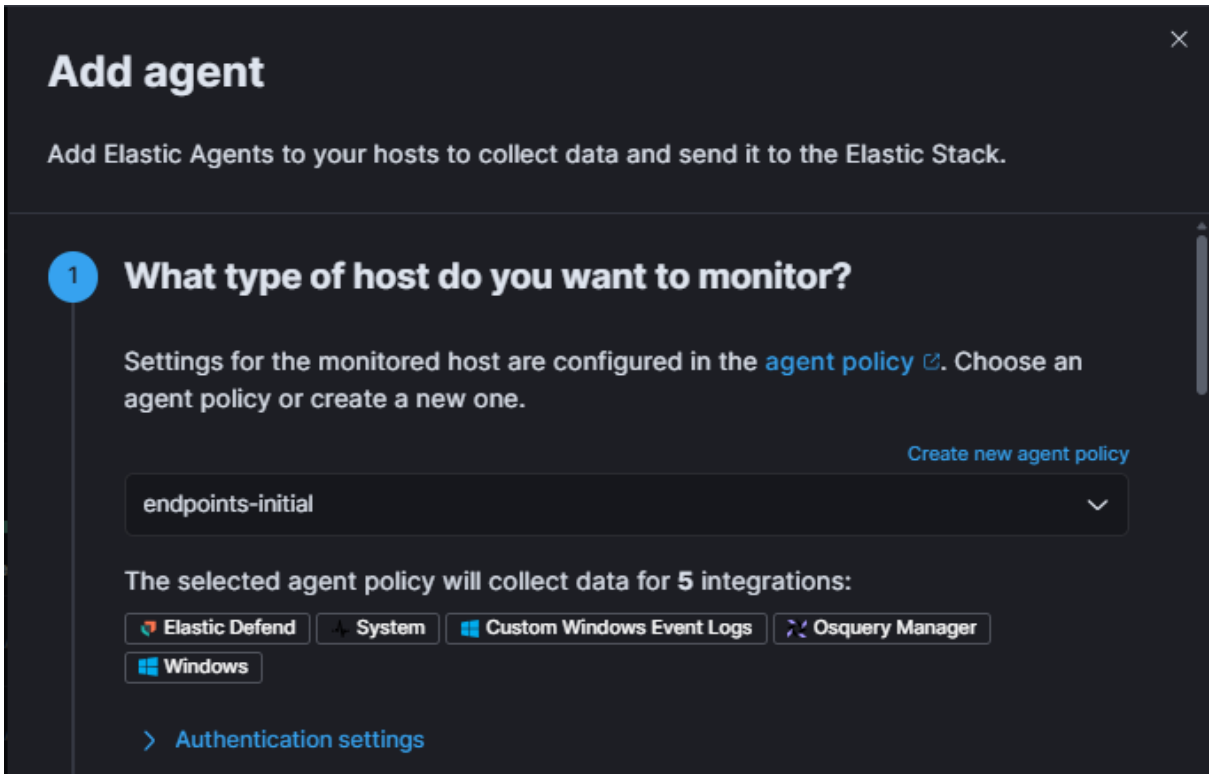
Anexo 24. Selección de política de agente



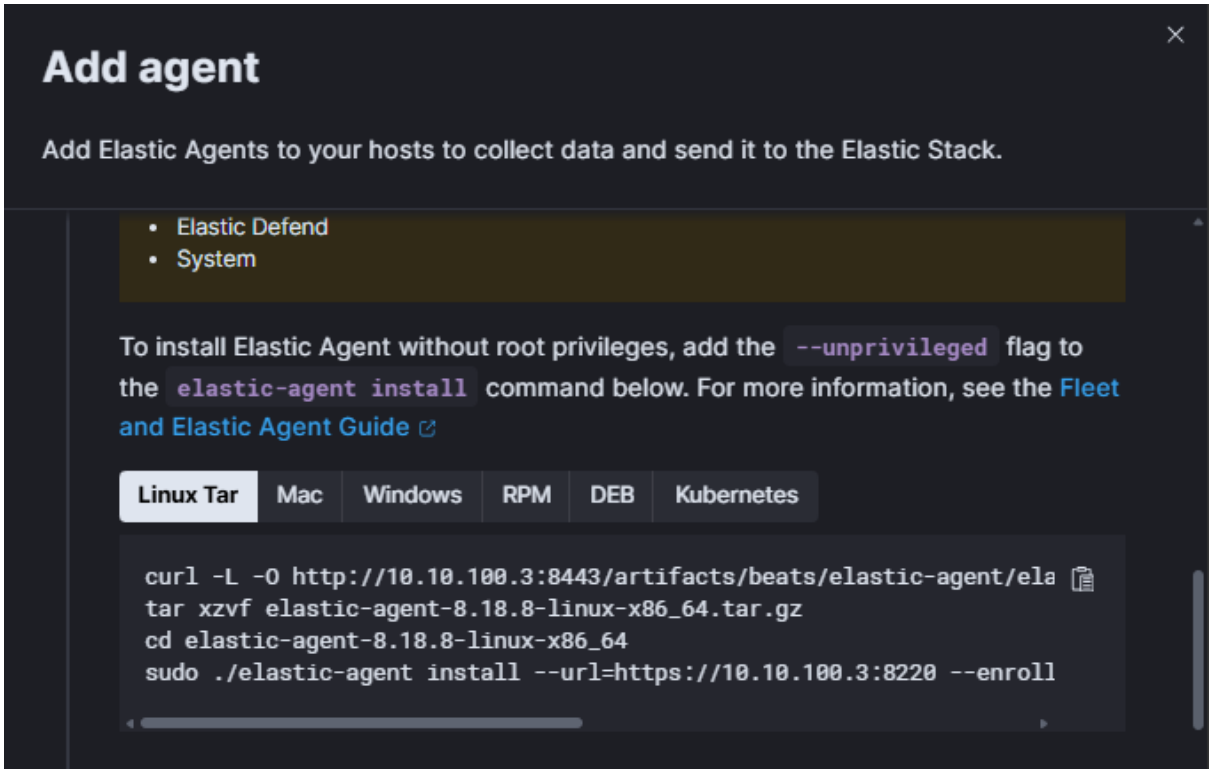
Anexo 25. Selección de sistema operativo del agente



Anexo 26. Instalación de Elastic Agent en Windows.

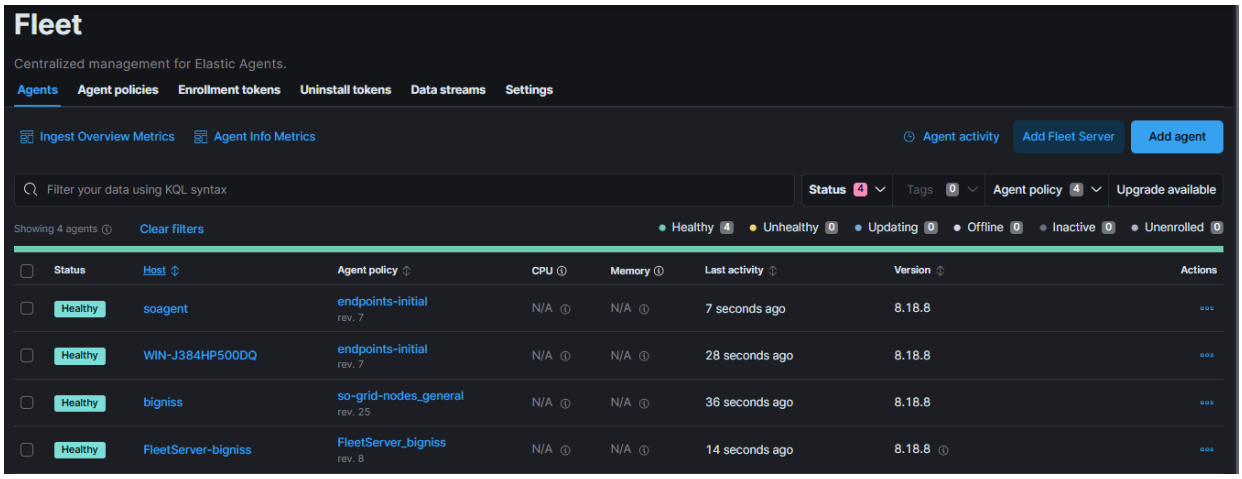


Anexo 27. Selección de política.

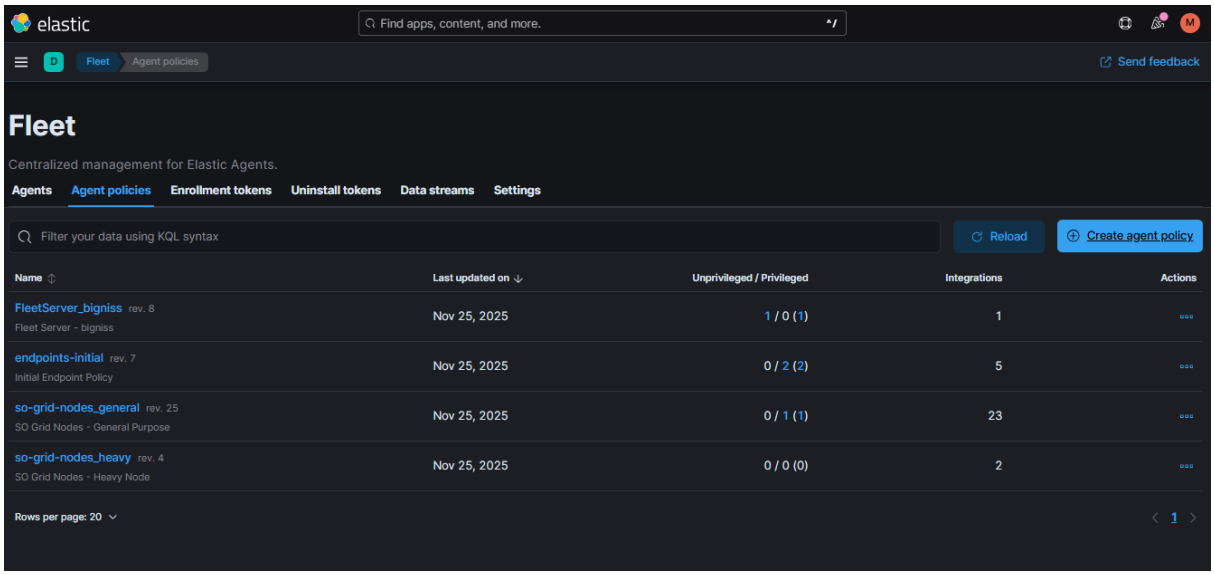


Anexo 28. Script de enrolamiento de Linux

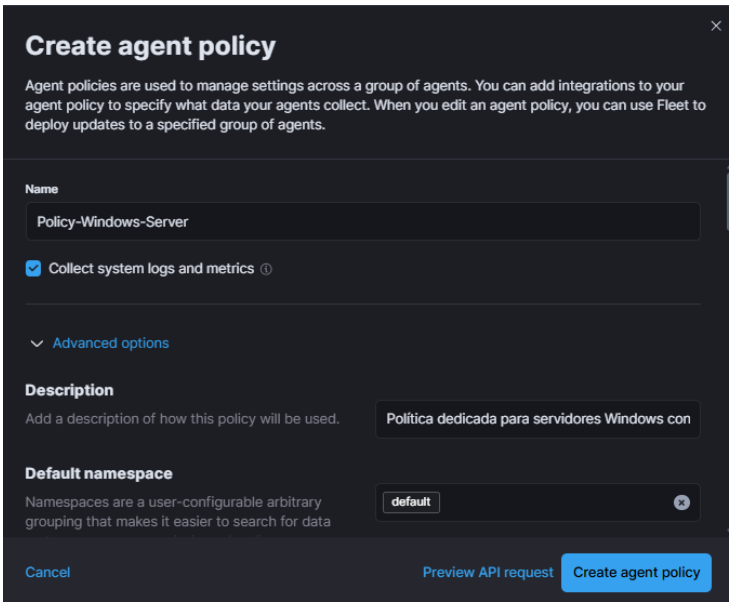
Anexo 29. Confirmación de la instalación exitosa del elastic agent en Linux



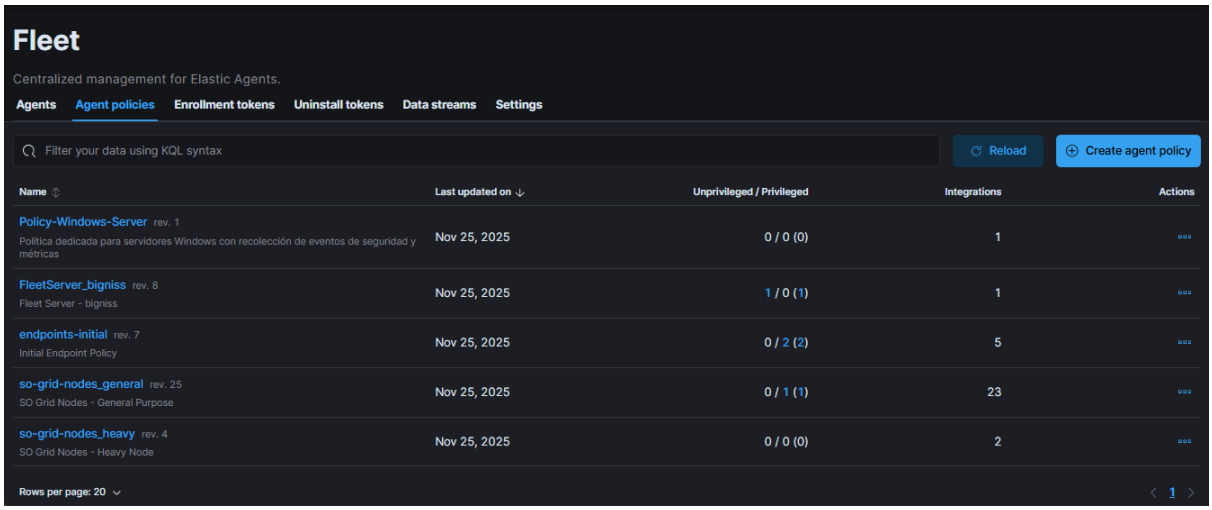
Anexo 30. Tabla donde se pueden ver los agentes



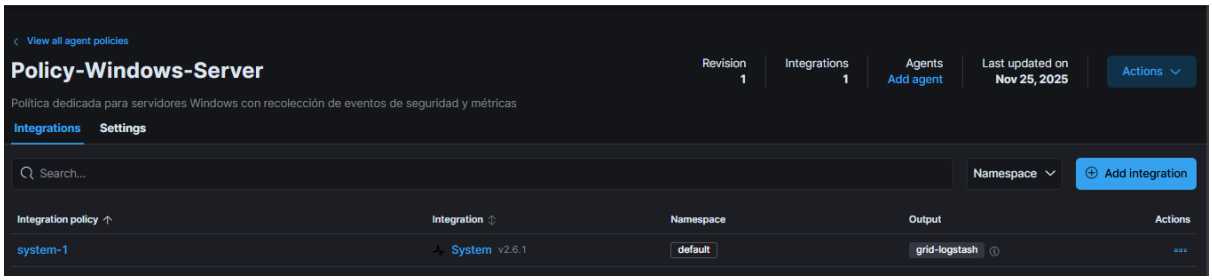
Anexo 31. Creación de nueva política.



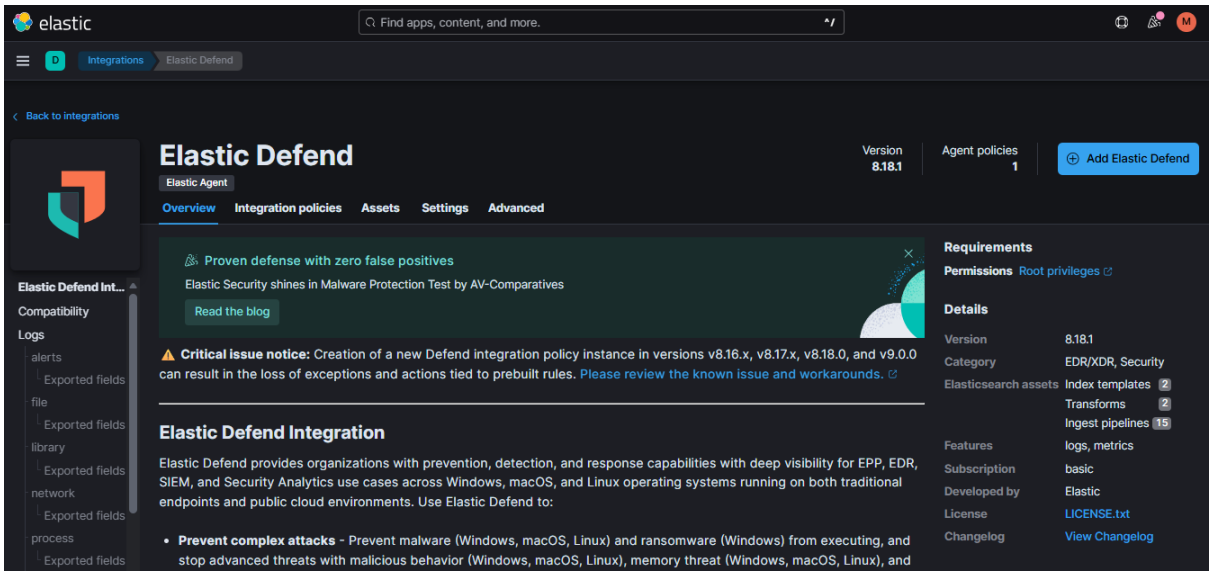
Anexo 32. Creación de política y descripción de esta.



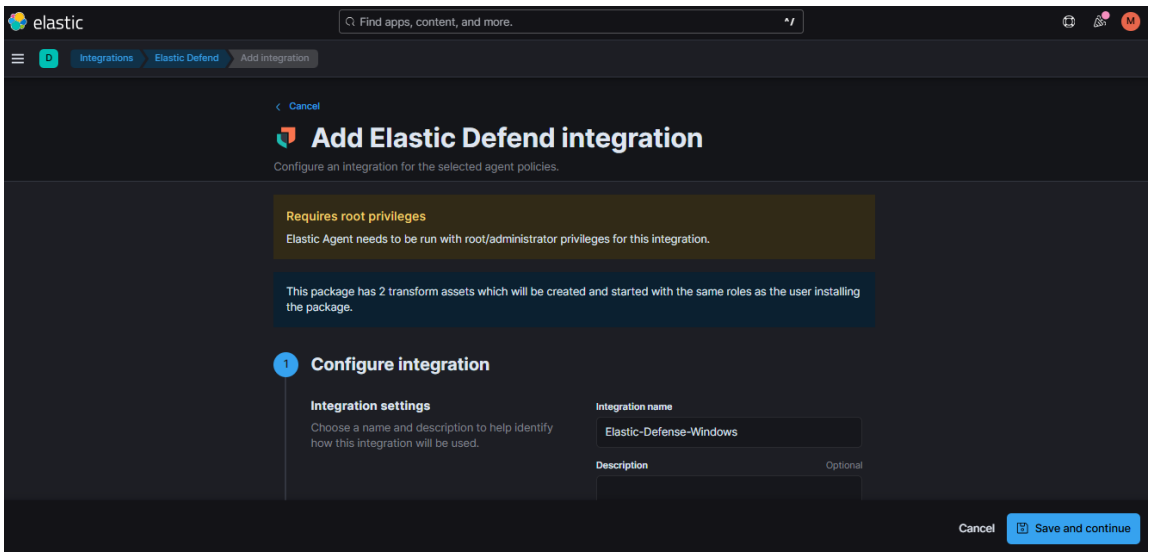
Anexo 33. Modificación de política de agente



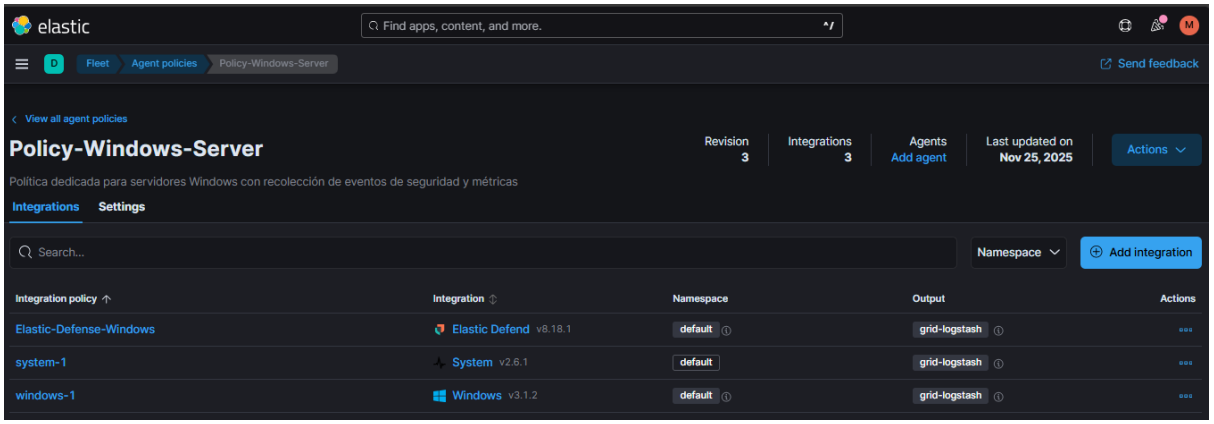
Anexo 34. Integraciones iniciales de la política creada.



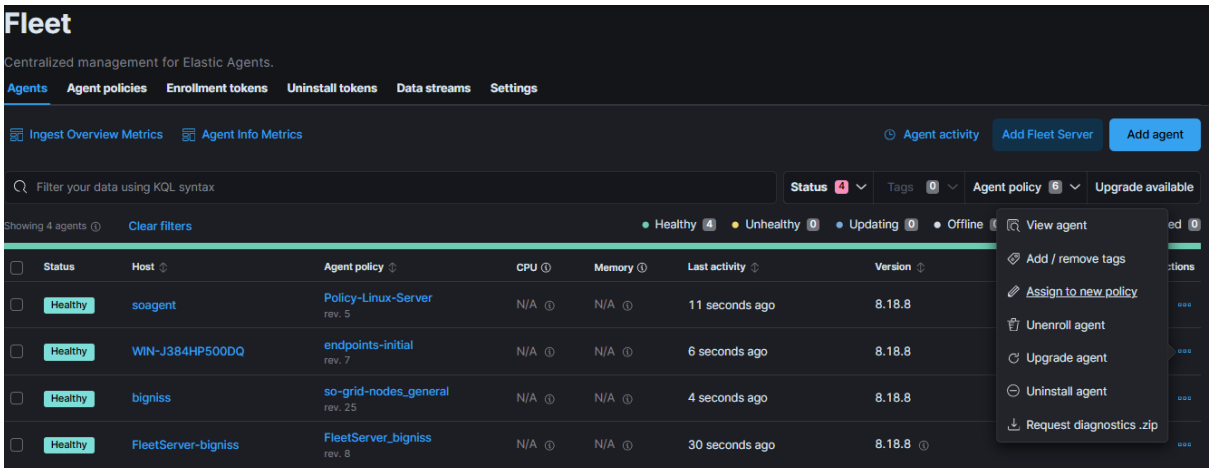
Anexo 35. Elastic Defend. Encargado de detección y respuesta en los endpoints



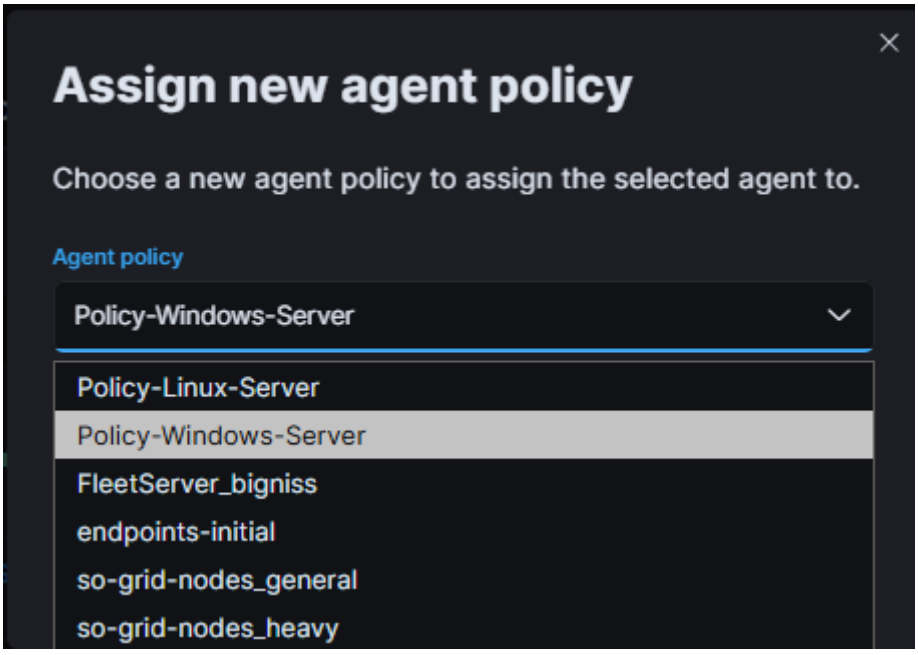
Anexo 36. Configuración de la integración



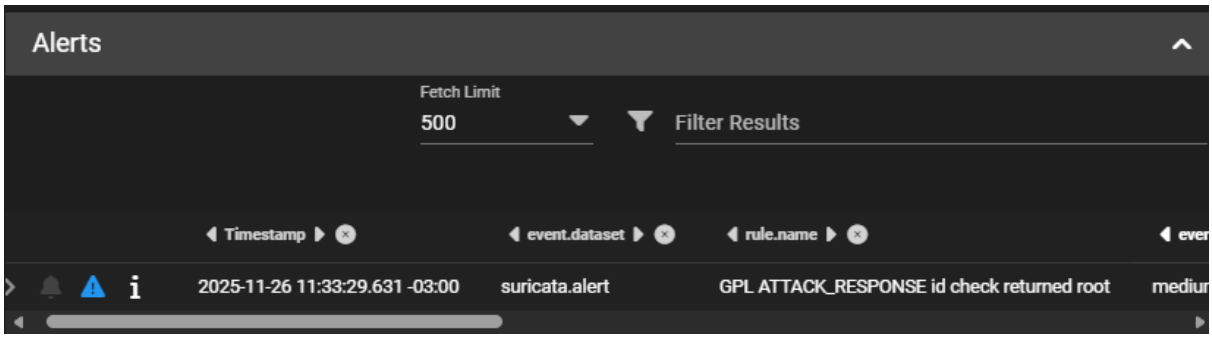
Anexo 37. Vista de integraciones realizadas en una política.



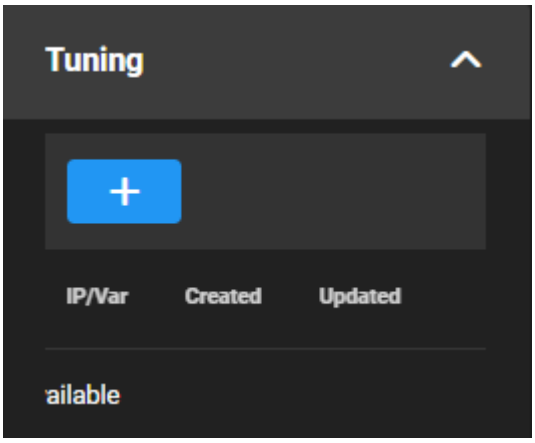
Anexo 38. Cambio de política a cada agente



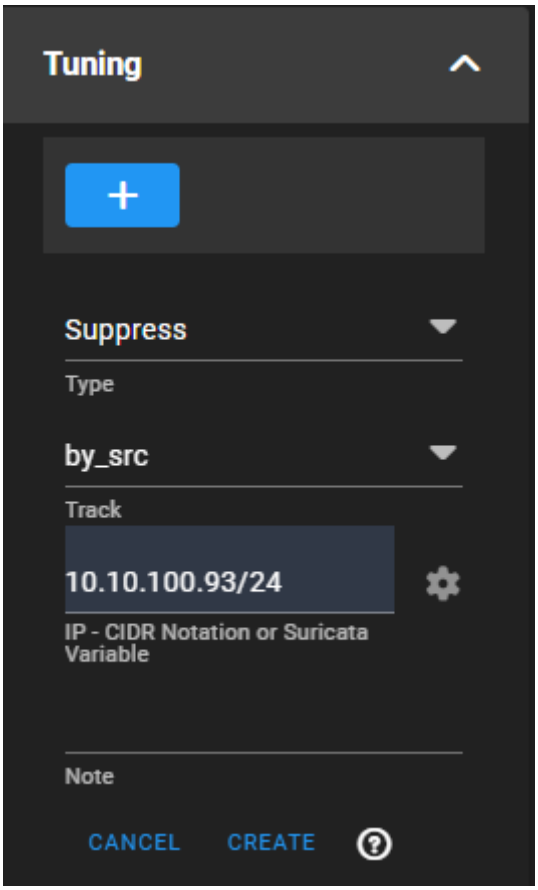
Anexo 39. Asignar la nueva política de agentes a Windows.



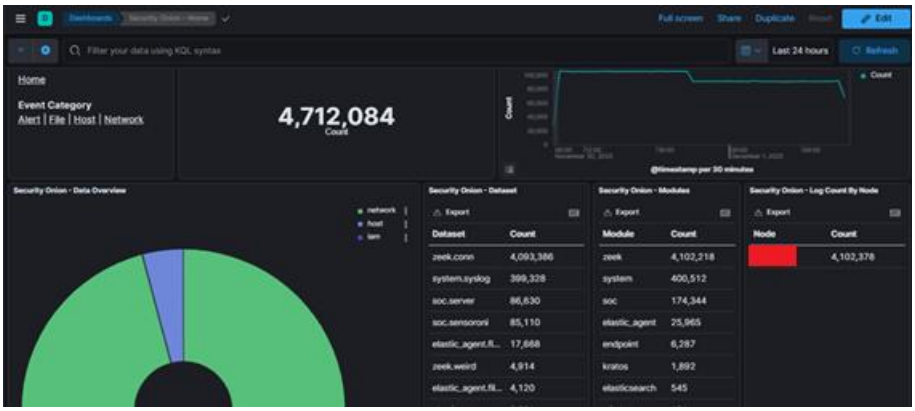
Anexo 40. Alerta de Security Onion con testmyids.



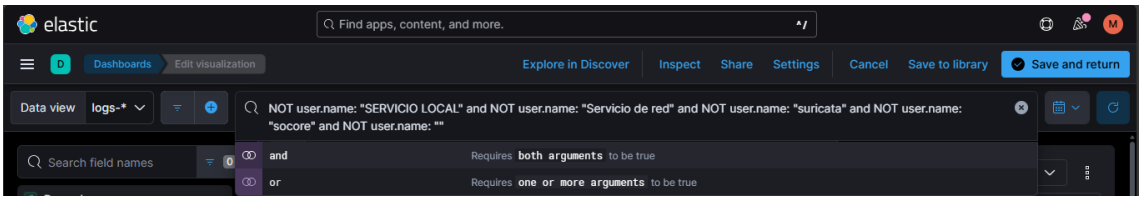
Anexo 41. Creación de regla para silenciar



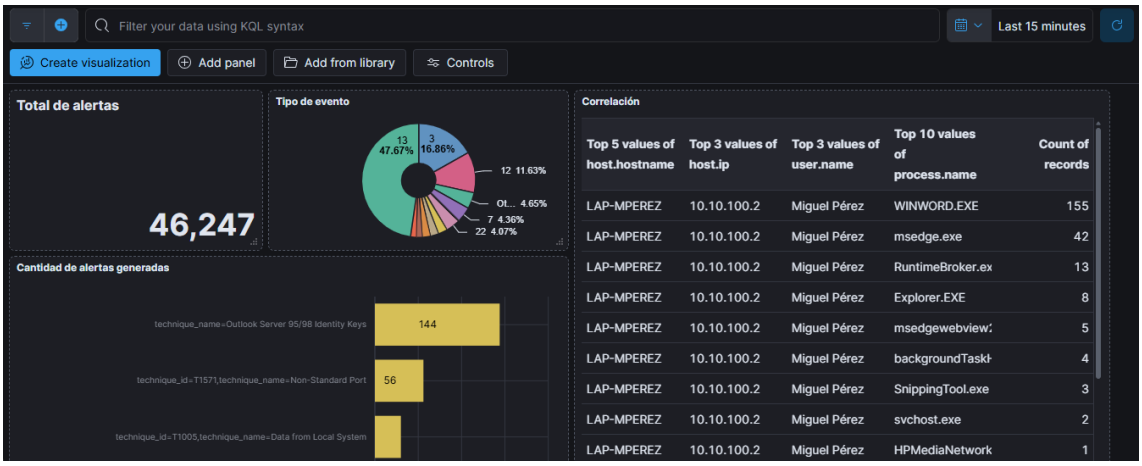
Anexo 42. Ajuste de dicha regla



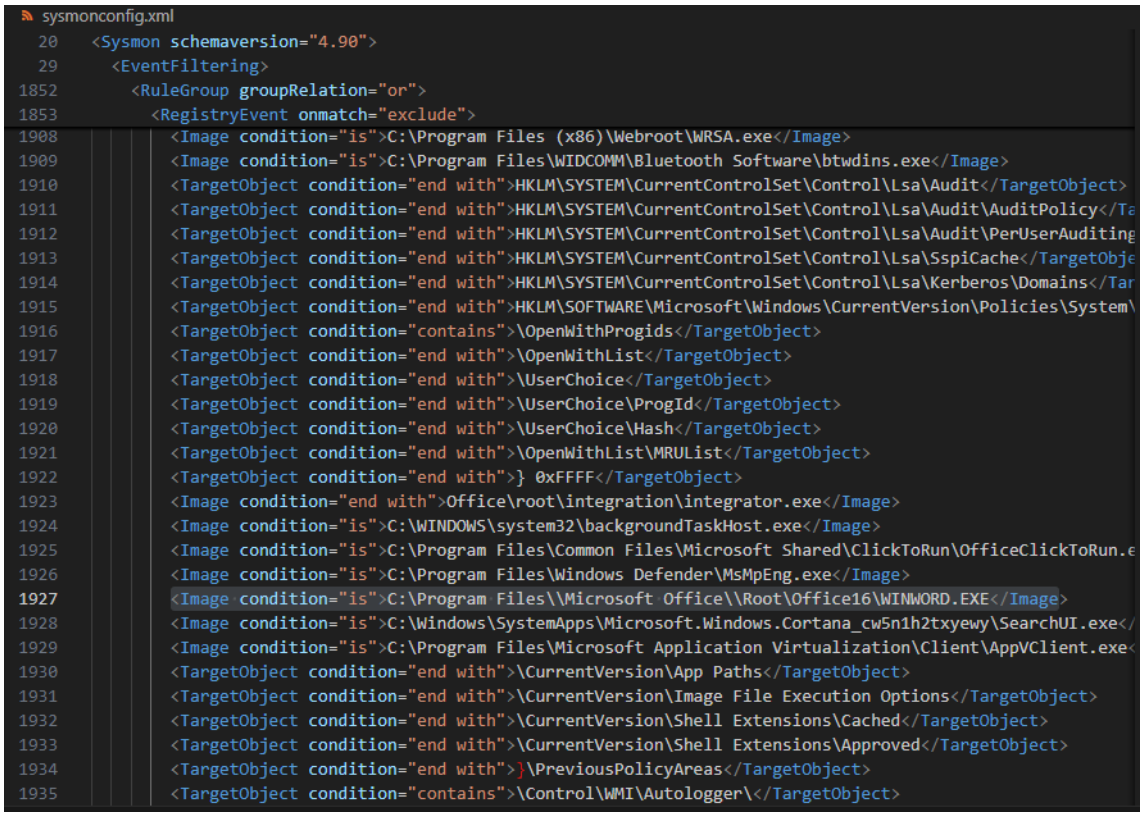
Anexo 43. Muestra del dashboard principal de Kibana.



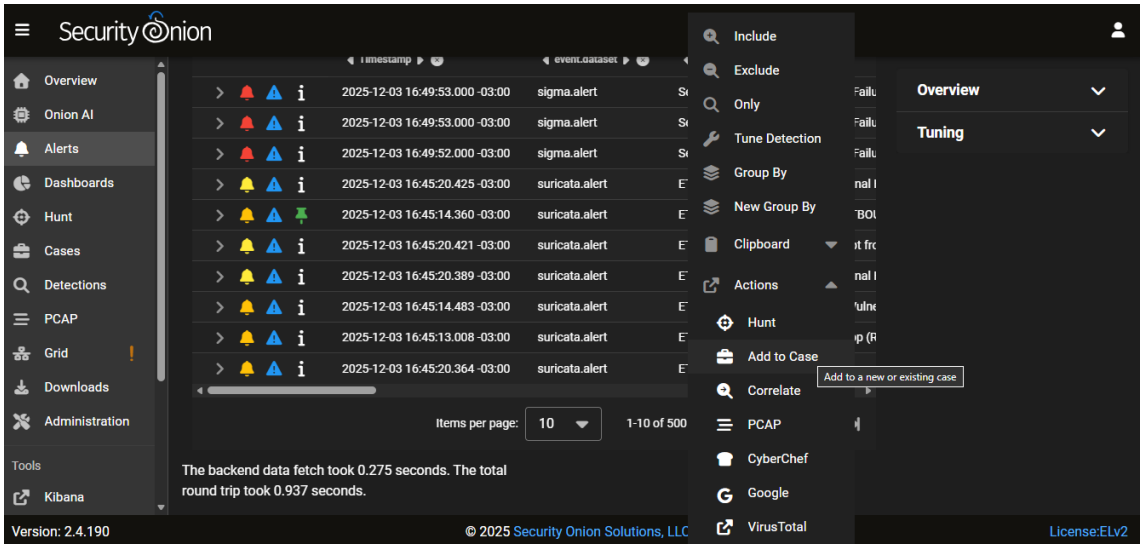
Anexo 44. Hace referencia a las queries que se pueden crear.



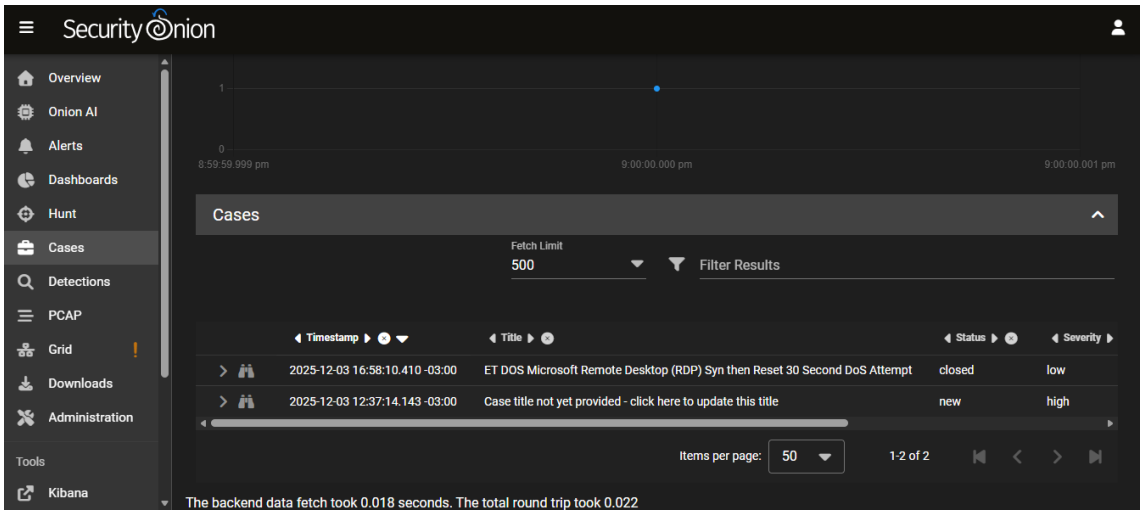
Anexo 45. Dashboard creado desde 0



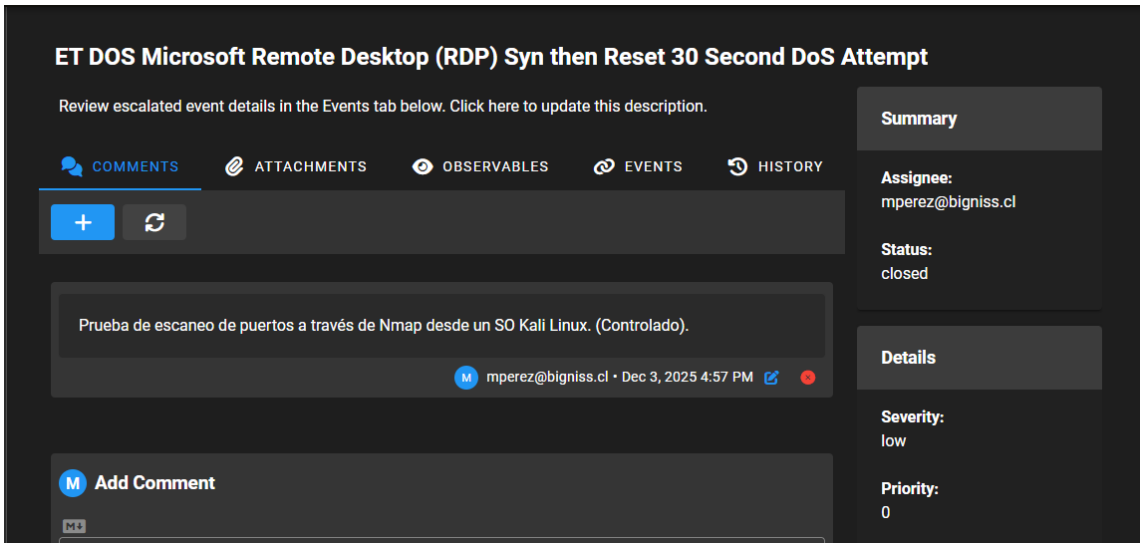
Anexo 46. Configuración de exclusión en Sysmon para optimización de logs.



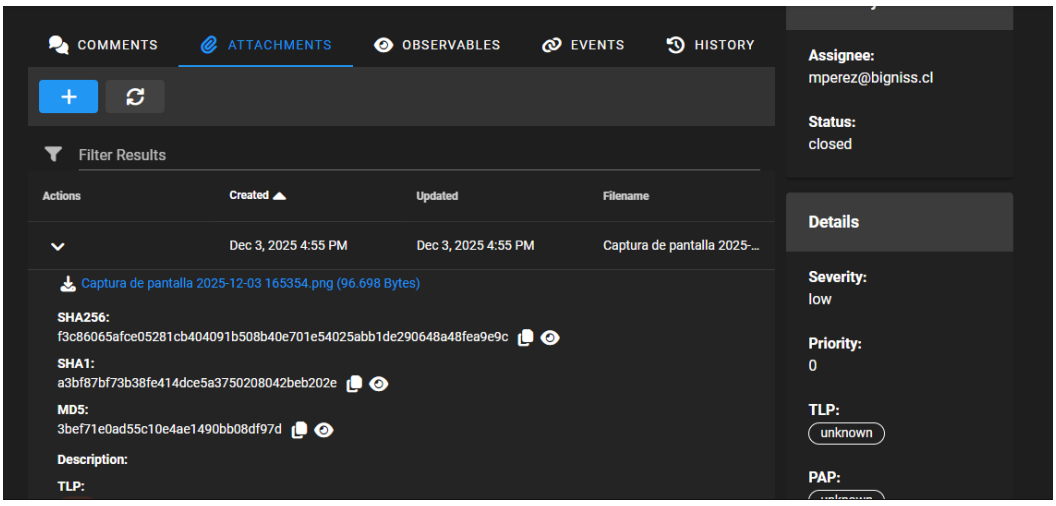
Anexo 47. Opciones de correlación.



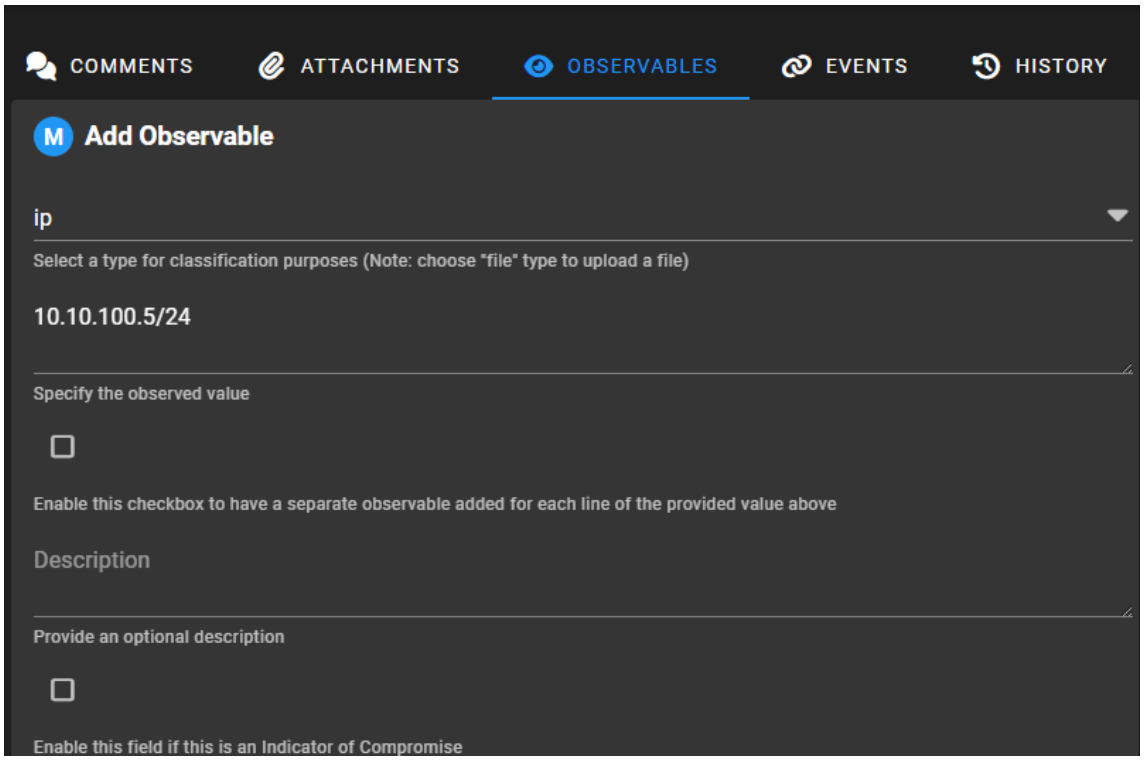
Anexo 48. Pestaña de casos



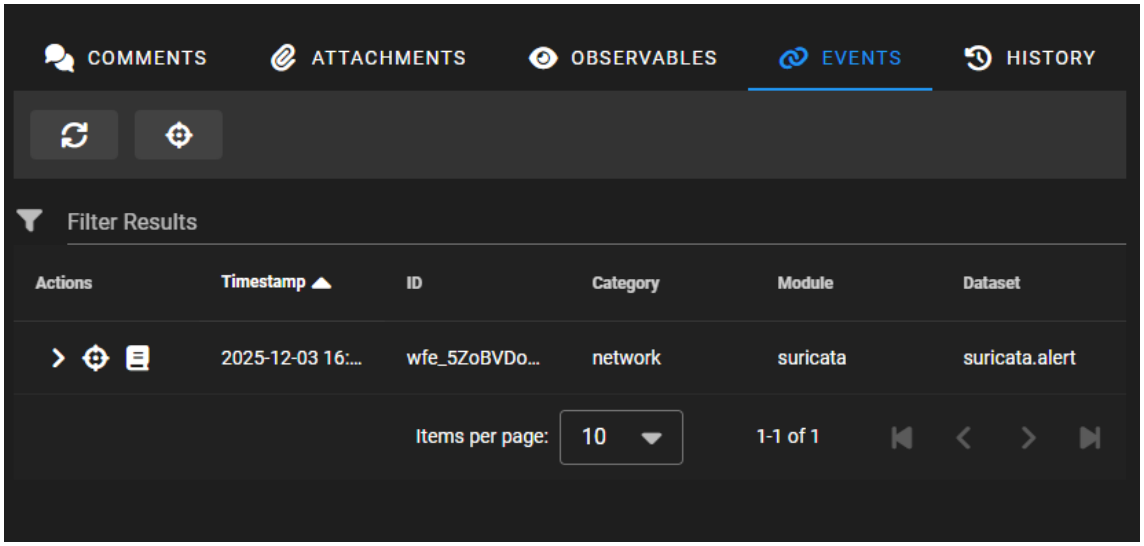
Anexo 49. Caso específico.



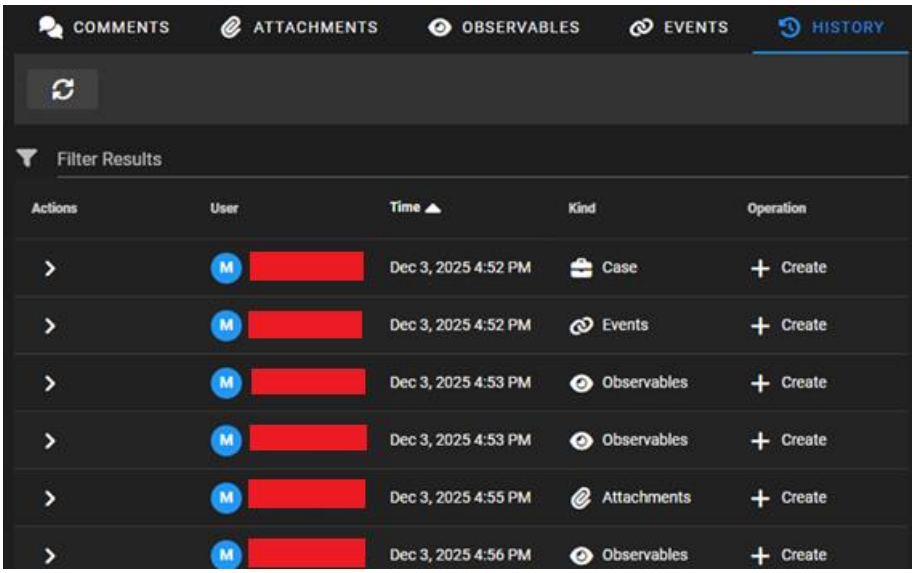
Anexo 50. Pestaña de adjuntado de evidencia.



Anexo 51. Pestaña de observables.



Anexo 52. Pestaña de Eventos relacionados con el caso



Anexo 53. Historial de acciones realizadas en este caso.