ANALISIS Y GESTION DE VULNERABILIDADES DE SISTEMAS INFORMATICOS CON SOFTWARE LIBRE (AGVISL)

José Luis López Fernández

Consultor: Miquel Colobran Huguet

PFC. Ingeniería Informática.

Administración de Redes y Sistemas Operativos.
Septiembre/Enero 2017/18

A Luis Alberto López Alvar por dedicarme parte de su tiempo a compartir y explicarme algunos conceptos de seguridad informática.

RESUMEN

El trabajo desarrollado en este PFC se basa en estudiar como en las pequeñas y medianas empresas se podría gestionar la protección ante las vulnerabilidades informáticas utilizando herramientas open source. De todos es conocida la importancia que ha adquirido la seguridad informática en todos los ámbitos de nuestra vida diaria; la seguridad se ha convertido en un factor crítico en el mundo empresarial dónde a menudo podemos escuchar y leer noticias sobre como ataques informáticos dejan inoperativos algunos negocios originando pérdidas económicas y de servicios a los clientes ; sirva de ejemplo uno de los últimos ataques, el ataque de ransomware Wanacray, realizado a escala mundial hace unos meses. El trabajo se centra en el uso de herramientas open source con objeto de aminorar la carga económica que supone el establecimiento de medidas de seguridad aun cuando también es cierto que a veces el ahorro que se consigue por un lado se desvanece por otro ya que las herramientas de seguridad normalmente necesitan de personal experto para ser manejadas o incluso si no se dispone de él la necesidad de externalizar el servicio.

En el desarrollo del trabajo se utiliza un enfoque formal de implantación de medidas de seguridad adaptándose lo máximo posible a los modelos actuales centrados en el SGSI empresarial, su implantación técnica en la parte informática a través de un SOC basado en OSSIM y teniendo en cuentas las pautas de normas y controles de la serie ISO/IEC 27001.

ÍNDICE

ÍNDICE DE FIGURAS

1. PLANIFICACIÓN PFC.

1.1 Justificación.

Con el crecimiento del uso de las tecnologías TIC en las empresas y la aparición de los problemas de seguridad que conlleva el uso de estas tecnologías, hoy en día se hace imprescindible el uso de técnicas que permitan controlar la seguridad de los mismos ya que un funcionamiento defectuoso de los mismos puede ocasionar un mal funcionamiento del sistema informático y por ende pérdidas económicas para las empresas. Los riesgos a que se encuentran sometidos los sistemas informáticos son muy variados. Este proyecto se centra en el análisis y gestión de uno de ellos, las vulnerabilidades de los sistemas hardware y software, incluyendo en este último no solamente el software de base sino también las distintas aplicaciones informáticas que se usan en una empresa.

A diario las empresas están sometidas a riesgos que ponen en peligro la integridad de la información que manejan y con ello la viabilidad de las mismas. Estos riesgos no solamente provienen del exterior sino también del interior de las mismas organizaciones por lo que para trabajar de forma segura se necesita asegurar los datos y la información de valor con ayuda de un Sistema de Seguridad de la Información.

Hoy en día a nadie se le escapa la importancia de los sistemas de seguridad en las empresas ya que servicios críticos tales como los servicios financieros, el control de la producción, suministro eléctrico, sanidad, abastecimiento de agua, gas... por citar algunos están soportados por sistemas y redes informáticas por lo que las empresas, a parte por supuesto de cumplir con la legalidad jurídica, necesitan Sistemas de Seguridad de la información. Estos sistemas persiguen, como se indica en la norma ISO/IEC 17799 preservar la confidencialidad, integridad y disponibilidad de datos y servicios. En este PFC se estudian algunas herramientas open source para llevar a cabo el estudio de una de las tareas que permiten conseguir los objetivos citados, el análisis de vulnerabilidades.

Se entiende por vulnerabilidad cualquier debilidad en el sistema informático que pueda permitir a las amenazas causarle daños y producir pérdidas en la organización.[3]

Los test de vulnerabilidades tienen como objetivo descubrir el mayor número posible de fallos de seguridad dentro de un sistema informático, con la finalidad de establecer una lista de prioridades que mitiguen las vulnerabilidades encontradas.

Los test de vulnerabilidades (tanto internos como externos) se consideran como el paso inicial para conocer el nivel de seguridad de la infraestructura informática de una organización lo que permite conseguir una visión global y específica de los errores/debilidades que se tienen que corregir y por lo tanto asegurar los activos informáticos de la organización.

Por lo tanto el punto de partida sería el de una organización que va a implantar por primera vez sistemas de seguridad informáticos (y uno de los aspectos a tener en cuenta son las vulnerabilidades informáticas) o bien de un pequeña/mediana empresa que ya posee algunos medios (zonas DMZ, cortafuegos, antivirus...). En ambos casos se pretende implantar un SGSI con el doble objetivo de hacer frente a las vulnerabilidades y el de cumplir con las normativas legales.

1.2 Objetivos.

El principal objetivo de este proyecto se centra en fijar un marco y normas de trabajo a seguir en el mantenimiento de la seguridad de un sistema informático y concretamente en las herramientas que se utilizan en el análisis y gestión de vulnerabilidades.

Con objeto de alcanzar el objetivo indicado se estudiarán herramientas para detectar y analizar vulnerabilidades en aquellos casos que se presenta más frecuentemente en los sistemas informáticos, tanto sistemas hardware como software. Se tendrá en cuenta para la realización práctica casos de uso que se presentan frecuentemente en las empresas.

Un segundo objetivo consistirá en que una vez detectadas las vulnerabilidades de un sistema se procede a realizar el trabajo de implementar las salvaguardas que disminuyan sus efectos para lo que se empleará como ayuda algunas de los numerosos repositorios de vulnerabilidades que existen tales como el repositorio de INCIBE o el NVD (National Vulnerability Database) del NIST.

1.3 Enfoque y método seguido.

El alcance de este PFC se centra concretamente en la gestión y análisis de vulnerabilidades así como aspectos relacionados con la misma tales como análisis de riesgos, tratamiento, seguimiento y monitorización de las vulnerabilidades de un sistema informático.

Se ha realizado un desarrollo metódico del trabajo con objeto de alcanzar los objetivos propuestos. Así se realiza un estudio lo más riguroso posible , dentro de las limitaciones de un PFC, teniendo en cuenta las normas legales existentes y los modelos de seguridad generalmente aceptados en las empresas, que hoy en día son los que giran en torno al SGSI (Sistema de Gestión de la Seguridad de la Información). Este método permite encuadrar el estudio de vulnerabilidades en el marco concreto que le asigna el SGSI lo que facilita su análisis, interpretación y su interacción con sistemas de bases de datos externas que ayudan a la detección y prevención de vulnerabilidades. También cabe citar que el usar marcos estandarizados por normativas ayuda a la empresa a cumplir con los requisitos legales en materia de seguridad.

El enfoque adoptado obliga a una manera de desarrollar las fases del PFC que se describen en el apartado 1.6.

La solución adoptada puede emplearse tanto si la empresa ya tiene implantado un Sistema de Gestión de Seguridad de la Información como si carece de él y hubiese que implementarlo.

1.4 Planificación temporal.

A continuación se muestran la secuencia de tareas a realizar así como el diagrama de Gantt correspondiente. El diagrama se ha realizado con el programa GanttProject.

Nombre	Fecha de inicio	Fecha de fin
FASE 1 : PLANIFICACION TFC	20/09/17	5/10/17
Tarea 1.1 : Motivación	20/09/17	22/09/17
Tarea 1.2 : Objetivos.Alcance	25/09/17	29/09/17
Tarea 1.3 : Planificación Temporal	2/10/17	5/10/17
Tarea 1.4 : Entrega PEC1	6/10/17	6/10/17
FASE 2: ESTUDIO,PLANIFICACION Y DISEÑO DE UN CENTRO DE SEGURIDAD	9/10/17	30/10/17
Tarea 2.1 : Introducción SGSI. NORMA UNE-ISO/IEC 27001	9/10/17	13/10/17
Tarea 2.1.1 : Objetivos de un Sistema de Segiuridad Informática	9/10/17	10/10/17
Tarea 2.1.2: El Sistema de Gestión de Riesgos. Amenazas y Vulnerab...	11/10/17	11/10/17
Tarea 2.1.3 : Marco Legal y Jurídico	12/10/17	13/10/17
Tarea 2.2 : Vulnerabilidades Informáticas	16/10/17	20/10/17
Tarea 2.2.1 : Introducción	16/10/17	16/10/17
Tarea 2.2.2 : ¿Qué es una vulnerabilidad?	17/10/17	18/10/17
Tarea 2.2.3 : Parámetros que definen una vulnerabilidad	17/10/17	18/10/17
Tarea 2.2.4 : ¿ Dónde se encuentran las vulnerabilidades ?	19/10/17	19/10/17
Tarea 2.2.5 : Estándares de Vulnerabilidades	19/10/17	20/10/17
Tarea 2.2.6 : Análisis y Gestión de Vulnerabilidades	20/10/17	20/10/17
Tarea 2.3 : Sistemas SIEM	23/10/17	26/10/17
Tarea 2.3.1: ¿Qué es un SIEM?. ¿Qué son los Logs?	23/10/17	25/10/17
Tarea 2.3.2: Capacidades de un SIEM	25/10/17	26/10/17
Tarea 2.4: Centro de Operaciones de Seguridad(SOC)	23/10/17	30/10/17
FASE 3: ALIENVAULT OSSIM	1/11/17	9/11/17
Tarea 3.1 : Introducción OSSIM.	1/11/17	2/11/17
Tarea 3.2: Instalación y Configuración OSSIM	2/11/17	3/11/17
Tarea 3.3: Funcionalidades de OSSIM	6/11/17	7/11/17
Tarea 3.4: Herramientas Integradas en OSSIM	8/11/17	9/11/17
Tarea 3.5: Entrega PEC 2	10/11/17	10/11/17
FASE 4: PARTE EXPRIMENTAL.CONCLUSIONES	10/11/17	14/12/17
Tarea 4.1: Gestión de Vulnerabilidades con OSSIM. Resumen	10/11/17	10/11/17
Tarea 4.2: Experimento 1. Gestión Vulnerabilidades de una LAN	13/11/17	21/11/17
Tarea 4.3 :Experimento 2. Gestión Vulnerabilidades de una LAN con DMZ	22/11/17	4/12/17
Tarea 4.4: Experimento 3. Gestión de ataques desde la red WAN	5/12/17	11/12/17
Tarea 4.5: Conclusiones	12/12/17	14/12/17
Tarea 4.6: Entrega PEC 3	15/12/17	15/12/17
FASE 5: TAREAS FINALES	20/09/17	2/01/18
Tarea 5.1 : Bilbliografía	20/09/17	2/01/18
Tarea 5.2 : Anexos	18/12/17	22/12/17
Anexo 1: Descripción del ataque realizado con Kali	18/12/17	22/12/17
Anexo 2: Configuración OSSEC	18/12/17	22/12/17
Anexo 3: Configuración Firewall Cisco ASA	18/12/17	22/12/17
Anexo 4: Configuración NDIS OSSIM	18/12/17	22/12/17
Anexo 5: Configuración monitorización de disponibilidad OSSIM	18/12/17	22/12/17
Anexo 6: Configuración Switch LAN	18/12/17	22/12/17
Tarea 5.3: Correcciones/Retoques	25/12/17	26/12/17
Tarea 5.4: Diapositivas PFC	25/12/17	29/12/17
Tarea 5.5: Documentación PFC	20/09/17	2/01/18

Fig. 1.1: Panificación PFC

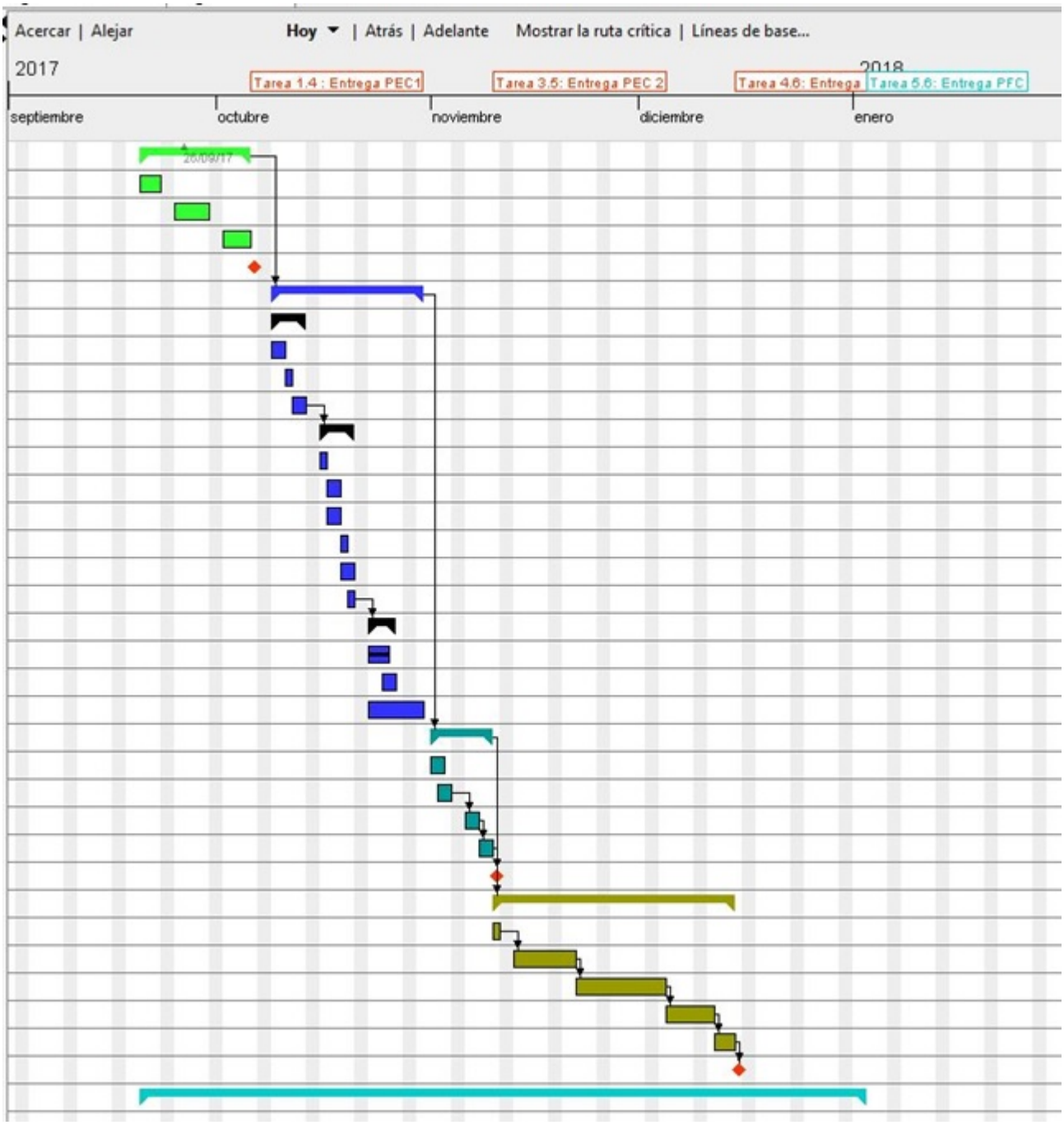


Fig. 1.2: Diagrama de Gantt PFC

1.4.1 Fases del Proyecto y sus Diagramas de Gantt

Nombre	Fecha de inicio	Fecha de fin
FASE 1 : PLANIFICACION TFC	20/09/17	5/10/17
FASE 2: ESTUDIO,PLANIFICACION Y DISEÑO DE UN CENTR...	9/10/17	30/10/17
FASE 3: ALIENVAULT OSSIM	1/11/17	9/11/17
FASE 4: PARTE EXPRIMENTAL.CONCLUSIONES	10/11/17	14/12/17
FASE 5: TAREAS FINALES	20/09/17	2/01/18

Fig. 1.3: Fases PFC

1.4.2 Fase 1: Planificación PFC

Nombre	Fecha de inicio	Fecha de fin
FASE 1 : PLANIFICACION TFC	20/09/17	5/10/17
Tarea 1.1 : Motivación	20/09/17	22/09/17
Tarea 1.2 : Objetivos.Alcance	25/09/17	29/09/17
Tarea 1.3 : Planificación Temporal	2/10/17	5/10/17
Tarea 1.4 : Entrega PEC1	6/10/17	6/10/17
FASE 2: ESTUDIO,PLANIFICACION Y DISEÑO DE UN CENTR...	9/10/17	30/10/17
FASE 3: ALIENVAULT OSSIM	1/11/17	9/11/17
FASE 4: PARTE EXPRIMENTAL.CONCLUSIONES	10/11/17	14/12/17
FASE 5: TAREAS FINALES	20/09/17	2/01/18

Fig. 1.4: Tareas Fase 1

Fig. 1.5: Diagrama Gantt Fase 1

1.4.3 Fase 2: Estudio, planificación y diseño de un centro de seguridad.

Nombre	Fecha de inicio	Fecha de fin
FASE 1 : PLANIFICACION TFC	20/09/17	5/10/17
FASE 2: ESTUDIO,PLANIFICACION Y DISEÑO DE UN ...	9/10/17	30/10/17
Tarea 2.1 : Introducción SGSI. NORMA UNE-ISO/...	9/10/17	13/10/17
Tarea 2.1.1 : Objetivos de un Sistema de Segi...	9/10/17	10/10/17
Tarea 2.1.2: El Sistema de Gestión de Riesgos....	11/10/17	11/10/17
Tarea 2.1.3 : Marco Legal y Jurídico	12/10/17	13/10/17
Tarea 2.2 : Vulnerabilidades Informáticas	16/10/17	20/10/17
Tarea 2.2.1 : Introducción	16/10/17	16/10/17
Tarea 2.2.2 : ¿Qué es una vulnerabilidad?	17/10/17	18/10/17
Tarea 2.2.3 : Parámetros que definen una vul...	17/10/17	18/10/17
Tarea 2.2.4 : ¿ Dónde se encuentran las vulne...	19/10/17	19/10/17
Tarea 2.2.5 : Estándares de Vulnerabilidades	19/10/17	20/10/17
Tarea 2.2.6 : Análisis y Gestión de Vulnerabili...	20/10/17	20/10/17
Tarea 2.3 : Sistemas SIEM	23/10/17	26/10/17
Tarea 2.3.1: ¿Qué es un SIEM?. ¿Qué son los L...	23/10/17	25/10/17
Tarea 2.3.2: Capacidades de un SIEM	25/10/17	26/10/17
Tarea 2.4: Centro de Operaciones de Seguridad(S...	23/10/17	30/10/17
FASE 3: ALIENVAULT OSSIM	1/11/17	9/11/17
FASE 4: PARTE EXPRIMENTAL.CONCLUSIONES	10/11/17	14/12/17
FASE 5: TAREAS FINALES	20/09/17	2/01/18

Fig. 1.6: Tareas Fase 2

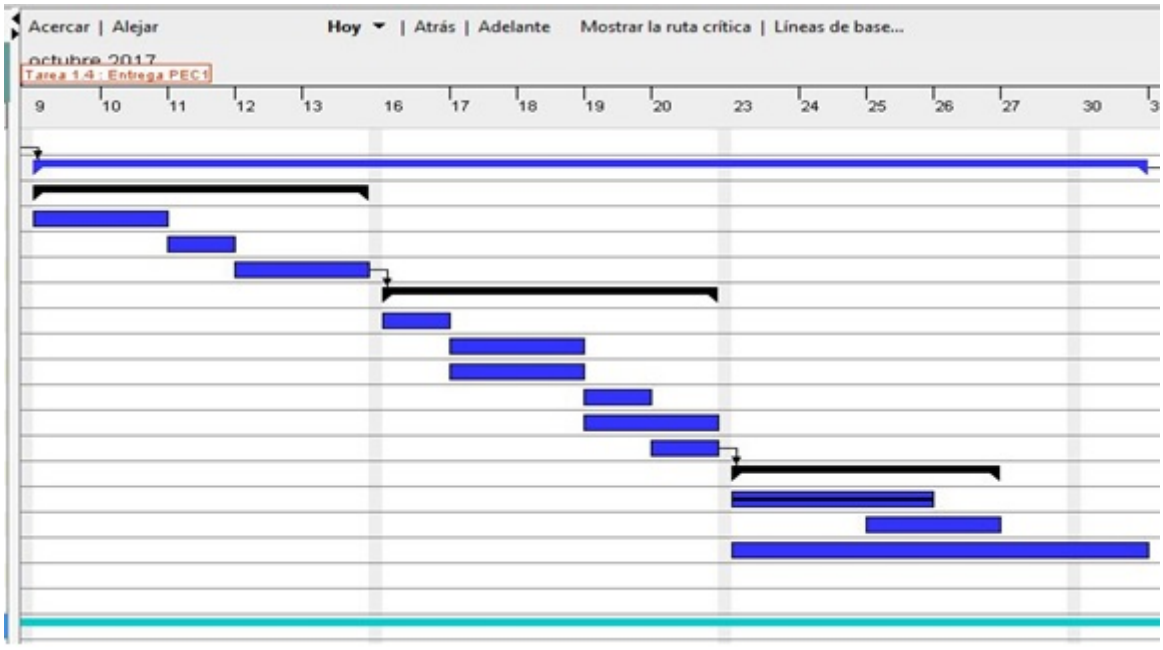


Fig. 1.7: Diagrama Gantt Fase 2

1.4.4 Fase 3: ALIENVAULT OSSIM.

Nombre	Fecha de inicio	Fecha de fin
FASE 1 : PLANIFICACION TFC	20/09/17	5/10/17
FASE 2: ESTUDIO,PLANIFICACION Y DISEÑO DE UN ...	9/10/17	30/10/17
FASE 3: ALIENVAULT OSSIM	1/11/17	9/11/17
Tarea 3.1 : Introducción OSSIM.	1/11/17	2/11/17
Tarea 3.2: Instalación y Configuración OSSIM	2/11/17	3/11/17
Tarea 3.3: Funcionalidades de OSSIM	6/11/17	7/11/17
Tarea 3.4: Herramientas Integradas en OSSIM	8/11/17	9/11/17
Tarea 3.5: Entrega PEC 2	10/11/17	10/11/17
FASE 4: PARTE EXPRIMENTAL.CONCLUSIONES	10/11/17	14/12/17
FASE 5: TAREAS FINALES	20/09/17	2/01/18

Fig. 1.8: Tareas Fase 3

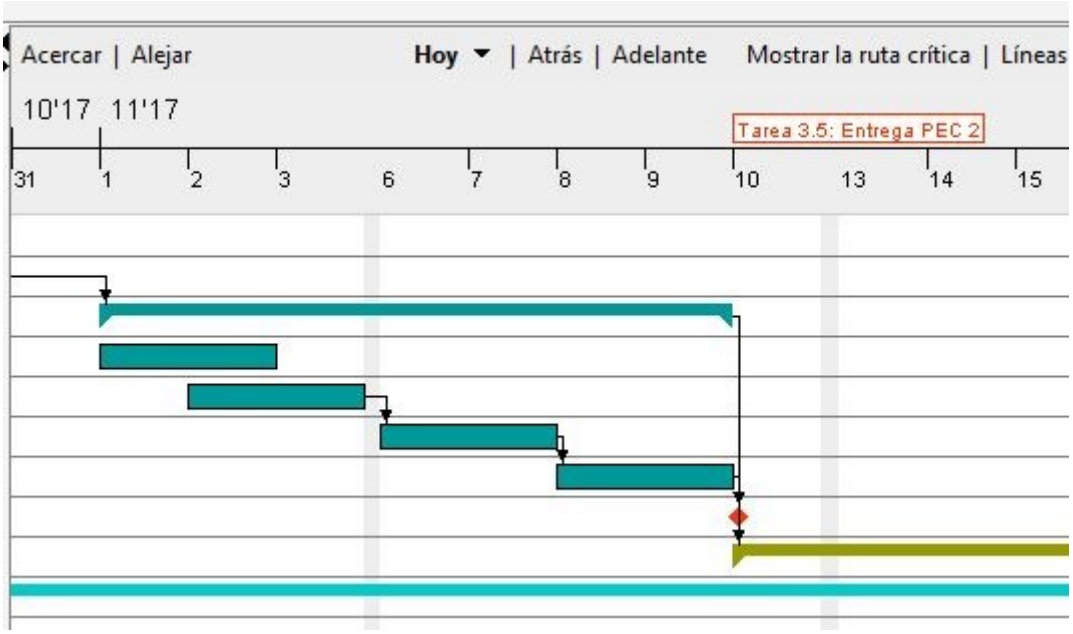
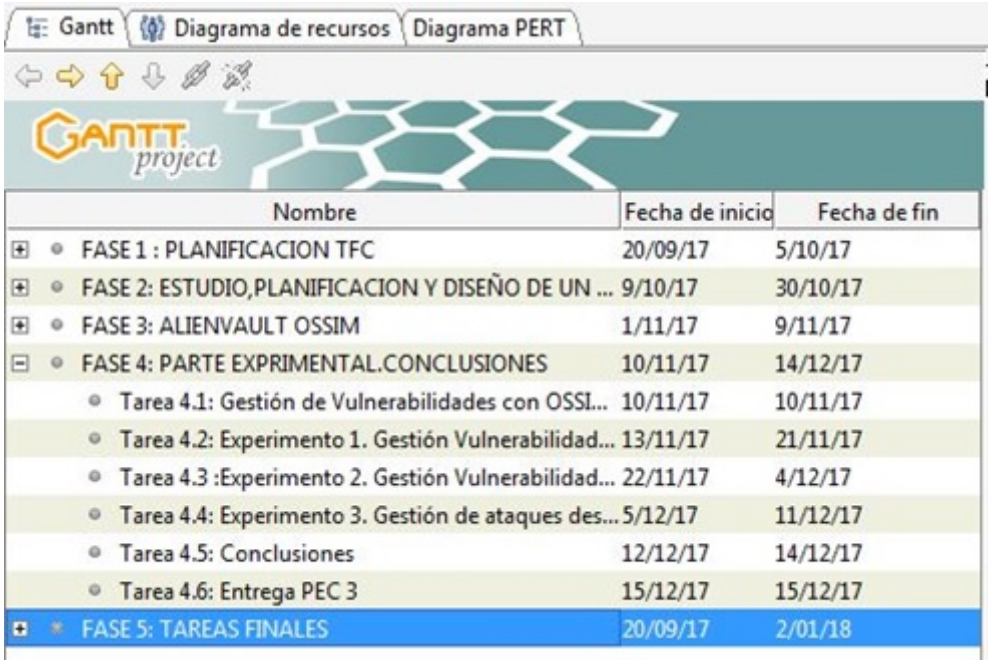


Fig. 1.9: Diagrama Gantt Fase 3

1.4.5 Fase 4: Parte Experimental.Conclusiones.



Nombre	Fecha de inicio	Fecha de fin
FASE 1 : PLANIFICACION TFC	20/09/17	5/10/17
FASE 2: ESTUDIO,PLANIFICACION Y DISEÑO DE UN ...	9/10/17	30/10/17
FASE 3: ALIENVAULT OSSIM	1/11/17	9/11/17
FASE 4: PARTE EXPRIMENTAL.CONCLUSIONES	10/11/17	14/12/17
Tarea 4.1: Gestión de Vulnerabilidades con OSSI...	10/11/17	10/11/17
Tarea 4.2: Experimento 1. Gestión Vulnerabilidad...	13/11/17	21/11/17
Tarea 4.3 :Experimento 2. Gestión Vulnerabilidad...	22/11/17	4/12/17
Tarea 4.4: Experimento 3. Gestión de ataques des...	5/12/17	11/12/17
Tarea 4.5: Conclusiones	12/12/17	14/12/17
Tarea 4.6: Entrega PEC 3	15/12/17	15/12/17
FASE 5: TAREAS FINALES	20/09/17	2/01/18

Fig. 1.10: Tareas Fase 4

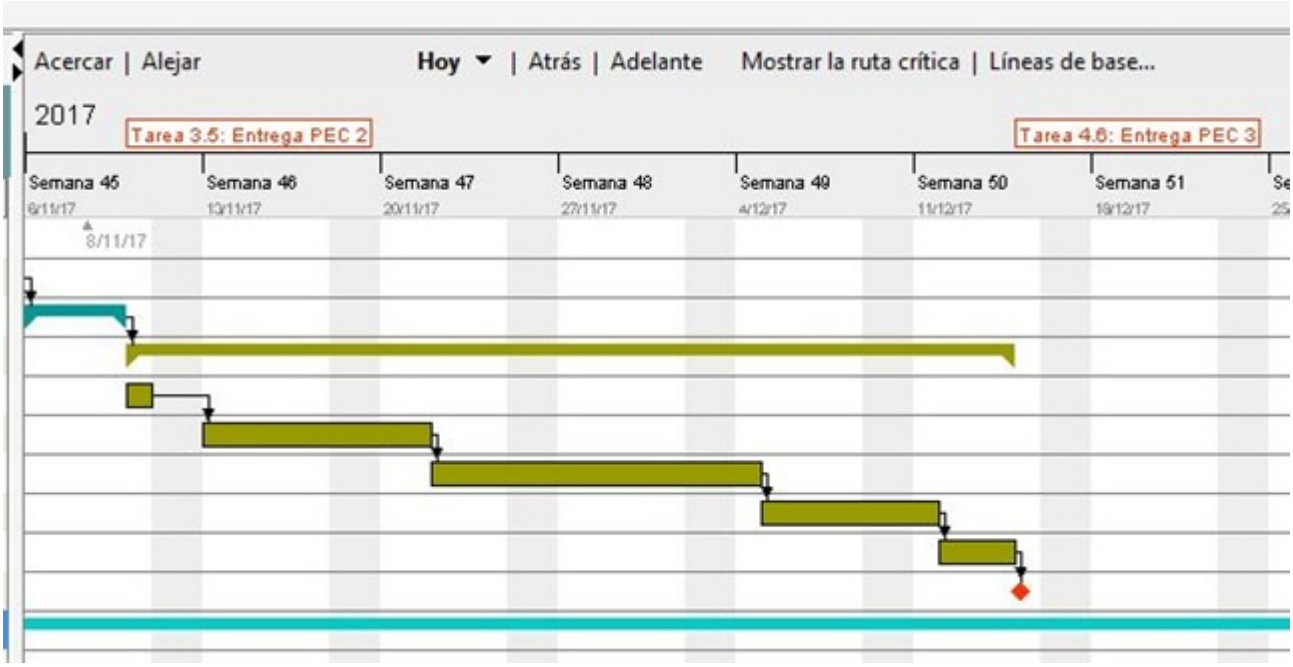


Fig. 1.11: Diagrama Gantt Fase 4

1.4.6 Fase 5: Tareas Finales.

Nombre	Fecha de inicio	Fecha de fin
FASE 1 : PLANIFICACION TFC	20/09/17	5/10/17
FASE 2: ESTUDIO,PLANIFICACION Y DISEÑO DE UN ...	9/10/17	30/10/17
FASE 3: ALIENVAULT OSSIM	1/11/17	9/11/17
FASE 4: PARTE EXPRIMENTAL.CONCLUSIONES	10/11/17	14/12/17
FASE 5: TAREAS FINALES	20/09/17	2/01/18
Tarea 5.1 : Bilbliografía	20/09/17	2/01/18
Tarea 5.2 : Anexos	18/12/17	22/12/17
Anexo 1: Descripción del ataque realizado co...	18/12/17	22/12/17
Anexo 2: Configuración OSSEC	18/12/17	22/12/17
Anexo 3: Configuración Firewall Cisco ASA	18/12/17	22/12/17
Anexo 4: Configuración NDIS OSSIM	18/12/17	22/12/17
Anexo 5: Configuración monitorización de di...	18/12/17	22/12/17
Anexo 6: Configuración Switch LAN	18/12/17	22/12/17
Tarea 5.3: Correcciones/Retoques	21/12/17	22/12/17
Tarea 5.4: Diapositivas PFC	25/12/17	29/12/17
Tarea 5.5: Documentación PFC	20/09/17	2/01/18
Tarea 5.6: Entrega PFC	3/01/18	3/01/18

Fig. 1.12: Tareas Fase 5

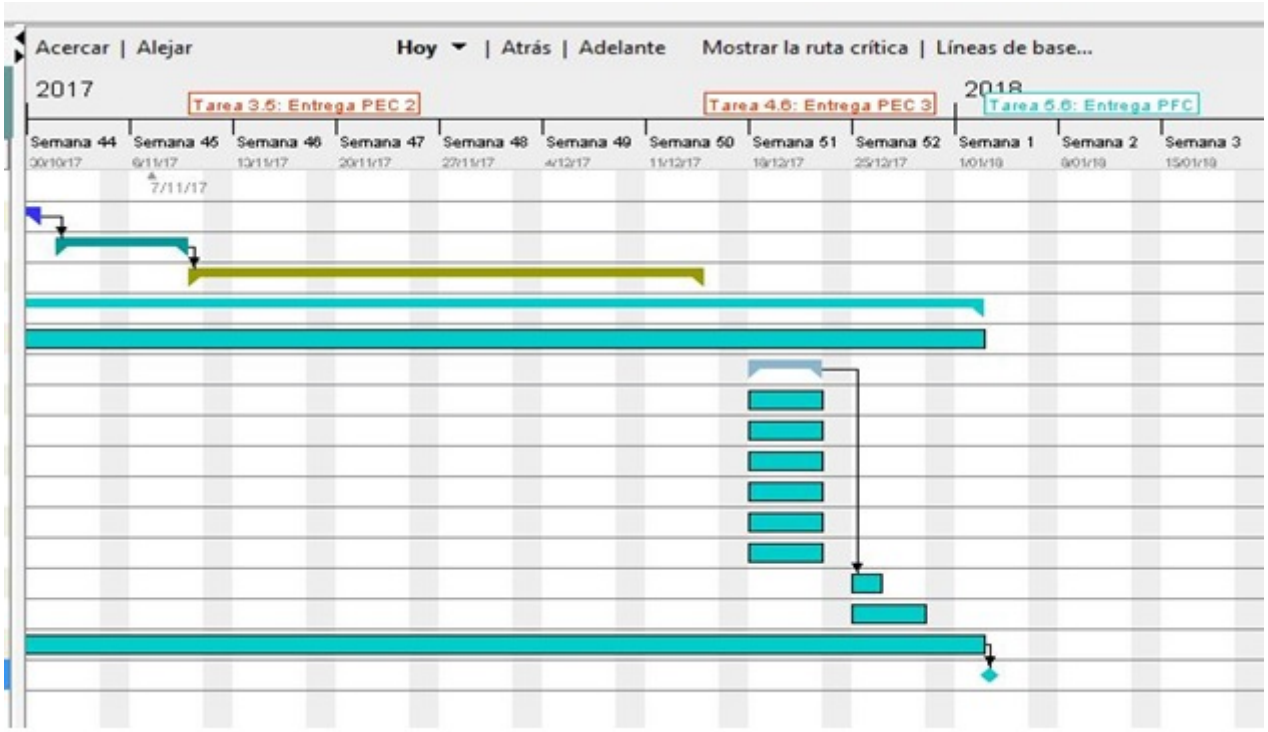


Fig. 1.13: Diagrama Gantt Fase 5

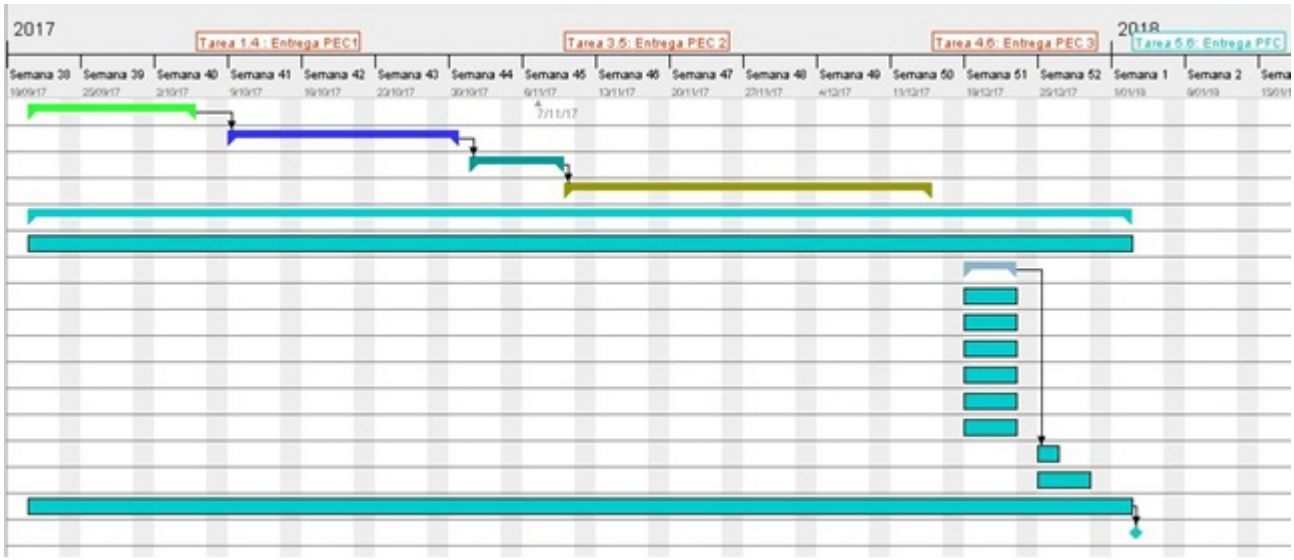


Fig. 1.14: Diagrama Gantt PECS

1.5 Productos obtenidos.

En este PFC los productos obtenidos están muy relacionados con las conclusiones del proyecto. Entre los objetivos más destacados está el aprendizaje obtenido de una herramienta de gestión de seguridad open source AlienVault OSSIM.

OSSIM tiene capacidad capacidad de poder adaptarse a nuevo software de detección de vulnerabilidades que pueda aparecer, aún cuando y esto puede en algunas organizaciones ser un punto débil, su configuración y puesta en marcha necesita de personal cualificado, necesitándose tiempo y medios económicos para su implantación. Siendo uno de los objetivos conseguidos la configuración de las herramientas que forman parte de OSSIM.

Otro hito formativo conseguido es el uso de las normas ISO 27001/27002/27004 y su adaptación al modelo de seguridad de una organización.

1.6 Estructura del proyecto.

Este trabajo está orientado al estudio de las vulnerabilidades de un sistema informático, no obstante la seguridad en una organización no es un capítulo ajeno al esquema organizativo de la empresa sino que está encuadrado como un elemento más de gestión dentro de la empresa, con objeto de garantizar que la información manipulada en la misma se realiza bajo la máxima seguridad. Así en el capítulo 2 se describe brevemente y se ve la necesidad del **Sistema de Gestión de Seguridad de la Información de la empresa**(SGSI).

En los capítulos posteriores se desarrollan de manera breve los medios y técnicas que nos permiten realizar de manera satisfactoria el estudio, control y respuesta frente a las vulnerabilidades informáticas. En el capítulo 3 se trata el **SOC**, elemento crítico para el desarrollo de las actividades del SGSI. El capítulo 4 está dedicado a la tecnología **SIEM**, al que sigue el capítulo 5 dedicado a **OSSIM**, ya que es el SIEM que va a utilizarse en este trabajo. El capítulo 6 trata aspectos de las **vulnerabilidades** y a continuación el capítulo 7 está dedicado a la **parte experimental** realizada con AlienVault OSSIM, parte del desarrollo experimental detallado se muestra en los **anexos** del capítulo 10.

Finalmente las **conclusiones** figuran en el capítulo 8 y alguna de la **bibliografía** consultada en el capítulo 9.

2. CONCEPTOS DE SEGURIDAD. SGSI.

2.1 Introducción. El SGSI.

Hoy en día muchas actividades dependen cada vez más del uso de sistemas y redes informáticas, de manera que Administraciones Públicas, empresas, muchas organizaciones y los mismos ciudadanos necesitan del adecuado funcionamiento de los sistemas que soportan sus actividades y de forma concreta en lo que atañe a la seguridad de los mismos.

La información constituye un recurso que a menudo no se valora adecuadamente, ya que aparentemente las medidas de seguridad no suelen contribuir a mejorar la producción de las redes informáticas, es más su aplicación suele reducir el rendimiento (de la misma red y aplicaciones), debido al uso de recursos computacionales y ancho de banda usado.

La información de las empresas normalmente se distribuye por los distintos computadores de la organización incluso ubicados en distintas sedes, lo que dificulta enormemente su control y su localización, esto da origen a que ocurran a menudo incidentes de seguridad, lo que sumado al mal uso por parte del personal de las empresas que también se da a menudo, aconsejen la implantación de un Sistema de Gestión de la Seguridad de la Información.

Además de los aspectos citados también se debe tener en cuenta el marco legal que obliga a las organizaciones al cumplimiento de distintas leyes tales como la LOPD (Ley Orgánica de Protección de Datos), la GDPR (General Data Protection Regulation), sustituta de la LOPD, LSSI (Ley de Prestación de Servicios de la Sociedad de la Información), PCI DSS (Payment Card Industry Data Securtiy Standard), la ley General de telecomunicaciones, la Ley de Firma electrónica...

Se puede definir la **Seguridad Informática** como cualquier medida que impida la ejecución de operaciones no autorizadas sobre un sistema o red informática, cuyos efectos puedan conllevar daños sobre la información, comprometer su confidencialidad, autenticidad e integridad, disminuir el rendimiento de los equipos o bloquear el acceso de usuarios autorizados al sistema.[3]

Para cumplir con los objetivos de seguridad informática una empresa debe contemplar las siguientes líneas de actuación:

- **Técnica:** en las vertientes física (hardware) y lógica (software).
- **Legal:** las leyes obligan a implantar determinadas medidas de seguridad.
- **Humana:** formación del personal de la empresa.
- **Organizativa:** definición e implantación de políticas/normas/procedimientos de seguridad y llevar a cabo buenas prácticas de actuación en seguridad.

Aún cuando el objetivo primordial de este proyecto se centra en el estudio y gestión de las vulnerabilidades software que pueden presentarse en los sistemas informáticos es necesario un encuadre del mismo en el marco de lo que se denomina el Sistema de Seguridad de la Información (SGSI) de la empresa.

Podemos definir un **Sistema de Gestión de Seguridad de la información** como una herramienta de gestión que nos va a permitir conocer, gestionar y minimizar los posibles riesgos que atenten contra la seguridad de la información de la empresa.[1][3]

Se entiende por información todos los datos que posee una entidad/empresa y que poseen un valor para ella independientemente de la manera que se almacena o se transmita.[2]

Al ser la información un activo de máxima importancia para el éxito y pervivencia de una empresa en el mercado se hace necesario implantar mecanismos para protegerla así como aquellos mecanismos que la almacenan. Se hace también necesario valorar la importancia de la información con objeto de arbitrar los mecanismos adecuados a cada situación.

2.2 Gestión SGSI. Normas de seguridad ISO/IEC.

Con objeto de gestionar el SGSI se necesita un mecanismo que lleve a cabo dicha gestión de forma metódica y documentada evaluando los riesgos a los que puede estar sometida. Distintos organismos nacionales y/o internacionales crean normas que constituyen los mecanismos a seguir. En lo que a seguridad de la información se refiere la norma ISO/IEC 27001:2013 proporciona un conjunto de estándares para la implantación de un SGSI. La implantación y posterior certificación de un SGSI aporta a la empresa:

- Reducción de riesgos ya que se establecen controles sobre ellos hasta reducir las amenazas a un nivel asumible, de esta manera si se produce una incidencia los daños se minimizan y se asegura la continuidad del negocio.
- Ahorro de costes al eliminar inversiones producidas por desestimar riesgos.
- Se convierte la Seguridad en una actividad de gestión pasando a realizarse de manera metódica y controlada.
- Permite un cumplimiento eficiente del marco legal evitando riesgos y costes innecesarios.
- La certificación del SGSI contribuye a mejorar la competitividad, diferenciando a las empresas, haciendo que sean más fiables e incrementando su prestigio.

En general, los aspectos a tener en cuenta al gestionar la seguridad de la información en una empresa hacen referencia a preservar la confidencialidad, integridad y disponibilidad de los datos. Por lo tanto esto nos obliga antes de establecer sistemas de protección a conocer qué información tenemos, en que dispositivos se encuentra, qué valor tiene en términos de confidencialidad, integridad y disponibilidad y evaluar qué riesgo supondría su incumplimiento para la continuidad del negocio o pérdidas que podría ocasionar caso materializarse alguna amenaza.

Fig. 2.1: Objetivos de la seguridad Informática

Para la realización de las citadas evaluaciones (durante la implantación del SGSI) se necesitan realizar una serie de etapas:

- Definición de Políticas de seguridad y Alcance del SGSI.
- Identificación y Registro de Activos.
- Análisis y Gestión de Riesgos.
- Selección e Implantación de Controles de Seguridad.
- Establecer un Plan de Mejora de la Seguridad.

Solamente se citan aquellas que están directamente relacionadas con los objetivos de este proyecto y que es necesario conocer para poder ubicarlo dentro del SGSI.

La gestión de la seguridad con un SGSI no se trata de un proceso estanco sino que los procesos para gestionar los riesgos deben realizarse con un seguimiento y revisión continua con objeto de lograr una mejora continua a lo largo del tiempo, por ello suele ser muy común la utilización del modelo PDCA, que se describe en la figura [2.2].

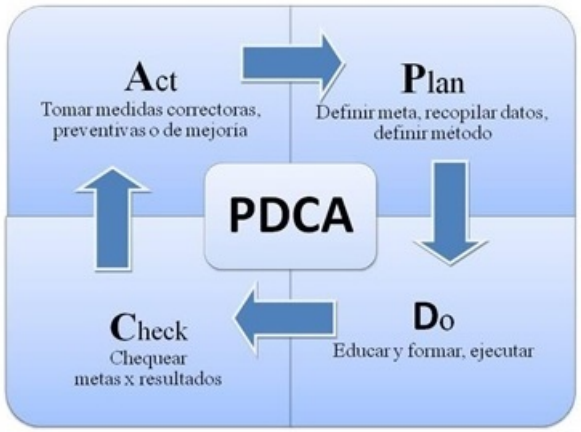


Fig. 2.2: Modelo PDCA

Una vez identificados los activos se tiene que realizar una valoración de los mismos para lo que se deben realizar las siguientes acciones:

- **Análisis de impacto.**
- **Estudio de Amenazas.**
- **Estudio de las Vulnerabilidades.**
- **Estudio del Riesgo.**

Con objeto de ir centrando el trabajo a continuación se definen algunos términos usados en el mismo:

- Los **activos** son los bienes, derechos y otros recursos de los que dispone una empresa. En este PFC los activos se refieren a activos informáticos: hardware, software, información.
- Los **recursos** del sistema son los activos a proteger del sistema informático.

- Una **amenaza** es cualquier evento accidental o intencionado que puede ocasionar algún daño en el sistema informático, provocando pérdidas materiales, financieras o de otro tipo a la empresa. Suelen clasificarse, cualitativamente, como muy baja, baja, media, alta, muy alta. La materialización de una amena se denomina a menudo **incidente de seguridad**.

- Una **vulnerabilidad** es cualquier debilidad en el sistema informático que pueda permitir a las amenazas causarle daños y producir pérdidas para la empresa. Cualitativamente se valoran como baja, media y alta.

- El **impacto** es la medición y valoración del daño que podría producir a la organización un incidente de seguridad.

- El **riesgo** es la probabilidad de que una amenaza se materialice sobre una vulnerabilidad del sistema informático, causando un determinado impacto en la empresa.

De esta manera en el análisis de impacto se estudian las consecuencias de que se concreten alguna/as de las amenazas, teniendo en cuenta la confidencialidad, integridad y disponibilidad.

En el estudio de amenazas se intenta averiguar todos los posibles eventos que puedan causar daño a un activo mediante la explotación de vulnerabilidades y en el estudio de vulnerabilidades se intentan identificar las vulnerabilidades que poseen los activos, que puedan ser explotadas por una amenaza y que tengan un impacto negativo para la empresa.

En el caso de las amenazas se hace un cálculo de la probabilidad de ocurrencia de explotación de una vulnerabilidad de un activo. Finalmente con todos esos datos se hace un estudio del riesgo. En la figura [2.3] se muestra un modelo de valoración del riesgo.

Fig. 2.3: Estudio del riesgo

2.3 Controles a aplicar sobre los activos. ISO/IEC 27001, 27002, 27004.

Acabados los distintos análisis hay que planificar e implementar los controles adecuados sobre los activos en riesgo con objeto de minimizarlos y llevarlos a los límites aceptables por la empresa y definidos en la Política de Seguridad. Hay que tener en cuenta que siempre habrá un riesgo residual y que cuando se le hace frente se trata de una relación coste-beneficio, siendo así que a veces puede darse el caso de aceptar el riesgo y no se implementan medidas.

Según sea el tratamiento a aplicar se seleccionan controles del Anexo A de la norma ISO/IEC 27001:2013. El Anexo A es una herramienta esencial para la gestión de la seguridad ya que proporciona una lista de controles (o medidas) de seguridad que pueden ser usados para mejorar la seguridad de la información. En una organización que siguiese la norma ISO 27001 debería justificarse aquellos controles que no se usen. El riesgo residual es el que resulta después de aplicar los controles y es el que hay que asumir y vigilar.

No se debe olvidar que proteger la información no es solamente protección de la información TI (tecnologías información) ya que por sí sola no sería suficiente también se necesitarían para garantizarla otros activos, por ejemplo recursos humanos... pero en este trabajo nos centramos en TI.

La manera de llevar a cabo los controles viene de forma detallada en la norma ISO 27002, pero no se puede usar solo norma ISO para gestionar la seguridad de la información ya que carece de elementos que se encuentran en la norma 27001 tales como controles a seleccionar, como medirlos... Y sobre todo no es posible obtener una certificación ISO 27002 porque no es una norma de gestión, es decir, no define cómo ejecutar un sistema, no obstante sí que es el caso de la norma 27001 por lo que sí es posible una certificación ISO 27001. La norma ISO 27002 es una guía de buenas prácticas.

La norma ISO/IEC 27001:2013 contiene 39 objetivos de control y 113 controles, agrupados en 11 dominios (detallados en 27002:2013). Los dominios son áreas funcionales de seguridad y en los objetivos de este PFC estarían implicados aspectos relacionados con los **dominios** siguientes (respetando la numeración con la que aparecen en la norma ISO):

- 8: gestión de activos.
- 9: control de accesos.
- 10: cifrado.
- 12: seguridad en la operativa.
- 13: seguridad en las comunicaciones.
- 14: adquisición, desarrollo y mantenimiento de los sistemas de información.
- 16: Gestión de incidentes en la seguridad de la información.
- 18: cumplimiento.

Concretamente en los dominios 12 y 13 de la norma figuran, entre otros, **los siguientes objetivos de control y controles**:

- 12.2 Protección contra código malicioso.
  - 12.2.1 Controles contra el código malicioso.
- 12.3 Copias de seguridad.
- 12.4 Registro de actividades y supervisión.

  - 12.4.1 Registro y gestión de eventos de actividad.
  - 12.4.2 Protección de los registros de información.
- 12.5 Control del software en explotación.
- 12.6 Gestión de la vulnerabilidad técnica.
  - 12.6.1 Gestión de las vulnerabilidades técnicas.
  - 12.6.2 Restricciones en la instalación de software.
- 13.1 Gestión de la seguridad en las redes.
  - 13.1.1 Controles de red.
  - 13.1.2 Mecanismos de seguridad asociados a servicios en red.
  - Segregación de redes.
- 13.2 Intercambio de información con partes externas.
  - 13.2.1 Políticas y procedimientos de intercambio de información.
  - 13.2.2 Acuerdos de intercambio.
  - 13.2.3 Mensajería electrónica.
  - 13.2.4 Acuerdos de confidencialidad y secreto.

Los controles son de dos tipos técnicos y organizativos, siendo los controles técnicos aquellos en los que se encuadraría el PFC.

Si la empresa estuviese obligada a cumplir con el estándar PCI-DSS, que es de obligado cumplimiento si la empresa vende productos o servicios o incluso si es una organización con o sin fines de lucro e independientemente de su tamaño, tendría que implementar una serie de requisitos, concretamente 12 requisitos agrupados en seis **secciones** (númeradas siguiendo el estándar PCI-DSS) denominadas objetivos de control:

1. Desarrollar y mantener una red segura.
   - Requisito 1: Instalar y mantener una configuración de cortafuegos para proteger los daos de los titulares de las tarjetas.
   - Requisito 2: No usar contraseñas del sistema y otros parámetros de seguridad predeterminados provistos por los proveedores.
2. Proteger los datos de los propietarios de las tarjetas.
3. Mantener un programa de gestión de vulnerabilidades.
4. Implementar medidas sólidas de control de acceso.
5. Monitorizar y probar regularmente las redes.
   - Requisito 10: Rastrear y monitorizar todo el acceso a los recursos de la red y datos de los propietarios de tarjetas.
   - Requisito 11: Probar regularmente los sistemas y procesos de seguridad.
6. Mantener una política de seguridad de la información.

Finalmente en un SGSI debe tenerse en cuenta la norma ISO/IEC 27004 ya que en la norma ISO/IEC 27001 impone ciertos controles que se deben cumplir para certificar pero es en la norma 27004 dónde se expone el programa de medición a llevar a cabo para así poder determinar el nivel de cumplimiento y efectividad de los controles, en esta norma se define un modelo de medición y además se indica que el programa de medición puede ser adaptado a las necesidades de cualquier organización ya sea grande o pequeña. A continuación se enumeran algunas recomendaciones de la norma 27004 para facilitar la implementación del modelo de medición:

- Como recolectar y analizar la medición.
- Como comunicar los resultados a las partes interesadas.
- Como analizar los resultados y transformarlos en acciones.
- Como adaptar los resultados de acuerdo al receptor de la información.
- Como realizar el seguimiento, control, revisión de los resultados.
- Como planificar la implementación de mejoras.

Como se observa el SGSI contiene una gran cantidad de controles e indicadores, tal como se ha visto en las normas descritas, siendo necesario para un manejo eficiente de los mismos el utilizar un cuadro de mandos, es decir, un centro de operaciones de seguridad (SOC) ya que esta herramienta nos ofrece una visión del estado de seguridad informática de la empresa así como de las alertas de los distintos indicadores en tiempo real y correlación de los eventos de manera centralizada. En el siguiente capítulo se describe el SOC y como lleva a cabo sus funciones.

3. EL CENTRO DE SEGURIDAD DE OPERACIONES/SOC (SECURITY OPERATIONS CENTER)

3.1 ¿Qué es un SOC?

Un Centro de Operaciones de Seguridad (SOC) es un equipo organizado y altamente cualificado cuya misión es monitorizar y mejorar continuamente el estado de seguridad de una organización mientras se previene, detecta, analiza y responde a incidentes de seguridad informática con la ayuda de tecnología así como de procesos y procedimientos bien definidos. Por lo tanto entendemos que el SOC es dentro de la seguridad informática el elemento neurálgico utilizado por el equipo informático para poder satisfacer los requisitos de análisis, monitorización y correlación de eventos informáticos tales como las vulnerabilidades de las aplicaciones.

Un SOC proporciona la información necesaria para que las organizaciones detecten eficazmente brechas de seguridad y las mitiguen posteriormente de manera que se tiene mayor capacidad de detección de amenazas con antelación y aumenta la efectividad en la respuesta. Aunque eliminar las amenazas a las que nos enfrentamos es un objetivo imposible, reducir el tiempo de respuesta y contenerlas es alcanzable.

El establecimiento de un SOC es un paso necesario para que una organización sea capaz de detectar y contener con eficacia una brecha de seguridad. También resulta ser una herramienta muy eficiente para el cumplimiento de normativas legales.

3.2 Implantación de un SOC.

Cuando se implanta un SOC con objeto de hacer frente a la seguridad informática una organización debe responder esencialmente a las siguientes cuestiones:

- ¿Cómo se puede lograr el objetivo de alcanzar una mayor eficacia en seguridad informática?
- ¿Cuáles son las amenazas a las que se enfrenta y cómo afectan a sus prioridades de inversión?

Una vez que se ha tomado la decisión de establecer un SOC es preciso estudiar las capacidades existentes en la organización centrándose de manera prioritaria, para la empresa, en los siguientes aspectos:

- ¿Qué tecnologías tiene ya implementadas que pudieran ser de utilidad?
- ¿Están las tecnologías a disposición del equipo de seguridad de la organización?
- ¿Qué capacidades tengo en mi equipo de seguridad?
- ¿Cuáles son las restricciones de tiempo actuales en dicho equipo?
- ¿Cuáles son las prioridades para este proyecto?

Hay que tener en cuenta que el establecimiento de un SOC cuenta a menudo con restricciones y que un SOC debe adaptarse a las necesidades de cada empresa.

3.3 Características de un SOC.

La manera más sencilla de entender las características de un SOC es respondiendo a lo que tenemos que asegurar en nuestra organización.

3.3.1 ¿Qué activos proteger?

Una vez que se decide implementar un SOC el primer paso es disponer de conocimiento de los activos implementados en la organización y de los servicios que ejecutan dichos activos ya qué contar con un buen conocimiento de los activos puede ayudar a priorizar una respuesta técnica a las amenazas que se produzcan. Una brecha de seguridad suele comenzar con una fase de descubrimiento o sondeo por parte de los hackers intentando escanear la red para explotar vulnerabilidades conocidas en servicios habituales. Por lo tanto para averiguar en qué activos se debe centrar la protección una organización, debe responder a las siguientes cuestiones:

- ¿Qué sistemas son críticos para que la empresa siga funcionando?
- ¿Qué sistemas son críticos para las tareas diarias?
- ¿De qué otros sistemas dependen dichos sistemas críticos?
- ¿Qué sistemas gestionan y almacenan información sensible?

En muchas ocasiones no es sencillo tener una visión precisa de los activos implantados y de los servicios en funcionamiento. La solución preferible y es la que puede realizarse de manera eficiente mediante un SOC es una solución automatizada ya que de esta manera se puede garantizar una adaptación precisa a los cambios.

Enfoques a seguir para descubrir activos de manera automatizada

- Monitorización pasiva de la red.
- Escaneo activo de la red.
- Inventario de software basado en Host.

En la monitorización pasiva de la red, se enumeran los hosts y los paquetes de software instalados, identificando los puertos y los protocolos usados en el tráfico capturado.

En el escaneo activo se sondea la red para intentar provocar respuesta de las máquinas y basándose en esa respuesta identificar la máquina y el software instalado en la máquina.

En el inventario basado en host se instala un agente en un host siendo este el encargado de realizar el inventario lo que normalmente da lugar a un inventario exhaustivo y preciso.

Estas técnicas suelen utilizarse de manera conjunta y usando un mecanismo centralizado que permita descubrir activos en segmentos remotos de la red.

3.3.2 ¿Cuáles de mis activos son vulnerables a ataques?

Una vez que se tiene un conocimiento de los activos implementados en una organización, el siguiente paso es comprender dónde se encuentran sus debilidades. Cuando se responde a un ataque o a una brecha de seguridad, comprender cómo puede atacarse su organización es un factor crucial a la hora de establecer prioridades. No siempre es factible eliminar una vulnerabilidad pero es posible conocer dónde se encuentra y qué impacto podría tener si se explotara.

Es por lo tanto necesario responder a las siguientes cuestiones:

- ¿Cómo están configurados los activos en funcionamiento?
- ¿Cómo se puede acceder a ellos?
- ¿Tiene alguno de ellos vulnerabilidades conocidas que pueda explotar un atacante?

Para responder a estas cuestiones se realizan los análisis de vulnerabilidades. Este proceso suele ser una tarea compleja ya que comprender cómo se podría explotar una vulnerabilidad de un software a menudo requiere saber cómo funciona, que tipos de datos acepta y qué función se supone que realiza. El análisis de vulnerabilidades se realiza sobre vulnerabilidades conocidas mediante procesos automatizados lo que permite realizar al SOC de manera precisa y centralizada una gestión centralizada de los escaneos de vulnerabilidades incluso en redes con topologías complejas.

Enfoques para automatizar el análisis de vulnerabilidades

- Escaneo activo de red.
- Análisis basado en host.

En el escaneo activo se exploran los hosts de manera activa usando tráfico de red cuidadosamente diseñado para suscitar una respuesta. La posterior combinación de tráfico dirigido, respuesta y un motor de análisis permite determinar la configuración del sistema remoto y de los paquetes de software que están funcionando en el mismo.

En el análisis basado en hosts se accede al sistema de archivos de la máquina para que un motor de análisis realice un estudio preciso y exhaustivo del mismo.

En ambos métodos se depende de bases de datos de vulnerabilidades conocidas que se actualizan periódicamente, labor que viene facilitada por las herramientas utilizadas en el SOC.

3.3.3 ¿De qué forma están siendo atacados mis activos?

Debido a la naturaleza misma de la redes y de Internet, los atacantes están constantemente escaneando a ciegas, atacando todos y cualquier tipo de sistemas que encuentran, por lo que a parte del análisis de vulnerabilidades se hace imprescindible el llevar a cabo una detección de amenazas con objeto de detectar los ataques que están dirigidos contra nuestras vulnerabilidades.

Tampoco en este caso es sencilla la detección de amenazas, dada la pericia que a menudo usan los atacantes, por ello las tecnologías usadas para la detección no deben implementarse solamente en el perímetro de la red empresarial, pensando que los ataques solo pueden venir del exterior sino que tampoco es extraño el que haya ordenadores comprometidos en la red interior ya que los empleados suelen usar computadores dentro y fuera del cortafuegos corporativo, por lo tanto la detección de amenazas debe implementarse de manera generalizada, empleando múltiples técnicas y bases de datos externas actualizadas, tareas que para una eficiente realización deben realizarse con un SOC.

Enfoques para automatizar la detección de amenazas

- Detección de intrusiones en red (NIDS).
- Detección de intrusiones basadas en Host(HIDS).

Mediante el uso de NIDS se analiza el tráfico de red para detectar firmas de ataques conocidos y patrones que indican actividad maliciosa por ejemplo malware, violaciones de políticas y escaneo de puertos. Con la técnica HIDS se analiza el comportamiento y la

configuración del sistema con objeto de detectar comportamientos anómalos, así se pueden reconocer rootkits comunes, modificación de archivos importantes.

De forma análoga a como se aplican a las redes cableadas también se hace con las intrusiones inalámbricas de manera que se puede monitorizar el tráfico wifi e identificar redes maliciosas, clientes wifi, redes asociadas y cifrado utilizado.

3.3.4 ¿Cómo sé si ha tenido lugar una brecha de seguridad?

No todas las brechas de seguridad son evitables. Es imposible hacerle frente de manera totalmente segura a todos los vectores de ataque, en consecuencia siempre tendremos que afrontar un problema de gestión de riesgos, lo que da lugar a la aparición de algún riesgo residual. Es necesario intentar que esta ventaja (el riesgo residual) sea lo menor posible para el atacante.

Es importante que se detecte una brecha de seguridad tan rápido como sea posible. Para ello se debería realizar una monitorización de comportamientos de manera continuada, centralizada para poder correlacionarlos y usando varias técnicas. Una vez más en vez de utilizar herramientas aisladas la mejor solución suele ser una integración de las mismas a través de un SOC. A continuación se indican las técnicas usadas para llevar a cabo dicha monitorización.

Monitorización de comportamientos

- Monitorización activa del servicio.
- Análisis de flujo de red (Netflows).
- Captura de paquetes.
- Detección de intrusiones basadas en Host.

3.3.5 Conclusión:

Cuando se implementan a escala, los controles esenciales descritos para proporcionar descubrimiento de activos, análisis de vulnerabilidades, detección de amenazas y monitorización de comportamientos, producen una cantidad ingente de datos. La comprensión y priorización de dichos datos debe automatizarse para tomar decisiones dentro de un marco temporal razonable. Además es importante que los datos generados se evalúen en conjunto con datos procedentes de otros controles de seguridad. La evaluación de cada flujo de datos de forma independiente conducirá a una mala priorización de los esfuerzos.

La capacidad de encontrar el sentido a estos datos requiere un sistema que los consolide y los gestione todos. Ese sistema debe proporcionar también capacidades de normalización para que los datos de fuentes diversas puedan relacionarse entre sí y pueda presentarse una visión completa al cliente final receptor de la información.

Hoy en día existe únicamente un enfoque que permita automatizar de manera eficiente la comprensión de los datos que genera un sistema en lo que atañe a su comportamiento de seguridad que es la implantación y adecuación a cada situación particular de un SOC.

En el siguiente capítulo se hace un repaso de lo que es una plataforma SIEM al ser esta tecnología un medio para conseguir realizar la implantación eficiente de SOC.

El SOC nos permite además realizar funciones de cumplimiento del marco legal ya que normalmente van almacenando los logs de manera cruda, esto es, tal y como se producen previamente a la realización de tareas de correlación para poder ayudar en la realización de las auditorías. Al mismo tiempo nos van generando informes en los que figura es estado de los distintos tipos de controles exigidos por las normas. OSSIM es capaz de generar informes acorde

con la norma ISO/IEC 27001 y la PCI DSS. Asimismo la versión comercial USM AlienVault guarda los logs en una base de datos específica con fines de auditoría.

4. TECNOLOGÍA SIEM (SECURITY INFORMATION AND EVENT MANAGEMENT)

4.1 ¿Qué es un SIEM?

Un SIEM es una tecnología que proporciona un análisis en tiempo real de las alertas de seguridad generadas por el hardware, software y en general por cualquier dispositivo de red. También suelen usarse con finalidad de log management y para generar informes de cumplimiento de normas de seguridad.

Se utilizan los SIEM entre otras razones porque el número de logs generados en un sistema informático es tan grande que resultaría imposible poder controlar todos los avatares posibles sin estos elementos que aportan centralización e inteligencia en el tratamiento de los eventos.

Un elemento básico en la tecnología SIEM son los logs. Se entiende por logs la información de distinto tipo que alimentan los SIEM. Los logs son registros, que se almacenan en un archivo o base de datos, de los acontecimientos (eventos o acciones) que afectan a un proceso (aplicación, dispositivo hardware...) con objeto de tener un mejor conocimiento del comportamiento de un sistema. Normalmente al menos se registra el momento en que se producen y su tipo o categorización. El primer sistema de log, Syslog, fue escrito en 1980, posteriormente el protocolo Syslog fue estandarizado (RFC3164, RFC5424). El principal problema de los logs atañe a su formato, que a pesar de los esfuerzos realizados, aún no se tiene un formato universalmente aceptado, por lo que en las herramientas de gestión de eventos uno de los problemas a resolver es la normalización de esos formatos. En este PFC y con objeto de ser prácticos entenderé que: *"A log file contains the information the developer of an application thought to be helpful and interesting in the current state of the software, together with the timestamp when this state occurred"*(tesis de Jens Kühnel[33]).

Los sistemas SIEM actuales son la evolución de una serie de tecnologías, entre las que están:

- LMS (Log Management System). Sistema basado en la recolección de archivos de log de sistemas operativos, aplicaciones... con objeto de consultarlos y analizarlos posteriormente.
- SLM/SEM (Security Log/Event Management).
- SIM (Security information management).
- SEM (Security event manager).

Los sistemas SEM se centran en el análisis histórico de los archivos de registro para generar informes y apoyar investigaciones forenses mientras que los sistemas SIM analizan los mismos eventos pero en tiempo real. Los sistemas SIEM incorporan ambas capacidades en una única solución incorporando funciones de correlación y gestión de registros más sofisticadas.

4.2 Arquitectura sistemas SIEM.

La arquitectura básica de estos sistemas se muestra en la figura [4.1].

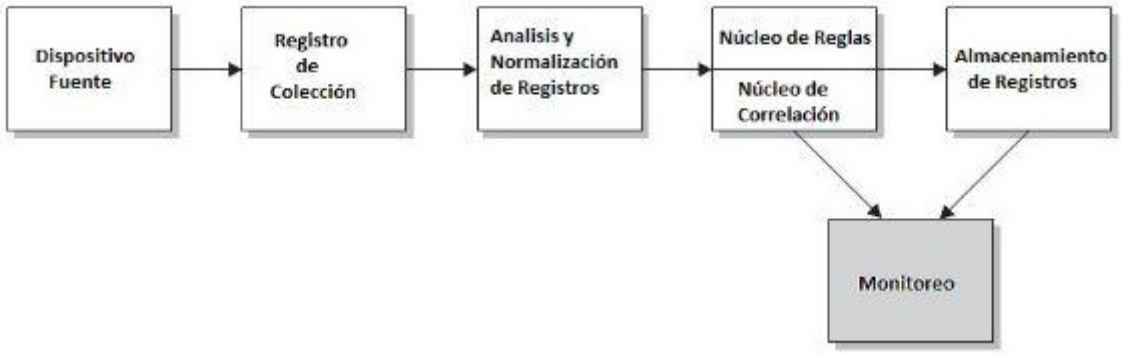


Fig. 4.1: Arquitectura SIEM

- **Dispositivo fuente:**
  Es la parte dónde (dispositivo, aplicación) se recuperan los registros que se almacenan y procesan en el SIEM. La comunicación con el resto del sistema se hace mediante protocolos estándar o propietarios dependiendo del tipo/fabricante. El dispositivo fuente podría ser: Router, switch, servidor, registros firewall, proxy, IDS, registros de base de datos... o de cualquier otra aplicación.

- **Registro de colección:**
  Es la parte del sistema en la que se recogen todos los registros con objeto de enviarlos al servidor del SIEM. Los métodos de recolección son esencialmente dos: El método de empuje (el dispositivo fuente envía sus registros al SIEM) y método de extracción (el SIEM recupera los registros del dispositivo de origen). Protocolos Push/Pull.

- **Análisis y normalización de registros:**
  En esta etapa los registros que se encuentran en su formato original se transforman en un formato estándar (Normalización). Con la normalización de eventos se consigue tener un formato que sea más fácil de leer antes de ser tratados por el SIEM y permiten además tener un formato adecuado para la generación de reglas. A partir de aquí todos los registros tienen el mismo formato. También se hace una copia de los registros en su formato original. Estos registros así pueden servir para usar en una posible auditoría, en el caso de OSSIM, los puede utilizar el aplicativo Logger, que solamente está activado en la versión propietaria.

- **Núcleo/Reglas de correlación:**
  Aquí aparecen dos componentes: Núcleo de Reglas y Núcleo de Correlación de Reglas. En el Núcleo de Reglas figuran condiciones específicas para los registros con objeto de activar alertas y aunque normalmente hay unas reglas predefinidas también se pueden definir reglas personalizadas y definir patrones.
  En la parte Núcleo de Correlación se comparan los eventos normalizados con las reglas de la parte de Reglas, es necesario tener precaución a la hora de poner reglas para no cargar en demasía la parte computacional que tiene lugar en la Correlación. En los SIEM actuales, como OSSIM, el núcleo utiliza lógica borrosa en la extracción de información.

- **Almacenamiento de registros:**
  La función principal es el almacenamiento en bases de datos para su posterior utilización con fines forenses e informes del SIEM.

- **Monitorización**
  Finalmente una vez procesados los datos y mediante una interface, ya sea de consola

o web se realizan los análisis y visualizaciones (gráficas, informes...) que usando el almacenamiento ya citado permiten al administrador hacer la gestión del sistema.

Por lo tanto, resumiendo y de manera sucinta las capacidades que un SIEM ofrece son: Agregación de datos, Correlación, Alertas, Dashboards, Cumplimiento y Retención.

4.3 Herramientas SIEM de mercado para el análisis y gestión de vulnerabilidades.

Dado que el proyecto se centra en herramientas de código abierto, se ha restringido el estudio a las que cumplen esté requisito aun cuando en algunos casos hay herramientas que disponen de una versión comercial. En el caso de software propietario también a menudo se ofrecen versiones de prueba gratuitas.

Lo ideal es que las herramientas se ajustasen lo más posible a la norma ISO/IEC27001:2013, o a cualquier otra norma estandarizada como PCI DSS, ya que eso permitiría a las empresas obtener las certificaciones correspondientes y tener el SGSI normalizado de cara a una auditoría de seguridad. A continuación se muestran algunos cuadros de comparación que se han obtenido de la bibliografía consultada.

Los sistemas de gestión de la seguridad de la información y eventos forman parte de la tecnología SIEM. Varias empresas ofrecen productos SIEM cada vez de más calidad. En la figura [4.2] se muestra unos de los últimos estudios realizados por la consultora Gartner enmarcados en lo que llama el cuadrante mágico de los SIEM.

Fig. 4.2: Cuadrante Mágico Gartner SIEM

El gráfico ofrece una buena comparación de los productos más importantes que ofrece el mercado tanto open source como propietarios. En el eje vertical se muestra la capacidad de ejecución y en el eje horizontal "la integridad" de visión.

A continuación se describen brevemente los parámetros que se utilizan para realizar la clasificación.

- Leaders: Proporcionan ofertas maduras que satisfacen la demanda de mercado, demuestran tener la visión necesaria para mantener su posición. Se distinguen en que centran e invierten en sus ofertas llegando a liderar el mercado y pueden afectar a su dirección general.

- Challengers: Pueden llegar a ser líderes si su visión se desarrolla. Los grandes proveedores pueden fluctuar entre Leaders y Challengers en función de su desarrollo y las necesidades del mercado.

- Visionaries: Se alinean con la forma de evolución del mercado. Suelen introducir nuevas tecnologías, servicios y/o modelos de negocio.

- Niche players: Hacen bien su trabajo en un segmento de mercado pero tienen una capacidad limitada para innovar y/o superar a otras empresas. Entre las causas de su limitación se suelen considerar en que centran su funcionalidad en una región geográfica o bien son nuevos en el mercado.

Como puede observar solamente grandes empresas figuran en el cuadrante LEADERS. En 2016 como en los años precedentes IBM destaca por encima de otros grandes proveedores tales como: Splunk, LogRhythm, HPE e Intel Security. Se observa que entre los productos analizados destaca el único producto open source OSSIM de AlienVAult. AlienVault presenta un producto comercial AlientVault USM y otro gratuito AlientVault OSSIM.

En la figura[4.3] se muestra la evolución de AlienVault entre los años 2011-2016 según los estudios de la consultora Gartner.

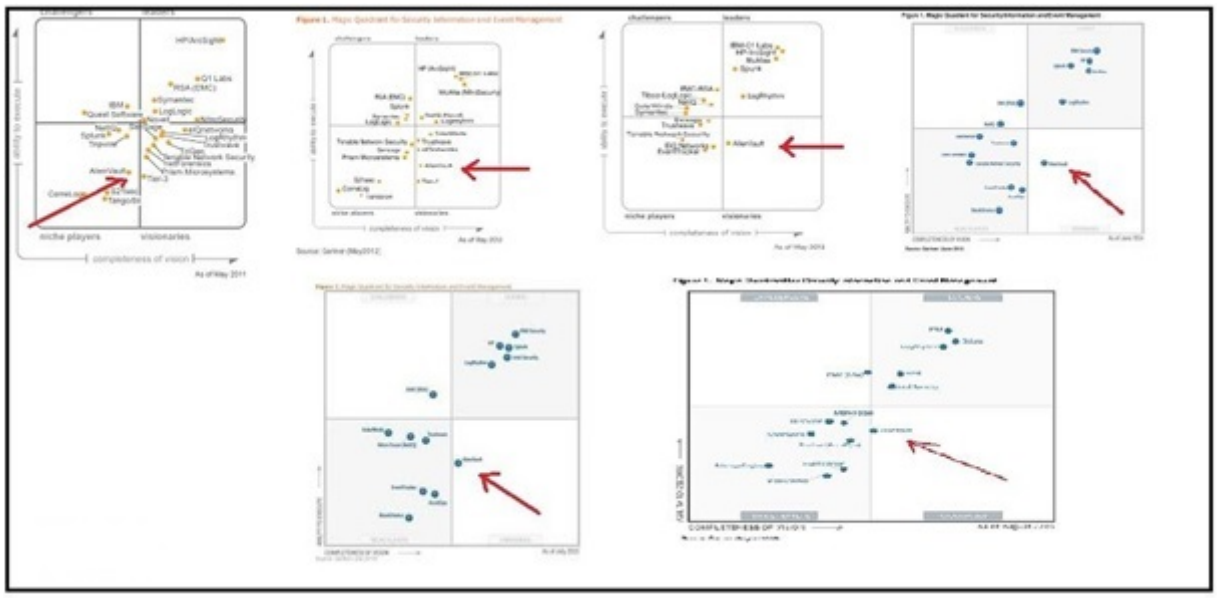


Fig. 4.3: Comparación cuadrantes Gartner SIEM

Como puede observarse ha conseguido mantenerse en un alto nivel durante los últimos años con el añadido del coste para las empresas aun cuando es una herramienta que necesita personal formado para su manejo.

Finalmente he decidido usar esta herramienta por las ventajas que presenta en cuanto a coste económico y por las posibilidades de adaptarlos a distintos tipos de situaciones, ya que se trata de código abierto y expansible mediante plugins.

5. PLATAFORMA OSSIM

5.1 Introducción OSSIM.

OSSIM se engloba dentro de las tecnologías SIEM (Security Information and Event Management). Sistemas de Gestión de la Seguridad de la Información y Eventos.

AlienVault OSSIM es un framework de monitorización y cumplimiento (compliance) gratuito y de código abierto, es decir, una plataforma de seguridad para pequeñas y medianas empresas. Está compuesto por una colección de herramientas todas con licencias GPL que permiten controlar los servicios ofrecidos por cada host de la red, incluidos switchs, routers, firewalls... También es capaz de analizar el tráfico entre los hosts de una LAN y entre una LAN y la WAN.

OSSIM se presenta como una buena herramienta para el estudio y gestión de las vulnerabilidades informáticas de una red, objeto de este trabajo, ya que además y como veremos en los siguientes apartados dispone de motor de correlación de eventos así como un sistema de generación de informes.

Respeto al Compliance, OSSIM tiene soporte para ISO/IEC 27001:20013 así como para diferentes versiones de PCI-DSS, es conocida la obligatoriedad de ajustarse a la normativa PCI 3.2 DSS a partir de febrero de 2018 para las empresas cuyo ámbito de negocio así lo requiera. OSSIM también presenta opciones para manejar el nivel de riesgo de la red y genera informes indicando las áreas fuertes y vulnerables del sistema.

5.2 Configuración de OSSIM.

La descarga de OSSIM puede hacerse desde las url que figuran en los puntos 8 y 10 de la bibliografía. Los aspectos más relevantes relacionados con la configuración de OSSIM se tratan en los anexos [2-5].

5.3 Capacidades/Funcionalidades de OSSIM.

Las principales funcionalidades de OSSIM son:

- Descubrimiento de activos e inventario.
- Gestión de vulnerabilidades.
- Detección de intrusiones.
- Monitorización del comportamiento de la red.
- Correlación de eventos SIEM.

Cabe resaltar que OSSIM no contempla la funcionalidad de log management que si presentan los productos comerciales de AlientVault: USM Appliance y USM Anywhere.

En la figura [5.1] puede verse un resumen más amplio de las funcionalidades de OSSIM.

Fig. 5.1: Funcionalidades OSSIM

5.4 Herramientas integradas en OSSIM.

OSSIM no es una herramienta que simplemente proporcione gestión de logs y almacenamiento de los mismos, sino que se trata de un sistema SIEM por lo que nos proporciona las características esenciales: descubrimiento de activos, evaluación de vulnerabilidades, detección de amenazas, supervisión de comportamiento (monitoring) e inteligencia de seguridad. Estas capacidades se consiguen con la utilización interna de herramientas de código abierto, algunas de las que se citan a continuación.

- Arpwatch usada para detección de anomalías en direcciones MAC.
- P0f utilizada para la detección pasiva de OS y detección de cambios de OS.
- Pads usada para la detección de anomalías de servicio.
- Tcptrack usada para obtener información de datos de sesión con objeto de correlacionarlos para un ataque.

- Nessus utilizada para la evaluación de vulnerabilidades y para la correlación cruzada (IDS vs. Escáner de vulnerabilidades). Actualmente se usa OpenVas. Nessus que en sus comienzos era software libre ahora es software privativo.
- Snort usada como un sistema IDS y para correlación cruzada con Nessus. Actualmente Snort ha dejado paso a Suricata un IDS con capacidad de multithreading.
- Ntop permite elaborar una base de datos con información de la red para usar en la detección de anomalías en comportamiento no usuales.
- Nagios se utiliza para monitorizar la disponibilidad de host y servicios. Utiliza una base de datos de activos de la red.
- Osiris un sistema HIDS.
- Snare un colector de logs para sistemas Windows.
- OSSEC un HIDS.

Además posee herramientas propias, por ejemplo el motor de correlación, que se describirán más adelante.

5.5 Arquitectura de OSSIM.

5.5.1 Descripción básica de la arquitectura

La arquitectura de OSSIM se puede diferenciar en dos partes, una parte se realiza a través de una arquitectura distribuida y la otra parte sobre una arquitectura centralizada. Las dos etapas diferentes del proceso:

- Preproceso: Que se realiza en los propios monitores y detectores distribuidos.
- Postproceso: Que se realiza en el servidor centralizado.

La figura [5.2] representa de una forma detallada la funcionalidad de cada uno de los dos procesos.

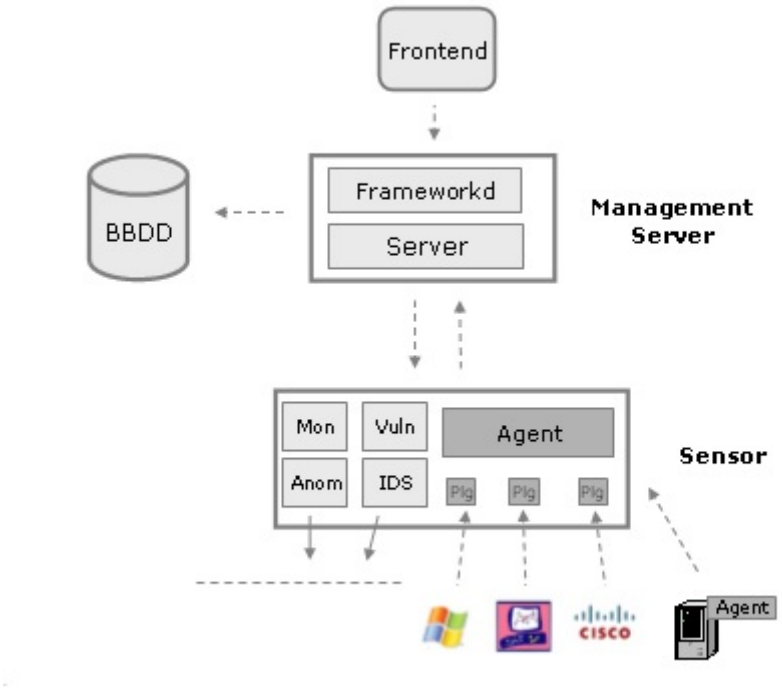


Fig. 5.2: Arquitectura OSSIM

Los programas principales de la arquitectura son **OSSIM-Server**, almacena en su base de datos los registros que le son enviados por los servidores o equipos de red mediante los agentes (tales como OSSEC) o bien mediante los Syslogs. Entre sus funciones están la recogida de los eventos, su catalogación, correlación, evaluación del riesgo y emisión de alarmas. El **OSSIM-Framework** tiene entre sus funcionalidades a parte de la ejecución de comandos externos la lectura y escritura de archivos en el filesystem evitando así que lo haga el servidor Web directamente con los problemas que ello podría ocasionar. El **OSSIM-Agent** son aplicaciones que se instalan en los equipos que se quieren monitorizar y que envían información de los diferentes eventos que ocurren en ellos. Ossim utiliza tres bases de datos heterogéneas para los distintos tipos de datos almacenados:

- EDB base de datos de eventos, la más voluminosa pues almacena todos los eventos recibidos desde los detectores y monitores.
- KDB base de datos del Framework, en la cual se almacena toda la información referente a la red y la definición de la política de seguridad.
- UDB base de datos de perfiles, almacena todos los datos aprendidos por el monitor de perfiles.

En la figura [5.3] se muestra una representación gráfica de las tres capas que componen OSSIM con sus niveles intermedios.

Fig. 5.3: Capas OSSIM

5.5.2 Detectores de patrones.

Son las aplicaciones capaces de escuchar el tráfico de la red, en busca de patrones malignos definidos a través de firmas o reglas, y producir eventos de seguridad.

Las aplicaciones más comunes son los sistemas de detección de intrusos (IDS). Se basan en el análisis detallado de tráfico de la red, comparando el tráfico con las firmas de ataques conocidos o reglas de comportamientos sospechosos, como puede ser el escaneo de puertos. Los IDS analizan tanto el tipo de tráfico como el contenido y el comportamiento de los paquetes de la red.

Cualquier otro dispositivo de la red, como puede ser un router, firewall, o el mismo sistema operativo de los hosts, tienen la capacidad de detectar patrones en la red como puede ser un escaneo de puertos, intentos de spoofing, o posibles ataques por fragmentación. Cada uno de ellos tiene su propio log de seguridad capaz de alertar de posibles problemas en la red, que podremos recolectar para su posterior tratado en los motores de correlación.

Ossim integra varios detectores de patrones de código abierto. El detector más común en Ossim es el Snort (NIDS, Network Intrusión Detection System), incluye varios preprocesadores de detección de ataques y anomalías.

Otros detectores incluidos son Snare y Osiris (HIDS Host Intrusión Detection System), instalados en los sistemas monitorizados de la red.

5.5.3 Detector de anomalías

Los detectores de anomalías gozan de una capacidad de detección mucho más compleja e innovadora que la de los detectores de patrones. En este caso al sistema de detección no tenemos que especificarle mediante reglas que es un comportamiento bueno o malo, sino que es capaz de "aprender" por sí solo y alertar cuando un comportamiento difiere del comportamiento normal.

Está técnica provee una solución para controlar el acceso de usuarios privilegiados y ataques internos, como puede ser un empleado desleal, o simplemente hacen un mal uso de los recursos y servicios de la empresa. Casos en los que los detectores de anomalías son útiles:

- Nuevos ataques para el que aún no existen firmas, puede definir anomalías obvias para un detector de anomalías, debido a que el detector de anomalías construye líneas base del comportamiento de un sistema.
- Un gusano que puede haber sido introducido desde la red interna, malware, ataque de spam, pueden generar un número de conexiones anómalas que son fáciles de detectar.
- Uso de servicios con origen y destino anormales.
- Uso en horarios anormales.
- Exceso de tráfico o de conexiones (programas P2P).
- Cambios de sistemas operativos, ips, macs.

Estas aplicaciones pueden generar un número de nuevas alertas elevado, que podrían empeorar la visibilidad del estado de la red por si solas, pero si tomamos estas alertas como información que acompaña al resto de alertas, los niveles superiores realizarán una correlación más fiable y les permitirá detectar nuevas anomalías. OSSIM integra una amplia gama de detectores de anomalías:

- **Spade:** Detecta conexiones no usuales por puertos y destinos utilizados. Usado para mejorar el reconocimiento sobre ataques sin firma.
- **Aberrant Behaviour:** Es un plugin para Ntop que aprende el uso de parámetros y alerta cuando dichos parámetros se salen de los valores esperados.
- **ArpWatch:** Utilizado para detectar cambios de mac.
- **Pof:** Utilizado para detección de cambios de sistema operativo.
- **Pads y Nmap:** Utilizado para detectar anomalías en los servicios de red.

5.5.4 Sistemas de colección y normalización

El proceso de colección y normalización se encarga de unificar todos los eventos de seguridad provenientes de cualquier sistema de la red en una única consola y formato.

La recolección de datos se puede hacer de dos formas distintas en el sensor. Se puede enviar los datos desde el equipo analizado usando protocolos nativos del equipo al gestor central, o instalando agentes en el equipo analizado que recopilan la información en el host y la envían seguidamente. OSSIM normalmente no utiliza agentes y utiliza las formas de comunicación naturales de los sistemas.

La normalización implica la existencia de un parser o traductor que conozca los tipos de formatos de alertas de los diferentes detectores, capaz de homogeneizar el tratamiento y la visualización de todos estos eventos en una única base de datos "EDB". EDB, es la base de datos que OSSIM utiliza para almacenar todos los eventos que colecciona, es la base de datos más voluminosa.

De esta forma se podrá visualizar en la misma pantalla y con el mismo formato los eventos de seguridad de un determinado momento, ya sean del router, firewall, IDS o de cualquier host. Al tener centralizado en la misma base de datos todos los eventos de la red se podrán desarrollar procesos a niveles superiores que permitan detectar patrones más complejos y distribuidos.

5.5.5 Políticas de priorización

La prioridad definida para una alerta será dependiente de la topología de la red, inventario de cada máquina y del rol que estas desempeñan en la organización. Si una alerta que se refiere a un ataque al servicio IIS de Microsoft, llega a una máquina con sistema operativos Unix y servidor Apache, la alerta debe de ser despriorizada. En cambio, si existe una conexión sospechosa de un usuario sobre un servidor, el sistema debe priorizar la alerta dependiendo de la ubicación del usuario y del uso de la conexión.

El proceso de priorización de alertas se realiza mediante contextualización, es decir la valoración de la importancia de una alerta depende del escenario de la red. Este escenario está descrito en una base de conocimientos sobre la red formada por:

- Inventario de máquinas y redes (ip, mac, sistema operativo, servicios, etc).
- Políticas de acceso (desde donde a donde está permitido o prohibido).

Todos estos parámetros son alojados en la base de datos "KDB", que es la base de datos que OSSIM utiliza para parametrizar el framework. De esta forma el sistema conocerá la topología de la red, características de las máquinas y las políticas de seguridad definidas.

A través de la valoración de alertas se realizará una de las partes más importantes del filtrado de alertas recibidas por los detectores. Desde el framework del sistema podremos configurar las siguientes características:

- Política de Seguridad.
- Inventario de las máquinas de la red.
- Valoración de activos.
- Valoración de amenazas.
- Valoración de fiabilidad de cada alerta.
- Definición de alarmas.

Para que el proceso de priorización sea efectivo se debe realizar una continua y detallada especificación de la situación de la organización.

5.5.6 Valoración de Riesgos

La arquitectura de OSSIM ha sido diseñada para que todas las decisiones que se tomen a la hora de actuar sobre una alerta, se apoyen en función de la valoración de riesgos calculada. Por lo que es necesario comprender el proceso de cálculo de valor de riesgo que OSSIM realiza sobre cada evento.

La importancia que se debe dar a cada evento será dependiente de los factores siguientes:

- El valor del activo (dispositivo) implicado sobre el evento.
- La amenaza que representa el evento o cuanto daño puede hacer al activo implicado.
- La probabilidad de que este evento ocurra.

Dada la capacidad que OSSIM ofrece para el trabajo en tiempo real, se podrá medir el riesgo asociado al esquema actual en tiempo real. En este caso el valor del riesgo se medirá como el daño que produciría el evento y la probabilidad de que esté ocurriendo en este momento la amenaza.

Está probabilidad, derivada de la imperfección de los detectores (falsos positivos), representará el grado de fiabilidad de estos en la detección de una posible intrusión. Por ello, el valor de riesgo instantáneo producido por la recepción de una alerta, dependerá del daño que produciría el ataque, la probabilidad de que este ocurra y la fiabilidad que el detector proporciona.

OSSIM calculará el riesgo instantáneo de cada evento recibido, que será la medida objetiva que se utilizará para valorar la importancia que un evento puede implicar. Y así poder descartar falsos positivos que las organizaciones reciben a miles cada día, y a través de estas medidas se valorará la necesidad de actuar.

OSSIM incluye un monitor de riesgos (descrito posteriormente), que valorará el riesgo acumulado en un rango de tiempo, sobre redes y grupos de trabajo relacionados en un evento.

5.5.7 El motor de correlación

La función de correlación se puede definir como un algoritmo que realiza una operación a través de unos datos de entrada y ofrece un dato de salida.

Los sistemas de correlación ayudan a aumentar la capacidad de detección de los detectores que hoy en día existen en la mayoría de las redes, aumentando así la sensibilidad, fiabilidad, escalabilidad y la visibilidad limitada de cada detector.

Se podría pensar que instalar un sistema único centralizado capaz de localizar toda la información de la red resultaría más fácil. Pero para ello se necesitaría una visibilidad completa desde un punto único de la red y una capacidad de almacenamiento y de memoria ilimitada.

El motor de correlación desarrollado en OSSIM se encarga de comprobar cada uno de los eventos recibidos y busca evidencias o síntomas que prueben la veracidad de un ataque o si se trata de un falso positivo.

En OSSIM se ha desarrollado un modelo de correlación que tiene la capacidad de:

1. Desarrollar patrones específicos para detectar lo conocido y detectable.
2. Desarrollar patrones ambiguos para detectar lo desconocido y no detectable.
3. Poseer una máquina de inferencia configurable a través de reglas relacionadas entre sí capaz de describir patrones más complejos.
4. Permitir enlazar detectores y monitores de forma recursiva para crear cada vez objetos más abstractos y capaces.
5. Desarrollar algoritmos que ofrezcan una visión general de la situación de seguridad de la red.

El motor de correlación de OSSIM se alimenta mayoritariamente de dos elementos claves en la red de datos:

- Los **monitores** que proporcionan indicadores del estado.
- Los **detectores** que proporcionan alertas.

Como salida, el motor de correlación podrá devolver tanto una alerta como un indicador con un grado de fiabilidad mayor que los eventos correlacionados.

Métodos utilizados en el proceso de correlación.

El proceso de correlación se rige mediante tres métodos heterogéneos pero con un mismo objetivo.

- *Correlación mediante secuencia de eventos (Correlación lógica).* Se centra en buscar ataques conocidos y detectables, relaciona a través de reglas que implementarán una máquina de estados, los patrones y comportamientos conocidos que definen un ataque.
- *Correlación mediante algoritmos heurísticos.* Este método detectará situaciones sin conocer patrones y comportamientos que definen un ataque. Implementa funciones que mediante algortimos heurísticos intentará descubrir situaciones de riesgo que se alejan del comportamiento cotidiano, intentará suplir la incapacidad del método anterior, además se podrá obtener una visión general del estado de seguridad de la red.
- *Correlación mediante inventariado.* Los ataques recibidos tienen siempre como objetivo un determinado sistema operativo, servicio específico etc. Con el inventario de la red podremos descartar falsos positivos a máquinas que no cumplen dichas características y priorizar las máquinas de mayor riesgo como los servidores.

Correlación mediante secuencia de eventos (Correlación lógica).

La correlación lógica se implementa a través del panel de secuencias, en el cual se definen reglas que representan árboles de nodos de condiciones lógicas (secuencia de eventos). Este tipo de estructuras se conoce como árbol de decisión (and/or tree) utilizados en sistemas de inteligencia artificial.

La variable de fiabilidad crece según el motor de correlación avanza a través de los nodos (eventos) cumpliéndose las condiciones de cada uno de ellos. Cuando se cumple la condición de un nodo, el motor de correlación salta al primer hijo, si la condición del hijo no se cumple, se buscará el hermano (nodo en el mismo nivel de dependencia del nodo anterior). De esta manera se implementa la operación "AND" en el eje "Y" y la operación "OR" en el eje "X", figuras [5.9][5.10]. Cuantas más evidencias tengamos, más posibilidades hay de que sea real el ataque.

El motor de correlación desarrollado en OSSIM goza las siguientes características:

- Fuente híbrida. Acepta tanto patrones procedentes de detectores como indicadores procedentes de monitores.
- Posibilidad de definir orígenes y destinos variables.
- Define una arquitectura recursiva. Permite que las alertas de salida se tomen como nuevos eventos que se pueden volver a correlacionar por otras reglas. Cada regla genera un nueva alerta con una prioridad específica, esta alerta de salida se puede tomar como un evento más de entrada (probablemente con una mayor fiabilidad), creando la recursión y posibilitando la implementación de n niveles de correlación.
- Se puede definir el nivel de prioridad y fiabilidad de las nuevas alertas.
- Utiliza variables "elásticas" o capaces de medir el grado de prioridad y fiabilidad (ej: Denegación de servicios: total $\rightarrow$ prioridad grave, 50% $\rightarrow$ prioridad media, 15% $\rightarrow$ prioridad baja)
- Define una arquitectura distribuida jerarquizada, que permite definir n niveles de correlación en una topología distribuida.

Correlación mediante algoritmos heurísticos.

OSSIM implementa un algoritmo heurístico de correlación por acumulación de eventos en un determinado tiempo, con el objetivo de obtener una imagen del estado general de seguridad de la red.

Desde el cuadro de mando obtendremos una visión a alto nivel de las situaciones de riesgo sin conocer en ningún momento detalle de las características del problema, pero con una rápida y clara visibilidad. Se mostrará el nivel acumulado de riesgo, que será sensible a la cantidad de riesgo acumulado en una ventana de tiempo. Irá subiendo proporcionalmente según la cantidad y la prioridad que tengan los eventos recibidos, e irá bajando con el paso del tiempo en caso de no recibir nuevos eventos. Se dará máxima prioridad a los eventos definidos como "riesgo instantáneo".

Este método de correlación quiere suplir con un punto de vista opuesto a la correlación mediante secuencias de eventos, donde intenta caracterizar al máximo nivel de detalle los posibles ataques

El objetivo de este método es:

- Ofrecer una visión general rápida de la situación.
- Detectar posibles patrones que al resto de sistemas de correlación puedan pasar por alto, ya sea por ser ataques desconocidos o por falta de capacidad.

Correlación mediante inventariado.

Todo ataque tiene como objetivo un determinado sistema operativo o servicio especificado. La correlación de inventario comprueba si el sistema atacado usa ese sistema operativo o servicio objetivo del ataque. Si lo usa, podremos determinar que existe riesgo, por lo contrario, se puede confirmar que el evento para dicha maquina es un falso positivo. Este tipo de correlación depende de la fiabilidad del inventario, OSSIM incorpora además del inventario manual, un método de inventario automático.

5.5.8 Inventario automático.

Se realiza en los sensores con detectores pasivos que permiten de forma pasiva ver el tráfico de la red. También se realiza de forma centralizada desde el servidor, mediante analizadores de red que de forma activa encuentran hosts y servicios.

Ambos métodos automáticamente rellenan la base de datos de inventarios con la siguiente información:

- Tipo de sistema operativo y versión.
- Tipo de servicio y versión.
- Dirección IP y MAC.

OSSIM realiza el inventariado mediante el uso de aplicaciones de código abierto como:

- Nmap, analizador de red sin necesidad de un agente.
- POf, detector pasivo de sistema operativo sin necesidad de agente.
- Pads detector pasivo de servicios sin necesidad de agente.
- ArpWatch, detector pasivo de paquetes ARP "cambios de mac y de ip" sin necesidad de agente.

5.5.9 Definición de la política.

Una gran parte del comportamiento de OSSIM se configura a través de las políticas. La política es esencialmente un grupo de ajustes que manejan el comportamiento y seguridad de la red, con ellas se puede definir que hacer para ciertos eventos o alarmas con origen y destino conocido.

Por defecto el servidor OSSIM no tiene ninguna política definida, ya que este tema es totalmente dependiente de cada entorno. A continuación se describe como se define una política y como se crean nuevas políticas cuando sea necesario.

Al insertar una nueva política deberemos establecer los siguientes campos de información:

- **Fuente.** Indica la dirección origen de los acontecimientos que queremos registrar. Los eventos que no tienen un objetivo solo tienen dirección de origen, como pueden ser (cambios de sistemas operativos, cambios de mac, nuevo servicio, identificación de una vulnerabilidad, etc.). En este campo podemos seleccionar cualquiera de los equipos definidos anteriormente e incluso seleccionar como origen una de las redes definidas.

- **Destino.** En este campo indicamos el objetivo del evento. En caso de que no tenga un objetivo como hemos visto anteriormente, marcaremos este campo como cualquiera (OSSIM insertará como destino la dirección 0.0.0.0). Al igual que en el campo fuente podremos agregar tanto los equipos definidos como las redes.

- **Puerto de destino.** Identifica el puerto de destino del evento, podemos seleccionar cualquiera de los puertos definidos en la sección de puertos, o une vez más, si el evento no tiene un puerto en relación definido podemos definirlo con la etiqueta cualquiera.

- **Prioridad.** Cada uno de los eventos llega con su propia prioridad y fiabilidad, que ha sido generada en el nivel de priorización. Además estos eventos pueden generar nuevas alarmas con sus propios parámetros de prioridad y fiabilidad. Ambos tipos pueden ser perfeccionados desde la creación de políticas, ajustando la prioridad en caso de concordancia entre el evento/alarma y la política definida. El campo prioridad puede tomar valores entre 0 y 5 o el valor "-1" que obtendrá la prioridad del evento o alarma. El valor 0 hace que el evento sea invisible dentro de OSSIM ya que no le da ninguna importancia, por lo contrario el valor 5 le da una gran importancia al evento o alarma.

- **Plugin grupos**. Los plugins son los tipos de eventos que cada detector o monitor envía a OSSIM. Estos plugins vienen ya definidos en la instalación de OSSIM y cada vez que desarrollan una nueva versión son actualizados. Los plugins se identifican con dos números.

  - **Plugin Id.** Es el identificador padre y se suelen asignar por agentes (snort 1001, nessus 3001, snare 1518, etc.).
  - **Plugin Sid.** Es el subgrupo identificador dentro de cada "Plugin Id" e identifica los diferentes tipos de eventos para cada agente.

  Desde la sección grupo de plugins podemos agrupar los diferentes identificadores por alguna razón que tengan en común (en OSSIM vienen configurados cinco grupos donde se distinguen anomalías, Cortafuegos, eventos Windows, eventos Linux y disponibilidad). Al insertar una nueva política podemos definir que grupos son apropiados para la misma.

- **Sensor.**Identifica el sensor que debe generar los eventos asociados a la política.Como en los anteriores casos podemos seleccionar los sensores definidos en la sección de sensores.

- **Rango de tiempo.** Permite definir en que rango de tiempo esta política va a ser válida.

- **Meta.** Para quien va a ser "instalada" esta política. En realidad no se instala en el objetivo seleccionado, pero especifica que el objetivo ha de tener esta política en cuenta. Podemos elegir tanto los sensores agregados como los servidores.
- **Acciones de la política.** Cuando los objetivos de la política son generados por algún evento, se pueden definir algunas acciones a realizar.
- **Correlacionar eventos.**Indica si debería la equiparación del evento con la política ser usada para correlacionar con nuevos eventos.
- **Correlación cruzada.** Indica si debería usarse para correlacionar con plugins de correlación cruzada "cross-plugin", detectores de sistemas "ids-os" o con detectores de servicios "ids-service".
- **Almacén de evento.** Indica si deberían de ser almacenados en la base de datos.
- **Calificar los eventos.** Indica si estos eventos afectan a la valoración de riesgos, si debe de modificar los niveles de C & A.
- **Reenviar alarmas.** Indica si las alarmas generadas deben de ser reenviadas al resto de servidores.
- **Reenviar eventos.** Indica si los eventos generados deben de ser reenviados al resto de servidores.
- **Descripción.**Es un simple campo de descripción usado para describir de forma breve la política.

5.5.10 Directivas

Las directivas en OSSIM son un tipo especial de "plugin". Cuando una directiva genera una alarma lo que hace es crear un tipo especial de evento, que al igual que cualquier otro caso debe de ser generado por algún plugin.

Una empresa tras una exhaustiva configuración podría crear miles de directivas en su red, que sin una buena organización podría volverse en su contra y ser un monstruo incapaz de manejar. Para que esto no ocurra se ha definido una serie de categorías en donde se alojan las directivas usando el rango definido para cada categoría.

En las nuevas versiones de OSSIM se pueden definir a través del interface Web por lo que el resto de la descripción sobre la creación de directivas se hace en la parte experimental y anexos en el momento en que se necesitan.

5.5.11 Evaluación del Riesgo.

El riesgo es una interesante manera de normalizar y determinar un valor que puede ser utilizado para tomar decisiones sobre una serie de ataques realizados en un plazo específico de tiempo.

OSSIM utiliza dos valores de riesgo diferentes, uno para el origen y otro para el destino de un evento. Por lo tanto cada caso tiene dos riesgos. Esto significa que como tenemos dos máquinas diferentes para calcular el riesgo, cada una de ellas será evaluada con su propio valor del activo.

Estas son las dos reglas que vamos a seguir:

1. Si estamos calculando el riesgo de C, utilizaremos el valor del activo de la dirección origen.

2. Si estamos calculando el riesgo de A, utilizaremos el valor del activo de la máquina destino.

Así que, gracias a esto, el valor de riesgo de cada directiva se utiliza para modificar el valor de C & A de cada máquina. Para normalizar el cálculo del valor de riesgo se utiliza la siguiente fórmula:

$$\text{Riesgo} = (\text{Activo} * \text{Prioridad} * \text{Fiabilidad}) / 25$$

Esta es la manera de normalizar el valor del riesgo para todos los casos posibles, siendo el riesgo mínimo un 0 y el máximo riesgo el valor de 10. Las variables internas de la formula pueden tomar los siguientes rangos:

- Prioridad entre 0 y 5.
- Fiabilidad entre 0 y 10.
- Activo entre 0 y 5.

De esta forma se justifica el valor máximo y mínimo que puede alcanzar el nivel de riesgo de una directiva.

5.6 Gestión Web de OSSIM. Panel de control.

Una vez OSSIM ha sido configurado, desde la misma interfaz web tenemos multitud de secciones desde donde podremos visualizar y gestionar el estado de la red. En este apartado vamos a detallar cada una de las secciones que nos permitirán realizar una gestión diaria de nuestra red.

El panel de control es el punto de partida de la aplicación OSSIM. Una vez se ha configurado el servidor, desde aquí podremos controlar el estado de la red de una forma gráfica con una rápida visibilidad.

Finalmente los procesos que tienen lugar en OSSIM son:

- Las aplicaciones generan eventos.
- Los eventos son reconocidos y normalizados.
- Los eventos son enviados a un servidor central.
- Se hace una valoración de riesgo para cada evento.
- Se realiza una correlación de eventos.
- Almacenamiento de eventos.
- Acceso a los eventos almacenados.
- Acceso a la configuración.
- Acceso a la métrica e informe.
- Acceso a información en tiempo real del estado de la red.

Un despliegue típico de OSSIM podría ser el que se muestra en la figura [5.4]:

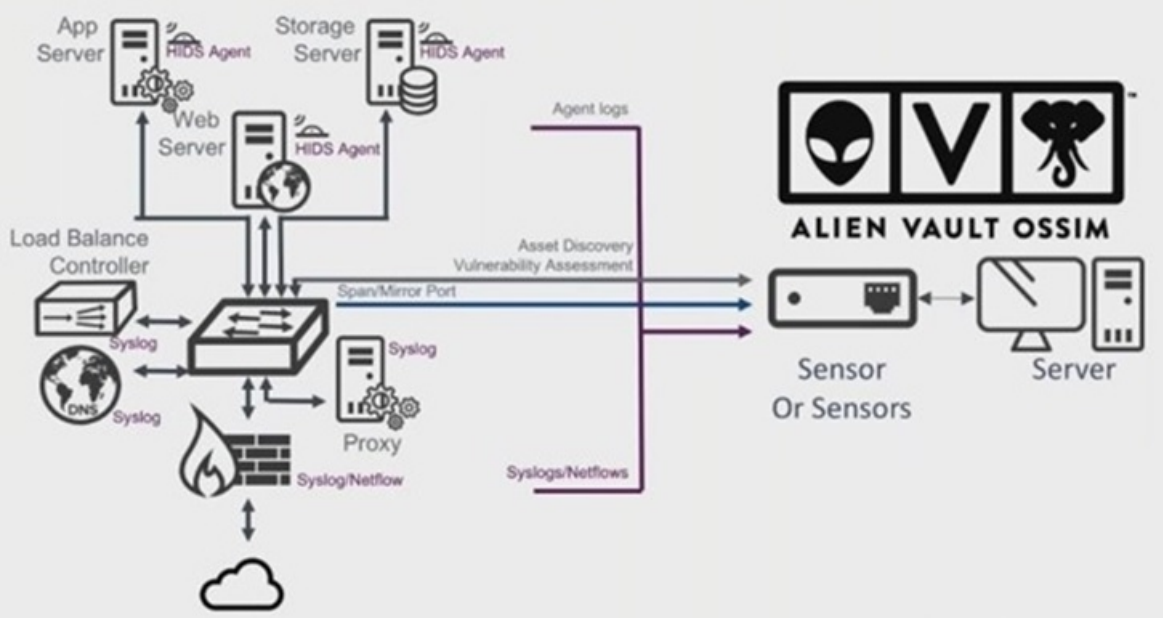


Fig. 5.4: Despliegue OSSIM

En la figura [5.5] puede verse un esquema del despliegue funcional de OSSIM en el que pueden apreciarse los bloques funcionales:

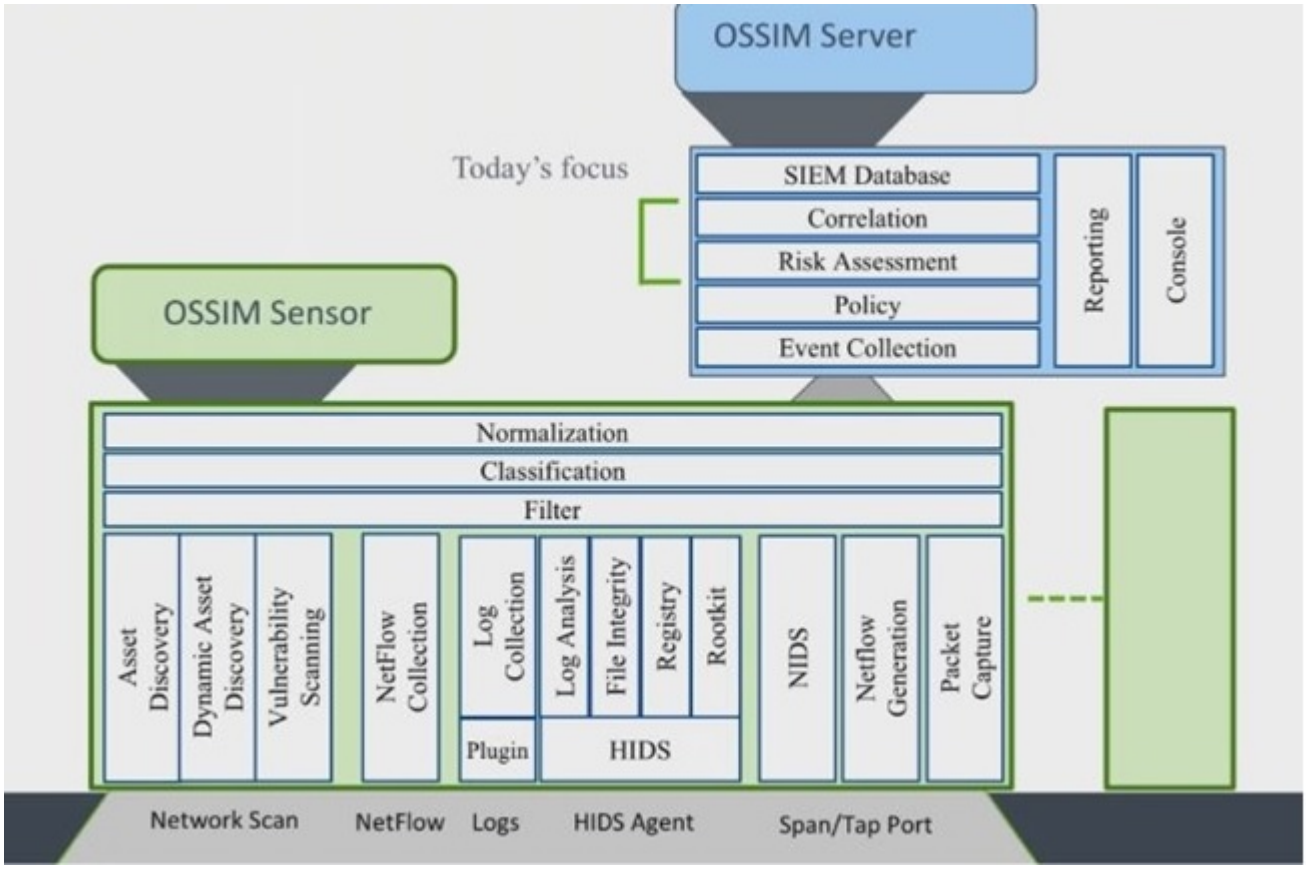


Fig. 5.5: Bloques funcionales OSSIM

El esquema básico de funcionamiento es el siguiente:

Cualquier log generado por cualquiera de las fuentes de datos (aplicación, sistema, red) es un evento. En la figura [5.6] se muestra el proceso seguido por los eventos después de su recogida.

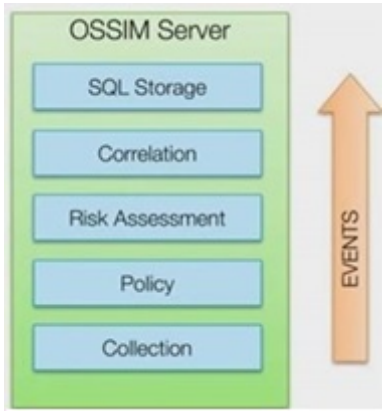


Fig. 5.6: Proceso seguido por los eventos

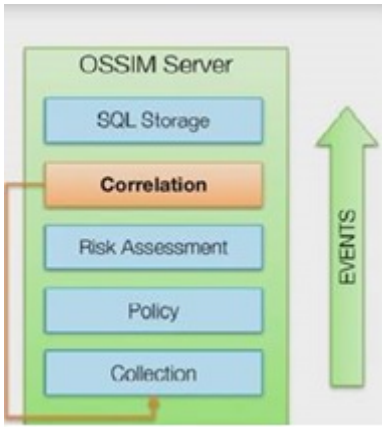


Fig. 5.7: Procesado de eventos con correlación

Otro proceso que tiene lugar en el funcionamiento de OSSIM es la correlación, como puede observarse en la figura [5.7] el motor de correlación genera nuevos eventos que son vueltos a inyectar en el Server de OSSIM y procesados como si llegasen del Sensor de OSSIM. El motor de correlación usa múltiples eventos para generar nuevos eventos con una mayor fiabilidad.

Los principales tipos de correlación que se llevan a cabo son: la correlación lógica y la correlación cruzada. A continuación se muestra la zona de la interface Web en dónde se configuran ambas, figura [5.8].

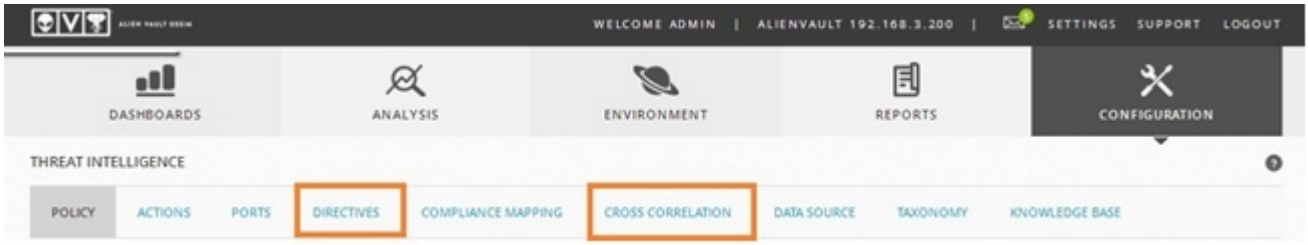


Fig. 5.8: Interfaz Web OSSIM: Directivas y correlación cruzada.

En la correlación lógica los eventos (creados por detectores y monitorización) una vez recibidos en el Server de OSSIM se analizan para buscar patrones de actividad maliciosa usando

las directivas de correlación, de manera que cuando un patrón es reconocido se generan nuevos eventos con nuevos valores de fiabilidad- Las directivas se definen usando árboles lógicos, tal como se muestra en la figura [5.9].

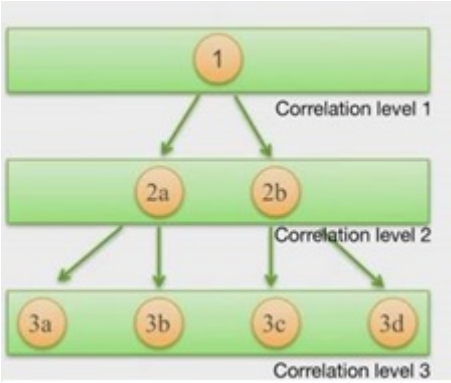


Fig. 5.9: Correlación lógica

Fig. 5.10: Niveles Correlación Lógica

En la que el eje horizontal define operaciones OR y el eje vertical operaciones AND. Los niveles de correlación pueden extenderse hasta dónde sea necesario, figura [30]. En la correlación cruzada se relacionan dos tipos de eventos diferentes, una gran mayoría de las relaciones cruzadas relacionan eventos IDS con vulnerabilidades previamente detectadas así por ejemplo en el caso de que un IDS detecte un ataque a un activo con una vulnerabilidad específica entonces la fiabilidad del evento se cambia a un valor de 10.

6. VULNERABILIDADES INFORMÁTICAS

6.1 Introducción.

En este capítulo se hace una descripción breve de lo que es una vulnerabilidad, sus tipos y estándares de identificación, dado que uno de los objetivos de este PFC consiste en el estudio e implantación de un sistema open source que permita detectar vulnerabilidades en un sistema informático.

Así mismo en el capítulo siete dedicado a la parte experimental, los distintos casos estudiados tratan distintos tipos de vulnerabilidades.

6.2 ¿Qué es una vulnerabilidad?

Se entiende por vulnerabilidad cualquier debilidad en el sistema informático que pueda permitir a las amenazas causarle daños y producir pérdidas en la organización. Se entiende que una vulnerabilidad es un fallo en un programa o sistema informático que da lugar a un fallo de seguridad. Ya que no todos los errores/fallos derivan en un fallo de seguridad así un error en un programa puede conducir a que su funcionamiento no sea correcto pero que este comportamiento anómalo no sea un fallo de seguridad.

Cualquier amenaza que ofrece un "camino" potencial para atacar un sistema aumenta el riesgo del sistema. Una vulnerabilidad de seguridad es una debilidad en un producto que puede permitir a un atacante comprometer la integridad, disponibilidad o confidencialidad de ese producto.

Una debilidad en un producto puede ocurrir de manera que esta se produzca por un fallo en el diseño del producto pero esto no supone una vulnerabilidad en seguridad. Un ejemplo sería: La elección de implementar un cifrado de 40 bits en un producto no constituye una vulnerabilidad aunque esta implementación sea inadecuada en algunos casos. En contraste, un método de cifrado que utilice 100 bits y que por un error en el diseño del producto descarte 60 de esos bits en la clave de cifrado si sería una vulnerabilidad.

Las vulnerabilidades son el resultado de un problema en un producto. Pero es importante destacar que los fallos asociados a los estándares no son vulnerabilidades de seguridad. Un ejemplo sería un navegador que se conecta a un servidor FTP enviando la información en texto plano, esto no es una vulnerabilidad ya que la especificación de FTP define el protocolo de esta manera. En cambio si el navegador envía en texto plano una sesión mediante SSL esto si es una vulnerabilidad ya que el estándar define que las sesiones SSL deben ser cifradas.

La integridad se refiere a lo fidedigno de un recurso. Un atacante que explota una debilidad en un producto para modificar silenciosamente este producto sin autorización está comprometiendo la integridad de este producto. Un ejemplo: La debilidad de que el administrador pueda modificar los permisos de todos los ficheros de un sistema no es una vulnerabilidad de seguridad, sin embargo que lo haga un usuario sin privilegios administrativos si puede constituir una vulnerabilidad.

La disponibilidad hace referencia a la posibilidad de acceder a un recurso. Un atacante que explota una debilidad en un producto, denegando el acceso a un usuario legítimo, está comprometiendo la disponibilidad de ese producto. Un ejemplo es un atacante que consigue

acceso a un sistema pudiendo permitir o no el funcionamiento de un servicio. En cambio un ataque de denegación de servicio no constituye una vulnerabilidad ya que el administrador aún controla el sistema.

La confidencialidad hace referencia al acceso limitado a la información por los usuarios autorizados. Un atacante que explota una debilidad en un producto para acceder a información no pública está comprometiendo la confidencialidad de este producto. Un ejemplo: El hecho de que un sitio web permita acceder a información no pública a un usuario no constituye una vulnerabilidad. Sin embargo la debilidad de que un atacante pueda conocer la existencia de un determinado fichero no constituye una vulnerabilidad en si misma pero si que puede ser útil para el atacante para labores de reconocimiento lo que le puede ayudar a explotar el sistema. Aún así sigue sin ser una vulnerabilidad ya que por sí misma esta debilidad no hace posible al atacante comprometer información.

6.3 Ciclo de vida de una vulnerabilidad

Una vulnerabilidad desde que se manifiesta por primera vez y hasta que se elimina pasa por una serie de etapas que constituye su ciclo de vida.

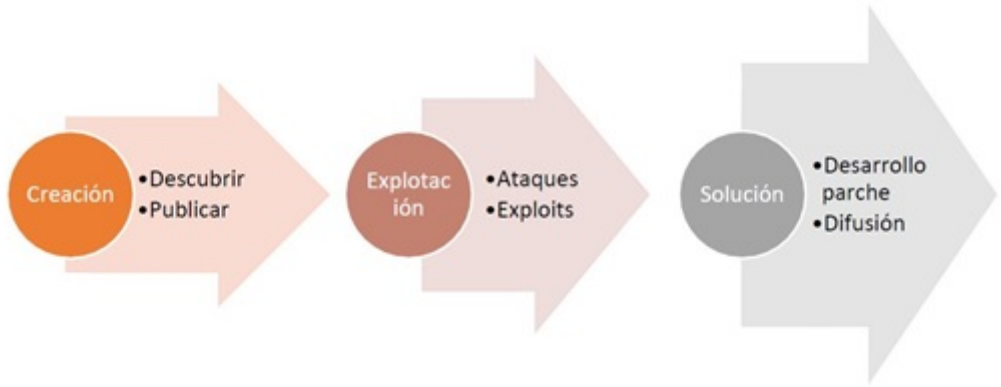


Fig. 6.1: Ciclo de vida de una vulnerabilidad

Se parte de la creación ya sea voluntaria o no de la debilidad siendo ésta producto de un error de desarrollo de software o de una actualización inadecuada. Tras crearse la debilidad se publicará una vez que se descubra. El descubrimiento puede ser tanto accidental como producto del testeo del desarrollador del software o de una empresa subcontratada para tal testeo. El objetivo es hacer pública la existencia de dicha vulnerabilidad para su rápida corrección y para que los usuarios sean conscientes del riesgo al que se exponen de tal forma que queda registrada y publicada en bases de datos de organismos como la "National Vulnerability Database" de EEUU[1] o "Security Focus"[2]. Esto es un arma de doble filo puesto que de esta manera se está también facilitando que los "crackers" conozcan la existencia de una vulnerabilidad y se propongan atacarla con fines ilícitos.

Es por ello que la siguiente fase del ciclo de vida de una vulnerabilidad cualquiera suele consistir en la "explotación" de la misma. Este término de nuevo es muy ambiguo y la explotación puede llevarse a cabo con fines benévolos o por el contrario con carácter delictivo. Se desarrollan por tanto "exploits" que ataquen dicha vulnerabilidad y la lleven al límite para poder desarrollar un parche que la corrija. Pero... ¿qué es un **exploit**?

La definición resulta sencilla: un exploit no es más que "un trozo de código (script), porción de software o conjunto de comandos que toman ventaja de un bug, error, o vulnerabilidad

[1] https://nvd.nist.gov/
[2] https://www.securityfocus.com/

para causar un comportamiento inesperado o indeseado del software o el hardware de un equipo electrónico."

Una vez que se ha experimentado lo suficiente con la vulnerabilidad es posible desarrollar un parche o una actualización que corrija de forma efectiva, y a ser posible permanente, el fallo existente. Este proceso no es fácil y requiere cierto tiempo por parte del fabricante para ponerse al día por lo que no siempre se soluciona directamente con un parche software. Es probable que mientras se busca una posible respuesta al daño causado, el fabricante sugiera tomar medidas preventivas como por ejemplo cerrar determinados puertos o deshabilitar ciertos servicios.

El objetivo final por parte de la empresa desarrolladora debe ser llegar a una solución para el agujero de seguridad creado y una vez lograda dicha solución difundirla mediante actualizaciones. El problema de los equipos no actualizados suele ser más grave por temas de seguridad que por características funcionales. El fin de las actualizaciones de Windows XP por ejemplo, hace que el exitoso sistema operativo de Microsoft sea ahora objetivo para aquellos que quieran crackear software.

6.4 Parámetros que identifican/definen una vulnerabilidad.

- **Producto.** Para identificar correctamente a una vulnerabilidad debe especificarse a qué productos o versiones de productos afecta concretamente.
- **Ubicación de la vulnerabilidad.** Los programas suelen estar compuestos de módulos que interactúan entre sí, de forma que una vulnerabilidad puede encontrarse en un módulo concreto o en una configuración concreta, pudiendo darse el caso de que no pueda activarse si el módulo no se encuentra activo. También puede suceder que se encuentre en un módulo intrínseco al programa y no se pueda desactivar como podría ser el caso de que se encontrarse en el núcleo de un sistema operativo.
- **Causa y consecuencia.** Este aspecto hace referencia al fallo técnico concreto cometido por el diseñador/programador de la aplicación concreta en la que aparece la vulnerabilidad. Las consecuencias técnicas de los fallos, normalmente, son diferentes pero suelen derivarse de los mismos fallos lo que da información a los buscadores de vulnerabilidades.
- **Impacto.** Se refiere a lo que puede conseguir un atacante que explote la vulnerabilidad. Así si no se han comprobado bien los permisos de acceso (causa) puede que se produzca un salto de restricciones (consecuencia) y que el atacante consiga elevar privilegios en el sistema (impacto). El impacto es una medida de la gravedad de una vulnerabilidad. Una vulnerabilidad de la mayor gravedad sería la posibilidad de ejecutar cualquier programa en el computador de la víctima.
- **Vector de ataque.** El vector de ataque se refiere a la manera que tiene el atacante de aprovechar la vulnerabilidad. Un vector de ataque de uso común consiste en el envío de información especialmente manipulada a un puerto de un sistema. Otro caso frecuente es conseguir que la víctima visite un enlace concreto.

6.5 ¿Dónde encontramos vulnerabilidades? Tipos de vulnerabilidades.

Se pueden encontrar vulnerabilidades en:

1. Aplicaciones web.
2. Sistemas operativos.

3. Dispositivos de red.
4. Debilidades en protocolos de red.
5. Configuraciones por defecto.
6. Errores humanos.

En primer lugar se realizará una clasificación muy genérica de los diferentes tipos de vulnerabilidades para, a través de ella, poder profundizar en casos más concretos. Las vulnerabilidades **en función de su origen** pueden ser las siguientes:

1. **Diseño:** Se basan en problemas basados en el planteamiento de las políticas de seguridad del sistema o en el desarrollo de los protocolos utilizados por la red.
2. **Implementación:** Basadas en fallos tanto en la planificación, como en la programación final del software que permiten por error del fabricante posibles "puertas traseras" que facilitan la manipulación de los equipos por individuos no deseados.
3. **Utilización:** Se debe a desconocimiento y falta de responsabilidad en la utilización de los equipos que, combinada con una mala configuración de los sistemas (ya sea por ignorancia o por negligencia), puede provocar una disponibilidad indeseada de herramientas que faciliten los ataques.
4. **Vulnerabilidad de día cero:** Se caracteriza por infectar equipos informáticos aprovechando vulnerabilidades desconocidas por los creadores y los usuarios de las aplicaciones. Es por tanto aquella vulnerabilidad que al estar recién descubierta no tiene solución pero sí da pie a experimentar con ella.

Una vez conocidos los diferentes tipos en función de su origen podemos centrarnos en detallar una a una las vulnerabilidades **en función de sus causas y** de los **efectos** que producen:

1. **Vulnerabilidad de validación de entrada:** Es aquella que se produce cuando la entrada que procesa un sistema no es comprobada adecuadamente y puede dar pie a una entrada corrupta.
2. **Vulnerabilidad de salto de directorio:** Se aprovecha la debilidad de la seguridad o su completa ausencia en un servicio de red para acceder a los diferentes directorios hasta llegar a la raíz del sistema.
3. **Vulnerabilidad de seguimiento de enlaces:** En esta ocasión y de forma muy parecida a la citada anteriormente, se produce el salto entre directorios a través de un enlace simbólico o un acceso directo.
4. **Vulnerabilidad de inyección de comandos en el sistema operativo:** No es más que la capacidad de un usuario para teclear instrucciones que puedan comprometer la seguridad.
5. **Vulnerabilidad de ejecución de código cruzado:** Se basa en la ejecución de un script de código por parte de un atacante en un dominio ajeno. Normalmente el aprovechamiento de esta vulnerabilidad se hace efectivo sobre aplicaciones web o funcionalidades del propio navegador. El objetivo que se persigue es la obtención de datos cruzados o incluso el control de sesiones. Se pueden presentar dos versiones, siendo la primera "reflejada", en la que se pasan variables entre dos páginas web para evitar el uso de sesiones de tal forma

que se aprovechan las cookies o incluso la cabecera HTTP para el ataque. La segunda versión sería la "persistente", en la que se localizan puntos débiles en los que se incrusta código.

6. **Vulnerabilidad de inyección SQL:** Esta vulnerabilidad se da directamente sobre las bases de datos basadas en lenguaje SQL. El objetivo es explotarla añadiendo código SQL sobre otro código SQL para cambiar el comportamiento del mismo. Se pueden cambiar consultas en las que se obtiene información por otras en las que se elimina o se pueden sobrescribir datos. Esta vulnerabilidad suele ser causa de la negligencia del administrador de sistemas que puede dejar la base de datos montada con problemas de seguridad y siendo vulnerable a la ejecución de código ajeno.

7. **Vulnerabilidad de inyección de código:**

   (a) **Inyección directa de código estático:** En este caso, un fallo en el software permite que se inyecte código en un archivo de salida que vaya a procesarse posteriormente. Puede llegar a almacenarse este código en una base de datos con lo que ésta quedaría corrupta y se debería considerar como tal.

   (b) **Evaluación directa de código dinámico:** La vulnerabilidad en el software permite que se puedan introducir directamente entradas que son ejecutadas directamente como código de tal forma que al tener tiempo de vida limitado es más difícil detectar el error.

   (c) **Inclusión remota en archivo PHP:** Este tipo de vulnerabilidad se debe a la función "include()" que permite enlazar archivos situados en otros servidores y ejecutar código PHP remoto en el servidor víctima. Debido a esta función y a otras como "include_once" o "require" es posible obtener una Shell para ejecutar comandos directamente sobre el servidor atacado.

8. **Vulnerabilidad de error de búfer:** Esta vulnerabilidad es una de las más comunes puesto que se aprovecha de la necesidad de las aplicaciones de utilizar búferes para almacenar información temporalmente mientras se procesa. Existen varios tipos:

   (a) **Desbordamiento de búfer (buffer overflow):** Se produce cuando se intentan meter en él más datos de los que el sistema es capaz de procesar. La consecuencia es que se almacenan los datos sobrantes en zonas de memoria adyacentes sobrescribiendo otros datos que no se deberían de ver afectados. Esto constituye un fallo de programación importante y puede ser utilizado con uso malintencionado como, por ejemplo, para tomar el control de una aplicación o provocar que alguna otra termine. La idea es que mediante un ataque de este tipo se almacena código arbitrario en los segmentos de memoria afectados con el objetivo de ejecutarlo posteriormente causando comportamientos inesperados en la ejecución de los programas. En lenguajes como C o C++ se puede comprobar bien esta vulnerabilidad puesto que permiten acceder directamente a la memoria de la aplicación. Se distinguen a su vez varios casos de overflow:

      i. **Desbordamiento de entero:** Se produce cuando el resultado de una operación aritmética es mayor del que se puede representar con la capacidad de almacenamiento disponible.

      ii. **Desbordamiento de pila:** Se produce cuando los datos se escriben una vez pasado el búfer.

      iii. **Desbordamiento de "montículo":** Aparece cuando los datos son escritos fuera del espacio que se les asignó.

(b) **Agotamiento de búfer:** Es una vulnerabilidad que aparece cuando en un búfer de comunicación entran tan pocos datos como para que la velocidad con que se leen sea mayor que la velocidad con que entran datos. Para evitar este fallo se necesita que el búfer detenga el proceso cuando esto ocurra.

9. **Vulnerabilidad por formato de cadena:** Sucede por cadenas cuyo formato es controlado externamente (como la función printf de C) y que pueden conducir a un problema de representación de datos o incluso un desbordamiento de buffer.

10. **Revelación o filtrado de información:** Es un tipo de vulnerabilidad que puede ser accidental o provocada. Se produce cuando se accede a información sensible.

11. **Gestión de credenciales:** Se debe a un mecanismo defectuoso para la gestión de usuarios y contraseñas y aquellos ficheros que almacenan dichas credenciales. Un fallo en este sistema puede dar pie a numerosos ataques como por ejemplo uno de fuerza bruta.

12. **Permisos:** Sucede cuando hay un problema con la administración de permisos derivado del mal funcionamiento del sistema y no debido a la gestión del administrador.

13. **Problema de autenticación:** Se debe a la incapacidad del sistema para autenticar a un usuario.

14. **De tipo criptográfico:** Fallos como la utilización repetitiva de ciertos algoritmos para generar números aleatorios en secuencias criptográficas o errores en la encriptación dan lugar a esta vulnerabilidad.

15. **Falsificación de peticiones en sitios cruzados:** Se produce cuando el agresor incrusta un código en una web para provocar una acción que no es la prevista por el administrador web. Así, uno de los ejemplos más típicos es la adición de una etiqueta HTML <img> de tal modo que en su interior se añade un código Javascript que lleva a ejecutar una acción en lugar de mostrar una imagen.

16. **Condición de carrera:** Se debe al acceso simultáneo a un recurso por parte de varios usuarios de tal forma que el orden de las acciones acarree una consecuencia diferente en función del mismo. Mediante esta vulnerabilidad se puede intentar obtener acceso al sistema.

17. **Error gestionando recursos:** El atacante provoca un consumo excesivo de CPU impidiendo que el sistema funcione adecuadamente.

18. **Error de diseño:** Fallo en la programación de la aplicación o en el diseño inicial de la misma que conlleva un agujero de seguridad una vez que la aplicación está en funcionamiento.

6.6 Análisis de vulnerabilidades. Herramientas automatizadas.

El análisis y escaneo de vulnerabilidades se hace en OSSIM de manera automatizada a través del dashboard tal y como se muestra en el anexo [7].

La gestión de vulnerabilidades es una funcionalidad usada para definir, identificar, clasificar y priorizar las vulnerabilidades de un sistema informático. La herramienta open source utilizada por OSSIM para realizar esta gestión es actualmente por defecto OpenVAS. OSSIM se encarga de realizar las siguientes tareas de configuración de manera automática:

1. Ejecución y programación de escaneos de vulnerabilidades.

2. Generación de informes de vulnerabilidades.

3. Actualización de las firmas de vulnerabilidades.

6.7 Estándares de vulnerabilidades

Se descubren decenas de vulnerabilidades cada día por lo que hace difícil su gestión. Su clasificación es compleja por lo que han surgido estándares con objeto de facilitar su clasificación e identificación.

- **CVE (Common Vulnerabilities and Exposures)[3].** El CVE es un estándar administrado por el MITRE que identifica las vulnerabilidades de manera unívoca. El formato CVE es: CVE-XXX-YYY, siendo XXXX el año en el que se asigna el código a la vulnerabilidad e YYYY un número de cuatro cifras. En el caso de grandes fabricantes el MITRE adjudica lotes de números. En otros casos el MITRE es el que se encarga de hacer la asignación según se van descubriendo vulnerabilidades. Por ejemplo la vulnerabilidad de OSSIM y NfSen (CVE-2017-6972) es una vulnerabilidad crítica encontrada en OSSIM en 2017.

- **CVSS (Common Vulnerability Scoring System)[4].** Este estándar asigna mediante fórmulas un grado de severidad a la vulnerabilidad, lo que permite tener una valoración cuantitativa de la gravedad de la vulnerabilidad. CVSS clasifica la facilidad de aprovechar el fallo y el impacto del problema teniendo en cuenta la confidencialidad, integridad y disponibilidad de los datos que pueden obtenerse aprovechando la vulnerabilidad. Mediante una calculadora online se puede calcular la gravedad de una vulnerabilidad encontrada en un producto (CVSS online Calculator)[5].

- **CVRF (Common Vulnerability Reporting Framework)[6].** Estándar que pretende dar uniformidad a la forma en la que se avisa de las vulnerabilidades de software a un programador o empresa que han creado el producto. Con este método se persigue que cuando un investigador o empresa hayan encontrado un fallo de seguridad en un programa, se le proporcione al fabricante la información precisa, rigurosa y adecuada para que pueda confirmarlo, entenderlo y parchearlo de forma eficaz.

[3] https://cve.mitre.org/
[4] https://www.first.org/cvss/
[5] https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
[6] https://www.icasi.org/cvrf/

7. PARTE EXPERIMENTAL

7.1 Resumen

Los experimentos se realizaron de manera incremental de manera que se va añadiendo elementos al experimento inicial para poder poner en práctica la amplia mayoría de funcionalidades que ofrece OSSIM y mostrar de forma práctica el contenido del trabajo. La topología de red usada para la realización de los experimentos es la mostrada en la figura [7.1]. En este diagrama se muestran dos WAN aun cuando no se utilizan simultáneamente. La primera WAN(192.168.0.0/24) se utiliza en los experimentos 1 y 2 para simular la conexión a Internet y la segunda WAN(192.168.2.0/24) se utiliza en el experimento 3 para simular un equipo atacante situado en Internet con objeto de estudiar el caso de acceso a una red LAN desde un equipo externo situado en Internet.

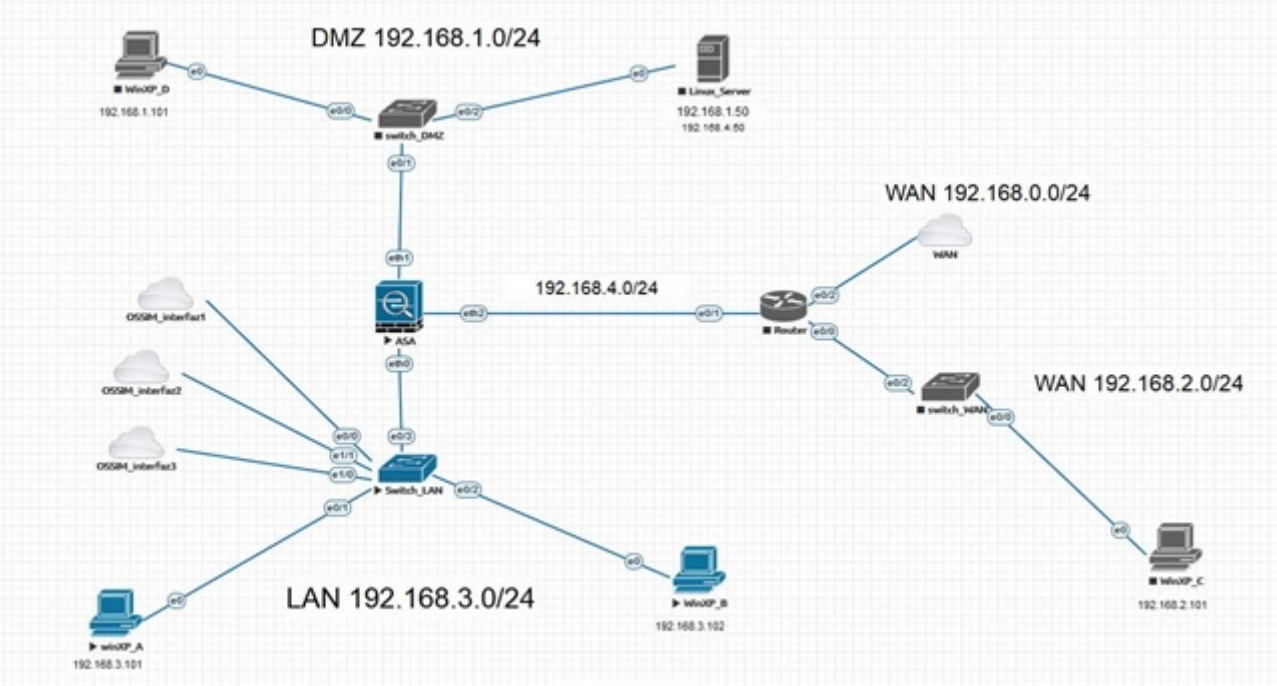


Fig. 7.1: Topología de red fase experimental

Se realizaron tres experimentos correspondientes a los casos:

- Red LAN Básica
- Red LAN con DMZ
- Red LAN con WAN

La división se justifica por el hecho de centrarse en la casuística que presenta cada una de las citadas zonas así como por los recursos necesarios para la virtualización simultánea de todas las áreas.

La virtualización de los equipos se ha llevado a cabo usando la versión gratuita de VMWARE Player y para la virtualización de red se ha usado la herramienta EVE-NG [1] La versión utilizada de OSSIM es la 5.4.0

7.2 Experimento 1: Red LAN básica

Caso en el que la empresa solamente presenta una LAN con acceso a Internet.

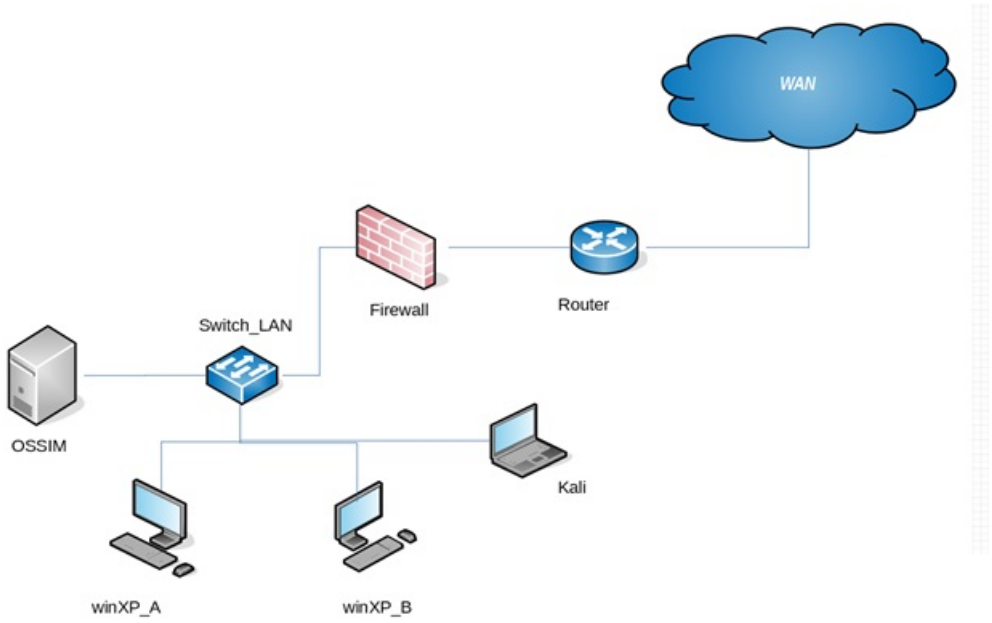


Fig. 7.2: Esquema topología LAN básica

Equipos	Interfaz	IP/MÁSCARA DE RED	OBSERVACIONES
Firewall	eth0	192.168.3.5/24	
	eth2	192.168.4.5/24	
OSSIM (Servidor + Sensor)	eth0	192.168.3.200/24	Interfaz de management
	eth1	192.168.3.201/24	Interfaz de recolección de logs
	eth2	No aplica	Este interfaz no tiene IP por recomendación de AlienVault porque es el interfaz de monitorización al que nada irá destinado en concreto sino que recibirá todo el tráfico. (modo promiscuo)
Windows XP A	E0	192.168.3.101/24	
Windows XP B	E0	192.168.3.102/24	
Router	E0/1	192.168.4.1	
	E0/2	192.168.0.X	
Kali	E0	192.168.3.103/24	
Switch LAN	E/0-6	No aplica	

Tab. 7.1: Detalle de topología: Experimento 1

[1] http://www.eve-ng.net/

Una vez configurado el NIDS, HIDS, Firewall y la monitorización de disponibilidad como se detalla en los anexos [1] y [2], se realiza una simulación de un ataque a los equipos de la red desde la máquina kali. Después de realizar este proceso se puede observar como resultado de las configuraciones realizadas con OSSIM, que es capaz de detectar estos ataques así como de las vulnerabilidades que subyacen de la instalación hecha. Esto confirma la importancia de uno de los aspectos que es de obligado cumplimiento en diferentes estándares y normas ISO, las auditorias de seguridad periódicas.

El proceso de ataque a la red con Kali Linux será el siguiente:

1. Escanear la red con nmap para descubrir los equipos de la red así como los servicios en cada uno de ellos.

2. En base a la información extraída en el paso 1 se lanzaran exploits desde la máquina kali con objeto de comprometer la confidencialidad, integridad o disponibilidad de los activos.

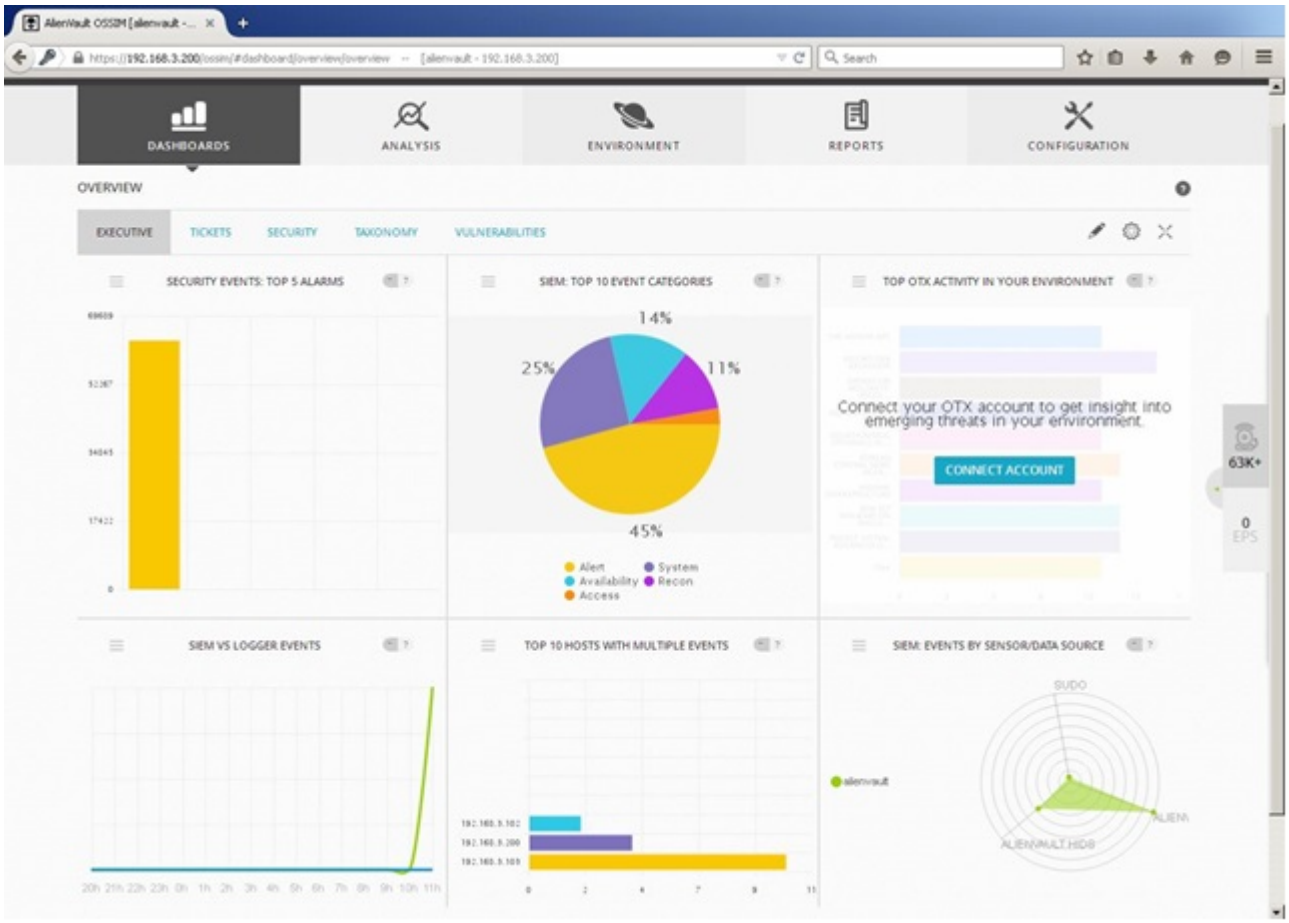


Fig. 7.3: Panorámica OSSIM después del escaneo NMAP.

Después de concluir el paso uno se puede observar el resultado en OSSIM en Availability donde el HIDS de Alienvault nos muestra como durante un breve período de tiempo el servidor OSSIM ha estado sobrepasando la capacidad de memoria.

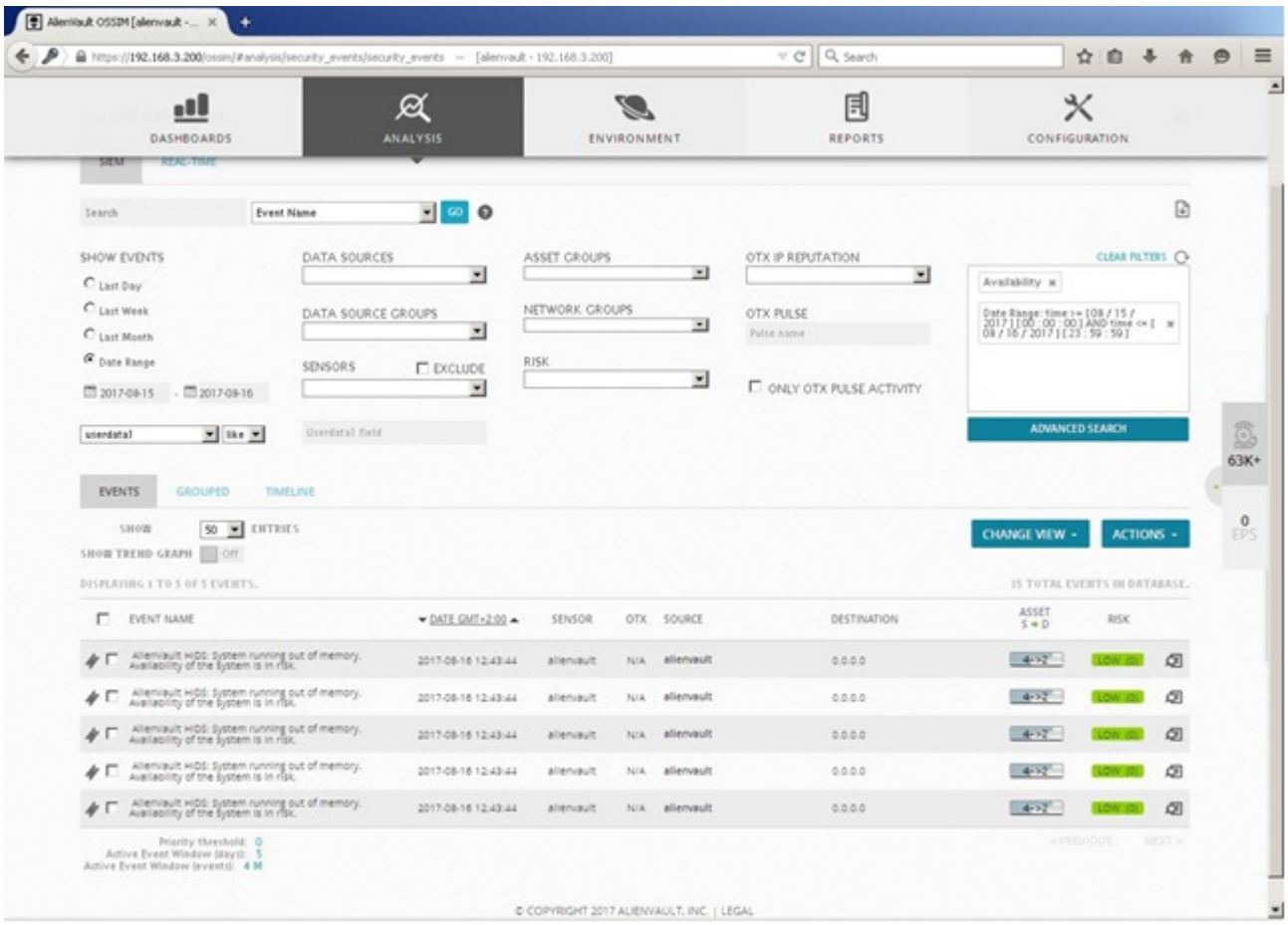


Fig. 7.4: Eventos generados por el HIDS de OSSIM durante el escaneo.

Navegando a **Alert**:

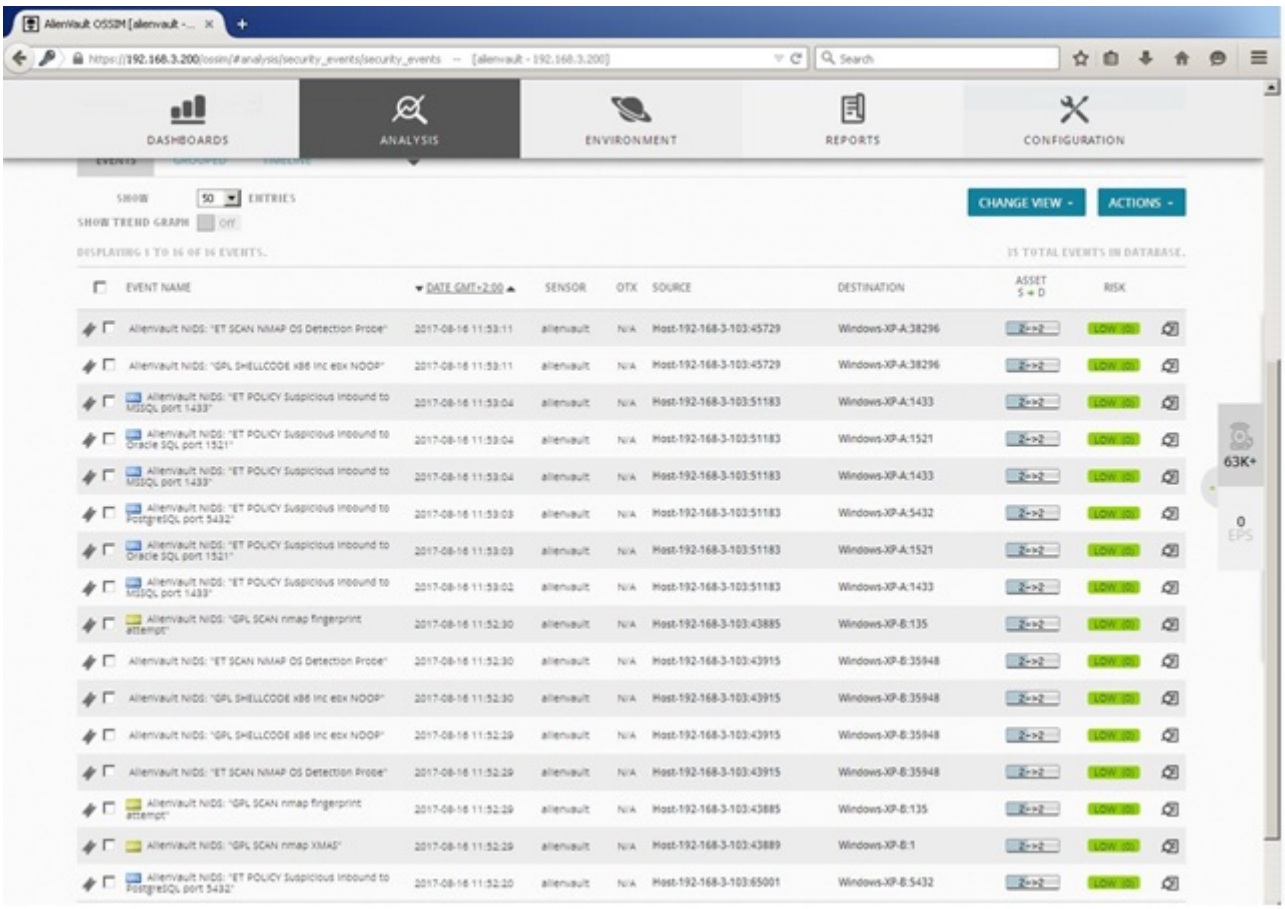


Fig. 7.5: Alertas mostradas por OSSIM tras escanear con NMAP

Navegando a **Recon**:

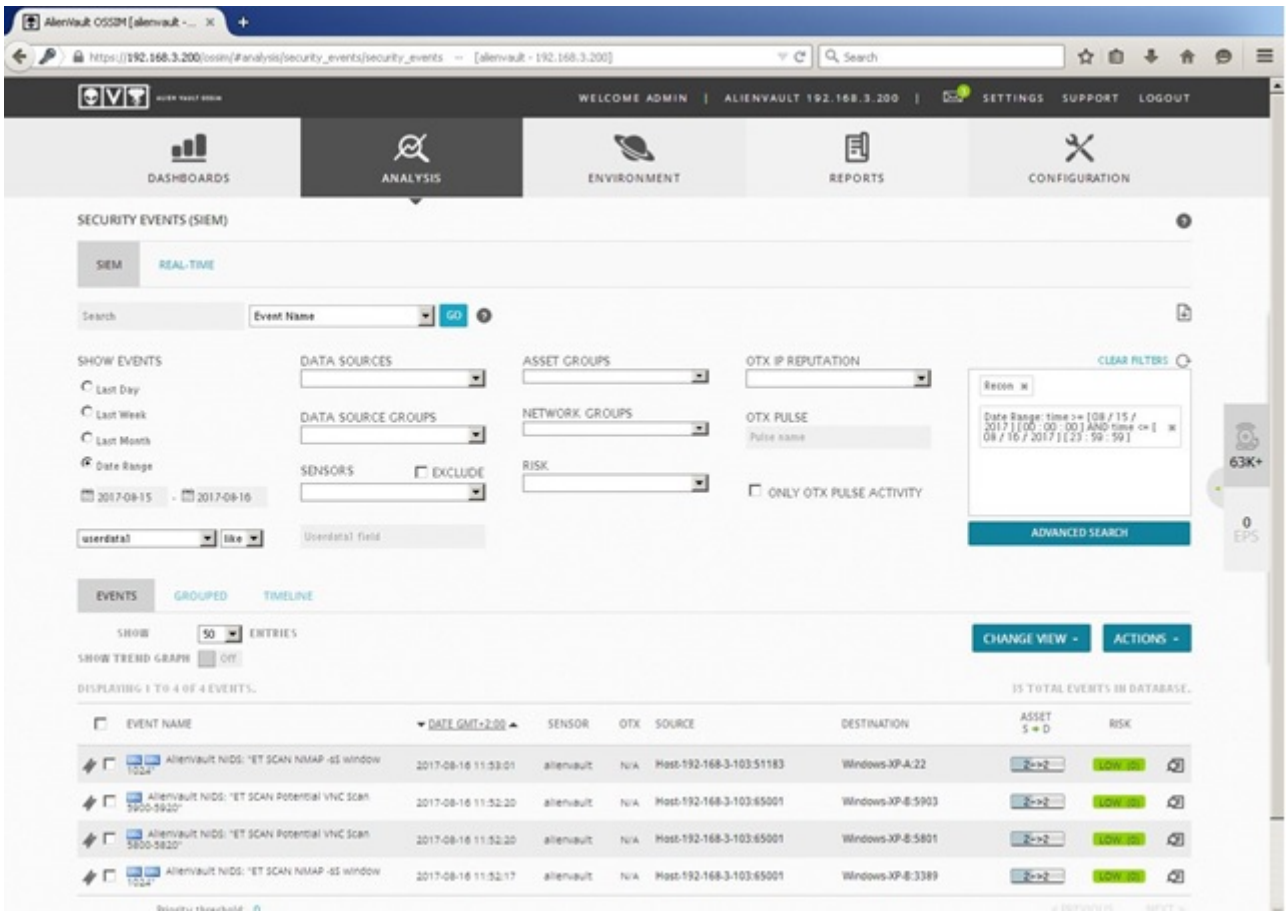


Fig. 7.6: Detalle del ataque de reconocimiento con NMAP

En la figura [7.6] se observa como el NIDS de OSSIM ha detectado los escaneos con nmap. Seleccionado uno de estos eventos podemos ver información relevante del evento:

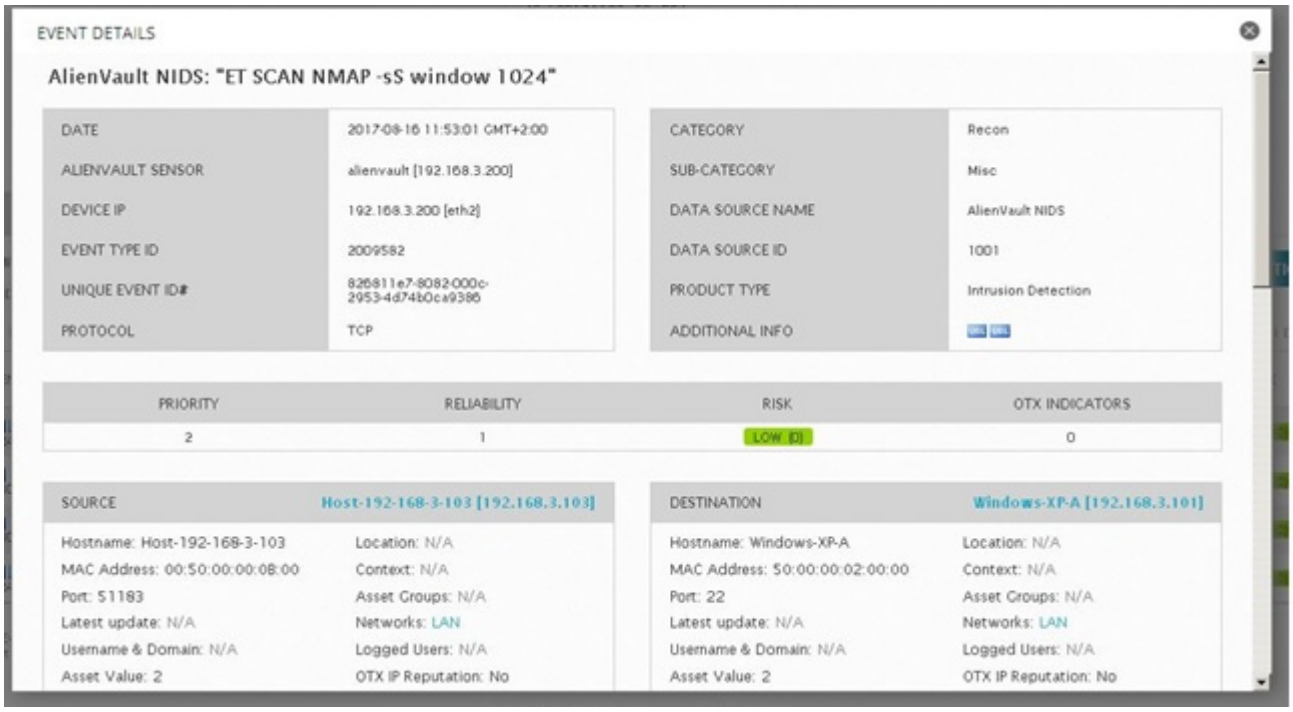


Fig. 7.7: Detalle del evento 1/3

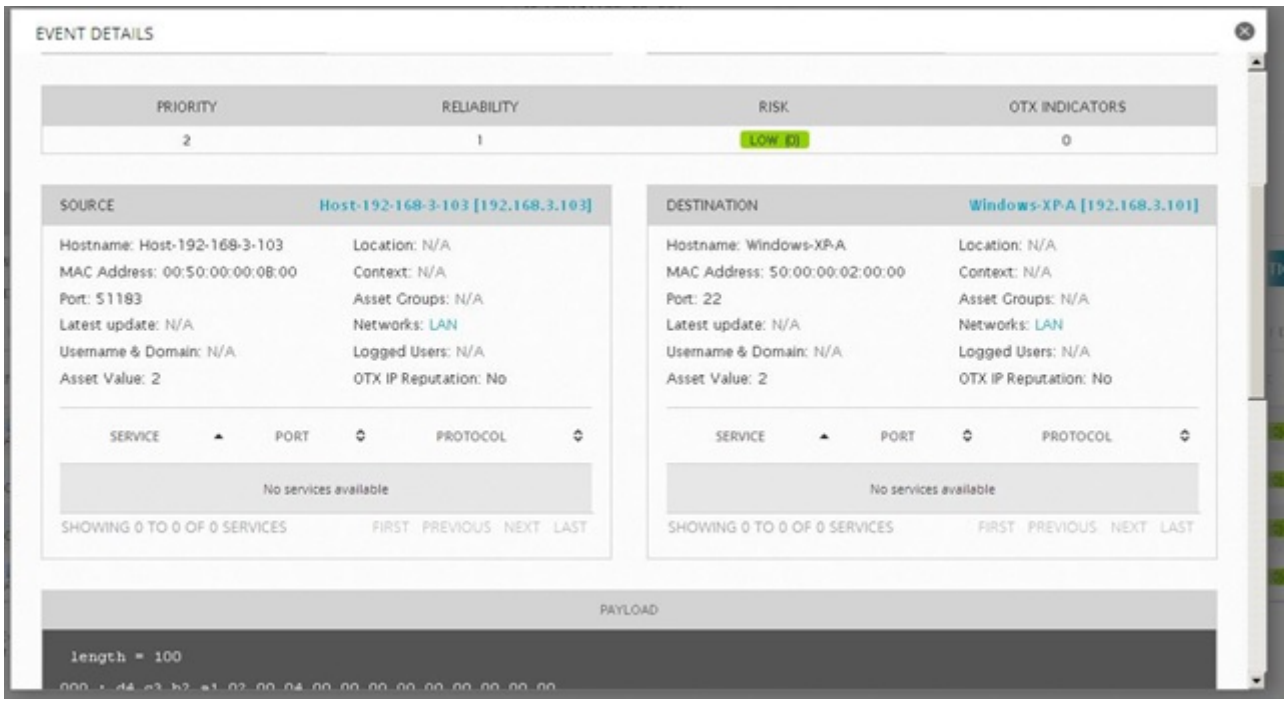


Fig. 7.8: Detalle del evento 2/3

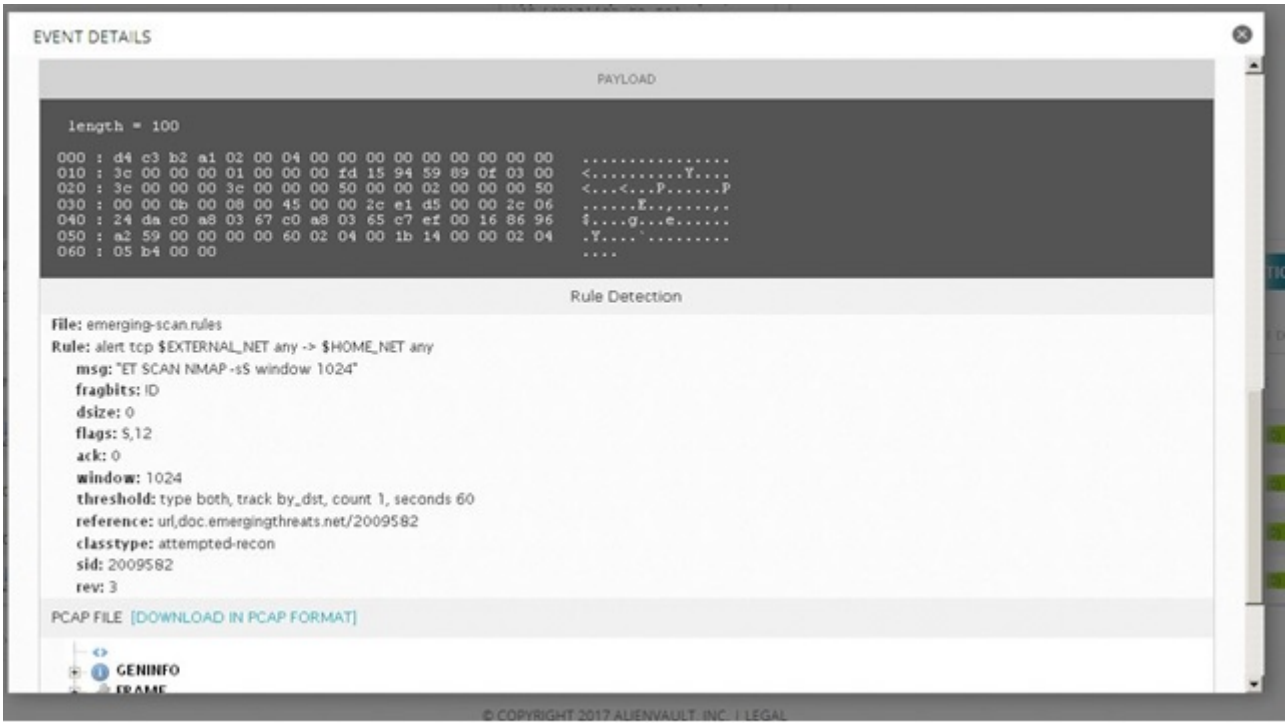


Fig. 7.9: Detalle del evento 3/3

En concreto y a diferencia de otros eventos, los eventos generados por el NIDS nos muestran la regla que ha provocado que se genere el evento. Además de un enlace en el que podemos encontrar más información acerca de este evento.

7.3 Experimento 2: Red LAN + DMZ

7.3.1 Descripción.

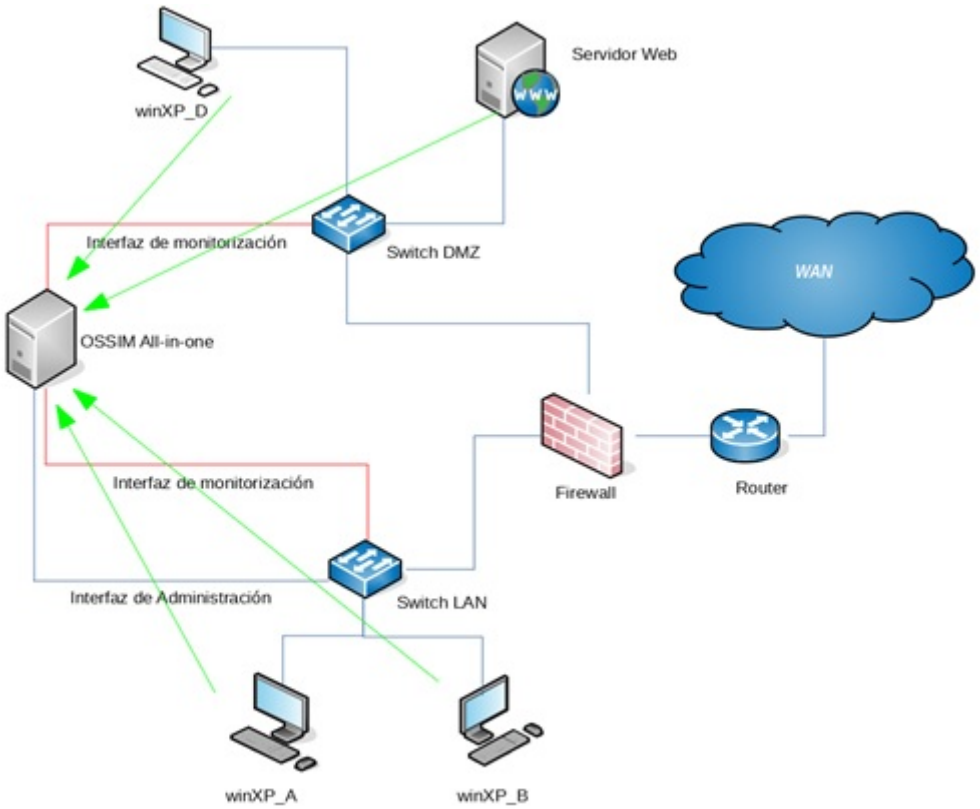


Fig. 7.10: Topología de Red Experimento 2: LAN + DMZ

Equipos	Interfaz	IP/MÁSCARA DE RED
Firewall	eth0	192.168.3.5/24
	eth2	192.168.4.5/24
	eth1	192.168.1.5/24
OSSIM (Servidor + Sensor)	eth0	192.168.3.200/24
	eth1	192.168.3.201/24
	eth2	No aplica
Windows XP A	E0	192.168.3.101/24
Windows XP B	E0	192.168.3.102/24
Router	E0/1	192.168.4.1
	E0/2	192.168.0.X
Windows XP D	E0	192.168.1.101/24
Servidor Web	E0	192.168.1.50/24
Switch DMZ	E/0-6	No aplica
Switch LAN	E/0-6	No aplica

Tab. 7.2: Detalle de topología: Experimento 2

El objetivo de este experimento es poner en práctica los conceptos previamente introducidos de correlación, acciones y política. Para ello se realizará un ataque de fuerza bruta al servicio SSH del servidor web, de manera que se correlacionen eventos que indiquen un intento de

conexión a SSH generando así un nuevo evento que indique que se está realizando un ataque al SSH. Posteriormente se configurará una acción, la acción será deshabilitar dicho servicio, y por último se configurará una política para ejecutar esta acción con el evento generado anteriormente.

En primer lugar se configurará el HIDS en la DMZ así como el interfaz de monitorización en dicha red. Posteriormente se introducirán las políticas, acciones y correlación.

Las acciones a realizar en el experimento son:

1. Configurar HIDS en la DMZ.
   - WinXP_D.
   - Servidor Web.
2. Configurar el NIDS en la DMZ.
3. Configurar la correlación en OSSIM.
4. Configurar Acciones en OSSIM.
5. Configurar Políticas en OSSIM.
6. Verificación de los pasos anteriores (Ataque)

7.3.2 Configurar HIDS en la DMZ. Configurar el HIDS en el servidor web.

En este apartado se configurará el HIDS en el Servidor Web. El equipo "Servidor Web" es un Linux Debian.

Para el despliegue del HIDS en Linux Alienvault recomienda realizar la descarga e instalación del agente desde la página oficial (https://ossec.github.io/downloads.html).

El proceso realizado ha sido el siguiente:

1. Descargar el agente
2. Descomprimirlo
3. Seguir el proceso de instalación

Una vez completado el proceso de instalación nos dirigimos a OSSIM y realizamos el siguiente proceso para añadir el agente a OSSIM:

1. Se navega a **Environment → Detection**
2. Se navega a **HIDS → Agents → Agent Control → Add Agent**
3. Se selecciona el agente de la lista de activos.
4. Se guarda.
5. Se extrae la clave.

Para finalizar el proceso en el equipo "Servidor Web", se realiza el siguiente proceso:

1. Se ejecuta: /var/ossec/bin/manage_agents
   Introduciendo la tecla "I" y insertando la clave previamente extraida en OSSIM.
2. Se edita el fichero /var/ossec/etc/ossec.conf modificando la dirección IP del servidor OSSIM.
3. Se inicia el agente ejecutando: service ossec start

Por último desde el equipo OSSIM se navega a **Environment → Detection** y se hace click en **HIDS Control** y después en **Restart**.

7.3.3 Configurar correlación en OSSIM

El objetivo de la correlación es generar un nuevo evento en base a otros eventos. En este caso se generaró el evento "Ataque fuerza bruta al servidor Web" como resultado de la correlación de eventos previos generados por el NDIS.

En este experimento se va a configurar una directiva para detectar un ataque de fuerza bruta por SSH en el servidor web. De este modo cuando se sucedan un número de accesos fallidos de login en servidor se generará un nuevo evento "Ataque de fuerza bruta en el servicio SSH en el servidor". Las directivas generan alertas

El proceso seguido es el siguiente:

1. Navegar a **Configuration** → **Threat Intelligence** y hacer click **Directives**. Click **New Directive**.
   Rellenar el formulario como se indica a continuación:
   (a) Name for the directive : " Ataque fuerza bruta Servidor Web"
   (b) Taxonomy:
   (c) Intent: "Reconnaissance & Probing"
   (d) Strategy: "WebServer Attack"
   (e) Method: "Attack"
   (f) Priority: 3
   (g) Click **Next**.
2. Después de configurar la directiva se añadirá una regla de primer nivel:
   (a) En **Name for the Rule**, escribir "Intento de ataque", y click **Next**.
   (b) En **Rule name** → **Plugin**, buscar "ALIENVAULT NIDS".
   (c) En **Rule name** → **Plugin** → **Event Type** buscar: "ET SCAN LibSSH Based Frequent SSH Connections Likely BruteForce Attack!"
   (d) En **Rule name** → **Plugin** → **Event Type** → **Network**,
      i. Seleccionar el activo destino "Servidor-Web"
      ii. Seleccionar el puerto destino 22
   (e) En **Rule name** → **Plugin** → **Event Type** → **Network** → **Reliability**, click 1.
   (f) Click **Finish**.
3. Por último se configurará una regla de segundo nivel
   (a) Click en el signo más (+) verde a la derecha de la regla uno en la columna **Action**.
   (b) Seguir los pasos 1 y 2 al añadir la regla de primer nivel.
   (c) **Rule name** → **Plugin** → **Event Type**, click **Plugin SID from rule of Level 1**.
   (d) **Rule name** → **Plugin** → **Event Type** → **Network**.
      i. Seleccionar **Source IP from level 1**
      ii. Seleccionar **Destination IP from level 1**
      iii. Seleccionar **Destination Port from level 1**
   (e) En **Rule name** → **Plugin** → **Event Type** → **Network** → **Reliability**, click **+2**.
   (f) Click **Finish**.

(g) En la columna **Timeout**, escribir "60" (segundos) en la segunda regla, y después click **OK**.

(h) En la columna **Occurrence** , click "1" en la segunda regla, escribir "15",y después click **OK**.

7.3.4 Configurar acciones en OSSIM

Para configurar la acción el proceso seguido ha sido el siguiente:

1. Navegar a **Configuration** → **Threat Intelligence** → **Actions** hacer click en **New**.
2. Cubrir los campos como se indica a continuación:
   (a) Name: " Notificación Ataque a Servidor Web"
   (b) Description: "Se ha producido un ataque al Servidor Web IP: DST_IP "
   (c) Type: "Open a ticket"
   (d) Condition: Any
   (e) To:
   (f) In Charge: User: "admin"

7.3.5 Configurar políticas en OSSIM

1. Navegar a **Configuration** → **Threat Intelligence** → **Policy**.
2. En la sección **Default Policy Group** , click en **New**.
3. A continuación se cubre la sección de condiciones de la polícica:
   (a) Event types: "Directive events"
   (b) Actions: "Notificación Ataque al Servidor Web"
   (c) Siem: se deja la configuración por defecto
   (d) Forwarding: no está disponible en la versión open source de OSSIM.

7.3.6 Verificación de los pasos anteriores (ataque)

En este caso se realizará un ataque por fuerza bruta al servidor SSH. Como consecuencia de la correlación de los eventos individuales generados por el NDIS de OSSIM se generará un nuevo evento: "directive_event: ataque de fuerza bruta servidor web". Después y como consecuencia de haber configurado una política para los eventos generados por la directiva anterior (correlación) se llevará a cabo la acción abrir un ticket.

El proceso seguido ha sido el siguiente:

En la máquina atacante (Kali) ejecutar:

hydra –l usuario –P diccionario.txt 192.168.1.50 ssh

Se puede observar como aparece el nuevo evento fruto de la correlación, figura[7.11].

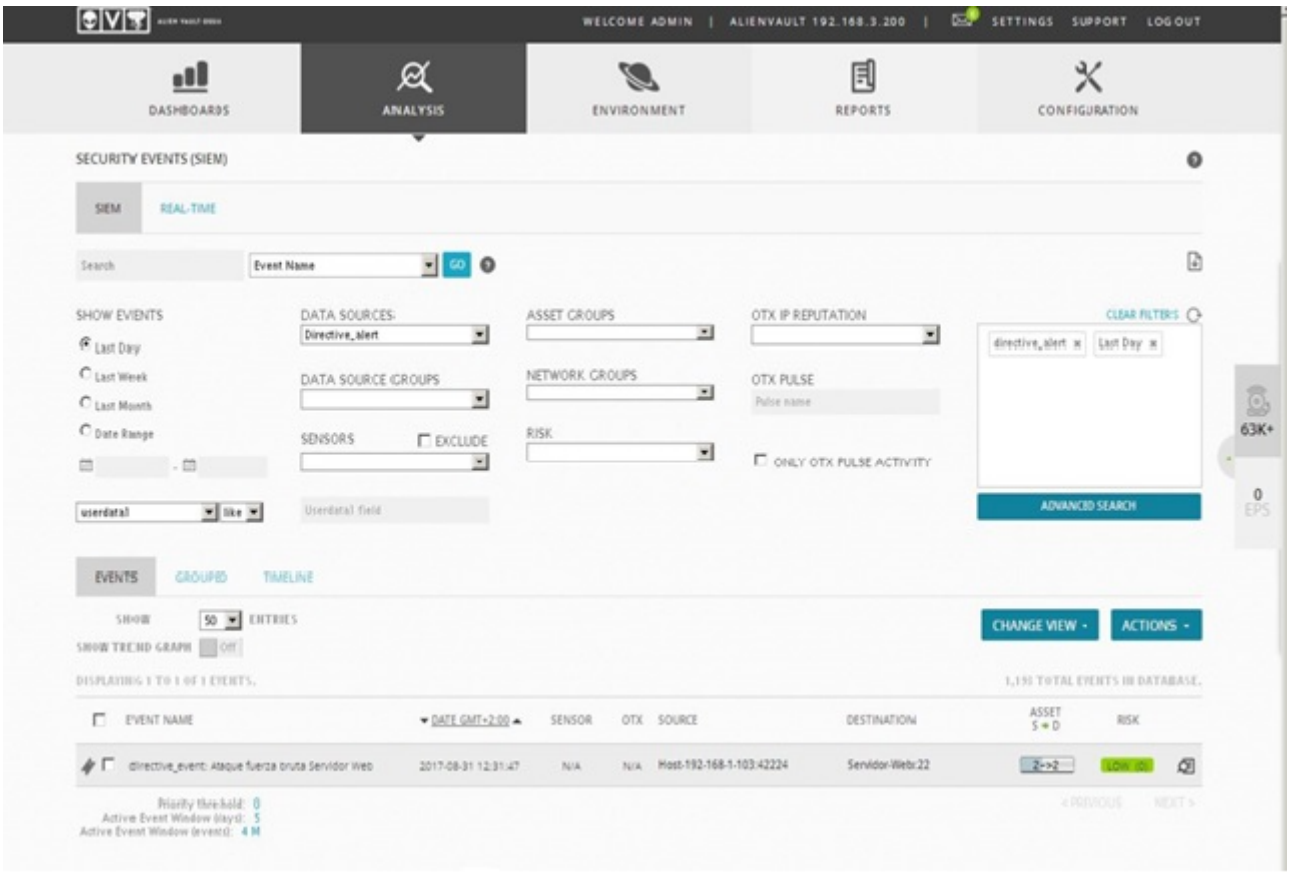


Fig. 7.11: Evento generado por una política de correlación tras el ataque de fuerza bruta SSH

A raíz de aplicar una política a este tipo de eventos se realiza la acción indicada: abrir un ticket, figura[7.12].

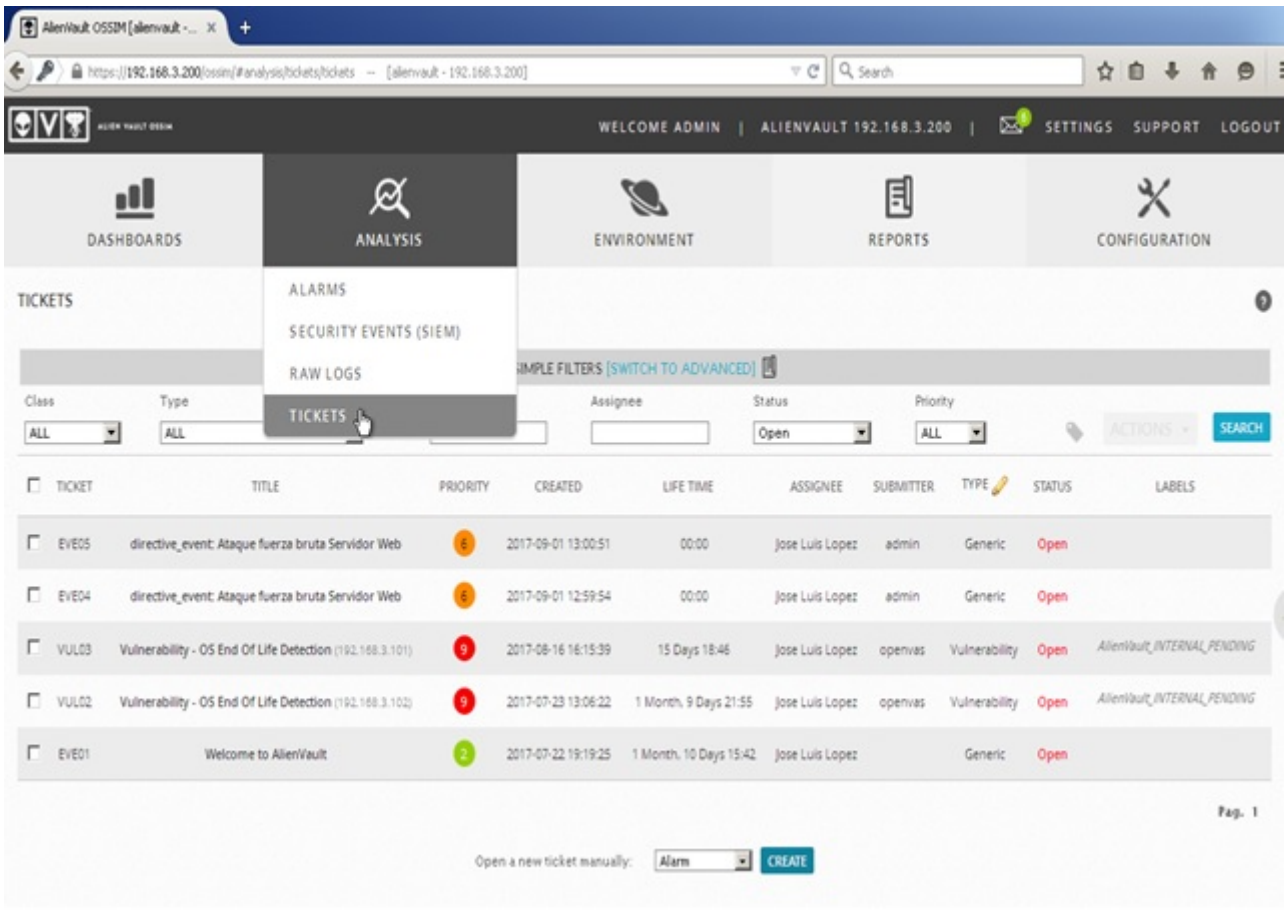


Fig. 7.12: Ticket abierto por OSSIM a consecuencia del ataque de fuerza bruta

7.4 Experimento 3: Detección de ataques en la red LAN/DMZ/WAN

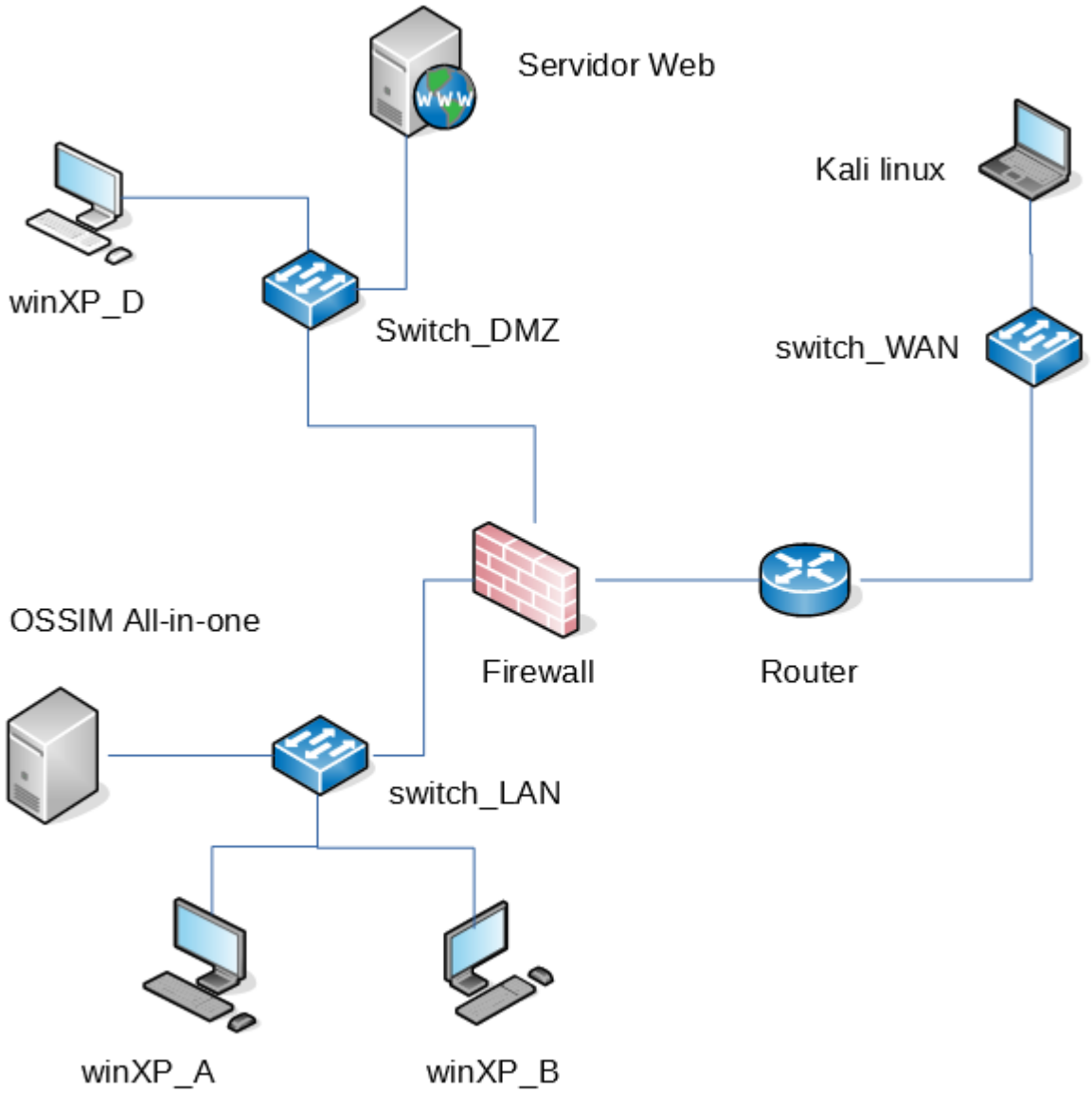


Fig. 7.13: Topología de la red experimento 3: LAN + DMZ + WAN

Equipos	Interfaz	IP/MÁSCARA DE RED
Firewall	eth0	192.168.3.5/24
	eth2	192.168.4.5/24
	eth1	192.168.1.5/24
OSSIM (Servidor + Sensor)	eth0	192.168.3.200/24
	eth1	192.168.3.201/24
	eth2	No aplica
Windows XP A	E0	192.168.3.101/24
Windows XP B	E0	192.168.3.102/24
Router	E0/1	192.168.4.1
	E0/2	192.168.3.1
	E0/0	192.168.2.1
Windows XP D	E0	192.168.1.101/24
Servidor Web	E0	192.168.1.50/24
Switch DMZ	E/0-6	No aplica
Switch LAN	E/0-6	No aplica
Switch WAN	E/0-6	No aplica
Kali Linux	E0	192.168.2.102

Tab. 7.3: Detalle de topología: Experimento 3

En este experimento se ha sustituido la WAN por un switch y un equipo con Kali Linux con el objetivo de simular un atacante exterior. Se realizan simulaciones de un ataque desde la WAN a la LAN.

En este experimento se detectará un ataque que intenta obtener acceso a la LAN desde la WAN, utilizando para ello la zona DMZ.

Se trata de representar las condiciones generadas por diferentes vectores de ataque como pueden ser un empleado descontento, la conexión de un usb infectado o un ataque de phishing al email.

Para ello se simula el comportamiento de un troyano del que previamente conocemos su compor-tamiento. Este troyano al que llamaremos "simulTroyan" al ser ejecutado en una máquina windows se intentará conectar al equipo atacante situado en la WAN a través del puerto 4444 TCP.

A continuación se muestra el comando para la generación del troyano:

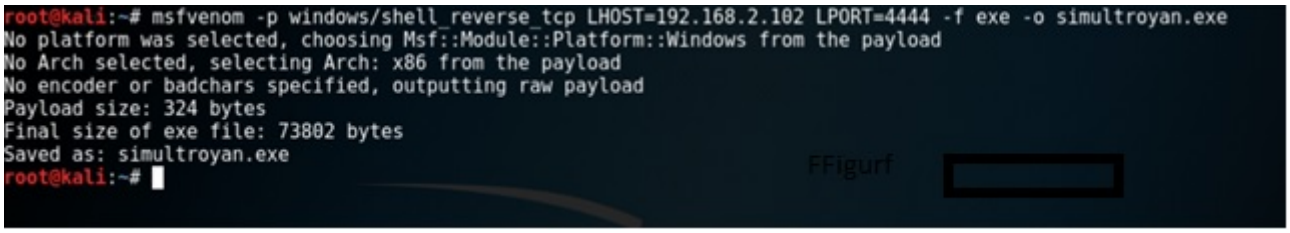


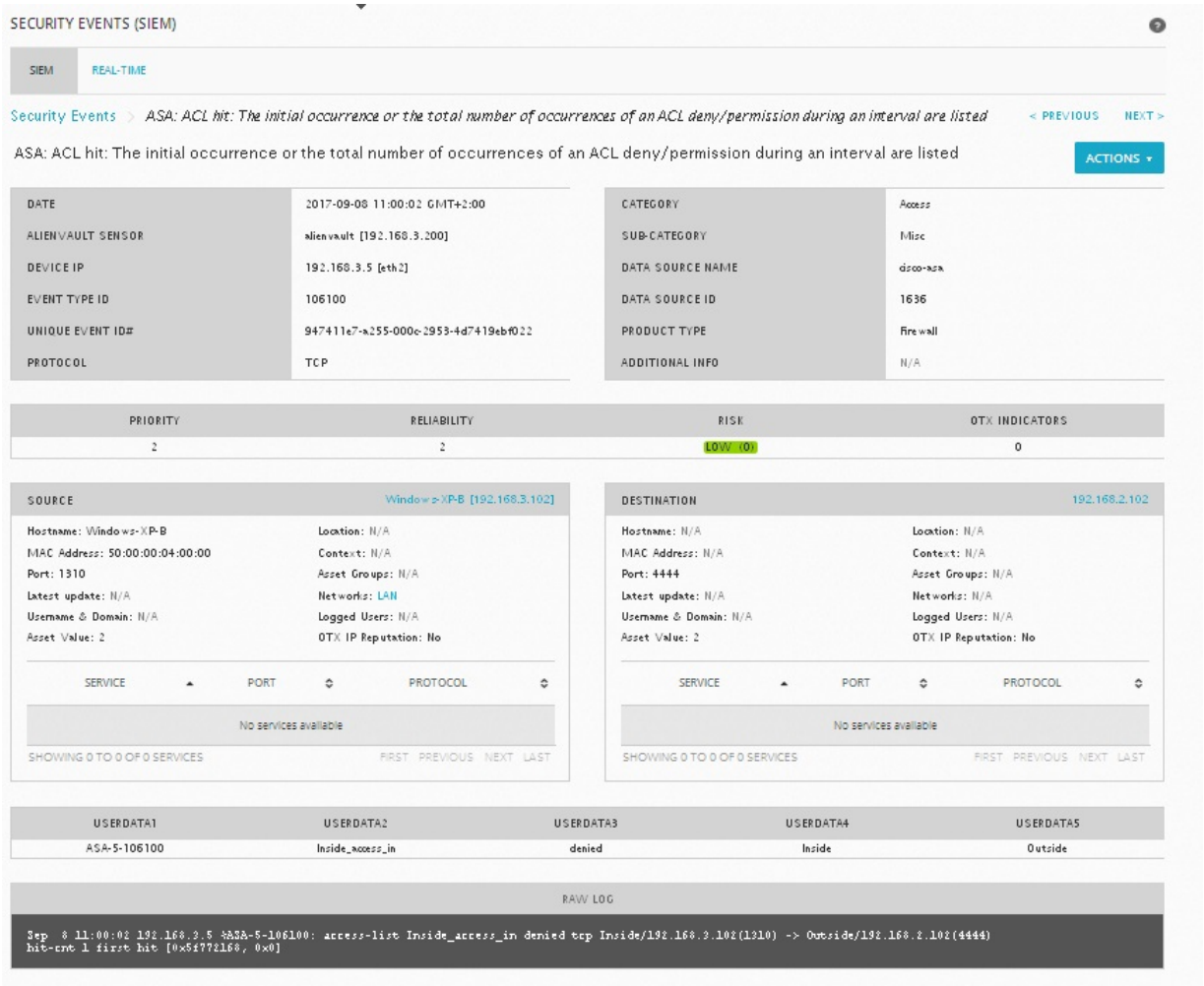
Fig. 7.14: Generación del Troyano

Y como OSSIM detecta la conexión utilizando el plugin de Cisco ASA:

sudo: Session opened	2017-09-08 11:00:53	alienvault	N/A	0.0.0.0	alienvault	2->2	LOW (0)	
ASA: ACL hit: The initial occurrence or the total number of occurrences of an ACL deny/permission during an interval are listed	2017-09-08 11:00:02	alienvault	N/A	Windows-XP-B:1310	192.168.2.102:4444	2->2	LOW (0)	
ASA: ACL hit: The initial occurrence or the total number of occurrences of an ACL deny/permission during an interval are listed	2017-09-08 10:59:40	alienvault	N/A	Windows-XP-B:1309	192.168.2.102:4444	2->2	LOW (0)	
ASA: A user made a configuration change	2017-09-08 10:59:14	alienvault	N/A	0.0.0.0	0.0.0.0	2->2	LOW (0)	

Fig. 7.15: Detección de la conexión por OSSIM

El evento generado en detalle es el siguiente:



SECURITY EVENTS (SIEM)

SIEM | REAL-TIME

Security Events > *ASA: ACL hit: The initial occurrence or the total number of occurrences of an ACL deny/permission during an interval are listed* < PREVIOUS NEXT >

ASA: ACL hit: The initial occurrence or the total number of occurrences of an ACL deny/permission during an interval are listed — ACTIONS

DATE	2017-09-08 11:00:02 GMT+2:00	CATEGORY	Access
ALIENVAULT SENSOR	alienvault [192.168.3.200]	SUB-CATEGORY	Misc
DEVICE IP	192.168.3.5 [eth2]	DATA SOURCE NAME	cisco-asa
EVENT TYPE ID	106100	DATA SOURCE ID	1636
UNIQUE EVENT ID#	947411e7-a255-000c-2953-4d7419ebf022	PRODUCT TYPE	Firewall
PROTOCOL	TCP	ADDITIONAL INFO	N/A

PRIORITY	RELIABILITY	RISK	OTX INDICATORS
2	2	LOW (0)	0

SOURCE — Windows-XP-B [192.168.3.102]

Hostname: Windows-XP-B | Location: N/A
MAC Address: 50:00:00:04:00:00 | Context: N/A
Port: 1310 | Asset Groups: N/A
Latest update: N/A | Networks: LAN
Username & Domain: N/A | Logged Users: N/A
Asset Value: 2 | OTX IP Reputation: No

SERVICE	PORT	PROTOCOL
No services available		

SHOWING 0 TO 0 OF 0 SERVICES — FIRST PREVIOUS NEXT LAST

DESTINATION — 192.168.2.102

Hostname: N/A | Location: N/A
MAC Address: N/A | Context: N/A
Port: 4444 | Asset Groups: N/A
Latest update: N/A | Networks: N/A
Username & Domain: N/A | Logged Users: N/A
Asset Value: 2 | OTX IP Reputation: No

SERVICE	PORT	PROTOCOL
No services available		

SHOWING 0 TO 0 OF 0 SERVICES — FIRST PREVIOUS NEXT LAST

USERDATA1	USERDATA2	USERDATA3	USERDATA4	USERDATA5
ASA-5-106100	Inside_access_in	denied	Inside	Outside

RAW LOG

```
Sep  8 11:00:02 192.168.3.5 %ASA-5-106100: access-list Inside_access_in denied tcp Inside/192.168.3.102(1310) -> Outside/192.168.2.102(4444)
hit-cnt 1 first hit [0x5f772168, 0x0]
```

Fig. 7.16: Detalles del evento generado tras la conexión

Para crear la política nos dirigimos a:
Configuration → **Thread Intelligence** → se hace click en **New**

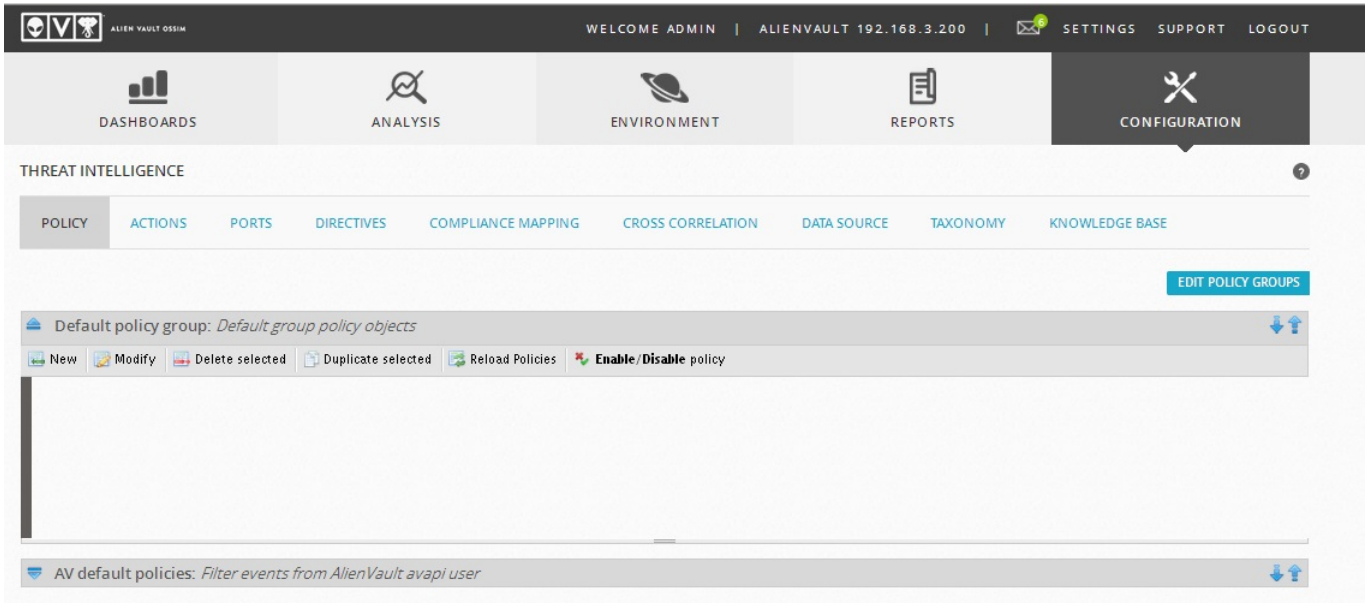


Fig. 7.17: Creando una política

A continuación cubrimos las condiciones de la política. Como primer paso el origen de los eventos a los que va a afectar la política, en nuestro caso el origen puede ser tanto la DMZ como la LAN.

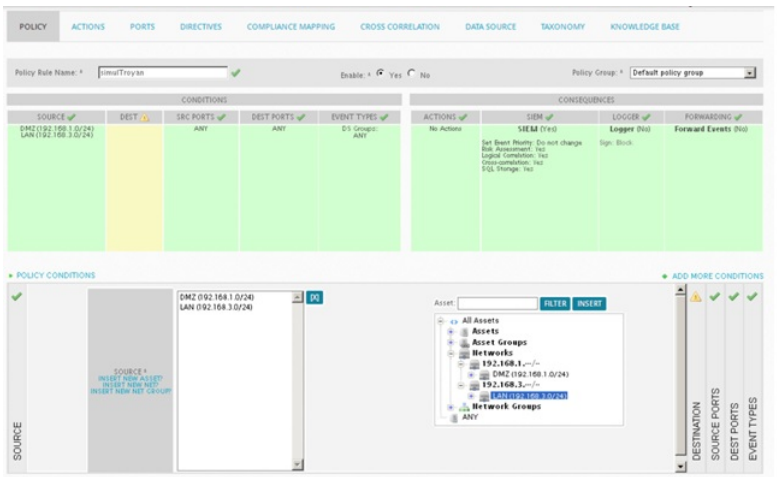


Fig. 7.18: Creando una política

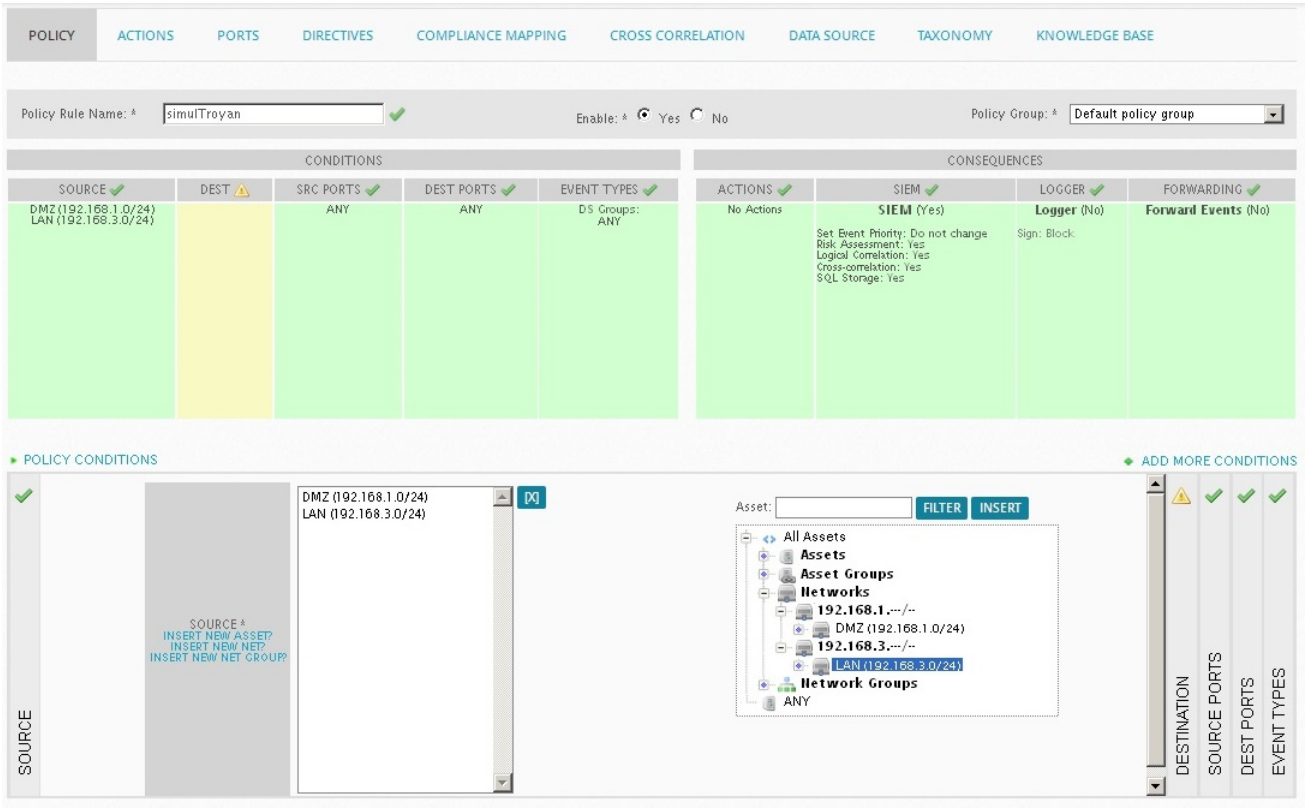


Fig. 7.19: Configuración condición dirección origen en la política.

El destino podrá ser cualquier dirección IP, ya que dependerá de la que use el atacante.

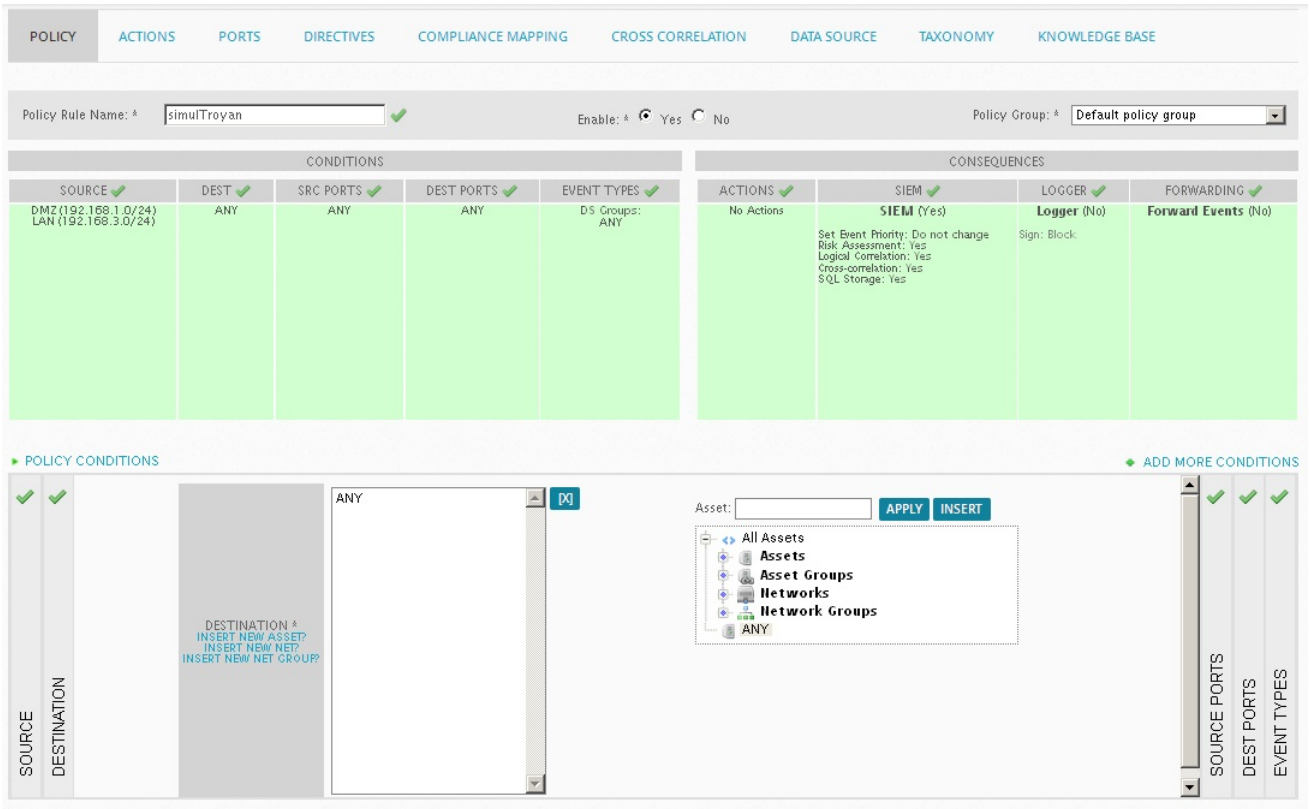


Fig. 7.20: Configuración condición dirección destino en la política.

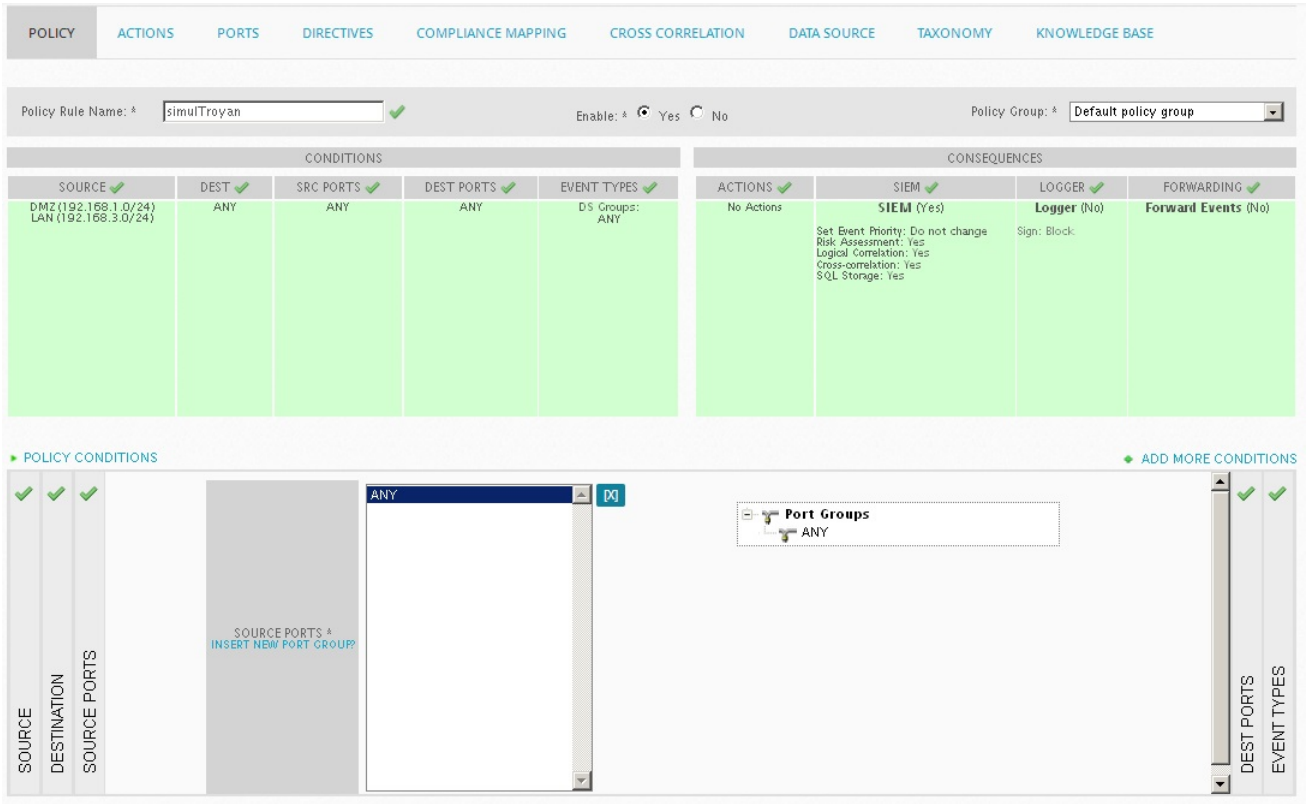


Fig. 7.21: Configuración condición puerto origen en la política.

El puerto destino será el TCP 4444 ya que es el dato del que disponemos para identificar este troyano, para ello debemos añadirlo a OSSIM para que posteriormente este se encuentre disponible para utilizarlo en la configuración de la política. Para añadir este puerto a la lista, y a continuación poder seleccionarlo debemos hacer click en **Insert New Port Group** para después cubrir los campos como se muestra a continuación.

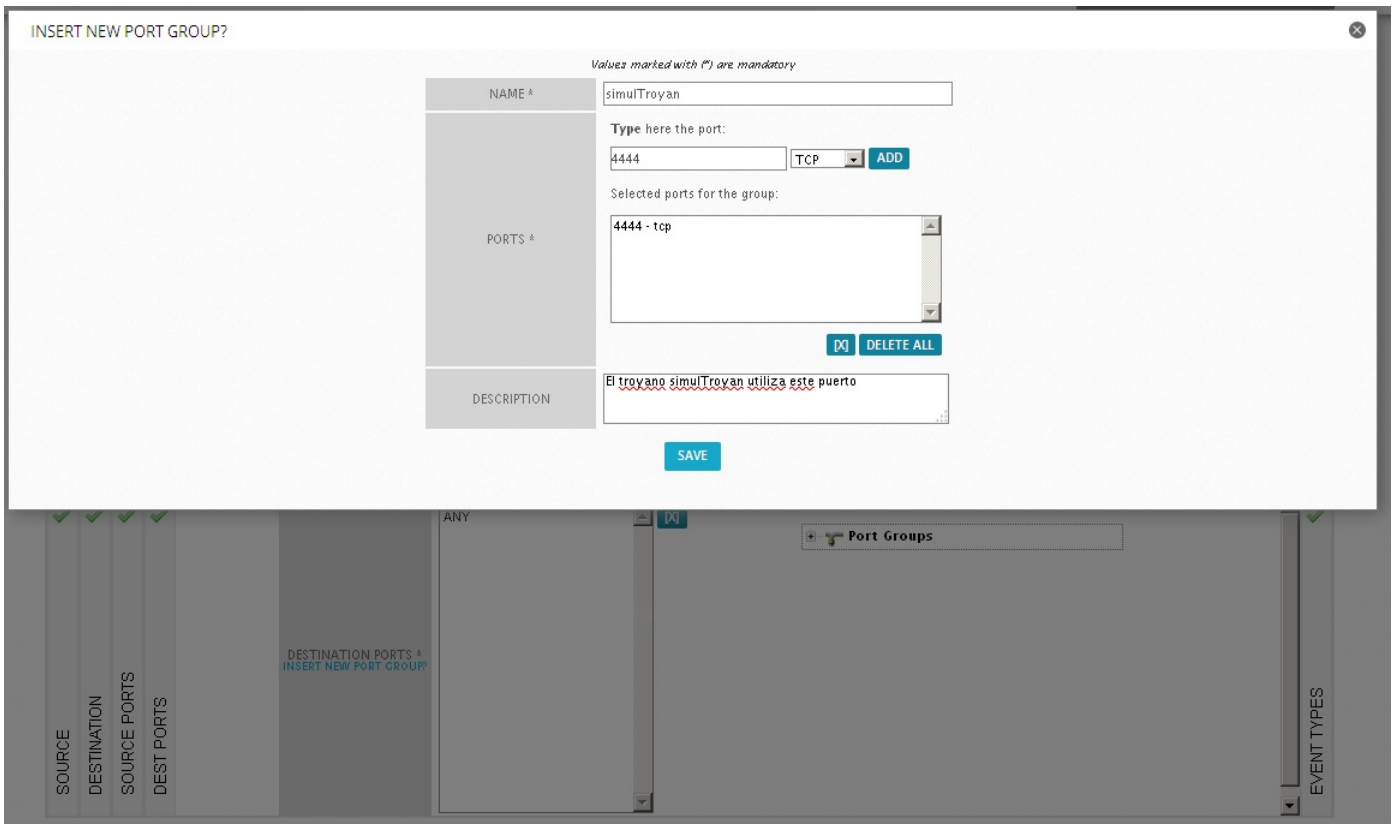


Fig. 7.22: Inclusión del puerto utilizado por "SimulTroyan" en OSSIM.

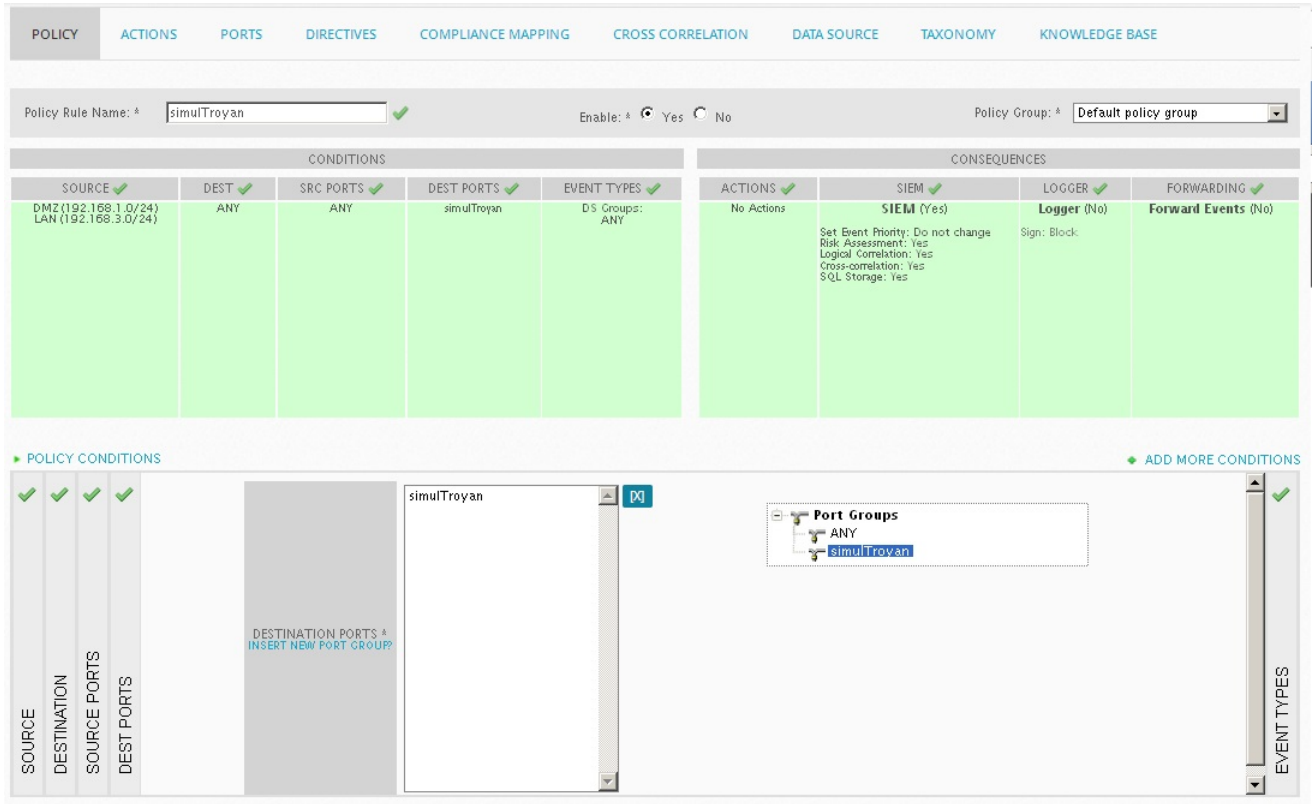


Fig. 7.23: Configuración condición puerto destino en la política.

Por último indicamos que a cualquier evento que encaje en estas condiciones se le aplique la política:

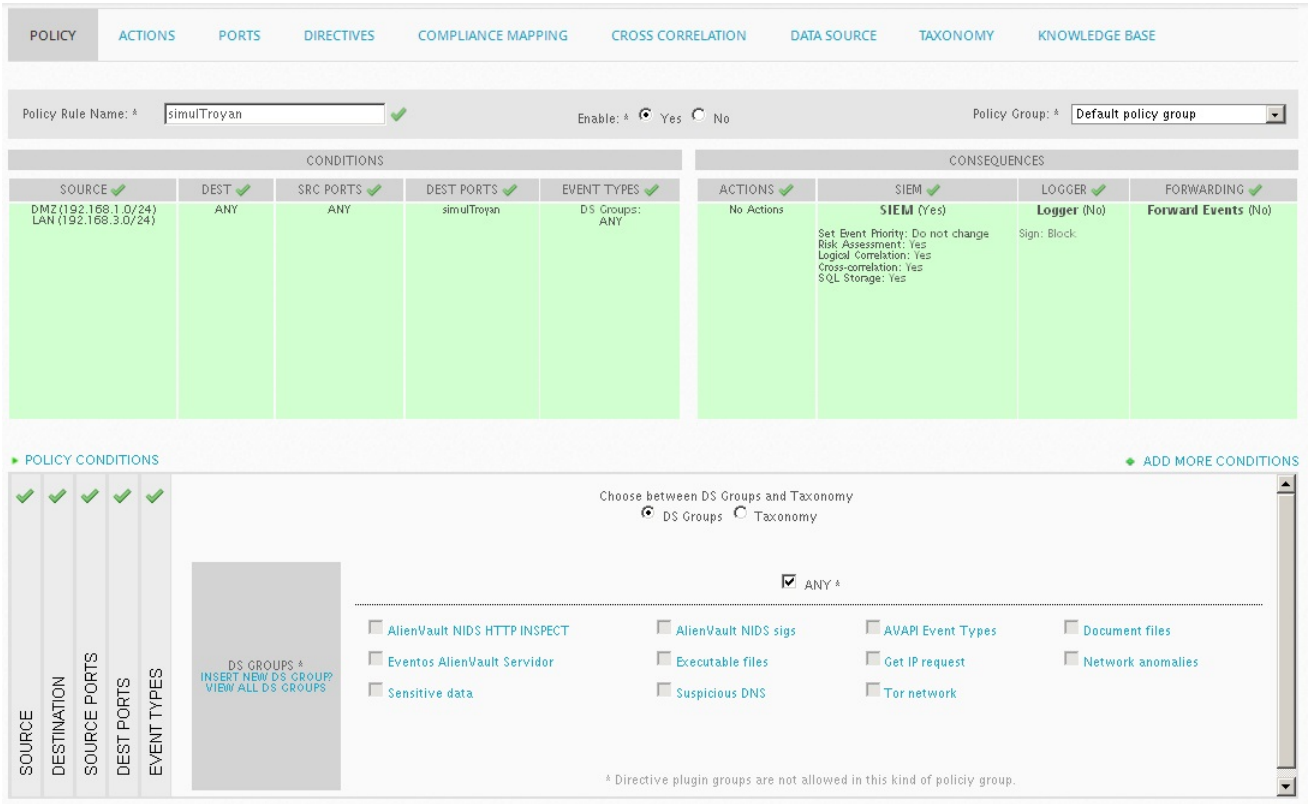


Fig. 7.24: Configuración condición tipo de eventos a los que aplicar la política.

En el siguiente paso se configuran las consecuencias de aplicar dicha política, como no se ha definido previamente la acción a realizar como consecuencia se hará click en **insert new action**.

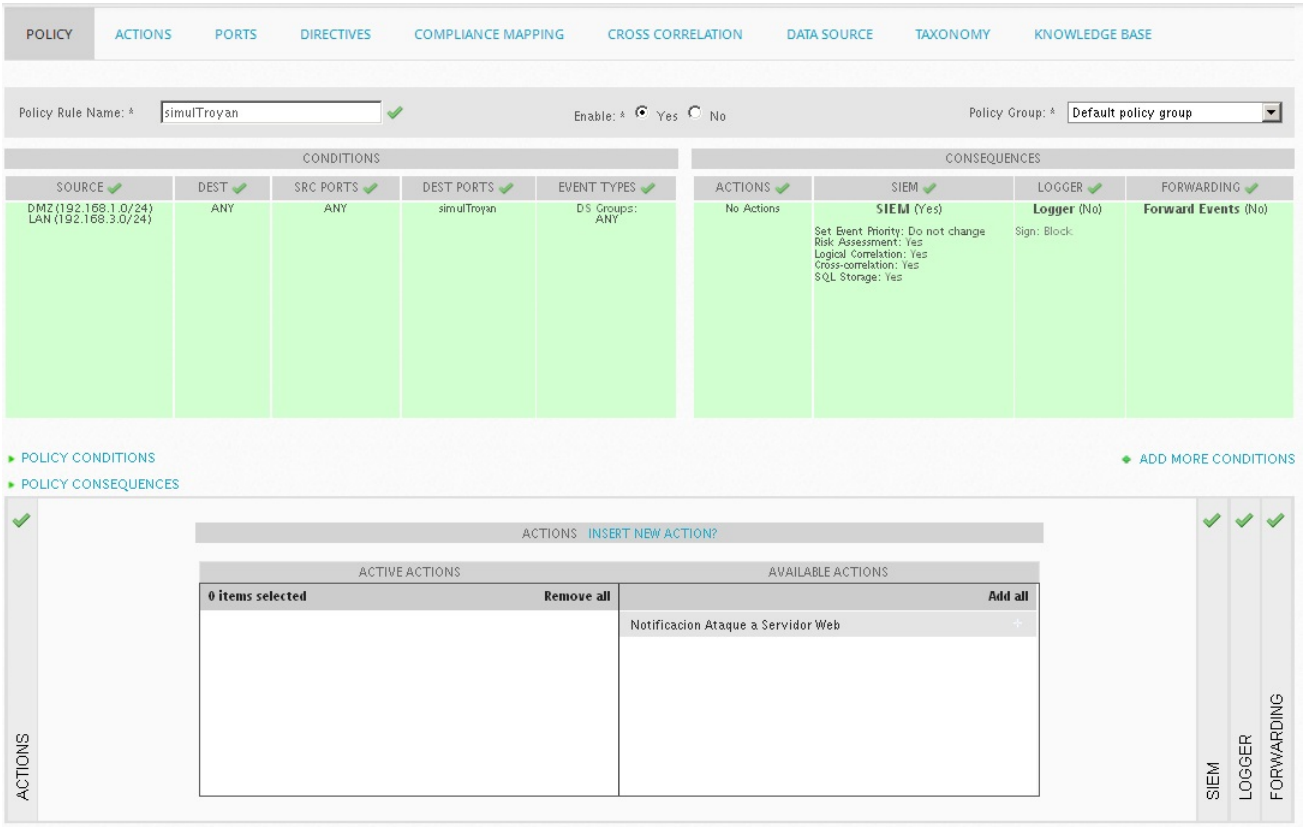


Fig. 7.25: Configuración acción consecuencia al aplicar la política.

Se crea una acción que abrirá un ticket, informándonos en la descripción del equipo que se ha visto afectado y la dirección del atacante.

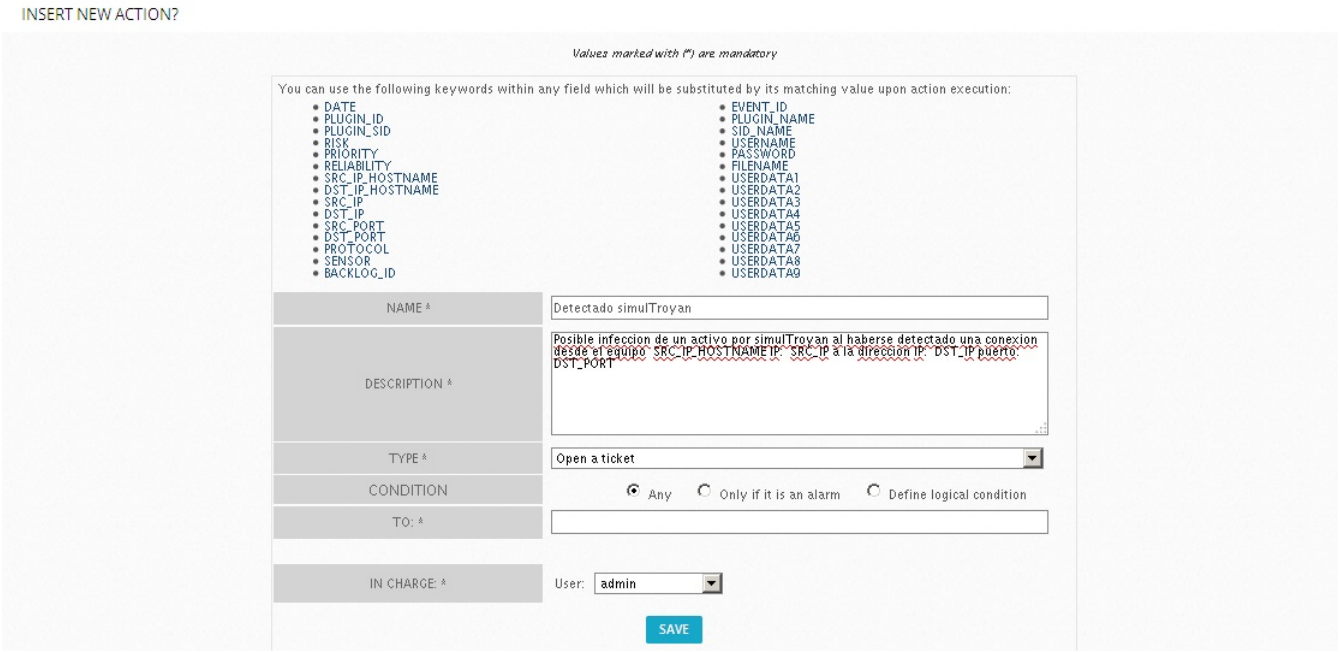


Fig. 7.26: Creando una acción para aplicar en la política.

Se selecciona la acción que acabamos de crear previamente.

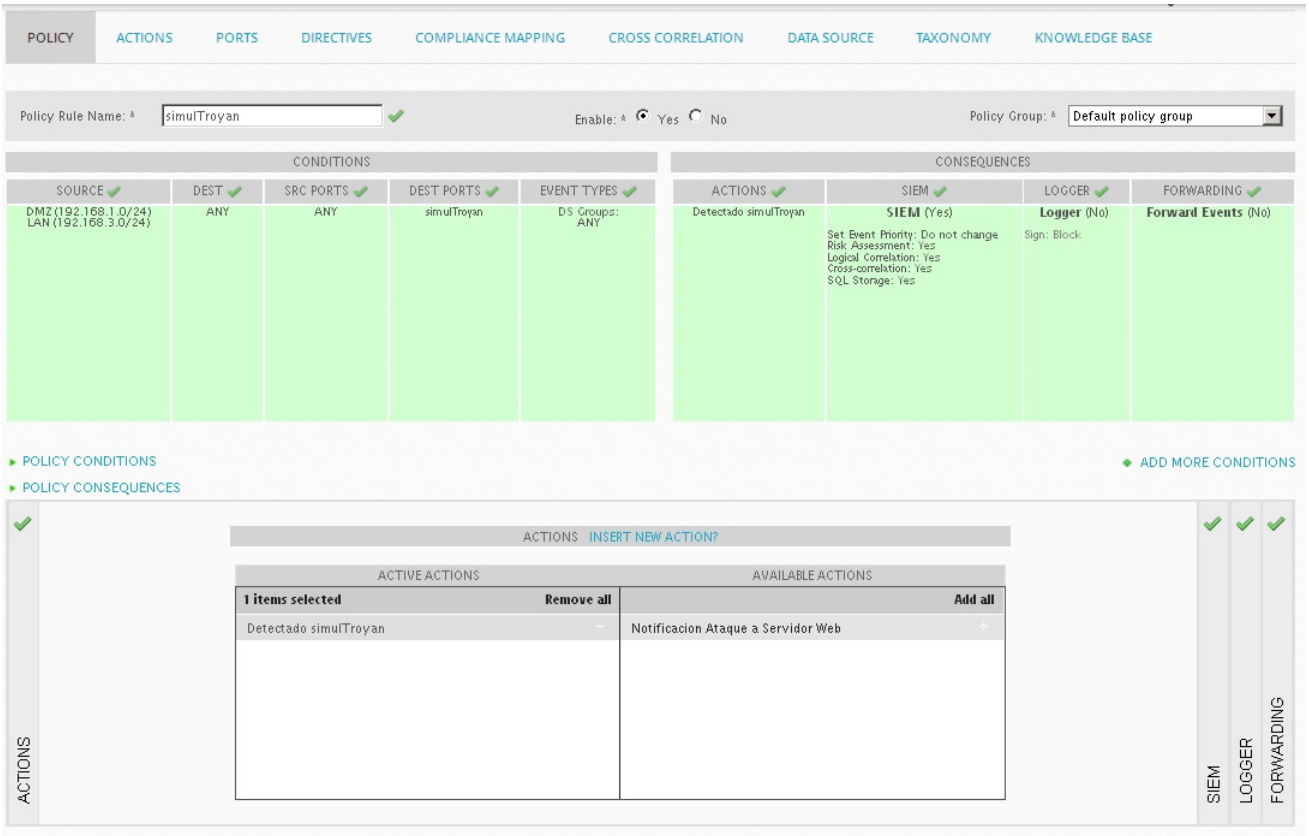


Fig. 7.27: Configuración acción consecuencia (creada previamente) al aplicar la política.

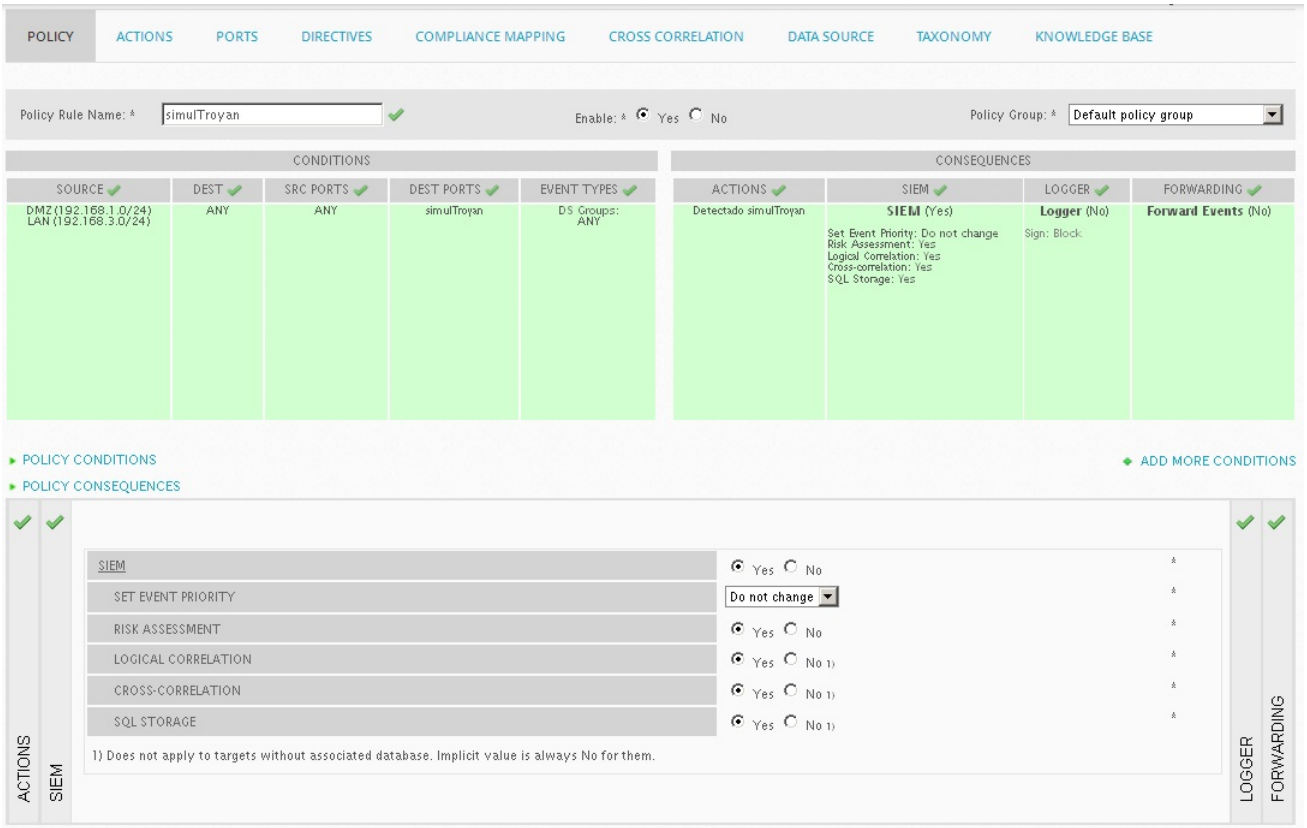


Fig. 7.28: Configuración SIEM al aplicar la política.

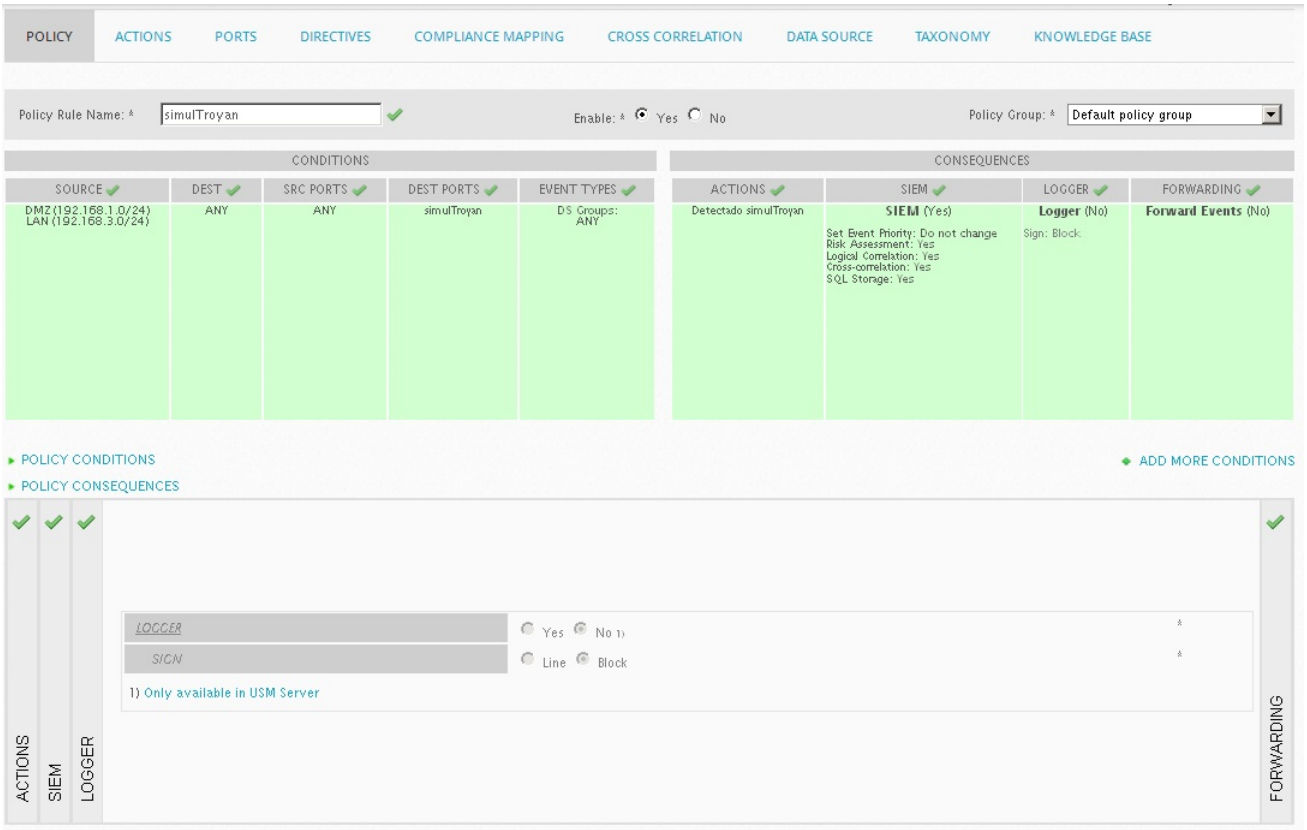


Fig. 7.29: Configuración LOGGER al aplicar la política.

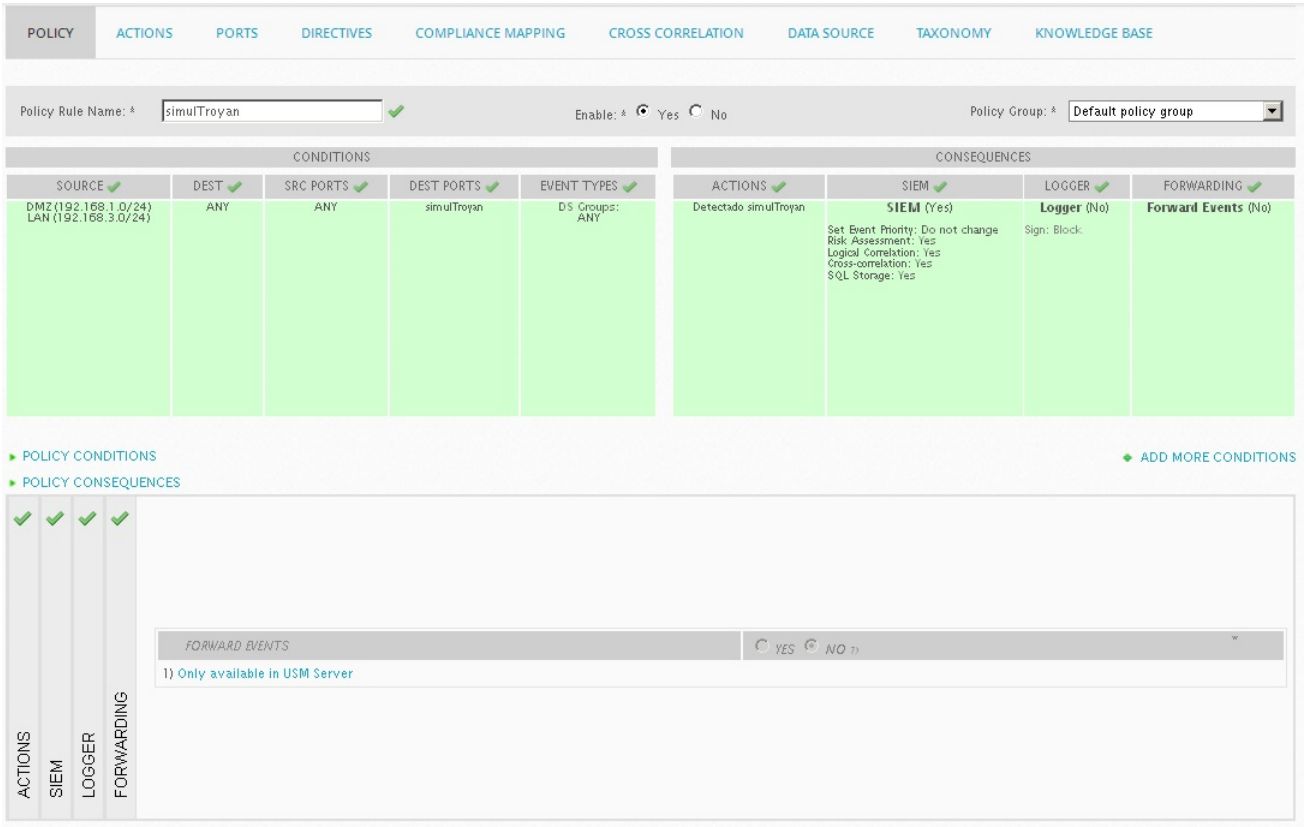


Fig. 7.30: Configuración FORWARDING al aplicar la política.

Por último se guarda y se aplica la política.

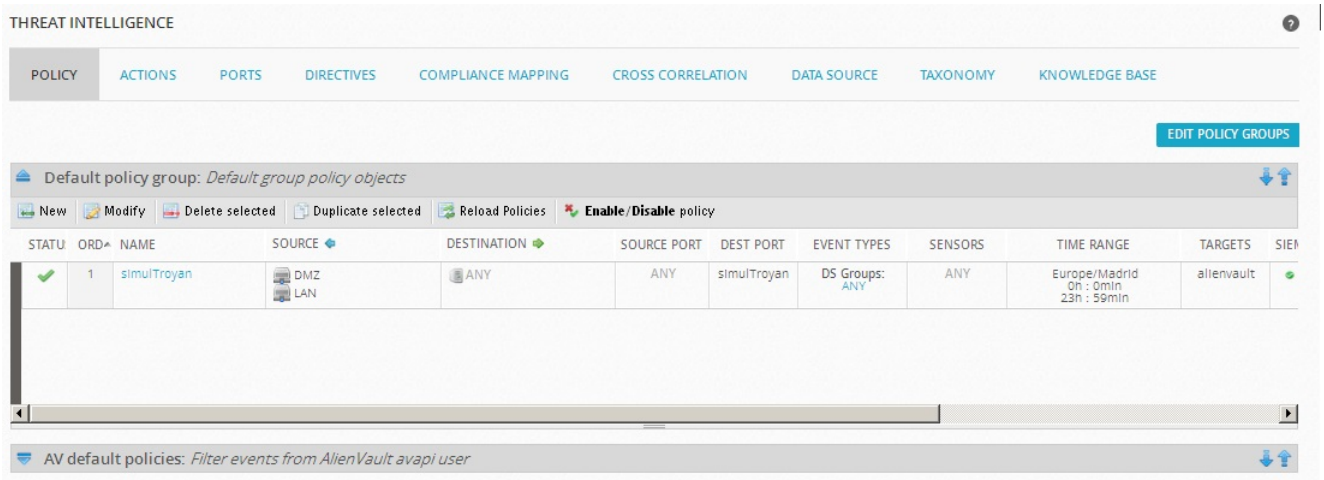


Fig. 7.31: Guardar y aplicar la política.

Una vez que se genera un evento que cumple las condiciones definidas en la política, se ejecutan las consecuencias definidas.

Fig. 7.32: Evento generado por "SimulTroyan".

El ticket puede verse en **Analysis** → **Tickets**

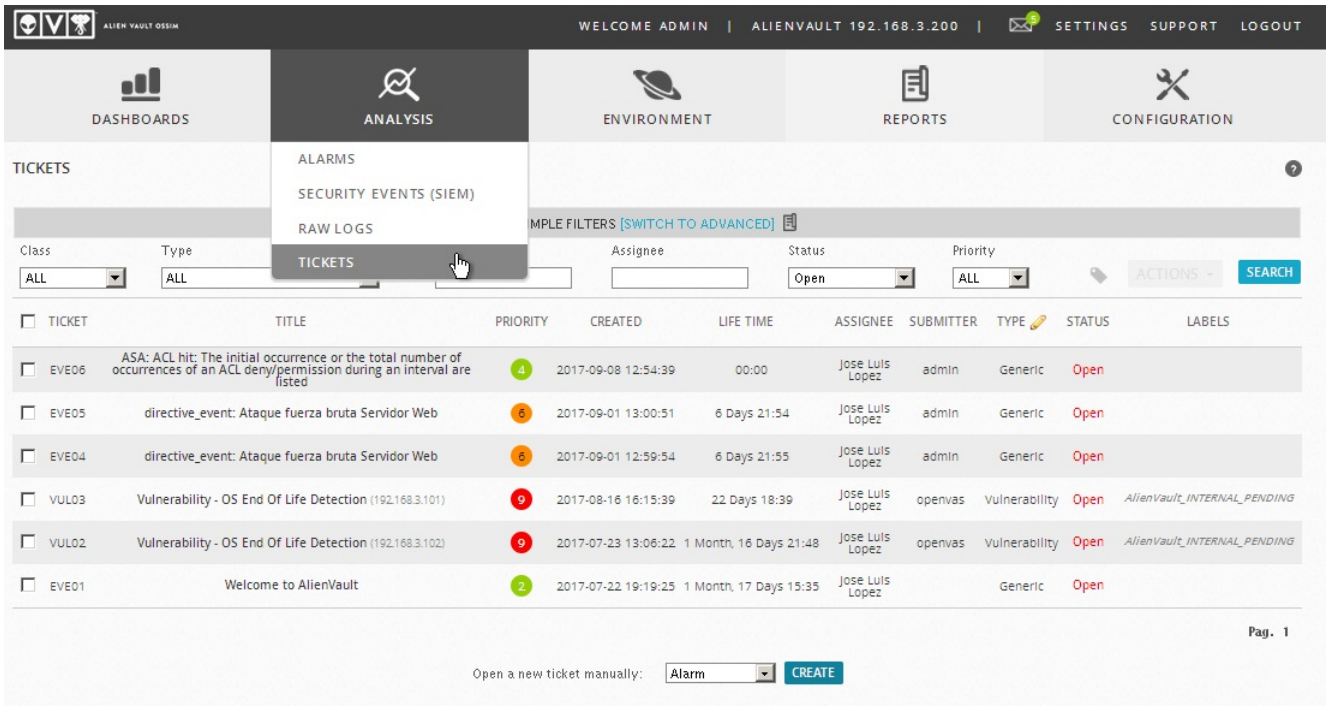


Fig. 7.33: Ticket generado por OSSIM.

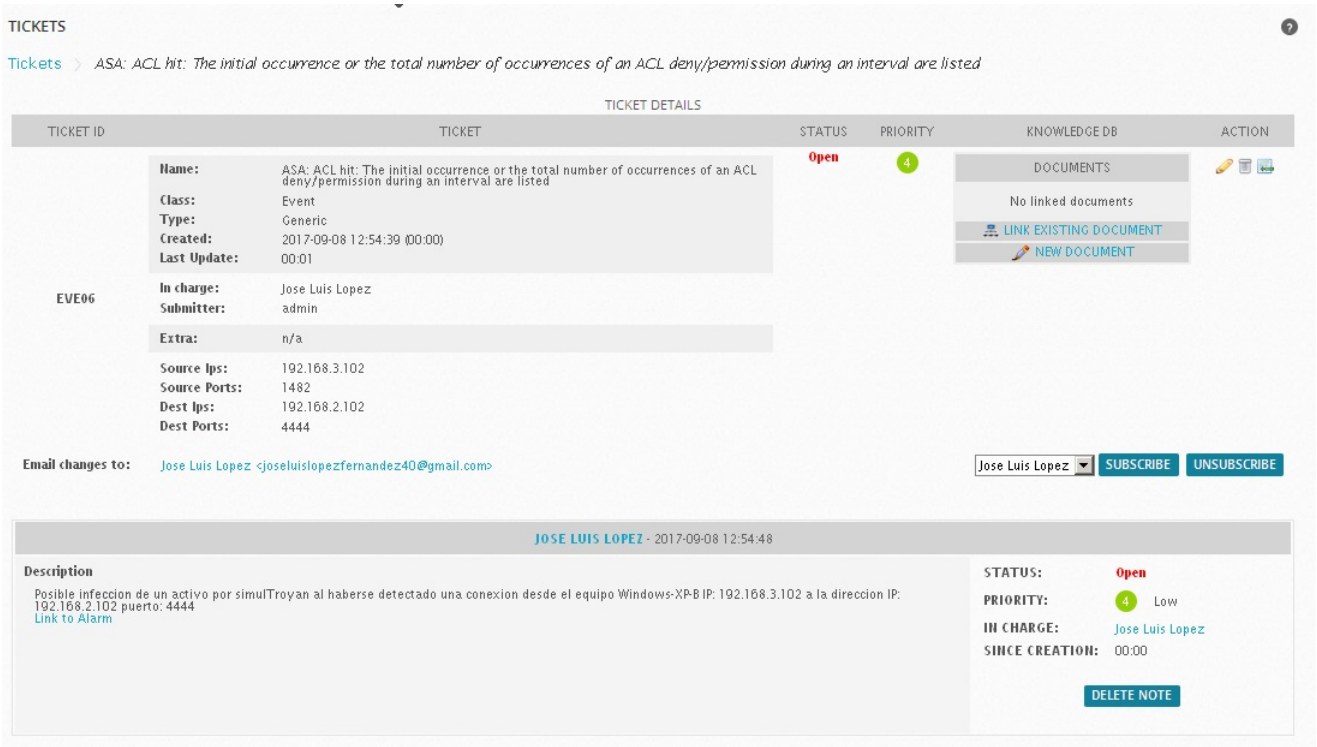


Fig. 7.34: Detalle del ticket generado por OSSIM.

8. CONCLUSIONES

- Se ha conseguido el aprendizaje/adiestramiento de una herramienta que ofrece Gestión de Vulnerabilidades además de una excelente protección de redes. OSSIM es un software modelo de los centros de seguridad que deben implantarse en las pequeña y medianas empresas.
- OSSIM es un sistema muy versátil adaptando con la adecuada programación a dispositivos propietarios (en la parte experimental se ha usado un Cisco ASA a modo de ejemplo).
- OSSIM permite un despliegue en distintos segmentos de red así como la centralización de la información en un punto único para su estudio.
- Una gran ventaja de OSSIM es que utiliza las mejores versiones de código libre existente para la detección de vulnerabilidades y monitorización de redes.
- Puede considerarse que OSSIM es una plataforma de administración de la Seguridad adecuada para realizar las funciones de seguridad esenciales que se necesitan para proteger los activos informáticos de una empresa, ya que además con su información ayuda al administrador de seguridad en la toma de decisiones.
- Con la herramienta OSSIM se tiene la posibilidad de llevar control de los requisitos de ISO 27001 y PCI DSS lo que ayuda en el cumplimiento de las normativas.
- La opción OTX (Open Threat Exchange) permite realizar una defensa colaborativa contra amenazas de seguridad informática. Esto permite ampliar los horizontes de actuación de OSSIM y mitigar su punto más débil que es el tratamiento de amenazas desconocidas.
- A pesar de ser OSSIM una herramienta open source la curva de aprendizaje de OSSIM es costosa para las empresas.
- Es necesario diseñar y programar algunas funcionalidades que no se ofrecen en su versión gratuita.
- No realiza almacenamiento de los raw logs lo que sería una opción deseable para auditorías.
- Por razones de tiempo y extensión no se han realizado experimentos con redes Wifi aunque el mecanismo sería análogo al de las redes cableadas.

9. BIBLIOGRAFÍA

[1] ISO
https://www.iso.org/home.html

[2] INCIBE
https://www.incibe.es

[3] Álvaro Gómez Vieites.
Enciclopedia de la Seguridad Informática.2ªEdición.Rama.

[4] GANTTPROJECT
http://www.ganttproject.biz/

[5] NVD(National Vulnerability Database)
https://nvd.nist.gov/

[6] OSSIM
https://www.alienvault.com/products/ossim?utm_source=google&utm_medium=c
pc&utm_term=kwd-2148725609&utm_campaign=BRAND-EMEA-GGLSE&
gclid=Cj0KEQiAuonGBRCaotXoycysvIMBEiQAcxV0nNLD6ollkqKnkzSc8sDnUbXnrmnOVSHv3V6aFh722Ea

[7] OSSIM
https://sourceforge.net/projects/os-sim/

[8] Descarga de OSSIM]
https://www.alienvault.com/products/ossim/download

[9] Proyecto OSSIM
https://github.com/alienfault/ossim

[10] Descarga OSSIM
https://sourceforge.net/projects/os-sim/

[11] Oracle VirtualBox
https://www.virtualbox.org/

[12] EVE-NG
http://www.eve-ng.net/

[13] Creación de un pluging para AlienVault USM
http://a3sec.com/creacion-rapida-de-un-plugin-para-alienvault-usm/

[14] Descripción PCI-DSS
https://es.wikipedia.org/wiki/PCI_DSS

[15] PCI-DSS versión 3
https://es.pcisecuritystandards.org/_onelink_/pcisecurity/en2es/minisite/en/docs/PCI_DSS_v3.pdf

[16] PCI Security
https://www.pcisecuritystandards.org/pci_security/

[17] Cuadrante Mágico para Gestión de Seguridad de la Información y Eventos de la Consultora Gartner 2016
https://www.gartner.com/doc/3406817/magic-quadrant-security-informationevent

[18] OSSEC
https://en.wikipedia.org/wiki/OSSEC

[19] OSSEC
https://ossec.github.io/docs/

[20] IDS
https://www.alienvault.com/solutions/intrusion-detection-system

[21] OSSIM
http://resources.infosecinstitute.com/alienvault-ossim-review-open-source-siem/

[22] Cuadrado mágico SIEM
http://www.splunk.com/goto/SIEM_MQ

[23] The CIS Critical Security Controls for Effective Cyber Defense
https://www.sans.org/critical-security-controls

[24] Automatic Testing of Program Security Vulnerabilities
https://www.researchgate.net/publication/224592929_Automatic_Testing_of_Program_Security_Vulnerabilities

[25] Guide to Computer Security LogManagement
http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf

[26] Guide to Computer Security Log Management
http://dx.doi.org/10.6028/NIST.SP.800-92

[27] SANS
https://www.sans.org/reading-room/whitepapers/analyst/security-spendingtrends-36697

[28] Costes del ciber Crimen
http://www.cnmeonline.com/myresources/hpe/docs/HPE_SIEM_Analyst_Report__ 2015_Cost_of_C

[29] Incident Response
https://www.lancope.com/sites/default/files/Lancope-Ponemon-Report-Cyber-Security-Incident-Response.pdf

[30] AlienVault Unified Security Management (USM) Asset Management Guide
https://www.alienvault.com/doc-repo/usm/asset-management/AlienVault-USM-5.1-5.2-Asset-Management-Guide.docx.pdf

[31] OSSIM
https://www.alienvault.com/documentation/

[32] Documento Gartner 2016
https://www.idrgrp.com/wp-content/uploads/2016/11/2016-Gartner-MQ-for-Security-Information-and-Event-Management.pdf

[33] Logs
Jens Kühnel. "Centralized and structured log file analysis with open source and free software tools". Bachelor Thesis. Fachhochschule Frankfurt am Main. University of Applied Sciences. 2013

[34] Vulnerabilidades
https://www.certsi.es/alerta-temprana/vulnerabilidades

[35] Vulnerabilidades
https://www.mitre.org/

[36] Thread Based Defense
https://www.mitre.org/capabilities/cybersecurity/threat-based-defense

[37] CVE
https://www.certsi.es/alerta-temprana/vulnerabilidades/cve-2014-4151

[38] CVE
https://www.certsi.es/alerta-temprana/vulnerabilidades/cve-2017-6972

[39] vulnerabilidades OSSIM
https://www.alienvault.com/forums/discussion/8698

[40] Vulnerabilidad OSSIM
https://www.exploit-db.com/exploits/42314/

[41] National Vulnerability Database
https://nvd.nist.gov

[42] CVSS Common Vulnerability Scoring System Version 3.0 Calculator
https://www.first.org/cvss/calculator/3.0

[43] CVRF Common Vulnerability Reporting Framework
https://www.icasi.org/cvrf/

10. ANEXOS

10.1 ANEXO 1: Descripción del ataque realizado con kali

1. Se configura la red en kali:

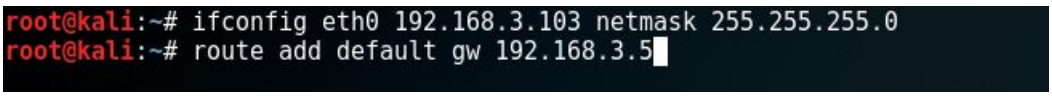


Fig. 10.1: Configuración de red de kali linux.

2. Se inicializa la la base de datos de metasploit:

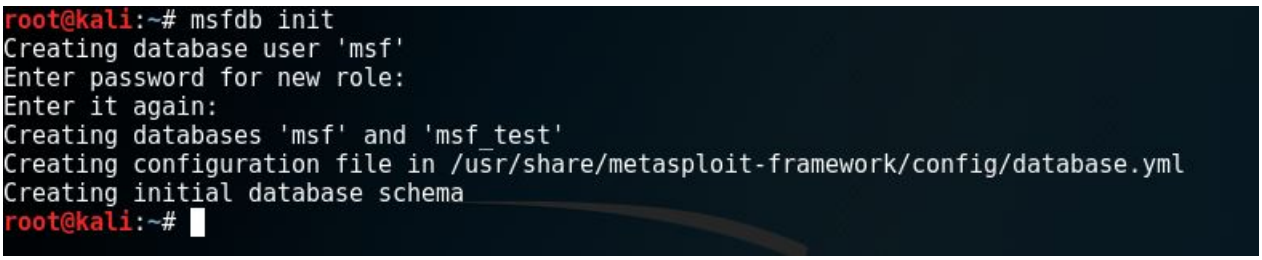


Fig. 10.2: Inicialización de la base de datos de metasploit.

3. Se ejecuta Armitage con conexión a la base de datos de metasploit:

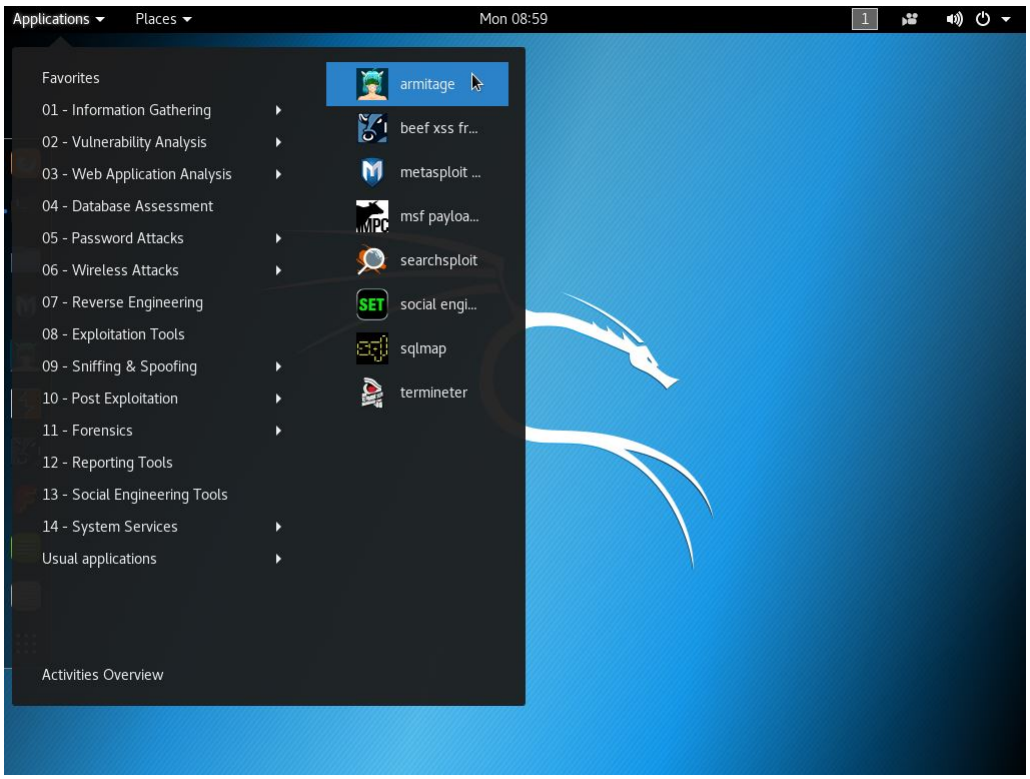


Fig. 10.3: Ejecución de Armitage 1 de 4.

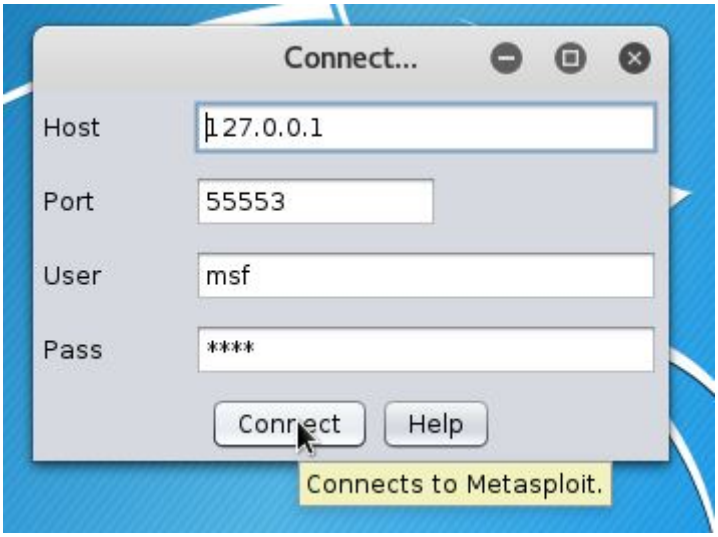


Fig. 10.4: Ejecución de Armitage 2 de 4.

Fig. 10.5: Ejecución de Armitage 3 de 4.

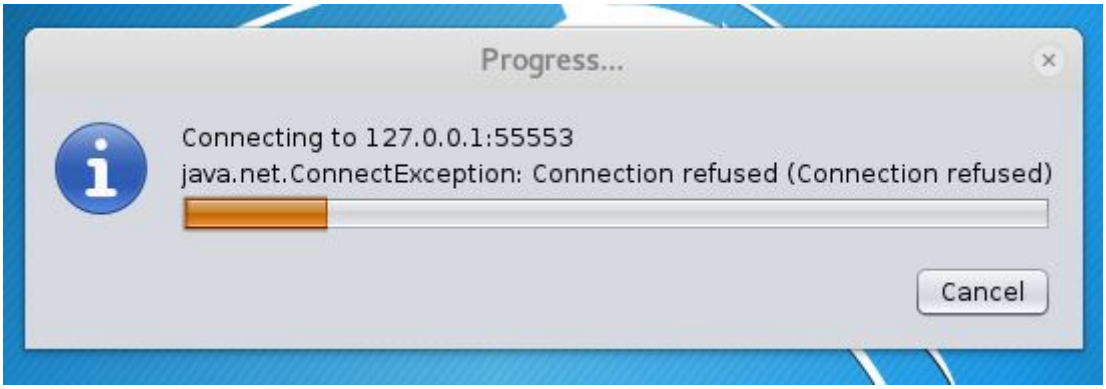


Fig. 10.6: Ejecución de Armitage 4 de 4.

4. Se ejecuta nmap para el descubrimiento de equipos y servicios en la red guardando los resultados en la base de datos:

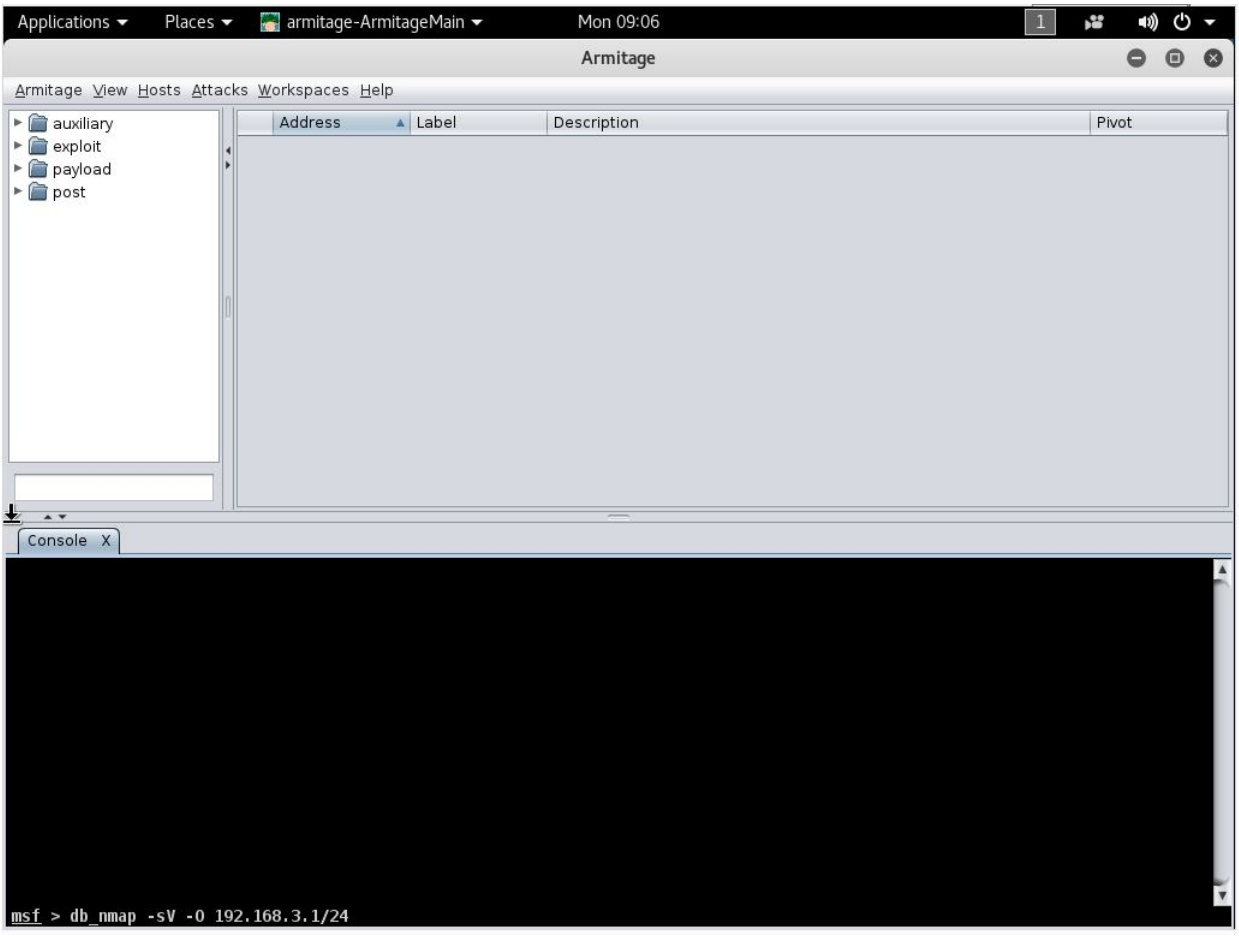


Fig. 10.7: Ejecución de escaneo con Nmap desde consola de Armitage.

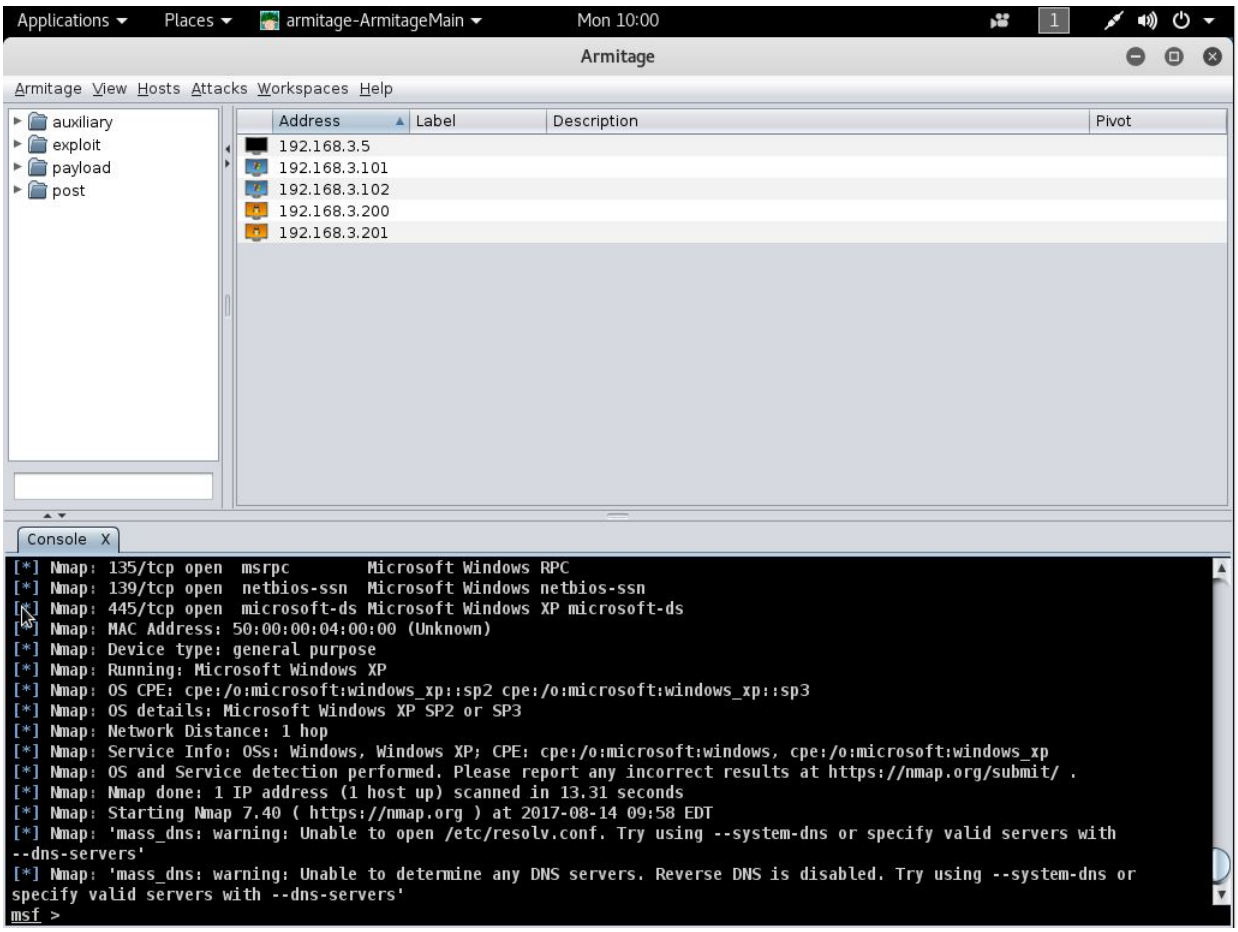


Fig. 10.8: Resultado de ejecución de escaneo con Nmap desde consola de Armitage.

5. Encontrar los ataques en base a la información recogida por nmap:

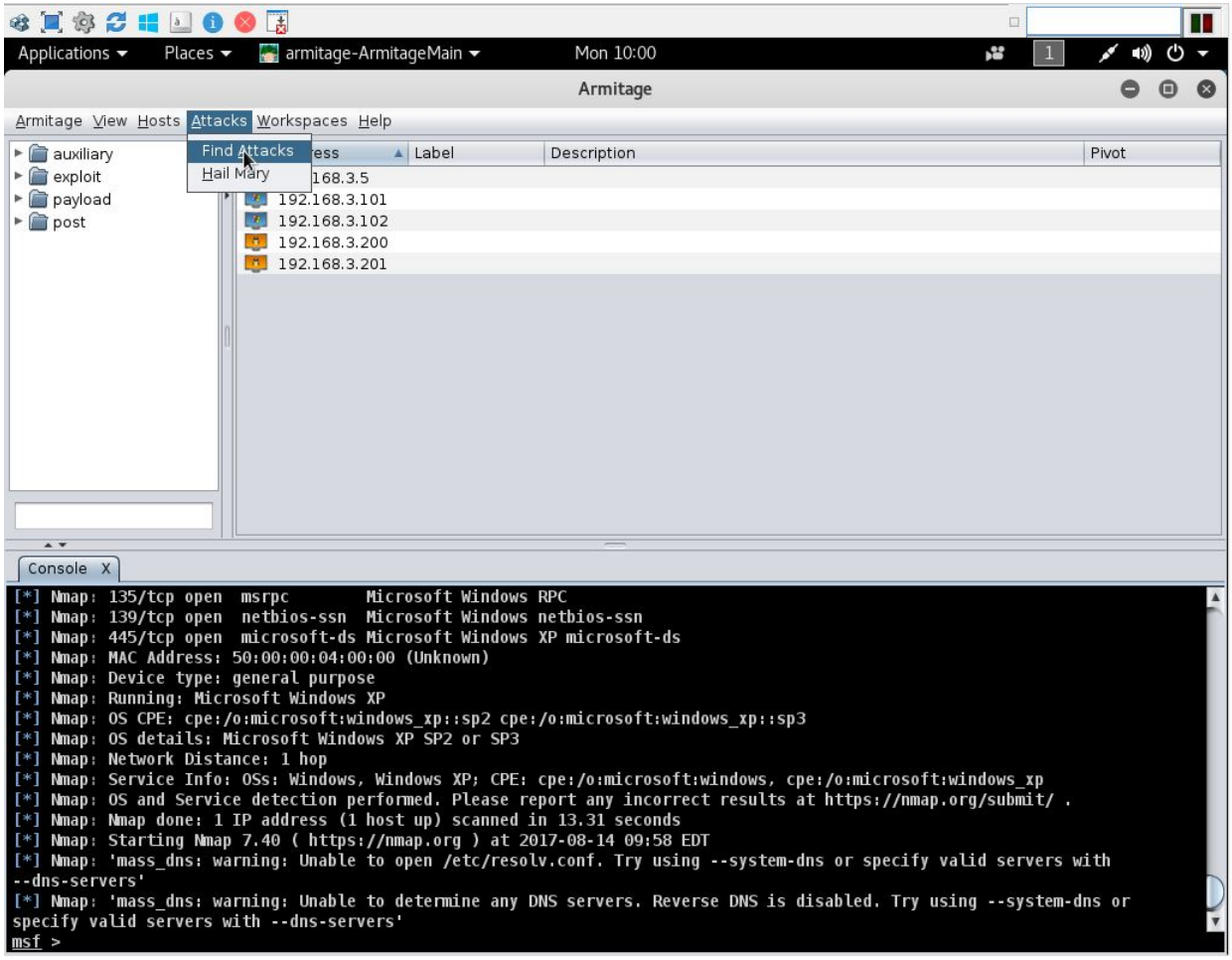


Fig. 10.9: Ejecución de búsqueda de ataques (exploits) con Armitage 1 de 2.

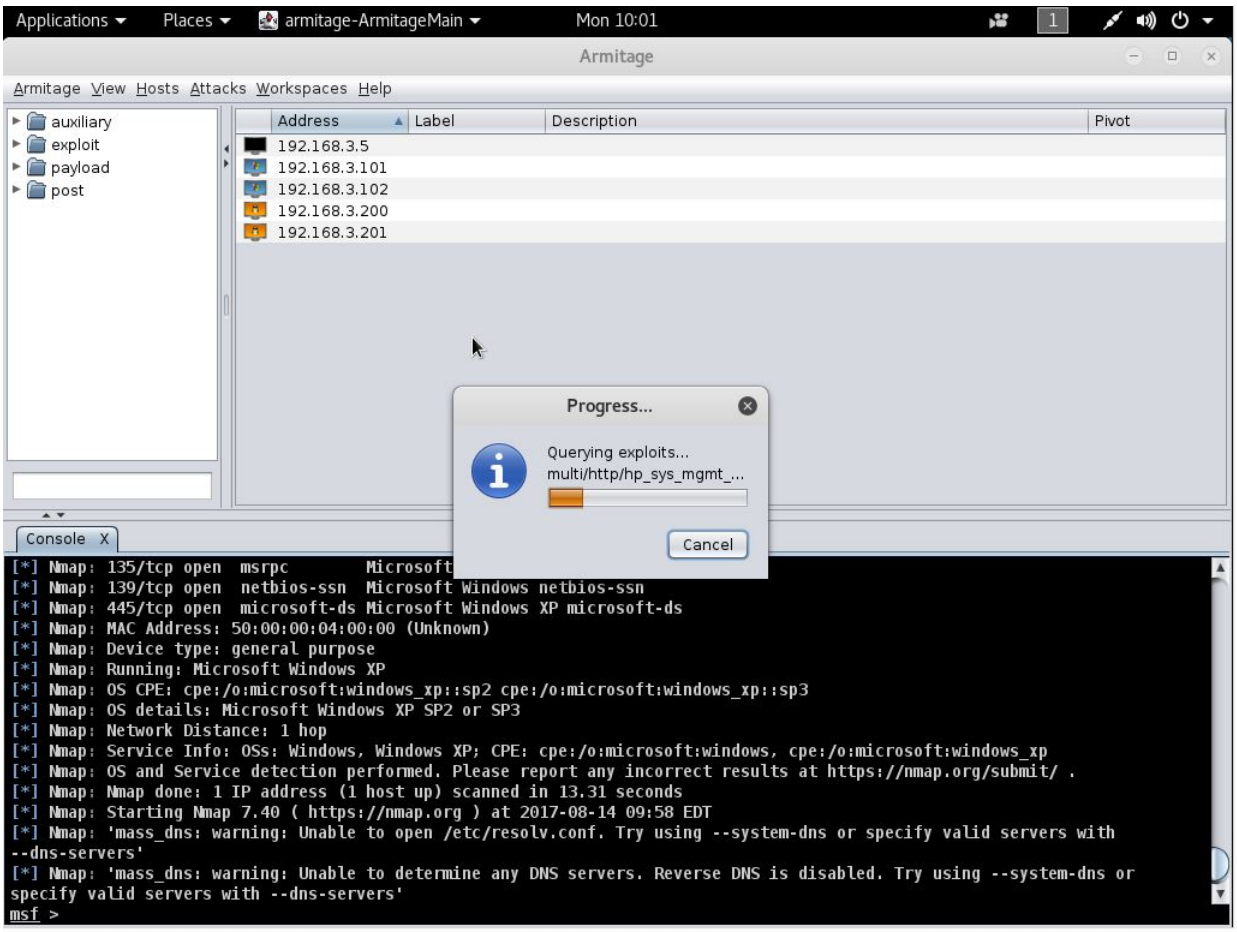


Fig. 10.10: Ejecución de búsqueda de ataques (exploits) con Armitage 2 de 2.

6. Se lanzan todos aquellos exploits susceptibles de tener éxito en base a la información previamente obtenida:

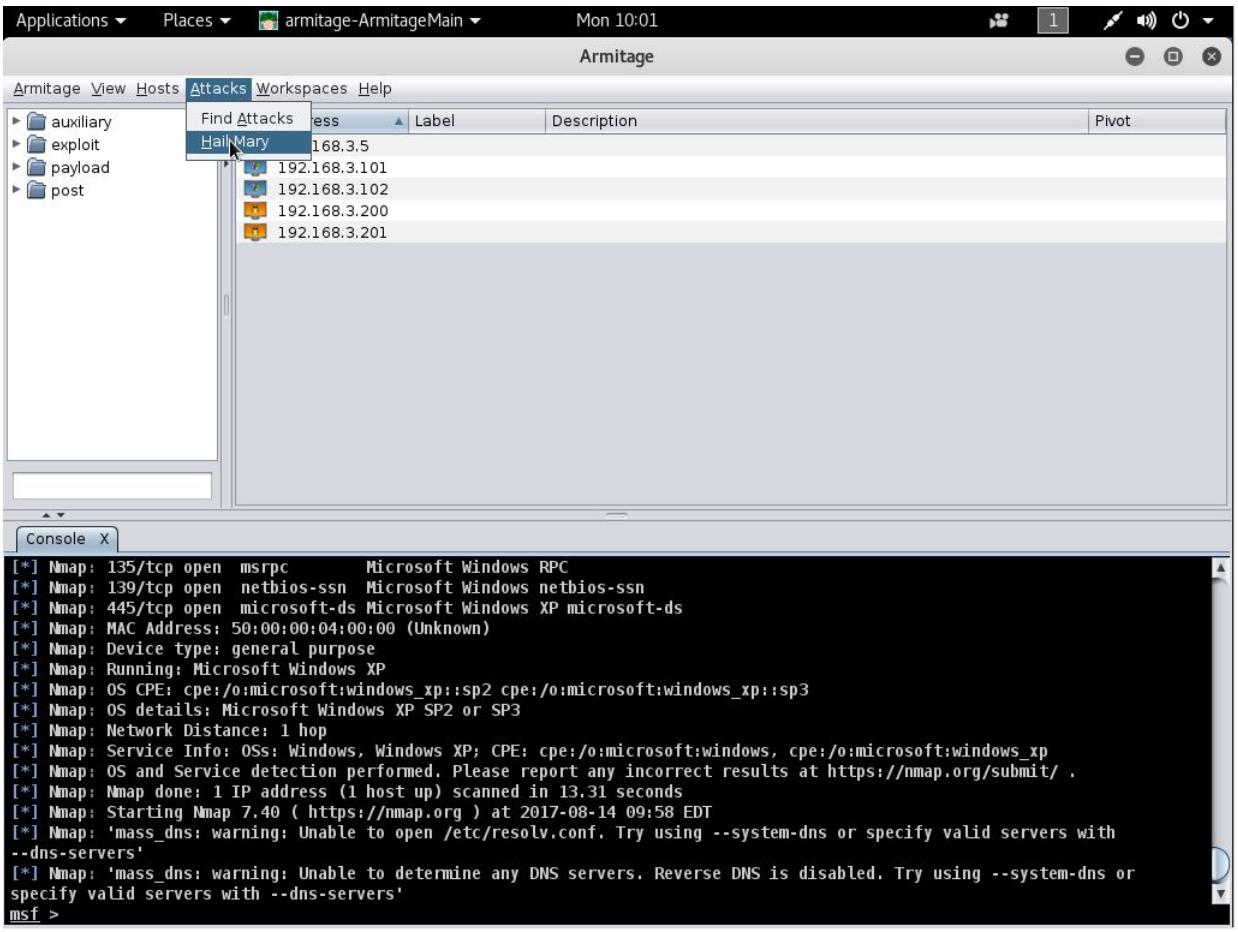


Fig. 10.11: Ejecución de ataques (exploits) encontrados con Armitage 1 de 3.

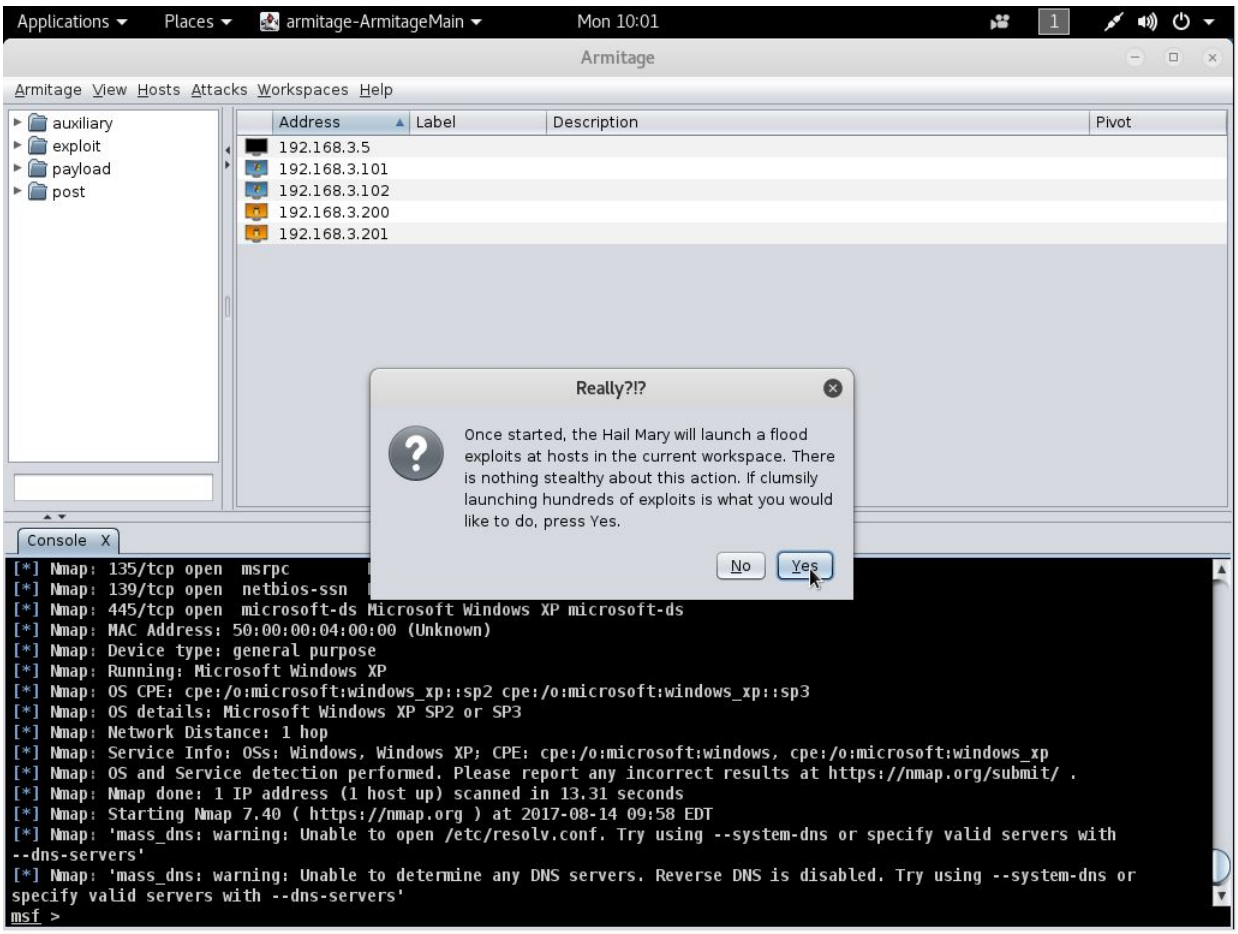


Fig. 10.12: Ejecución de ataques (exploits) encontrados con Armitage 2 de 3.

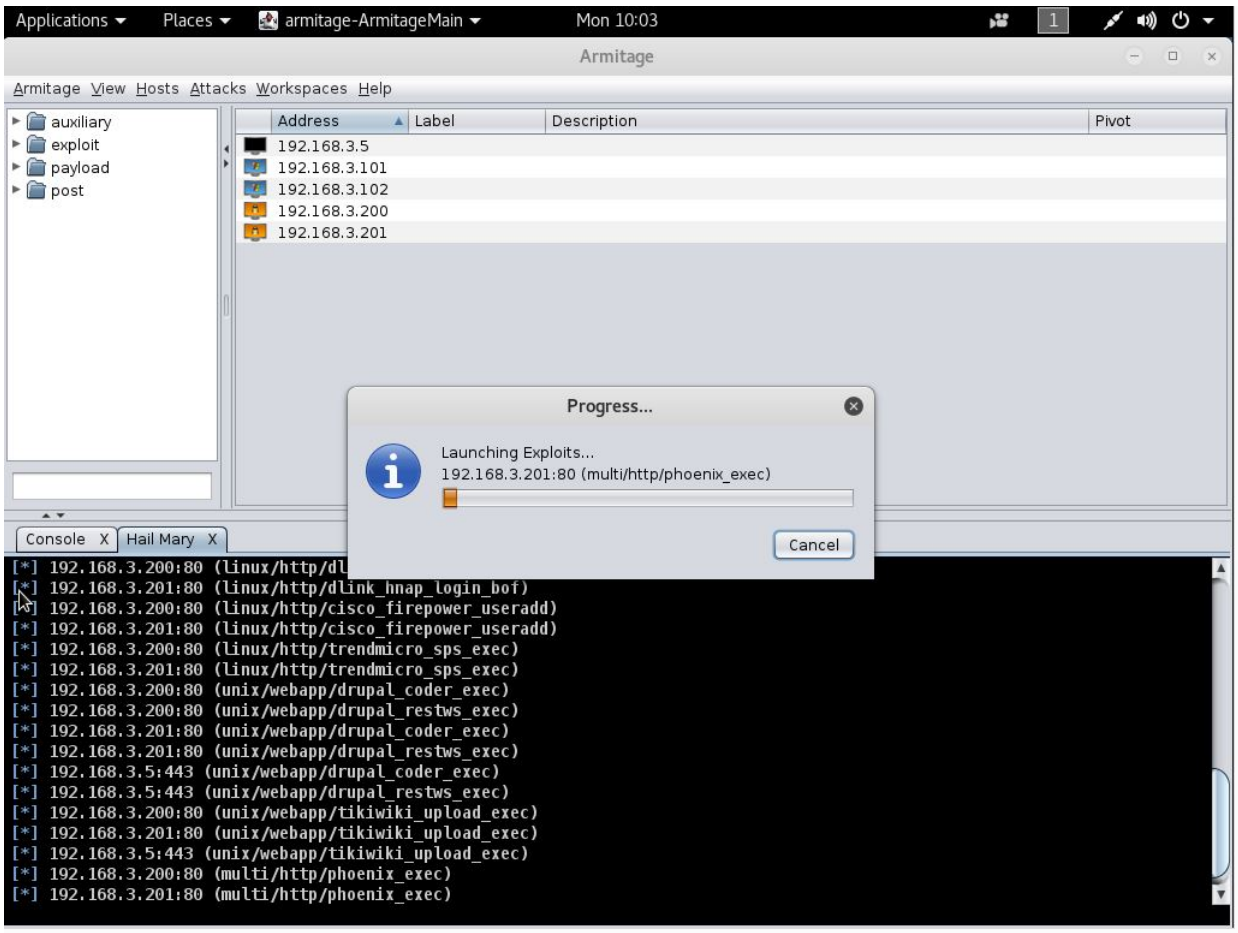


Fig. 10.13: Ejecución de ataques (exploits) encontrados con Armitage 3 de 3.

10.2 ANEXO 2: Configuración OSSEC (el HIDS en el Servidor Web)

En este apartado se configurará el HIDS en el Servidor Web. El equipo "Servidor Web" es un Linux Debian.

Para el despliegue del HIDS en Linux Alienvault recomienda realizar la descarga e instalación del agente desde la página oficial (https://ossec.github.io/downloads.html).

El proceso realizado ha sido el siguiente:

1. Descargar el agente.

2. Descomprimirlo

```
root@WEBserverDMZ:/home/jluis/Descargas# wget https://github.com/ossec/ossec-hids/archive/2.9.1.tar.gz
--2017-08-30 12:50:16--  https://github.com/ossec/ossec-hids/archive/2.9.1.tar.gz
Resolviendo github.com (github.com)... 192.30.253.113, 192.30.253.112
Conectando con github.com (github.com)[192.30.253.113]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Localización: https://codeload.github.com/ossec/ossec-hids/tar.gz/2.9.1 [siguiendo]
--2017-08-30 12:50:16--  https://codeload.github.com/ossec/ossec-hids/tar.gz/2.9.1
Resolviendo codeload.github.com (codeload.github.com)... 192.30.253.120, 192.30.253.121
Conectando con codeload.github.com (codeload.github.com)[192.30.253.120]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1686377 (1,6M) [application/x-gzip]
Grabando a: "2.9.1.tar.gz"

2.9.1.tar.gz                    100%[=================================================>]   1,61M  1,40MB/s    in 1,2s

2017-08-30 12:50:18 (1,40 MB/s) - "2.9.1.tar.gz" guardado [1686377/1686377]

root@WEBserverDMZ:/home/jluis/Descargas# tar zxf 2.9.1.tar.gz
root@WEBserverDMZ:/home/jluis/Descargas# ls
2.9.1.tar.gz  ossec-hids-2.9.1
root@WEBserverDMZ:/home/jluis/Descargas#
```

Fig. 10.14: Descarga y descompresión de OSSEC

3. Seguir el proceso de instalación.

```
root@WEBserverDMZ:/home/jluis/Descargas/ossec-hids-2.9.1# ./install.sh

  ** Para instalação em português, escolha [br].
  ** 要使用中文进行安装，请选择 [cn].
  ** Fur eine deutsche Installation wohlen Sie [de].
  ** Για εγκατάσταση στα Ελληνικά, επιλέξτε [el].
  ** For installation in English, choose [en].
  ** Para instalar en Español , eliga [es].
  ** Pour une installation en français, choisissez [fr]
  ** A Magyar nyelvű telepítéshez válassza [hu].
  ** Per l'installazione in Italiano, scegli [it].
  ** 日本語でインストールします．選択して下さい．[jp].
  ** Voor installatie in het Nederlands, kies [nl].
  ** Aby instalować w języku Polskim, wybierz [pl].
  ** Для инструкций по установке на русском ,введите [ru].
  ** Za instalaciju na srpskom, izaberi [sr].
  ** Türkçe kurulum için seçin [tr].
  (en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]: es
```

Fig. 10.15: Proceso de instalación de OSSEC 1 de 4.

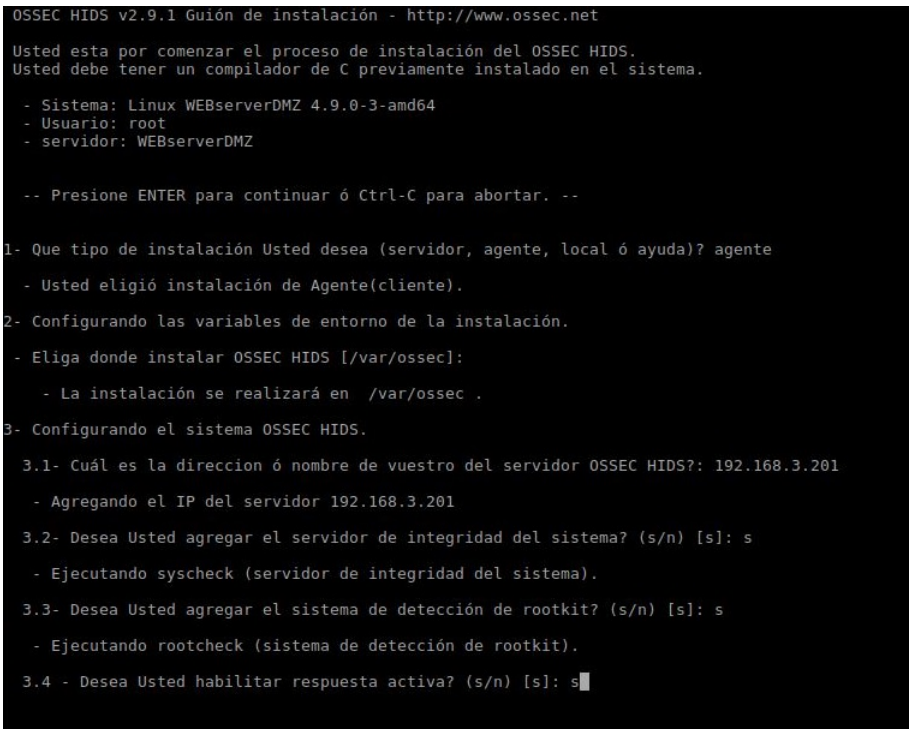


Fig. 10.16: Proceso de instalación de OSSEC 2 de 4.

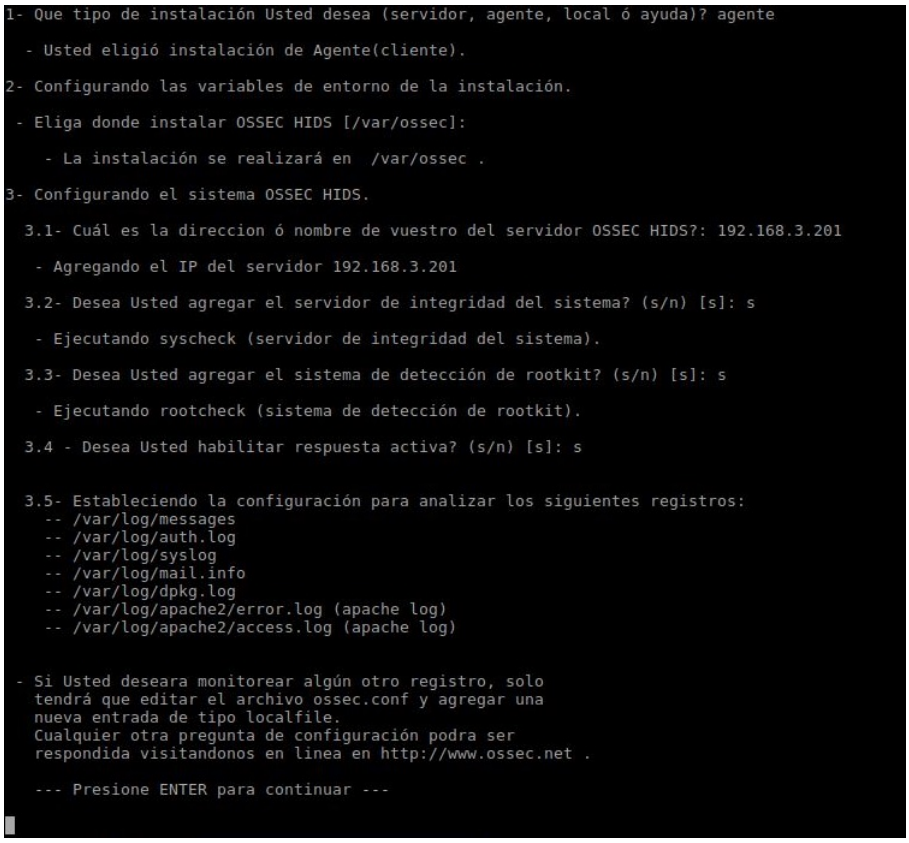


Fig. 10.17: Proceso de instalación de OSSEC 3 de 4.

```
install -m 0550 -o root -g ossec ../active-response/*.sh /var/ossec/active-response/bin/
install -m 0550 -o root -g ossec ../active-response/firewalls/*.sh /var/ossec/active-response/bin/
install -d -m 0550 -o root -g ossec /var/ossec/var
install -d -m 0770 -o root -g ossec /var/ossec/var/run
./init/fw-check.sh execute
install -m 0550 -o root -g 0 ossec-agentd /var/ossec/bin
install -m 0550 -o root -g 0 agent-auth /var/ossec/bin
install -d -m 0750 -o ossec -g ossec /var/ossec/queue/rids


 - El sistema es Debian (Ubuntu or derivative).
 - Init script modificado para empezar OSSEC HIDS durante el arranque.

 - Configuración finalizada correctamente.

 - Para comenzar OSSEC HIDS:
     /var/ossec/bin/ossec-control start

 - Para detener OSSEC HIDS:
     /var/ossec/bin/ossec-control stop

 - La configuración puede ser leída ó mofificada en /var/ossec/etc/ossec.conf


    Gracias por usar OSSEC HIDS.
    Si tuviera Usted alguna duda, sugerencia ó haya encontrado
    algun desperfecto, contactese con nosotros a contact@ossec.net
    ó usando nuestrs lista pública de correo en ossec-list@ossec.net

    Más información puede ser encontrada en http://www.ossec.net

            ---  Presione ENTER para finalizar. ---
       (Tal vez encuentre más información a continuación).


 - Usted debe de añadir este agente en el servidor así podran
   comunicarse el úno con el ótro. Una vez culminada la tarea
   podra ejecutar la herramienta 'manage_agents' para importar
   la autentificación de llaves extraidas del servidor.

   /var/ossec/bin/manage_agents

   Más información en:
   http://www.ossec.net/en/manual.html#ma

root@WEBserverDMZ:/home/jluis/Descargas/ossec-hids-2.9.1#
```

Fig. 10.18: Proceso de instalación de OSSEC 4 de 4.

Una vez completado el proceso de instalación, desde el equipo OSSIM se realiza el siguiente proceso para añadir el agente a OSSIM:

1. Navegar a **Environment → Detection**
2. Navegar a **HIDS → Agents → Agent Control → Add Agent**

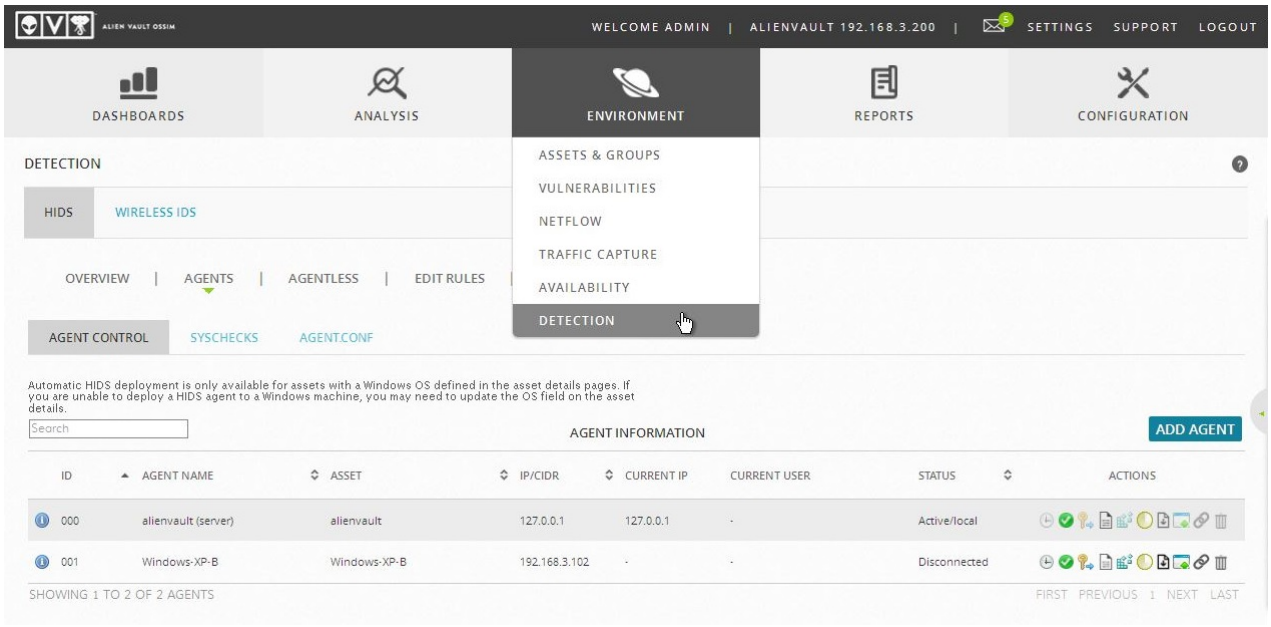


Fig. 10.19: Proceso de añadir agente a OSSIM 1 de 3.

3. Se selecciona el agente de la lista de activos.

4. Se guarda.

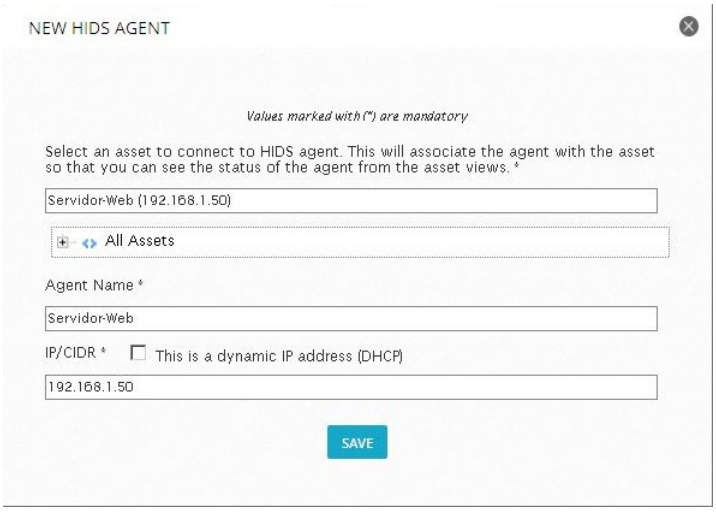


Fig. 10.20: Proceso de añadir agente a OSSIM 2 de 3.

5. Se extrae la clave.

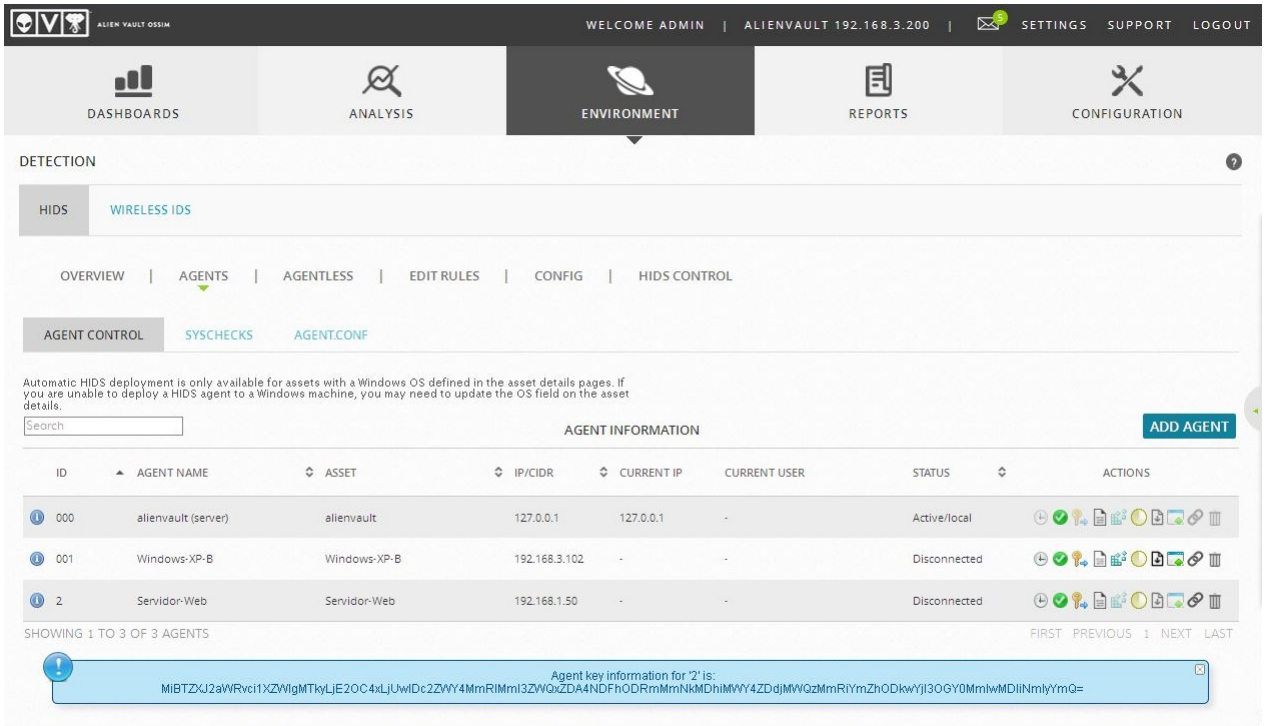


Fig. 10.21: Proceso de añadir agente a OSSIM 3 de 3.

Para finalizar el proceso en el equipo "Servidor Web" se realiza el siguiente proceso:

1. Se ejecuta:
   /var/ossec/bin/manage_agents
   Introduciendo la tecla "I" y insertando la clave previamente extraída en OSSIM.

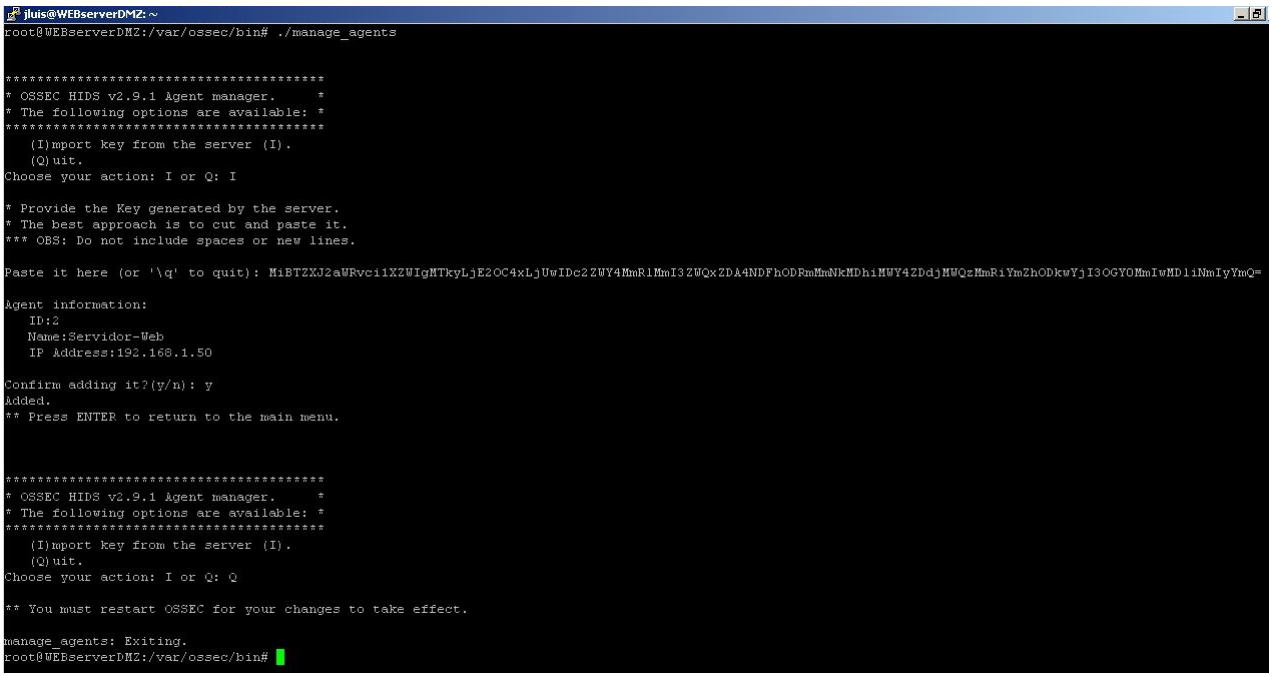


Fig. 10.22: Configuración de la clave del agente.

2. Se edita el fichero /var/ossec/etc/ossec.conf modificando la dirección IP del servidor OSSIM.

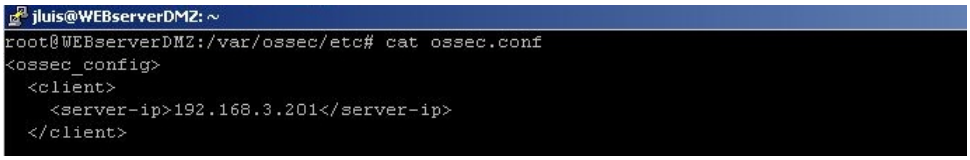


Fig. 10.23: Contenido del fichero de configuración después de editarlo.

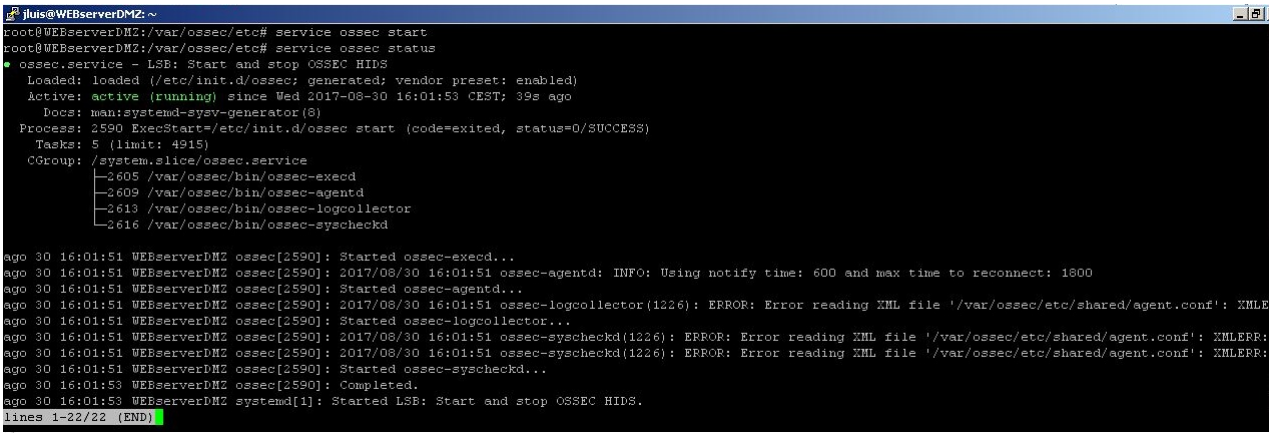


Fig. 10.24: Inicialización de OSSEC.

Por último vamos a OSSIM y navegamos a **Environment** → **Detection** y hacemos click en **HIDS Control** y después en **Restart**.

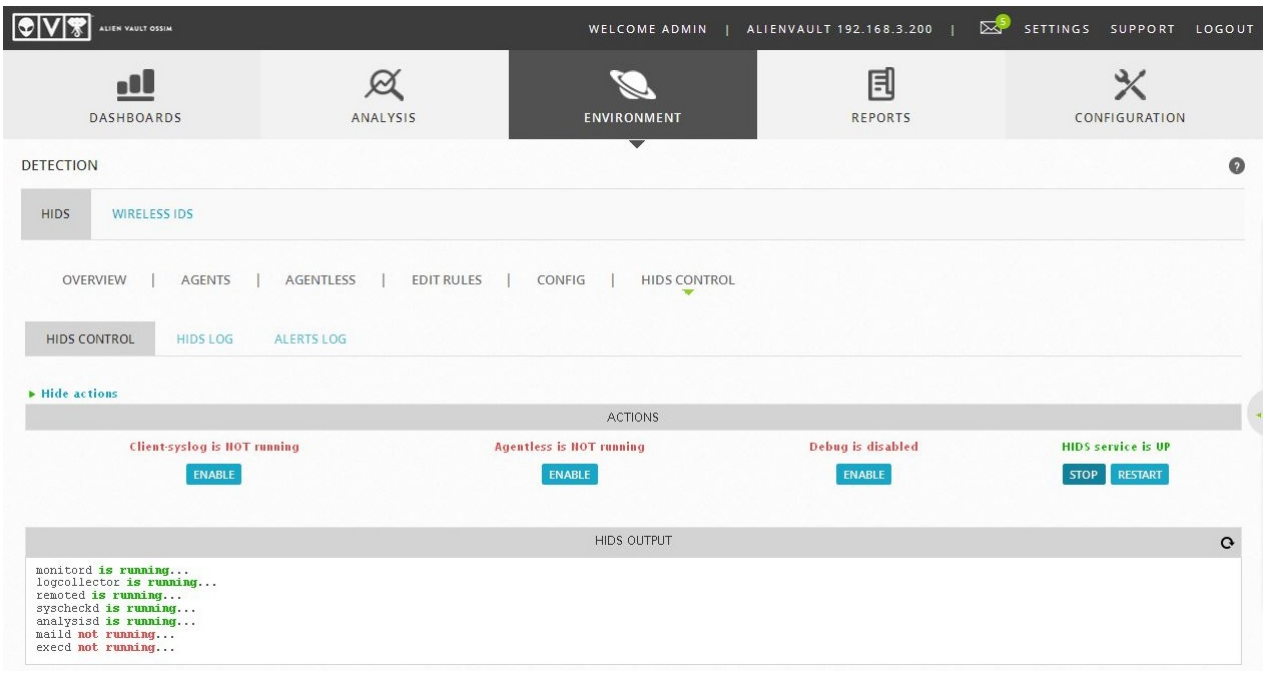


Fig. 10.25: Actualización del estado de OSSIM.

10.3 ANEXO 3: Configuración Firewall Cisco-ASA

Los plugins en OSSIM procesan la información recibida de diferentes "data-sources" (fuentes de datos) , analizan esa información (parse) y la normalizan . Siendo después guardada esta información en forma de eventos en la base de datos de OSSIM.

En este contexto los plugins definen dos cosas:

1. Como recoger la información de una aplicación o dispositivo.
2. Como normalizar esta información recogidad en forma de eventos estándar.

Existen a su vez dos tipos de plugins:

1. "Detector Plugins" que recogen y extraen eventos, principalmente de archivos de log
2. "Monitor Plugins" que recogen información ejecutando comandos, como "nmap" "tcptrack"

También existe la posibilidad de habilitarlo usando el asistente de instalación.

Para configurar/ habilitar un plugin ("detector plugin") en un activo previamente descubierto se sigue el siguiente proceso no así en el caso de los plugins de monitorización, en los que se ha de seguir otro proceso :

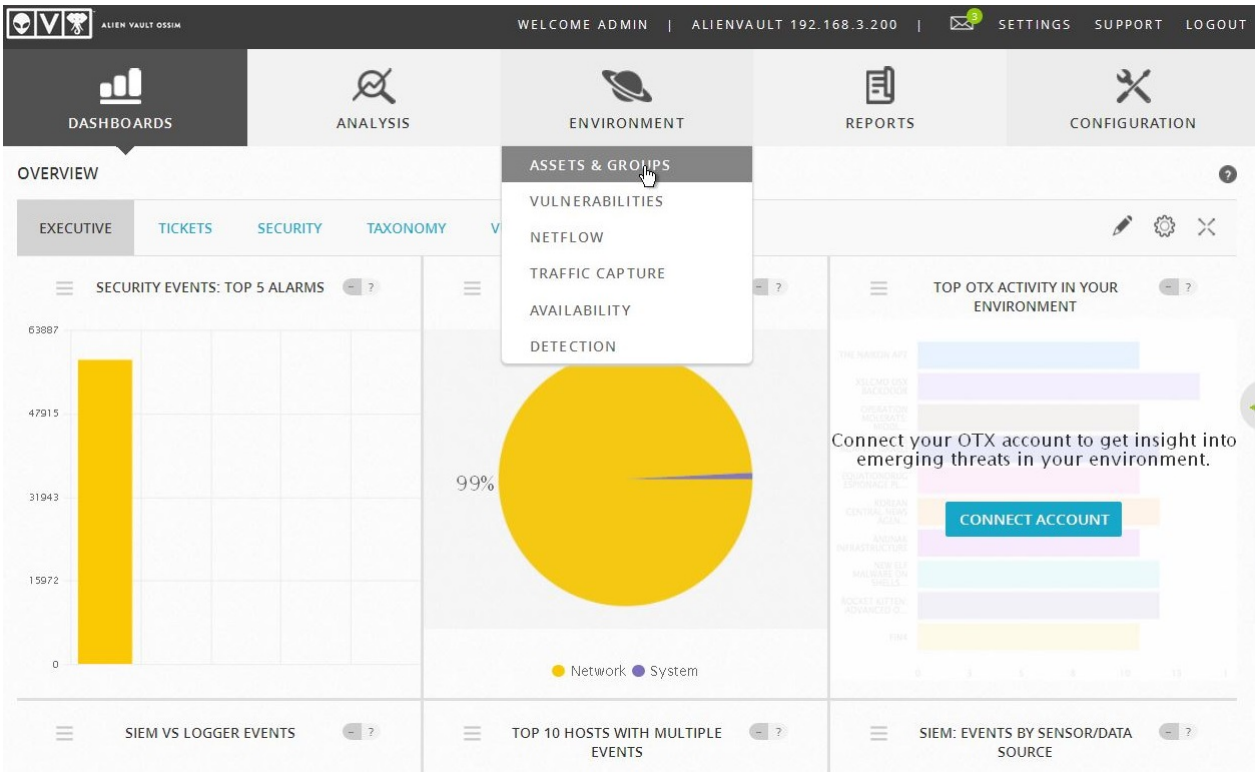


Fig. 10.26: Proceso de configuración de monitorización en OSSIM 1 de 6.

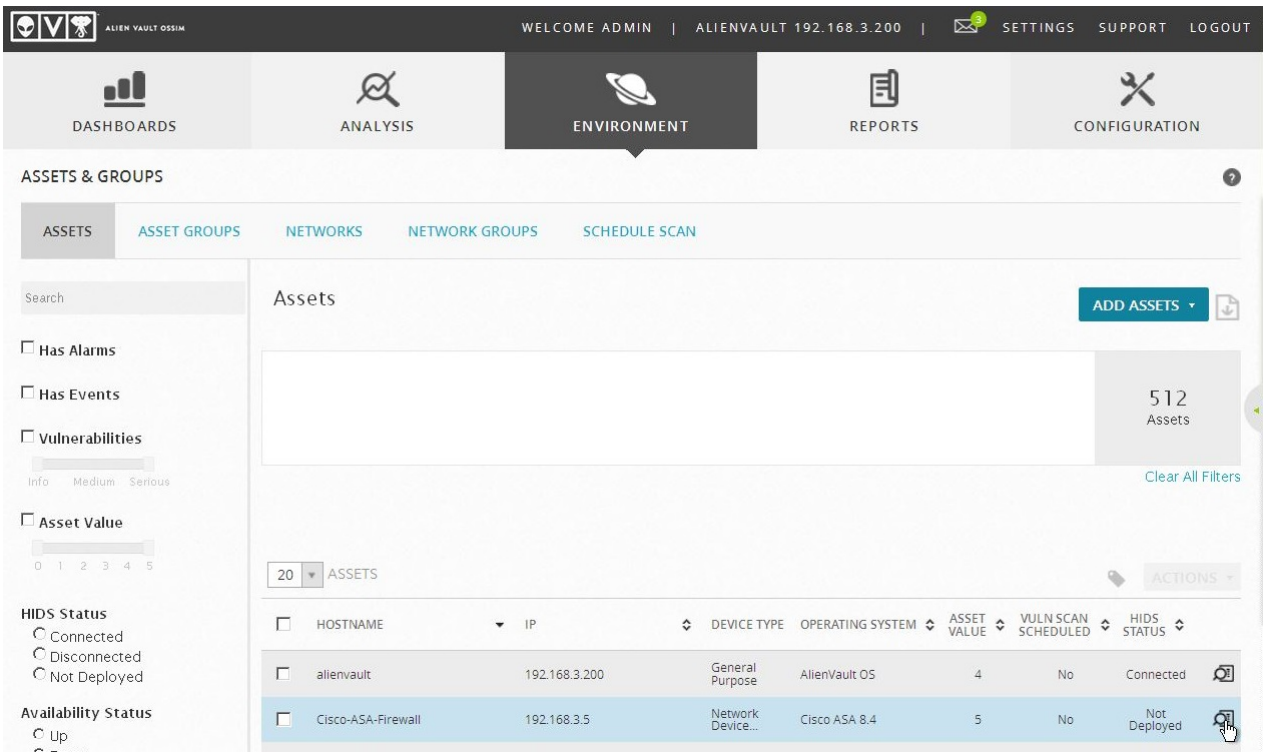


Fig. 10.27: Proceso de configuración de monitorización en OSSIM 2 de 6.

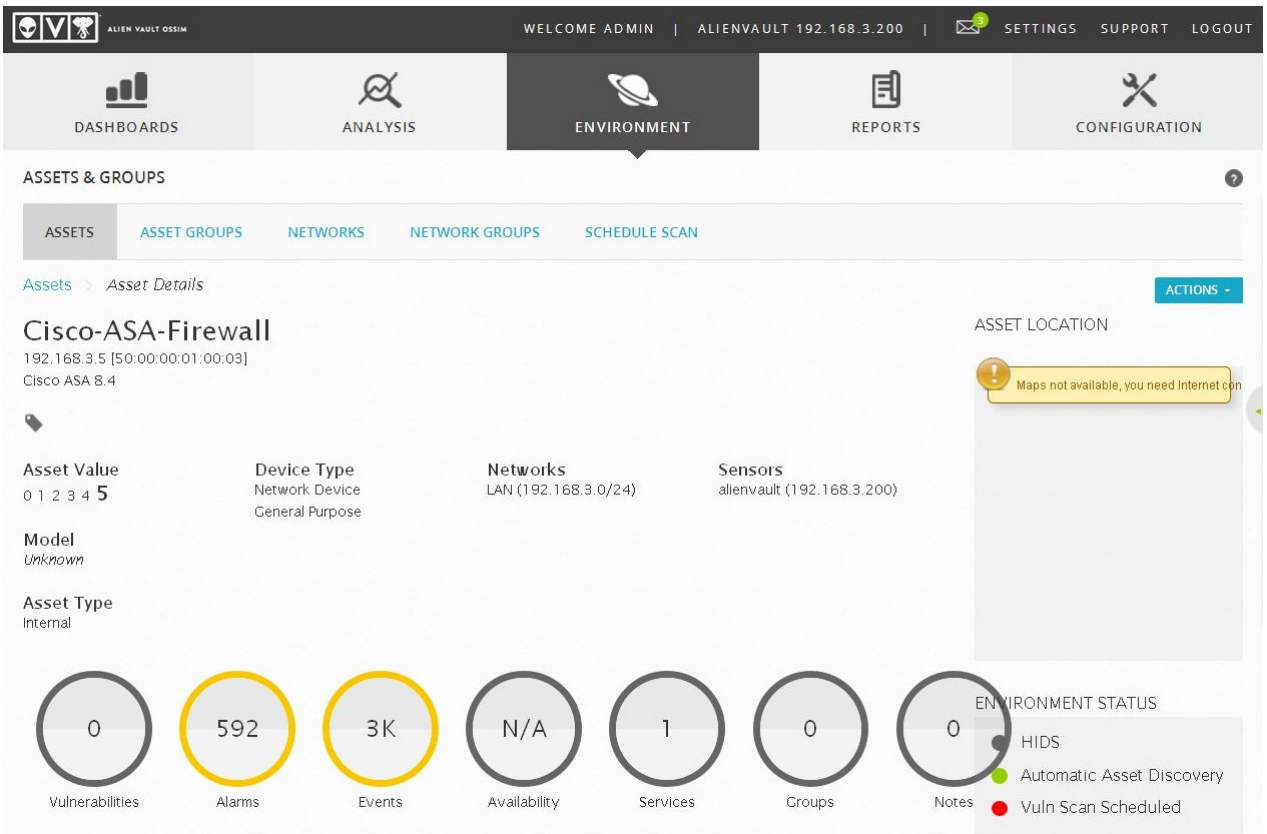


Fig. 10.28: Proceso de configuración de monitorización en OSSIM 3 de 6.

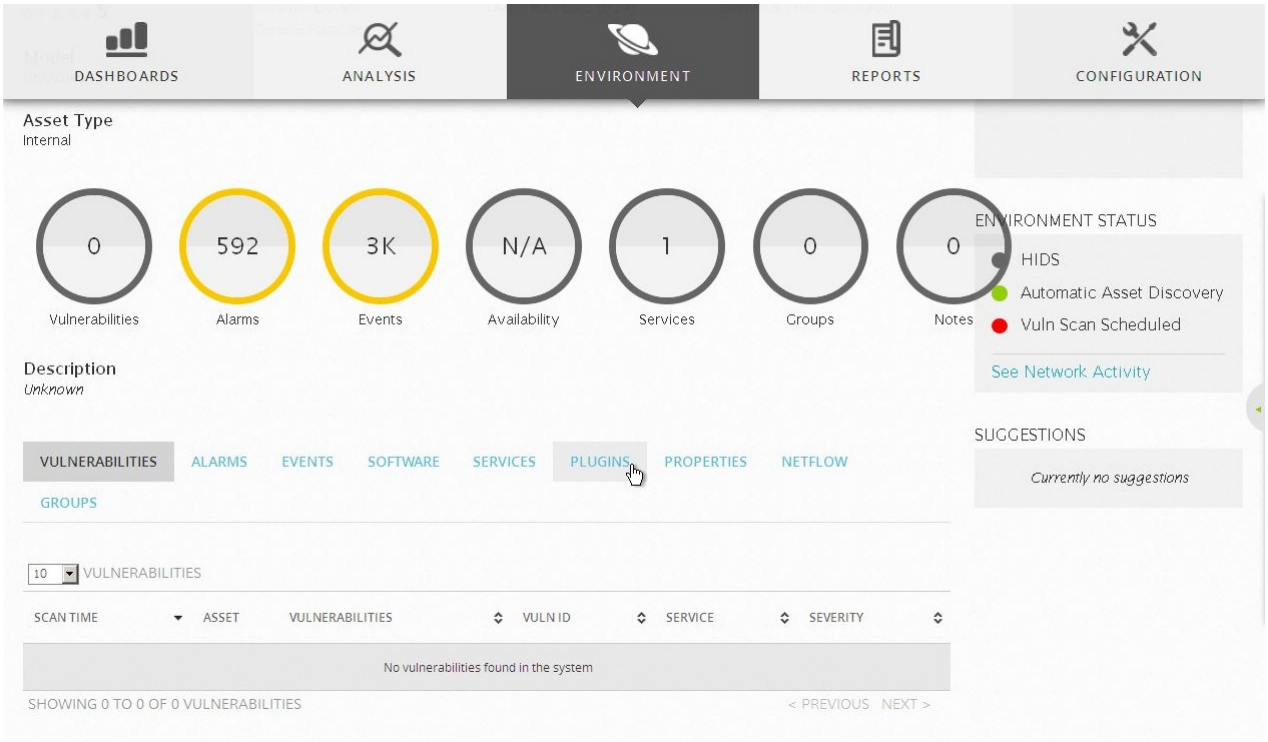


Fig. 10.29: Proceso de configuración de monitorización en OSSIM 4 de 6.

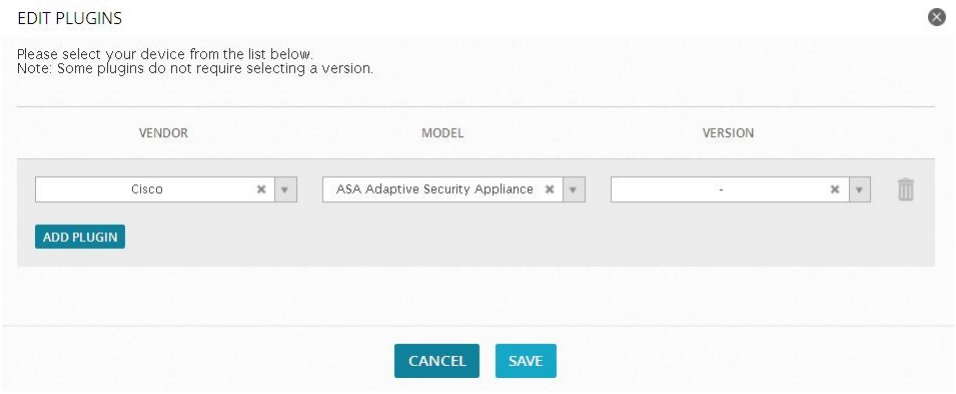


Fig. 10.30: Proceso de configuración de monitorización en OSSIM 5 de 6.

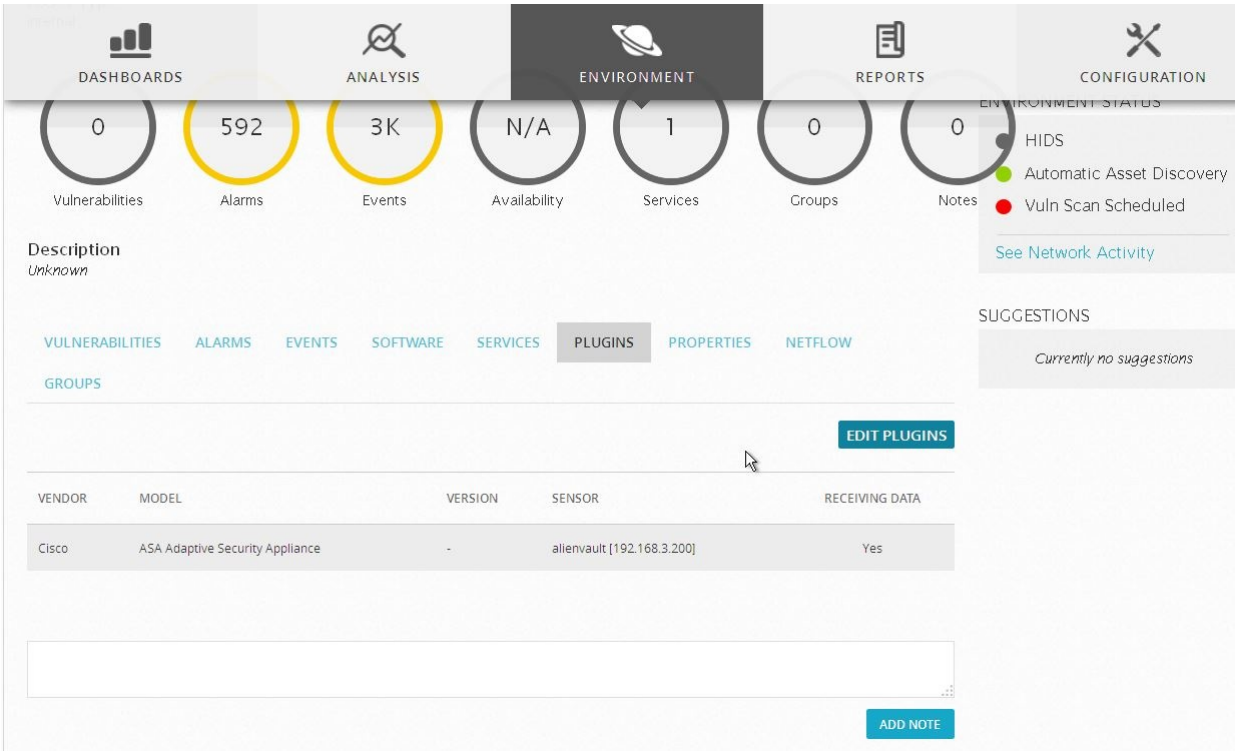


Fig. 10.31: Proceso de configuración de monitorización en OSSIM 6 de 6.

10.4 ANEXO 4: Configuración NDIS OSSIM (Suricata)

Para configurar el NDIS existen dos posibilidades que se describen a continuación

1. Desde el menú de la máquina OSSIM

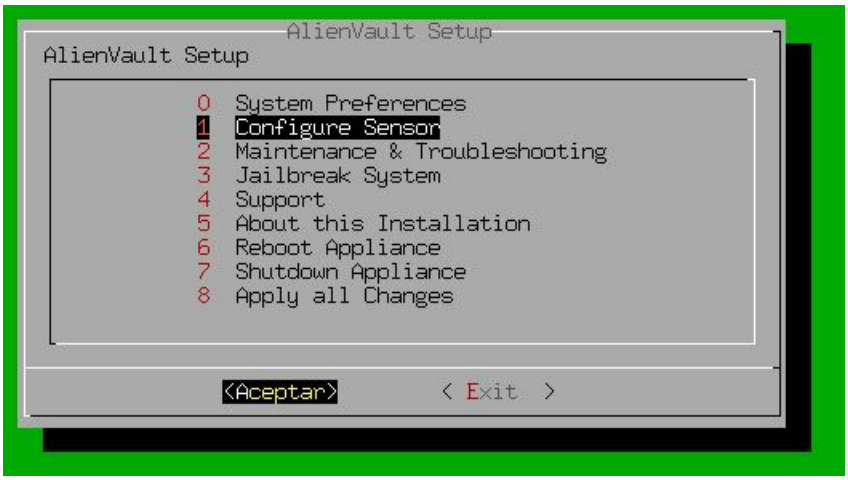


Fig. 10.32: Configuración de NDIS en menú máquina OSSIM 1 de 3.

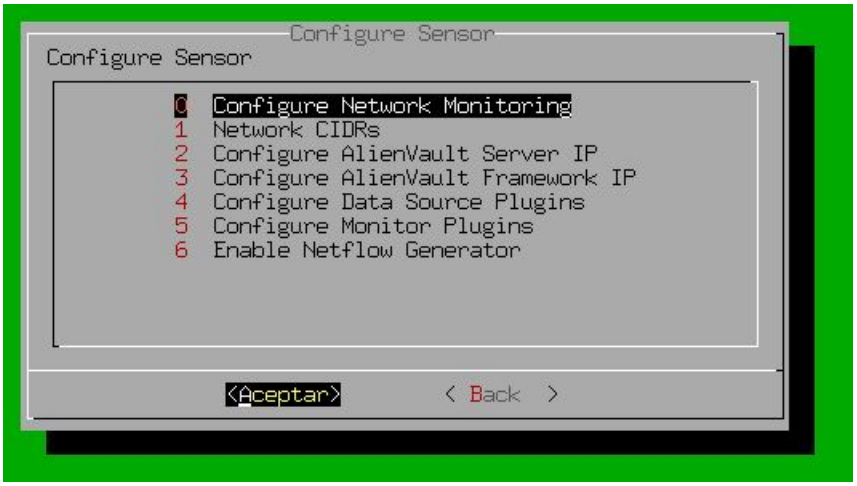


Fig. 10.33: Configuración de NDIS en menú máquina OSSIM 2 de 3.

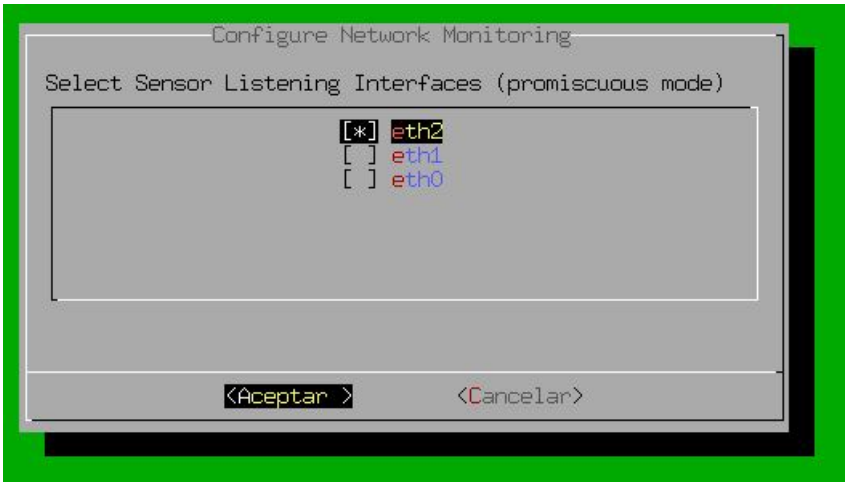


Fig. 10.34: Configuración de NDIS en menú máquina OSSIM 3 de 3.

2. A través del interfaz web: **Configuration → deployment**

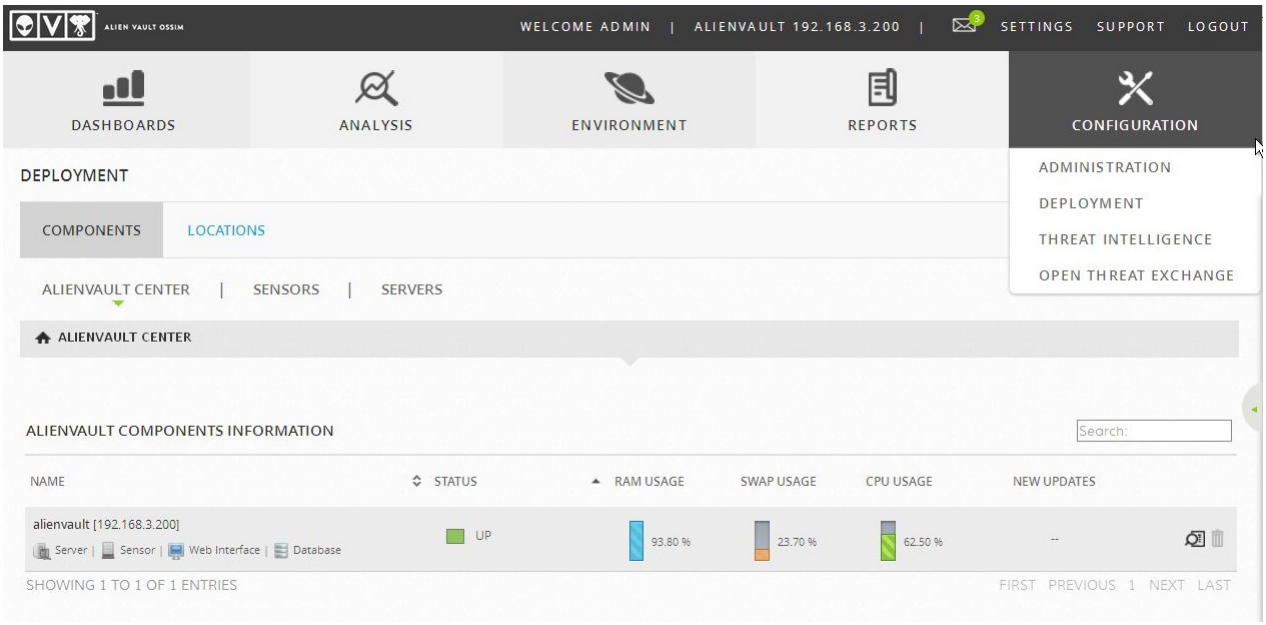


Fig. 10.35: Configuración de NDIS en interfaz web OSSIM 1 de 6.

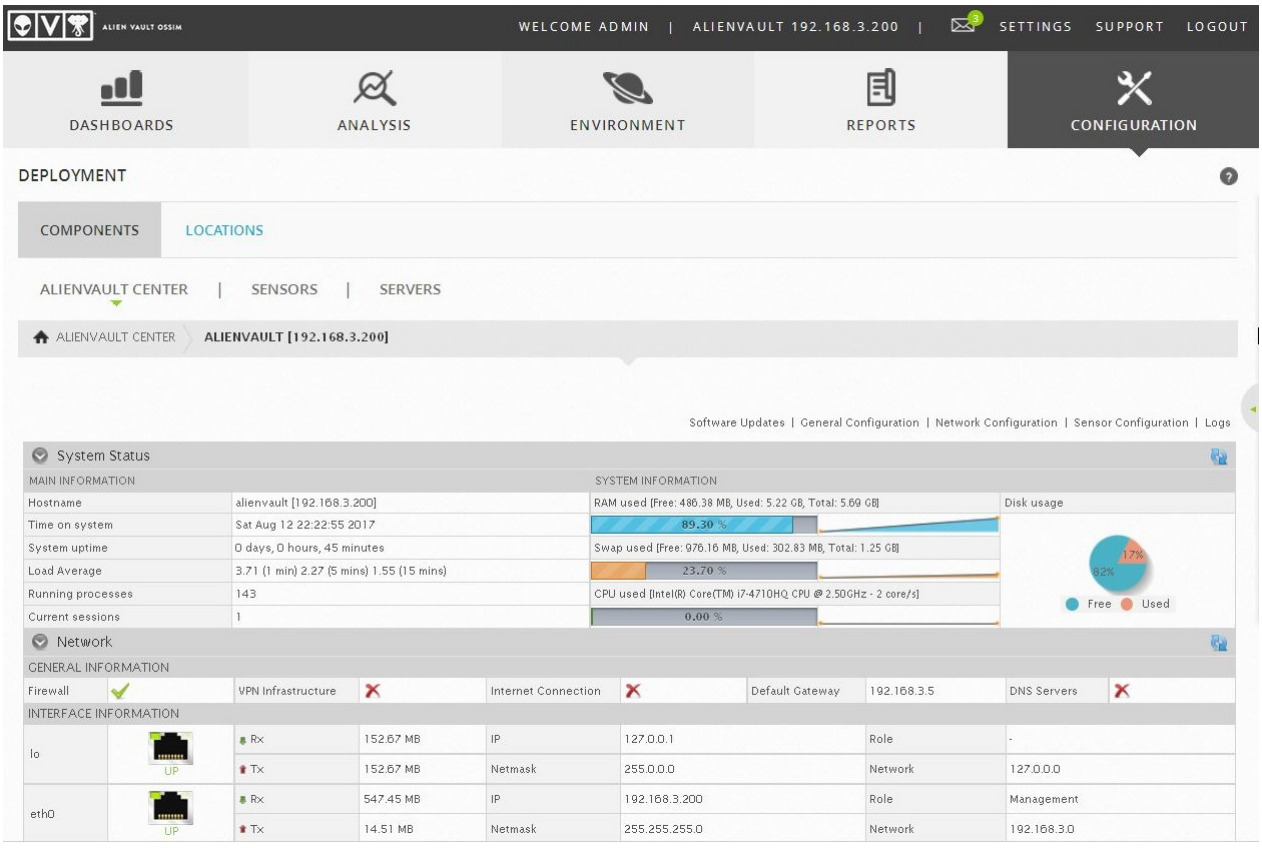


Fig. 10.36: Configuración de NDIS en interfaz web OSSIM 2 de 6.

Doble click en el servidor OSSIM.

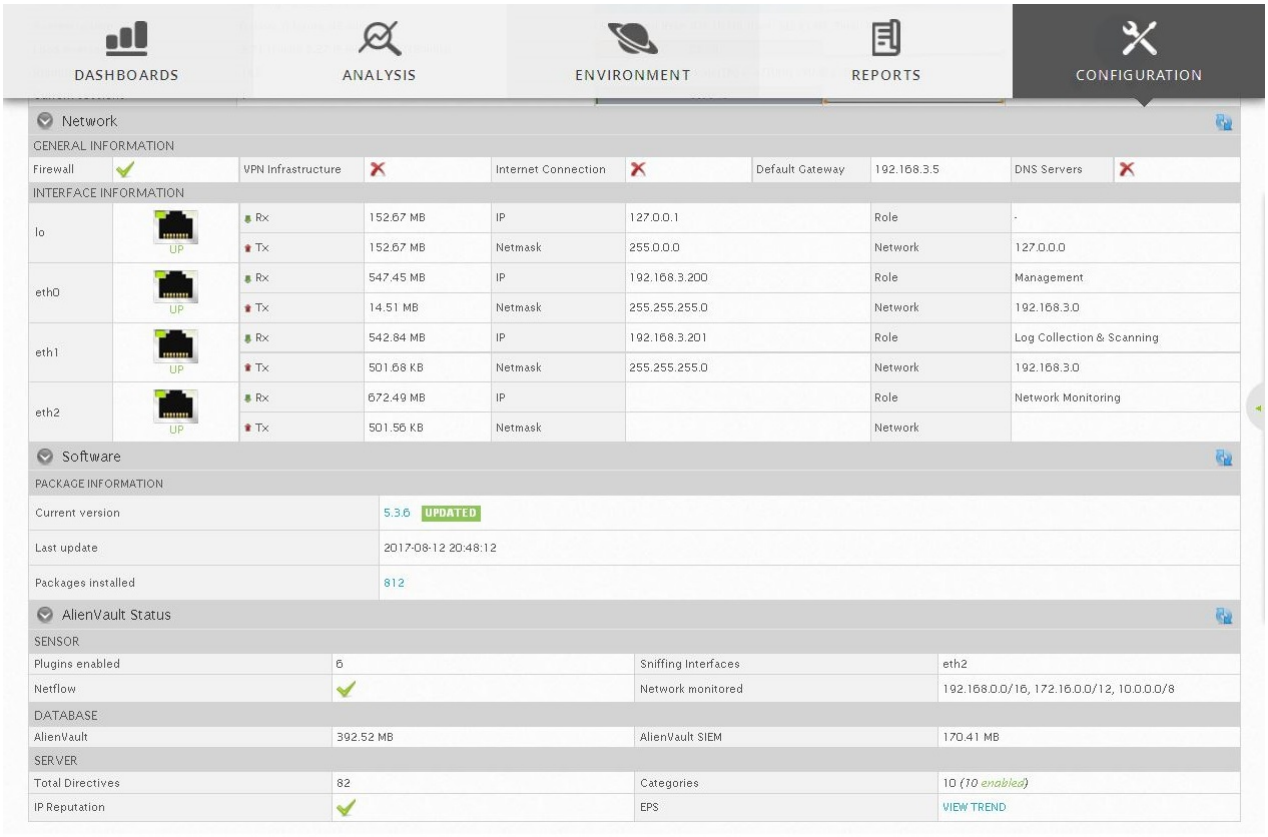


Fig. 10.37: Configuración de NDIS en interfaz web OSSIM 3 de 6.

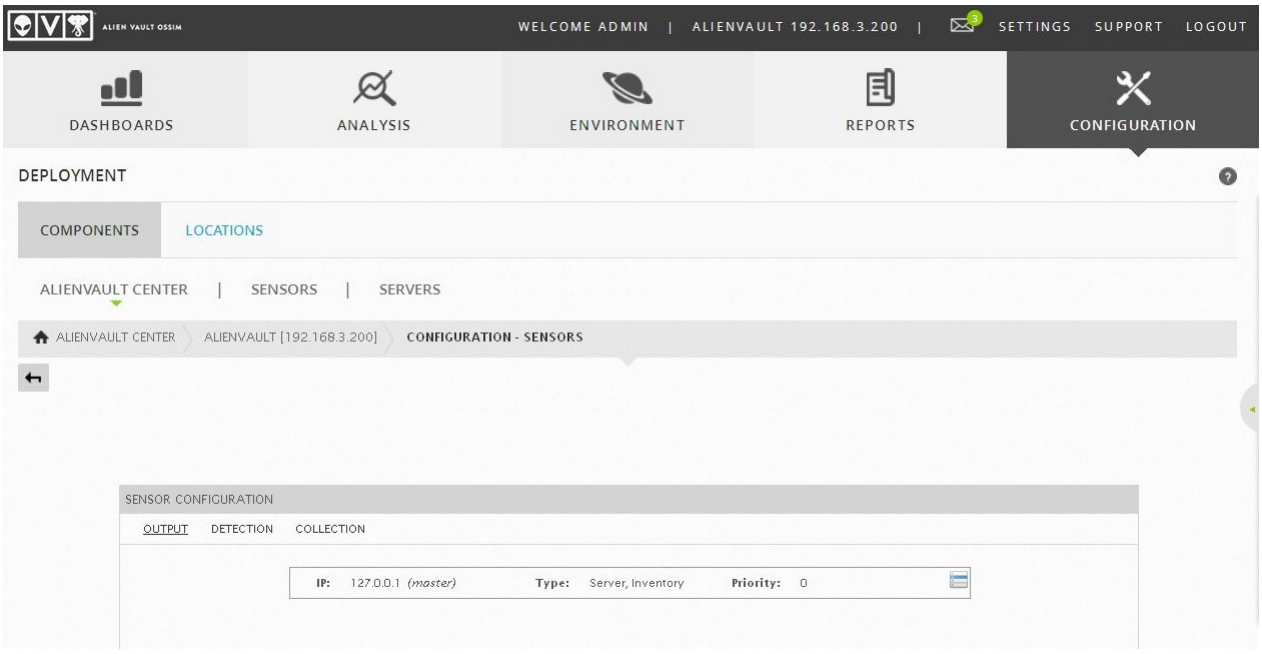


Fig. 10.38: Configuración de NDIS en interfaz web OSSIM 4 de 6.

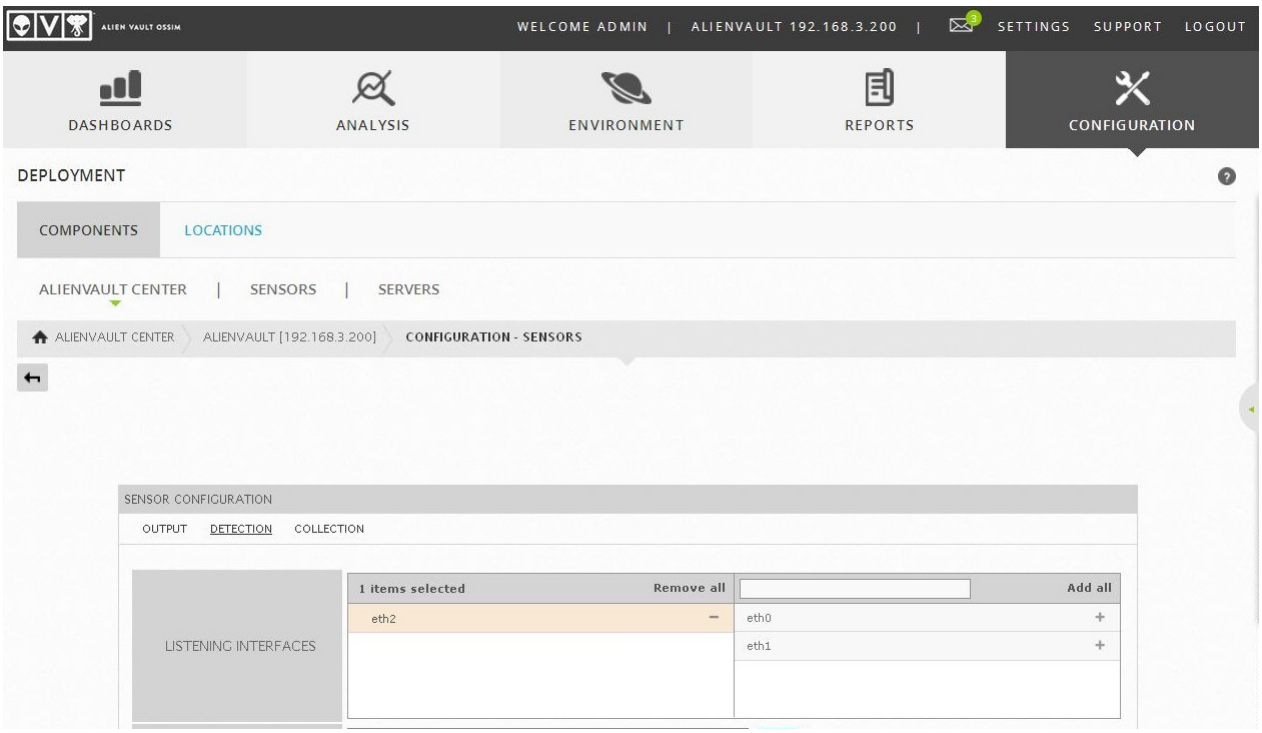


Fig. 10.39: Configuración de NDIS en interfaz web OSSIM 5 de 6.

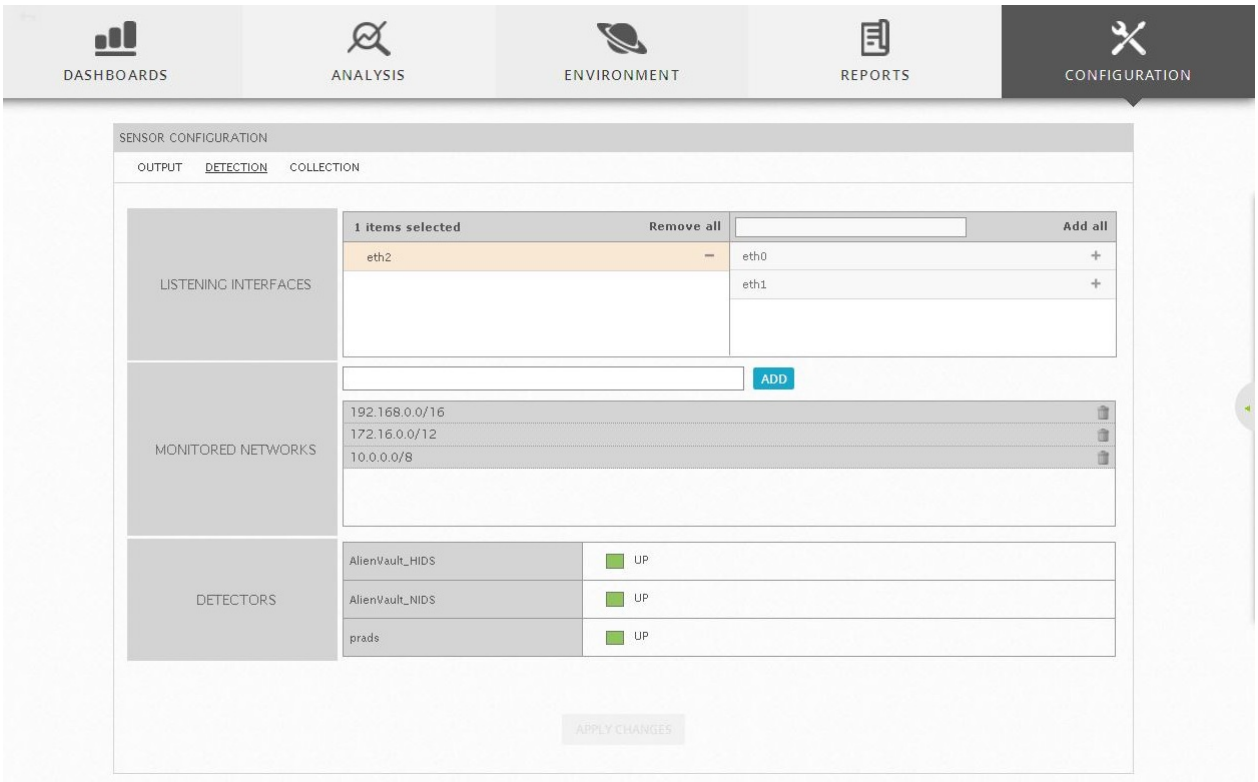


Fig. 10.40: Configuración de NDIS en interfaz web OSSIM 6 de 6.

Por defecto el NDIS de OSSIM solo detecta los ataques que provienen de redes externas a internas. Por ello se ha modificado la configuración para permitir que detecte ataques con independencia del origen del ataque.

El proceso seguido ha sido el siguiente:

1. Ir al fichero de configuración de suricata: /etc/suricata/suricata.yaml

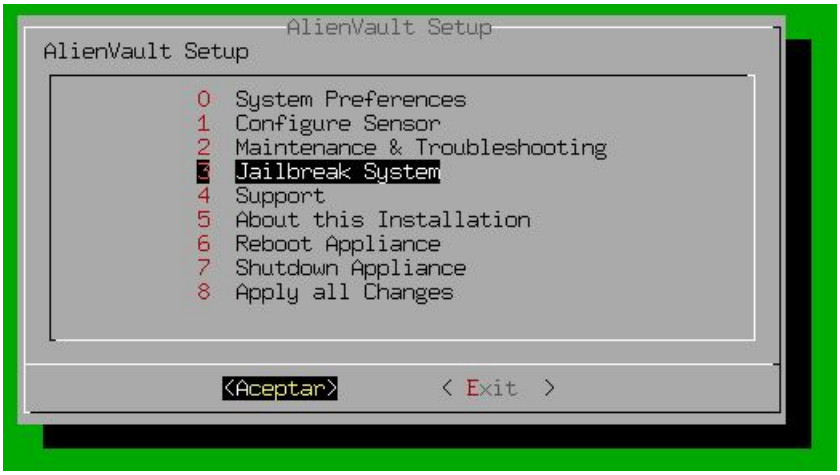


Fig. 10.41: Acceso al terminal de la máquina OSSIM 1 de 2.

Fig. 10.42: Acceso al terminal de la máquina OSSIM 2 de 2.

Fig. 10.43: Localización de los archivos de configuración de Suricata.

2. Cambiar en este fichero la variable “EXTERNAL_NET” cambiando su valor de “!$HOME_NET” a “any”.

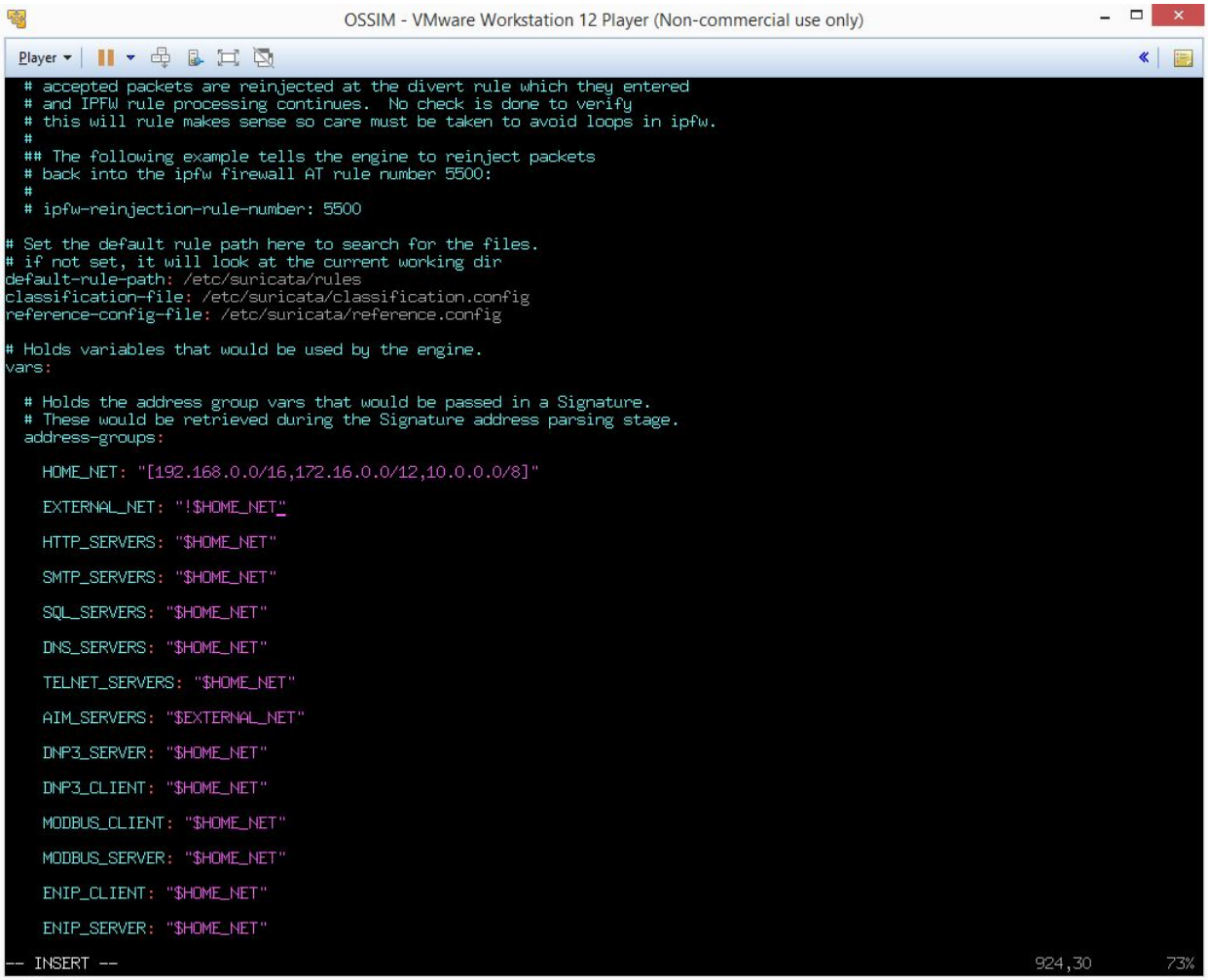


Fig. 10.44: Configuración por defecto de Suricata en OSSIM.

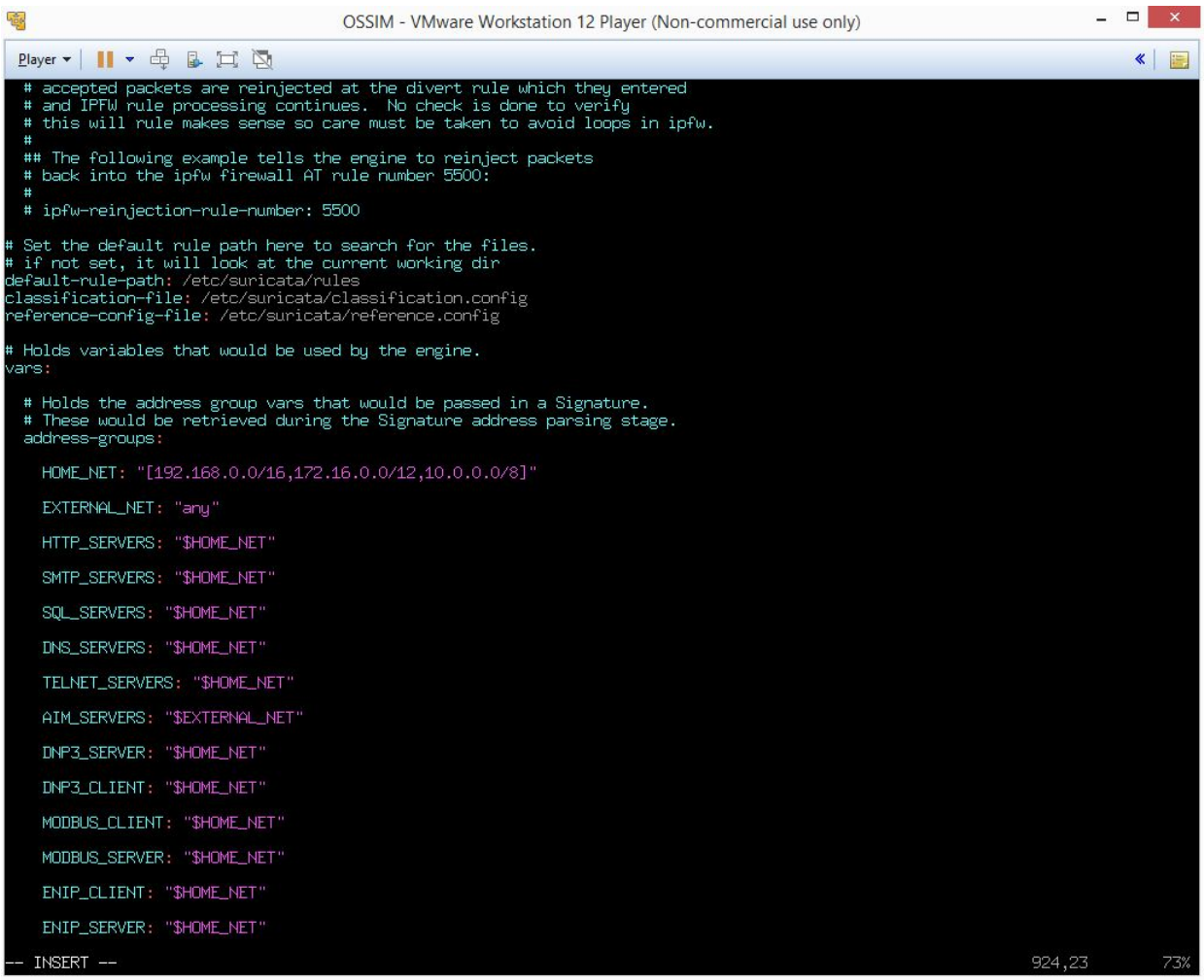


Fig. 10.45: Configuración modificada de Suricata en OSSIM.

Siguiendo el proceso recomendado por AlienVault de configuración del NDIS de OSSIM se ha configurado el switch_LAN de manera que reenvíe el tráfico de todos los puertos al puerto en el que se ha conectado el NDIS de OSSIM. También se ha configurado el interfaz de monitoreo de OSSIM sin dirección IP ya que así lo recomienta AlienVault.

10.5 ANEXO 5: Configuración monitorización de disponibilidad OSSIM

Dado que uno de los aspectos que entran en juego en las vulnerabilidades es la disponibilidad, OSSIM cuenta con la posibilidad de monitorizar si un servicio se encuentra disponible. Para usar esta funcionalidad de OSSIM se debe realizar el siguiente proceso (Se presupone que ya se ha añadido o detectado el activo):

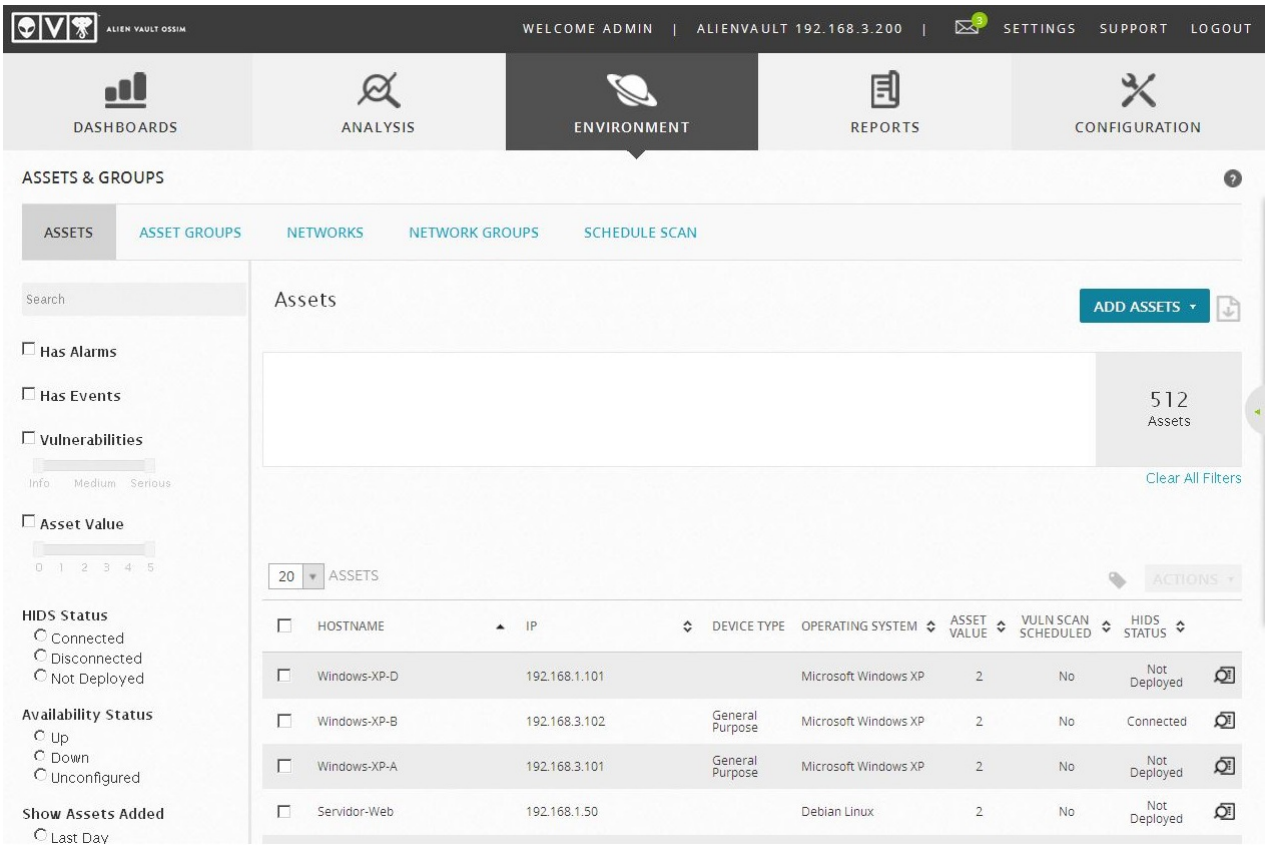


Fig. 10.46: Configuración monitorización de disponibilidad en OSSIM 1 de 6.

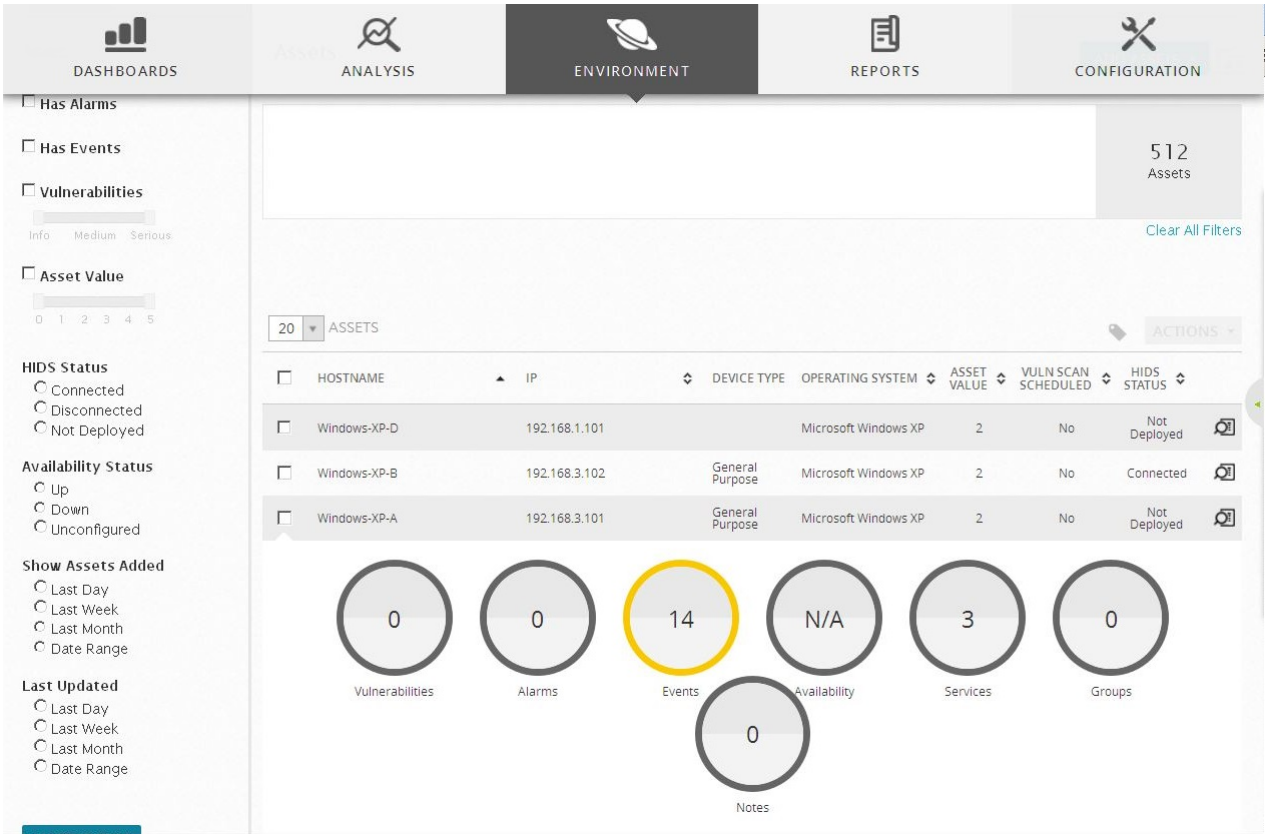


Fig. 10.47: Configuración monitorización de disponibilidad en OSSIM 2 de 6.

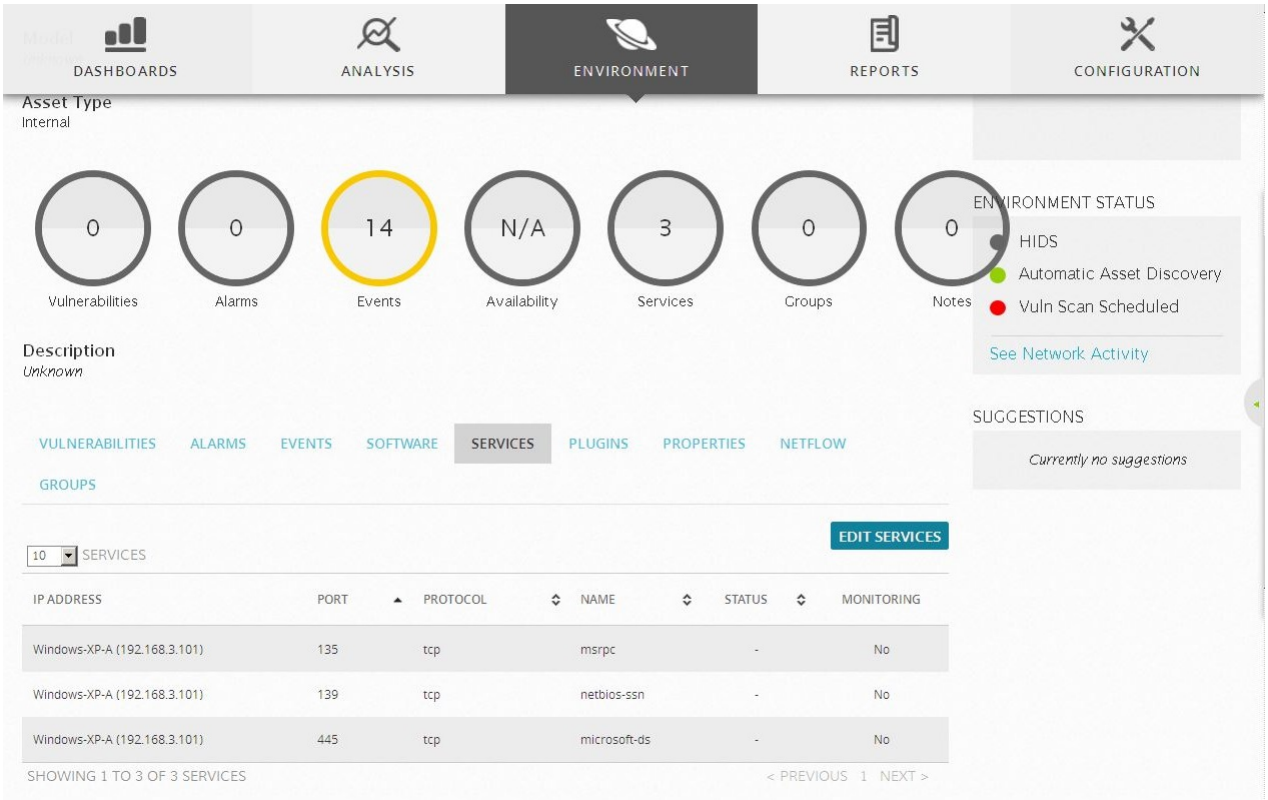


Fig. 10.48: Configuración monitorización de disponibilidad en OSSIM 3 de 6.

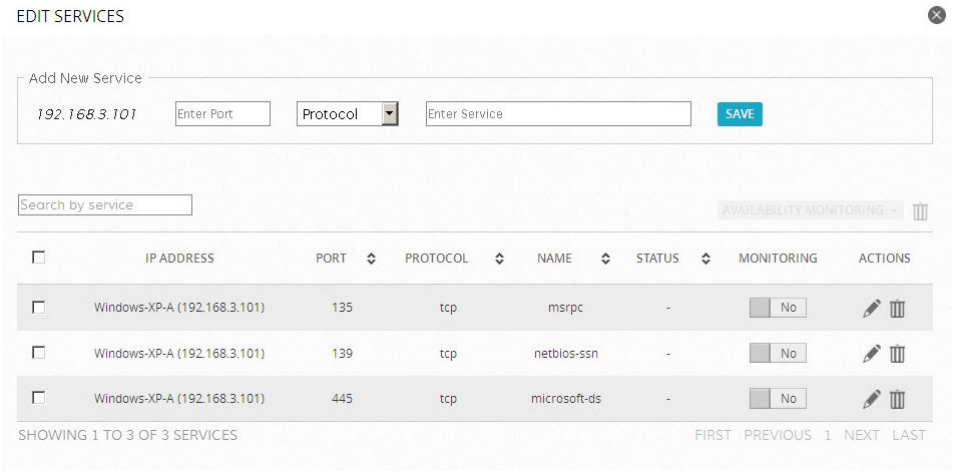


Fig. 10.49: Configuración monitorización de disponibilidad en OSSIM 4 de 6.

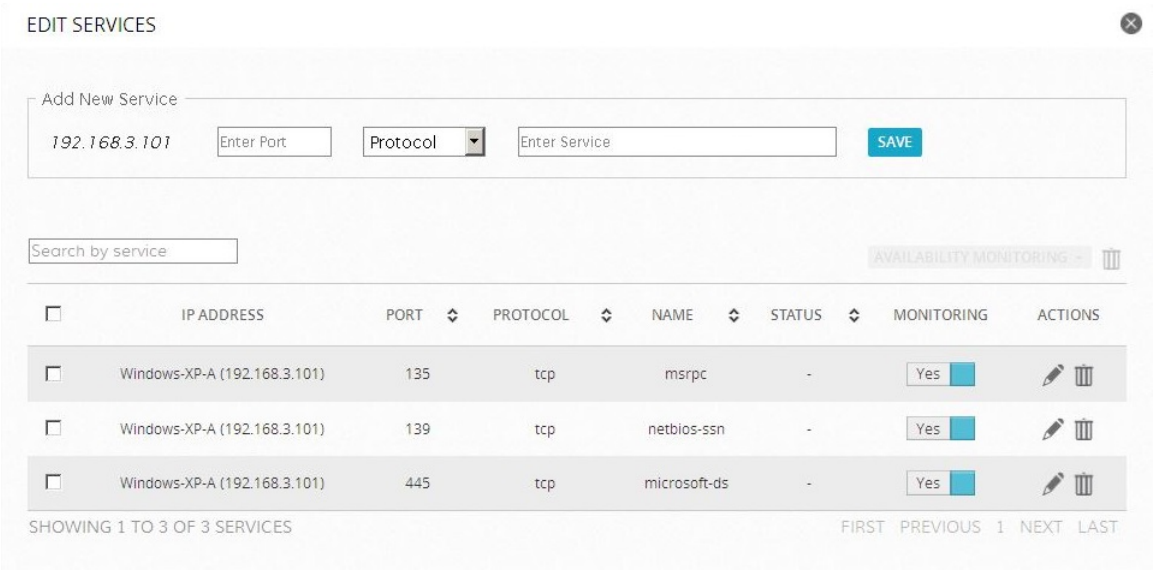


Fig. 10.50: Configuración monitorización de disponibilidad en OSSIM 5 de 6.

Fig. 10.51: Configuración monitorización de disponibilidad en OSSIM 6 de 6.

10.6 ANEXO 6: Configuración Switch LAN

Como switch se ha decidido utilizar un Lubuntu 16.04 configurado para actuar como switch.

Esta decisión se ha debido a que aunque si que es posible simular de manera virtual switches Cisco por ejemplo, no se ha podido utilizar en estos la funcionalidad de port mirroring o hacer SPAN en los puertos debido a que estos switches realizan esta función por hardware lo cual hace imposible este tipo de simulación. Por ese motivo finalmente se ha optado por una máquina linux configurada de manera que actúe como un switch, reenviando el trafico al interfaz de monitoreo de OSSIM.

Para ello e ha utilizado la herramienta bridge-utils.

10.7 ANEXO 7: Escaneo de Vulnerabilidades

Para la detección y escaneo de vulnerabilidades en OSSIM utilizando OpenVAS se ha realizado el proceso que se muestra a continuación.

1. Navegar a **Environment** → **Vulnerabilities**.

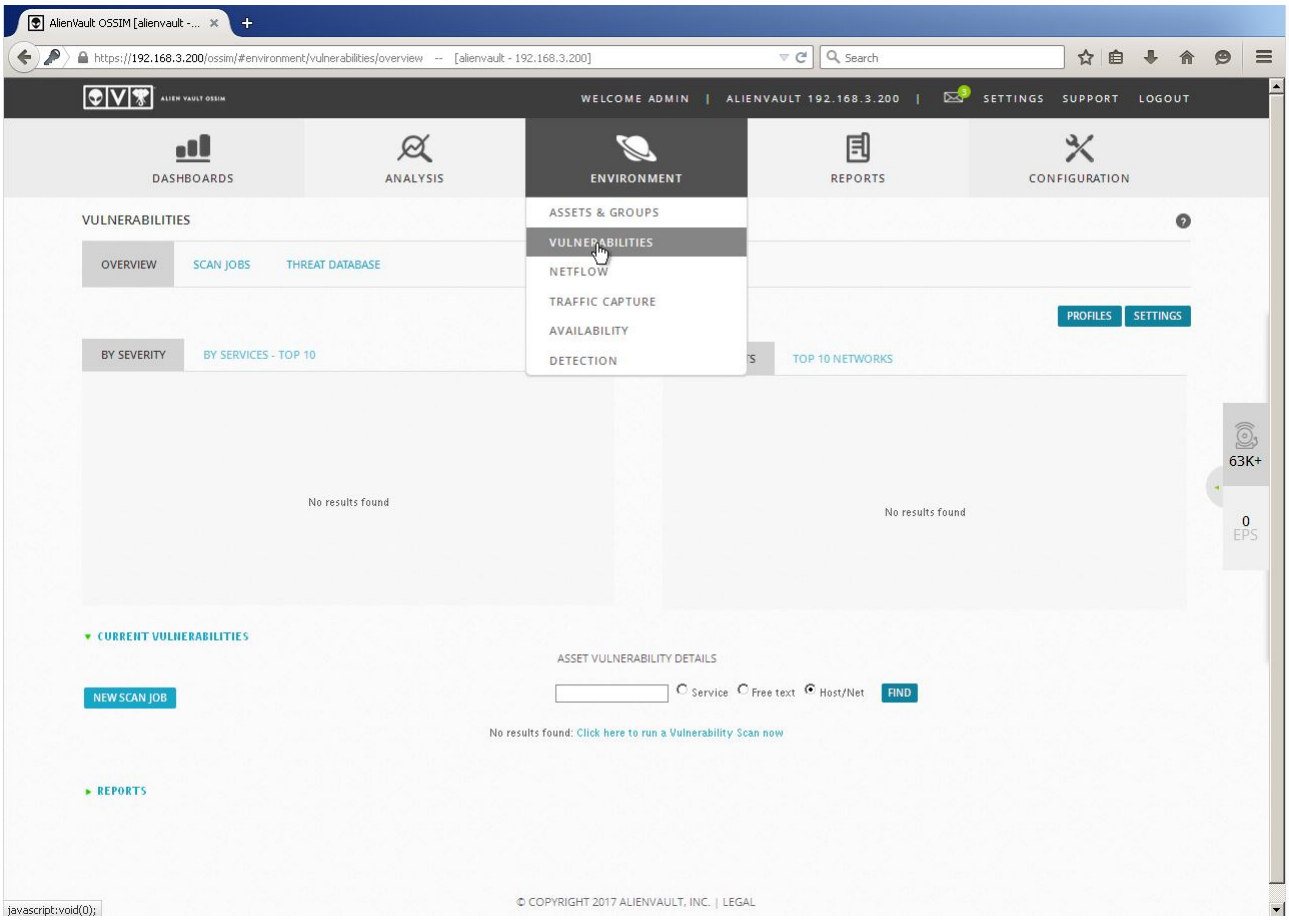


Fig. 10.52: Navegar a la sección de vulnerabilidades en OSSIM.

2. Hacer click en **New Scan Job**.

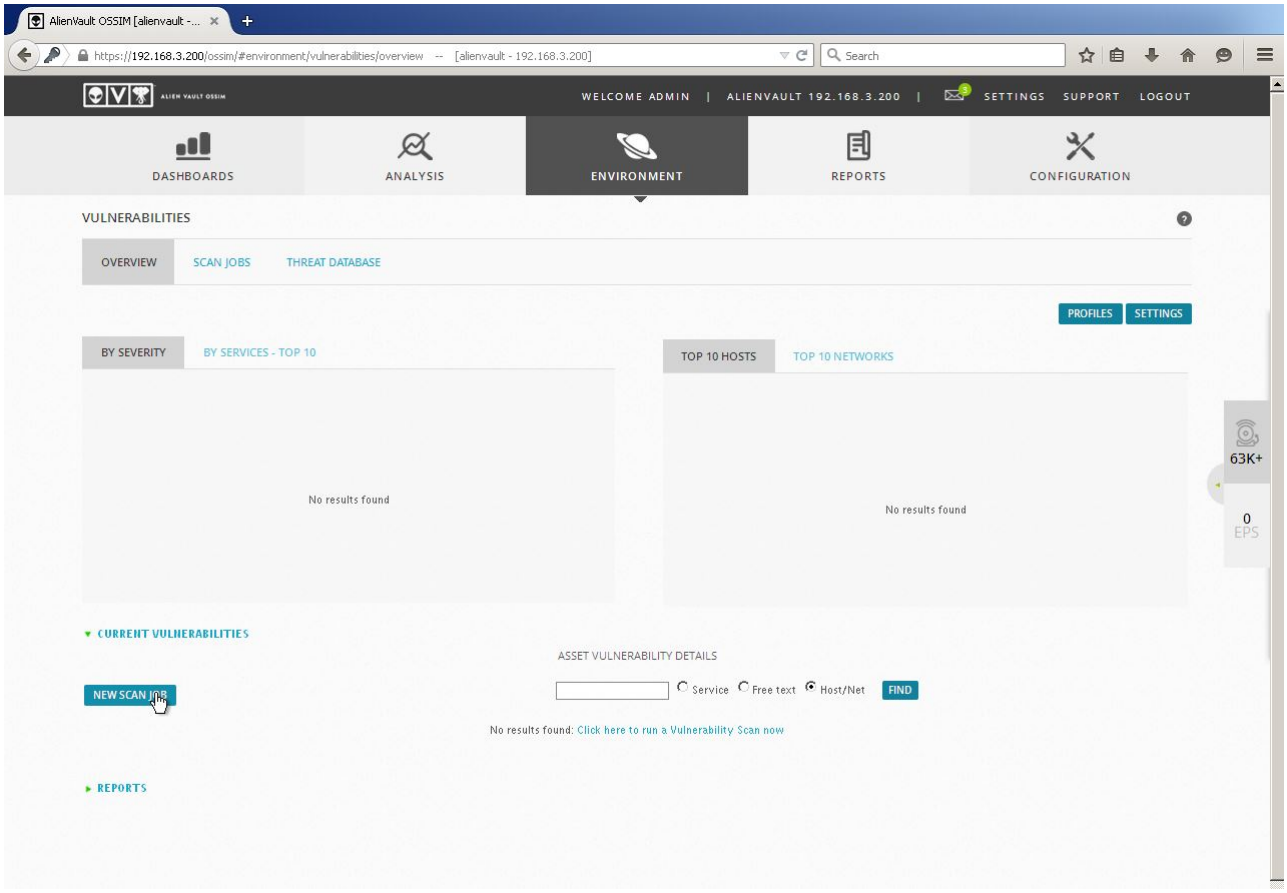


Fig. 10.53: Inicio de un escaneo de vulnerabilidades en OSSIM.

3. Seleccionar los parámetros del escaneo.

4. Una vez seleccionados los parámetros deseados para realizar el escaneo, hacer click en **save** para guardar el trabajo del escaneo.

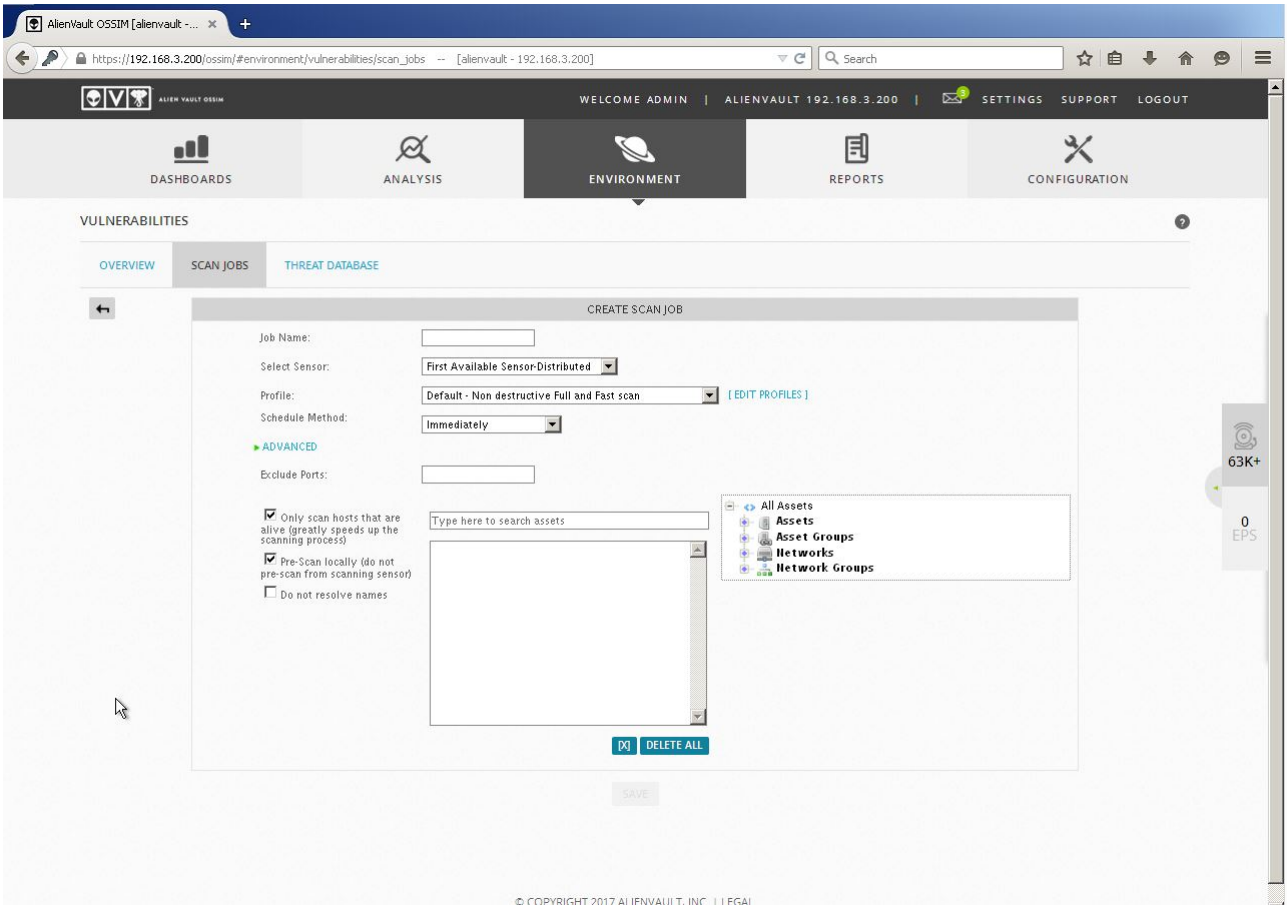


Fig. 10.54: Parámetros necesarios para realizar el escaneo de vulnerabilidades.

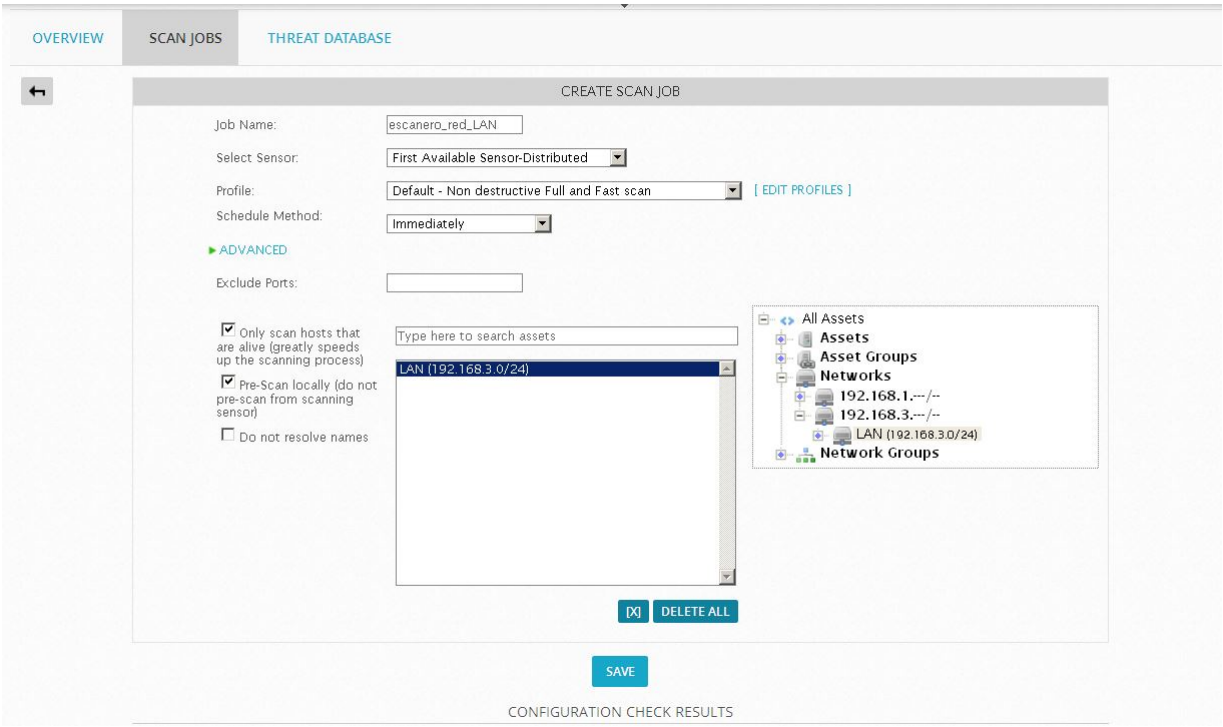


Fig. 10.55: Parámetros configurados para realizar el escaneo de vulnerabilidades en la red LAN.

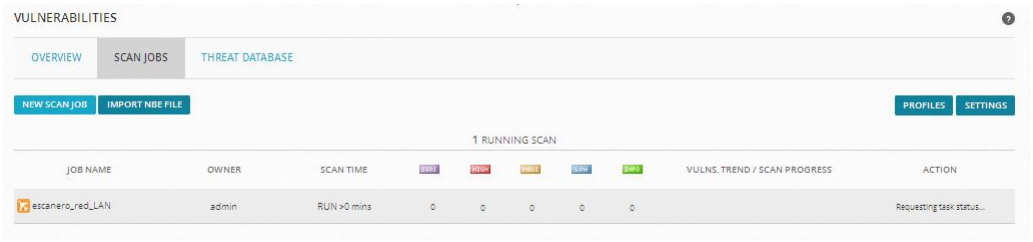


Fig. 10.56: Estado escaneo de vulnerabilidades en estado pendiente de ser realizado.

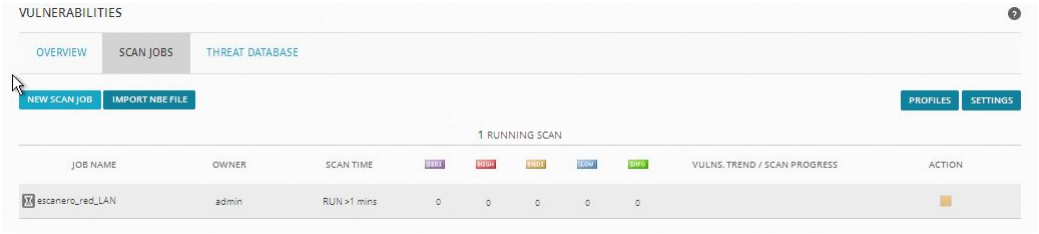


Fig. 10.57: Estado del escaneo de vulnerabilidades en los momentos en los que se esta realizando.

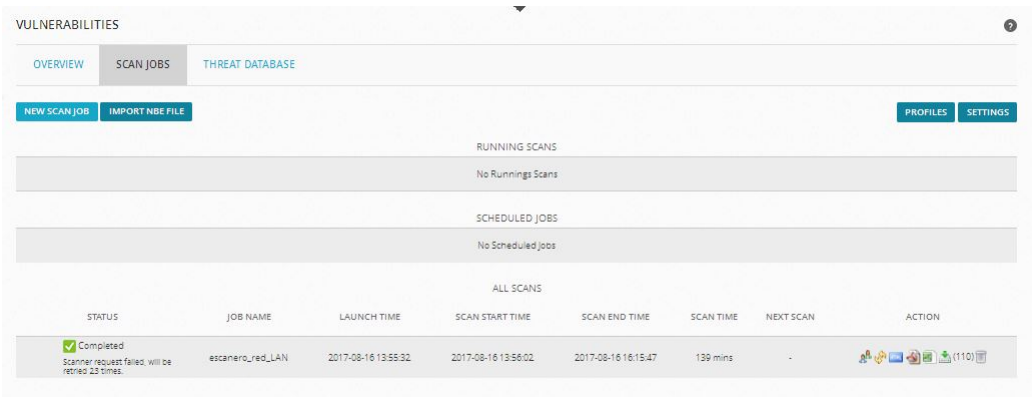


Fig. 10.58: Estado del escaneo de vulnerabilidades una vez completado.

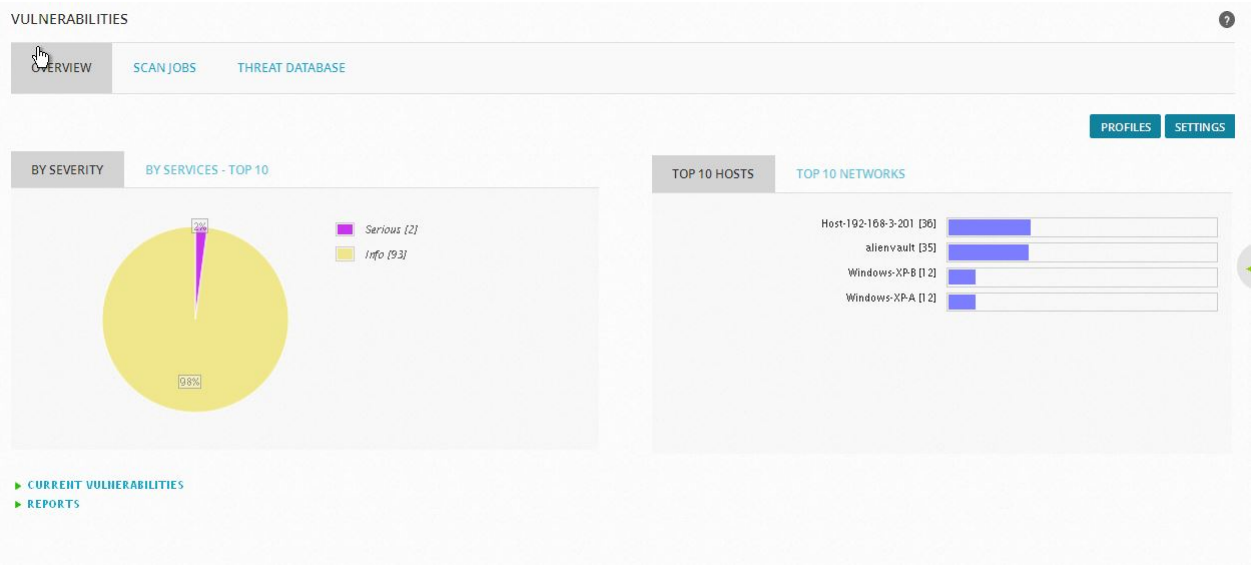


Fig. 10.59: Gráficos de vulnerabilidades encontradas clasificadas por severidad y hosts afectados.

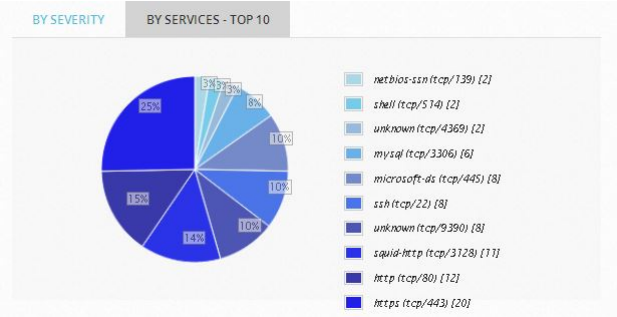


Fig. 10.60: Gráfico de vulnerabilidades encontradas clasificadas por servicio.

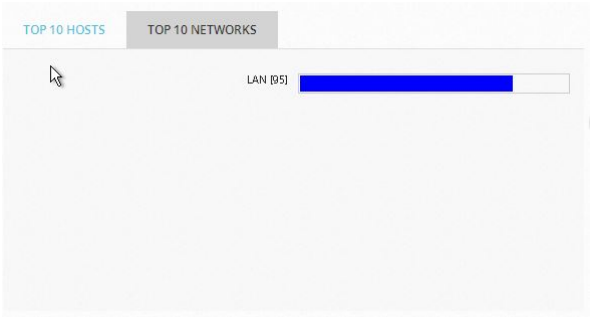


Fig. 10.61: Gráfico de vulnerabilidades encontradas clasificadas por red.

5. Una vez se ha completado el proceso se pueden ver los resultados. Para ver los resultados se navega a **Environment** → **Assets Groups** y se hace click en el activo del que se quieren ver las vulnerabilidades.

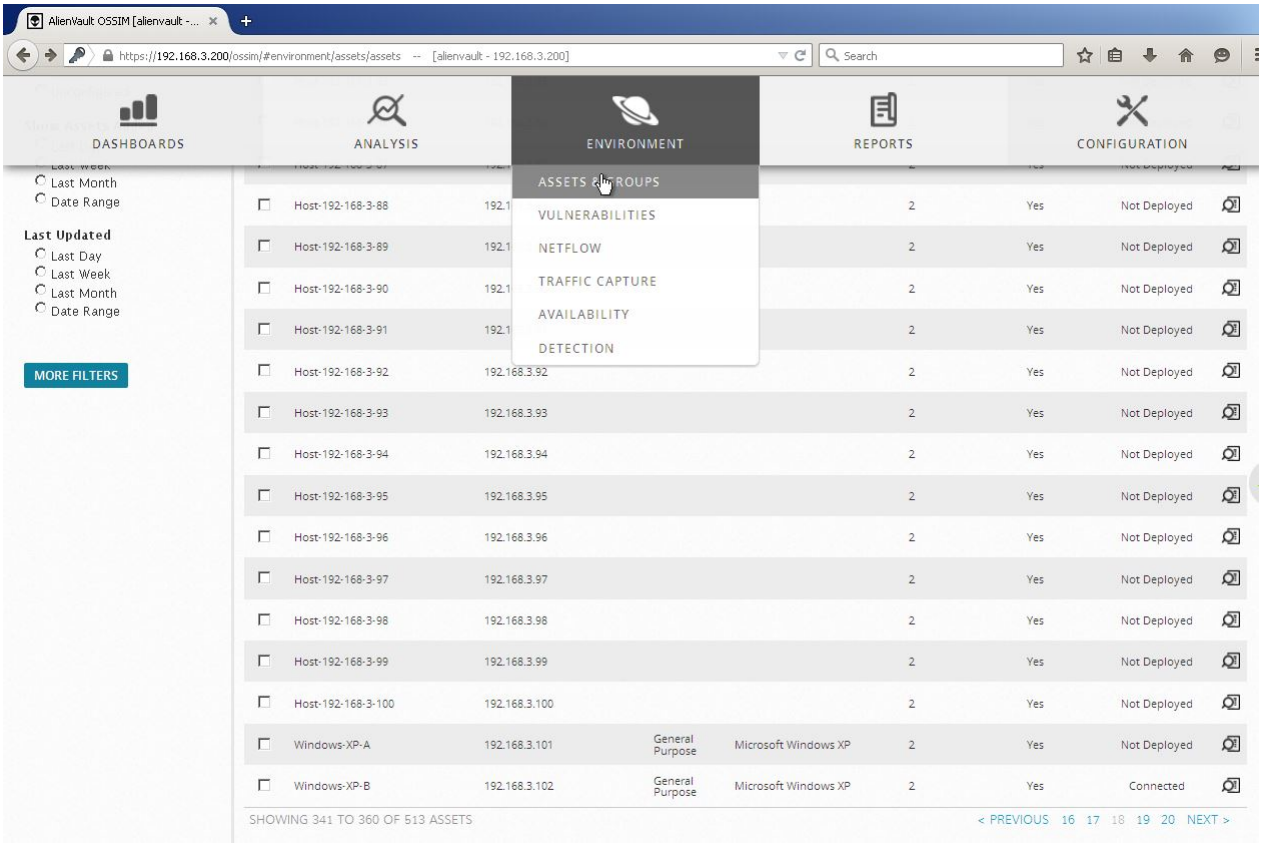


Fig. 10.62: Lista de activos.

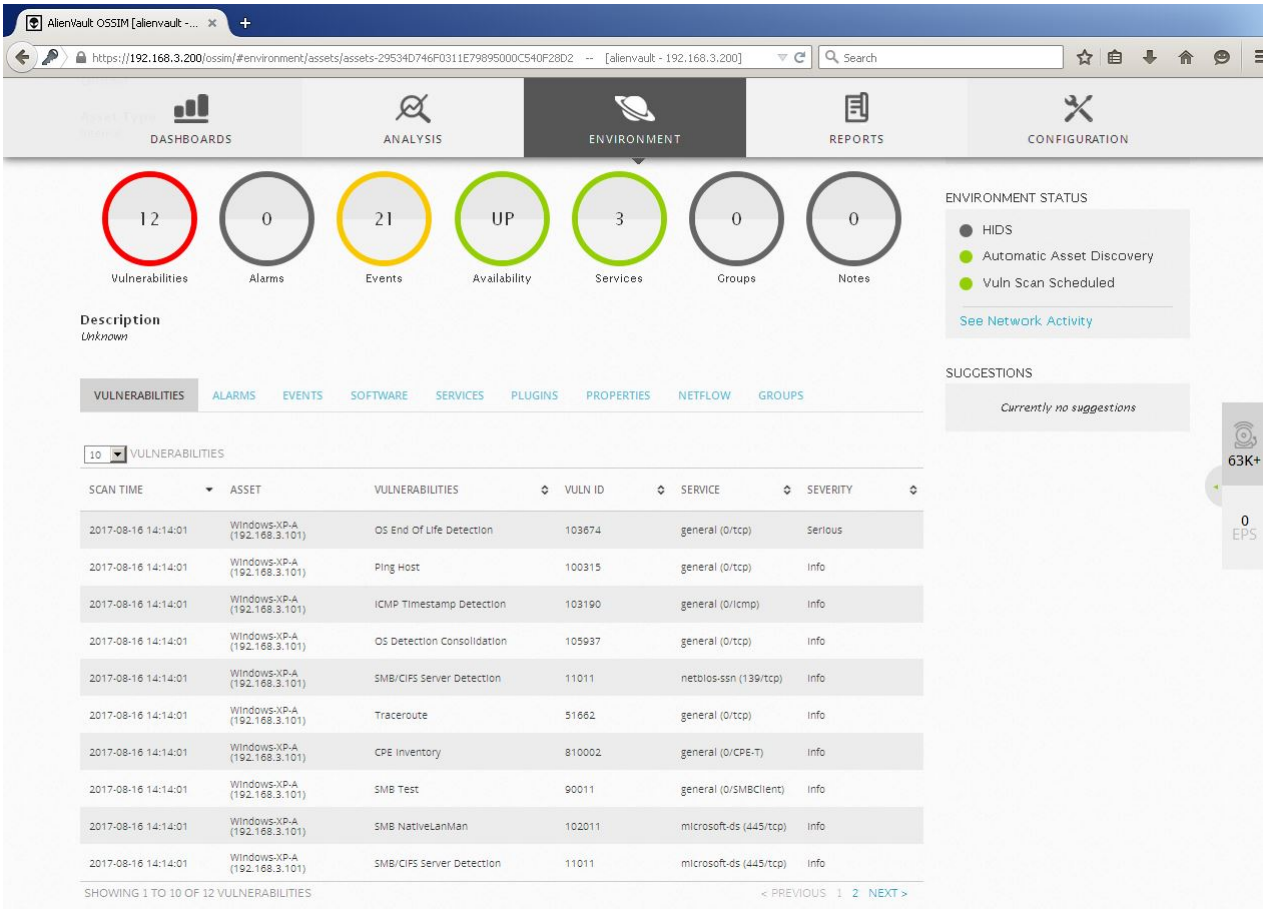


Fig. 10.63: Vulnerabilidades detectadas para el activo seleccionado en la lista de activos.

6. Por último, OSSIM permite exportar los resultados en varios formatos (HTML, PDF, EXCEL) tanto para activos concretos como para escaneos.

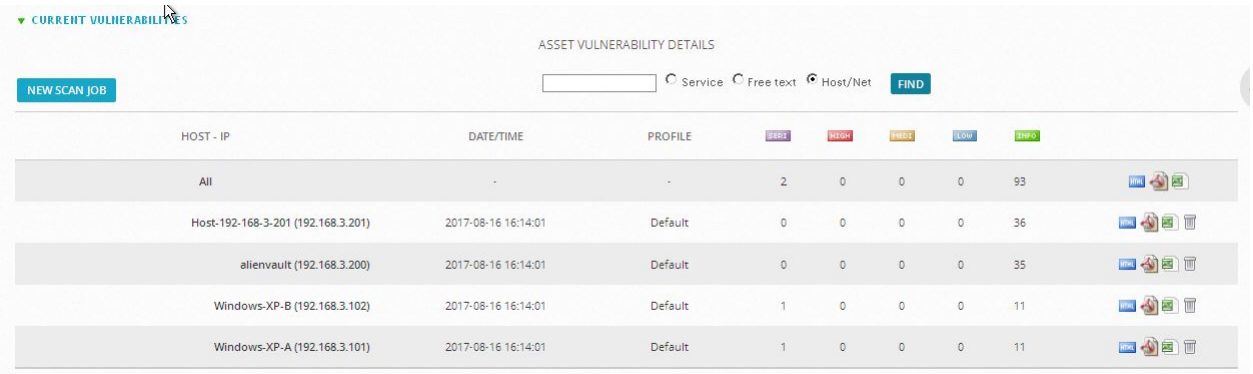


Fig. 10.64: Lista de escaneos realizados para cada activo.

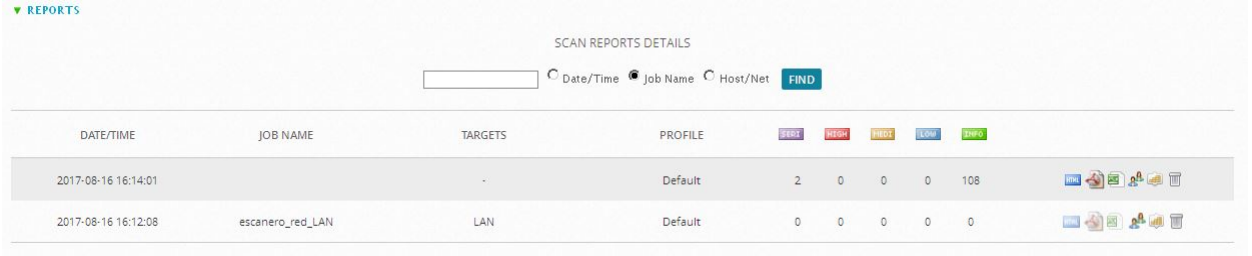


Fig. 10.65: Lista de escaneos realizados.

10.8 ANEXO 8: Configuración correlación en OSSIM.

El proceso seguido es el siguiente:

1. Navegar a **Configuration** → **Threat Intelligence** y hacer click **Directives**. Click **New Directive**.

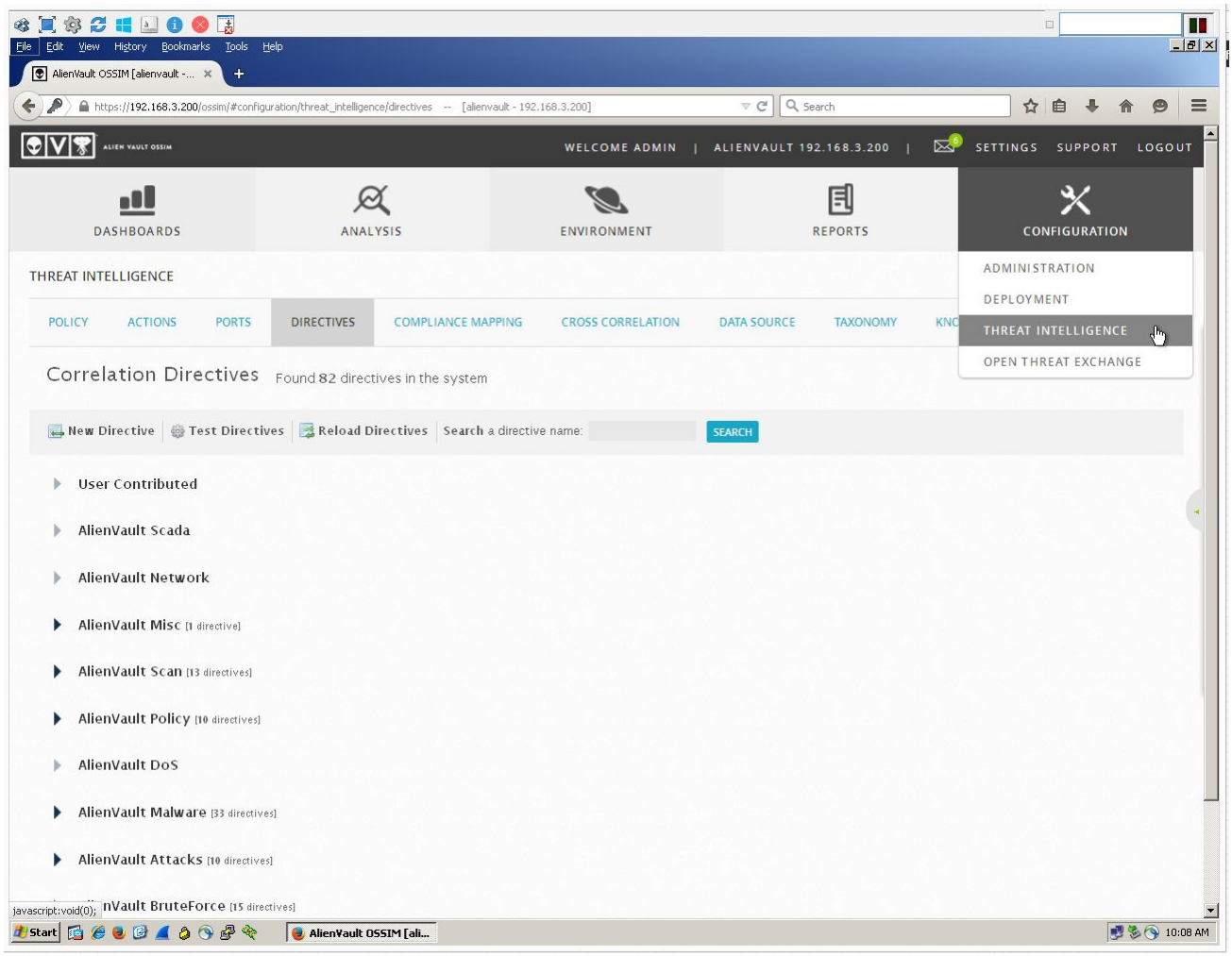


Fig. 10.66: Navegar a la sección de directivas en OSSIM.

Rellenar el formulario como se indica a continuación:

(a) Name for the directive : " Ataque fuerza bruta Servidor Web"
(b) Taxonomy:
(c) Intent: "Reconnaissance & Probing"
(d) Strategy: "WebServer Attack"
(e) Method: "Attack"
(f) Priority: 3
(g) Click **Next**.

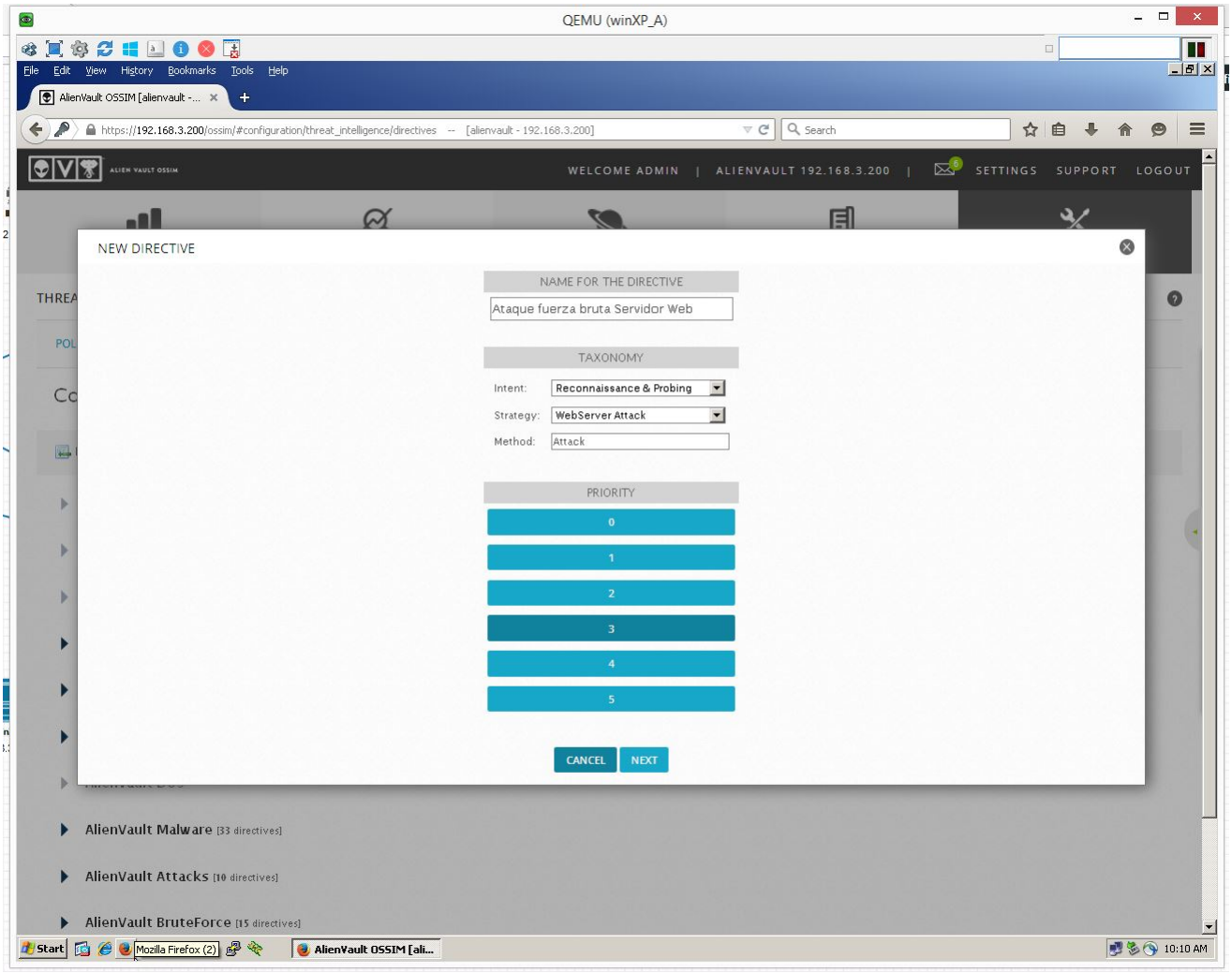


Fig. 10.67: Configuración de la directiva de correlación de primer nivel 1 de 6.

2. Después de configurar la directiva se añadirá una regla de primer nivel:

   (a) En **Name for the Rule**, escribir "Intento de ataque", y click **Next**.

Fig. 10.68: Configuración de la directiva de correlación de primer nivel 2 de 6.

   (b) En **Rule name** → **Plugin**, buscar "ALIENVAULT NIDS".

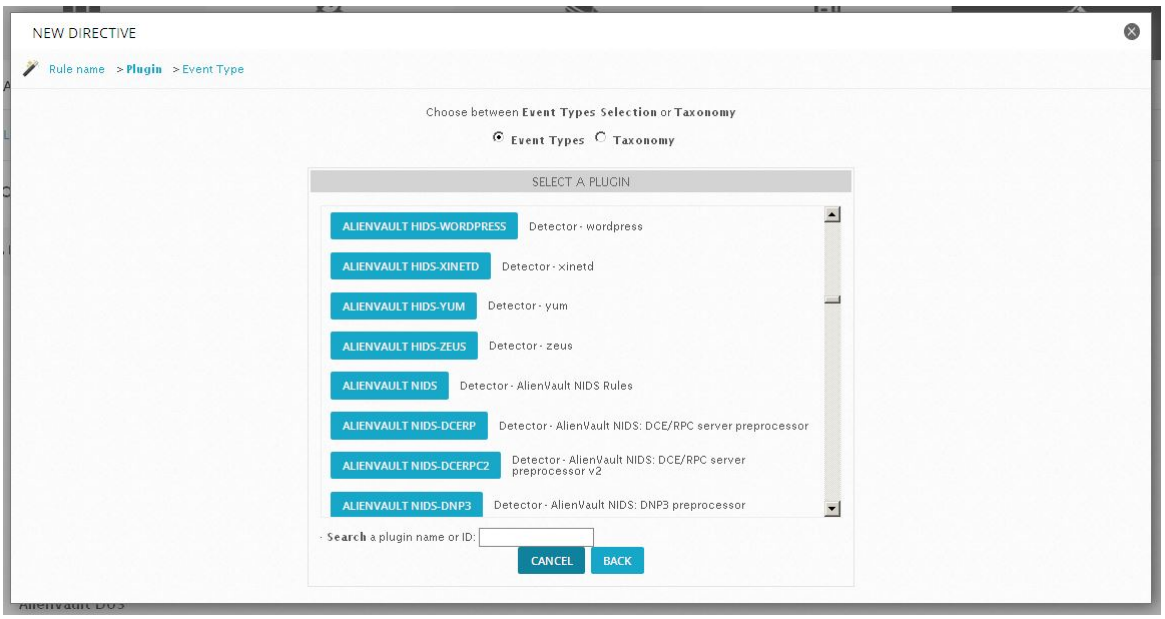


Fig. 10.69: Configuración de la directiva de correlación de primer nivel 3 de 6.

(c) En **Rule name** → **Plugin** → **Event Type** buscar: "ET SCAN LibSSH Based Frequent SSH Connections Likely BruteForce Attack!"

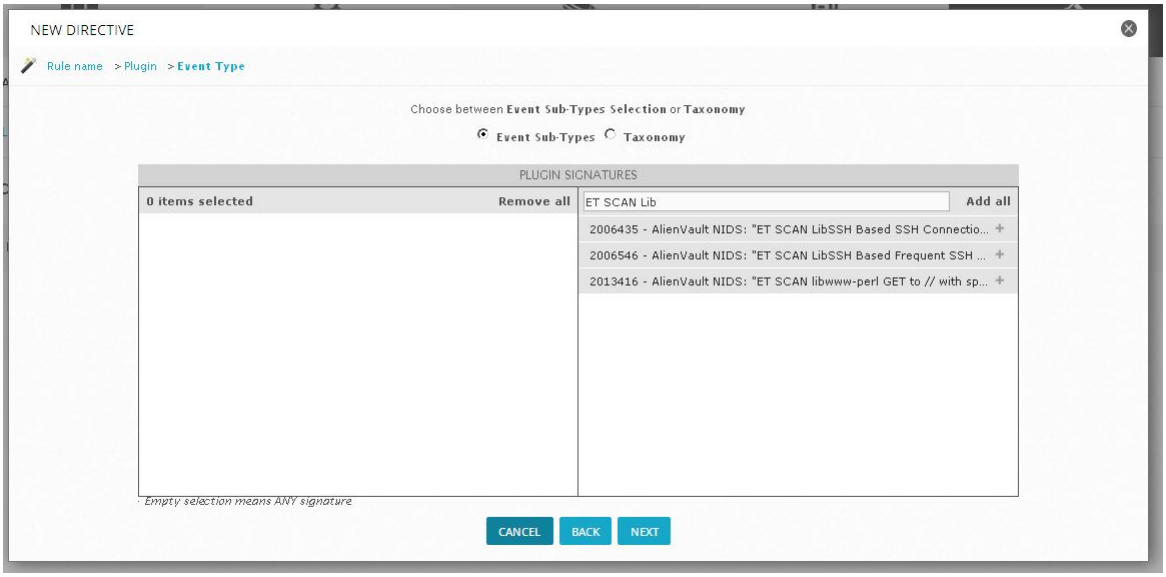


Fig. 10.70: Configuraciónde la directiva de correlación de primer nivel 4 de 6.

(d) En **Rule name** → **Plugin** → **Event Type** → **Network**,

i. Seleccionar el activo destino "Servidor-Web"

ii. Seleccionar el puerto destino 22

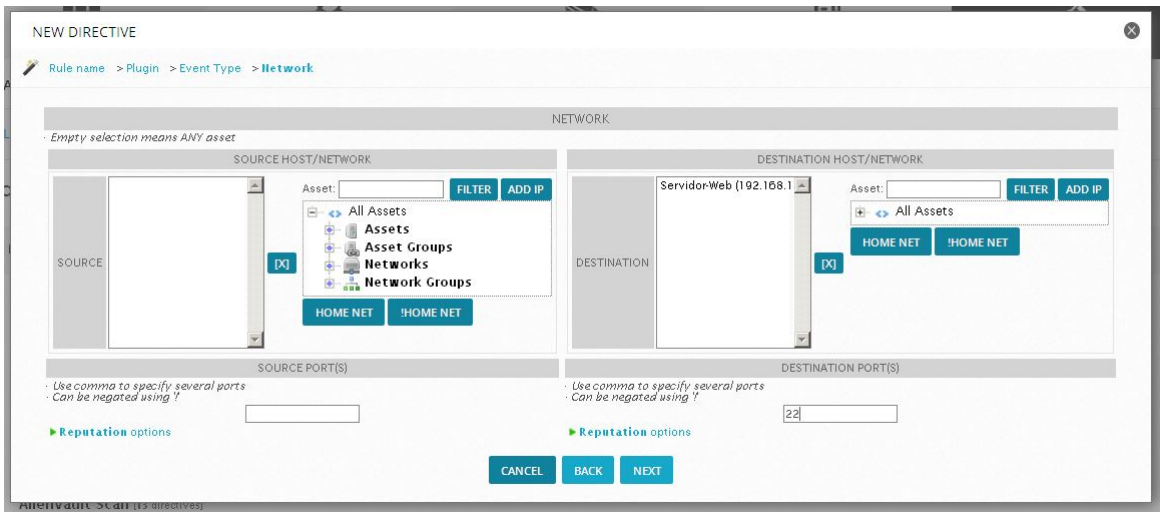


Fig. 10.71: Configuraciónde la directiva de correlación de primer nivel 5 de 6.

(e) En **Rule name** → **Plugin** → **Event Type** → **Network** → **Reliability**, click 1.

Fig. 10.72: Configuración de la directiva de correlación de primer nivel 6 de 6.

(f) Click **Finish**.

Fig. 10.73: Finalización de la inclusión de la directiva a OSSIM.

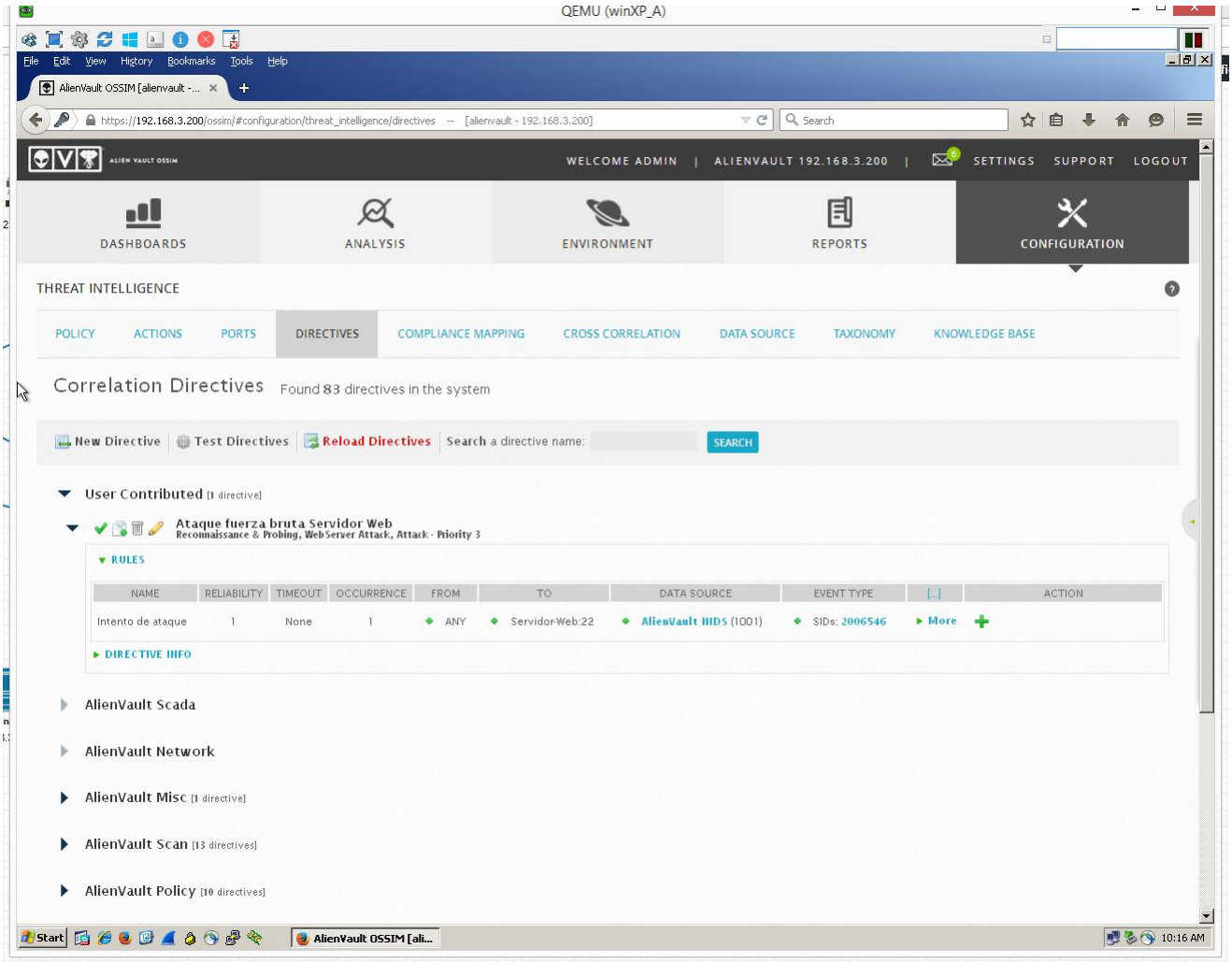


Fig. 10.74: Lista de directivas de correlación en OSSIM.

3. Por último se configurará una regla de segundo nivel

   (a) Click en el signo más (+) verde a la derecha de la regla uno en la columna **Action**.

   (b) Seguir los pasos 1 y 2 al añadir la regla de primer nivel.

   (c) **Rule name** → **Plugin** → **Event Type**, click **Plugin SID from rule of Level 1**.

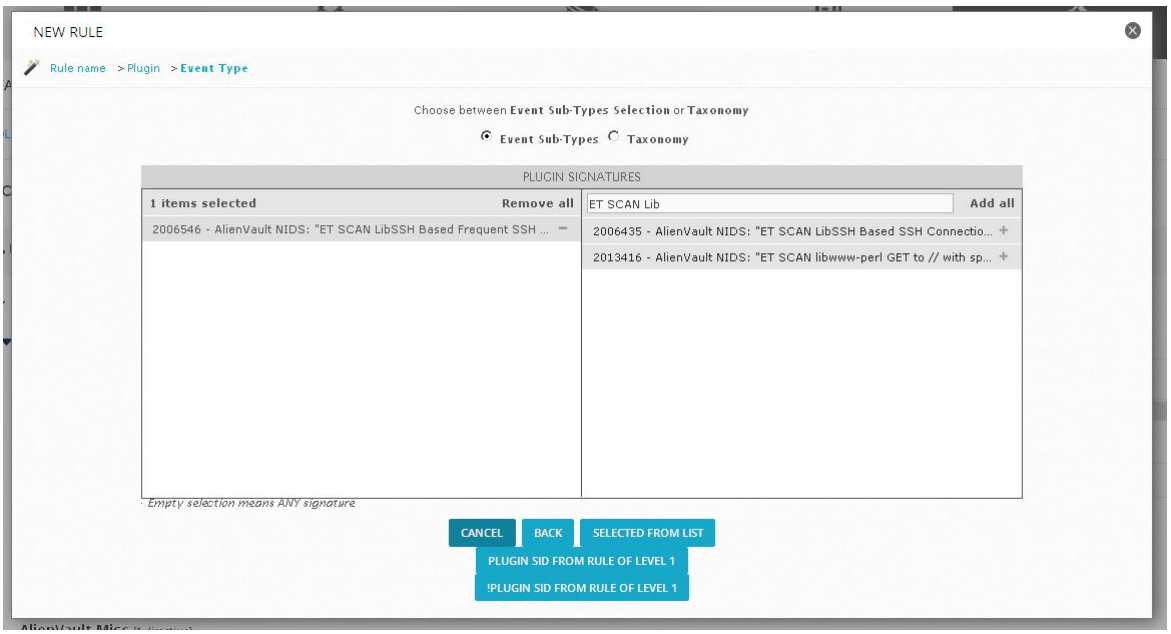


Fig. 10.75: Configuración de la directiva de correlación de segundo nivel 1 de 4.

(d) **Rule name → Plugin → Event Type → Network**.

i. Seleccionar **Source IP from level 1**

ii. Seleccionar **Destination IP from level 1**

iii. Seleccionar **Destination Port from level 1**

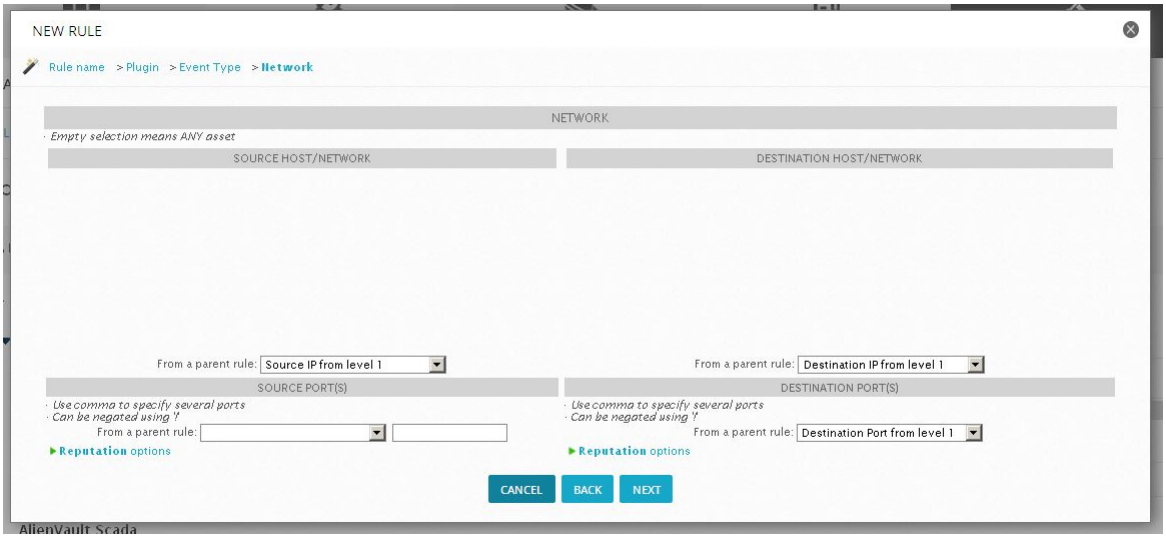


Fig. 10.76: Configuración de la directiva de correlación de segundo nivel 2 de 4.

(e) En **Rule name → Plugin → Event Type → Network → Reliability**, click **+2**.

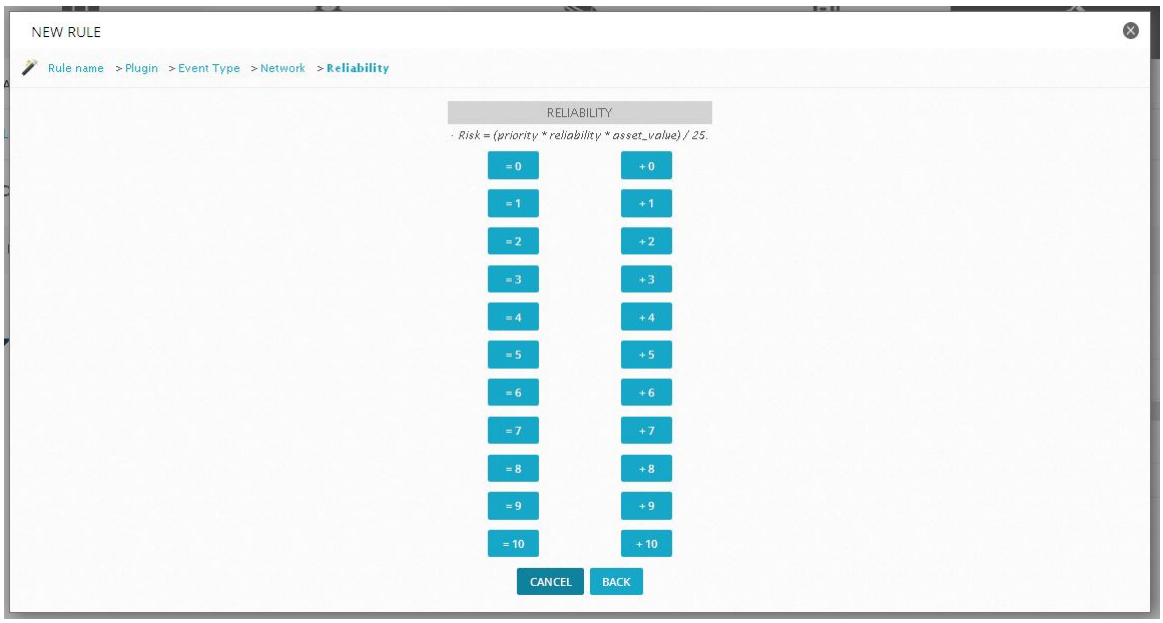


Fig. 10.77: Configuración de la directiva de correlación de segundo nivel 3 de 4.

(f) Click **Finish**.

Fig. 10.78: Configuración de la directiva de correlación de segundo nivel 4 de 4.

(g) En la columna **Timeout**, escribir "60" (segundos) en la segunda regla, y después click **OK**.

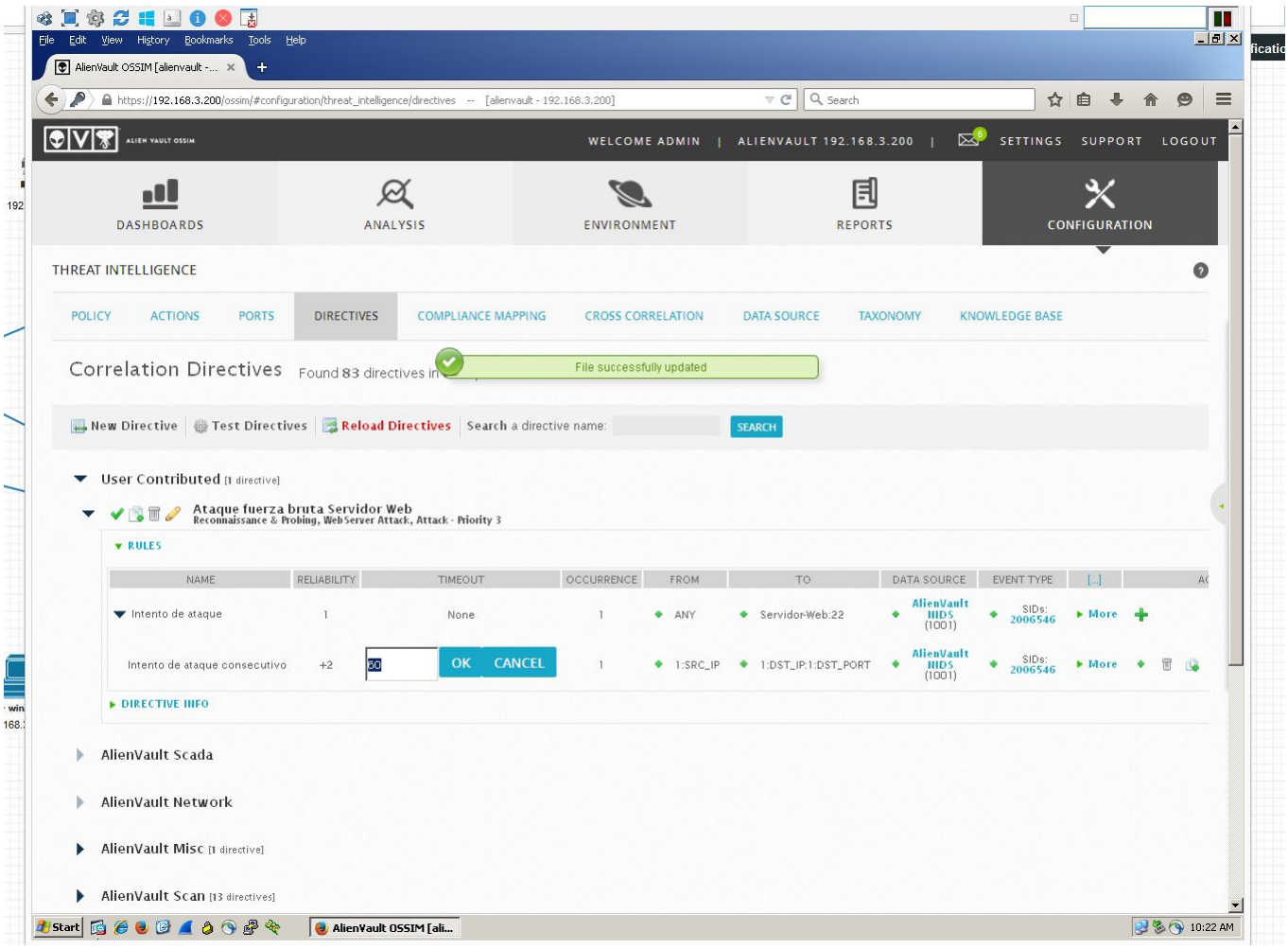


Fig. 10.79: Configuración del Timeout de la directiva de correlación de segundo nivel.

(h) En la columna **Occurrence** , click "1" en la segunda regla, escribir "15",y después click **OK**.

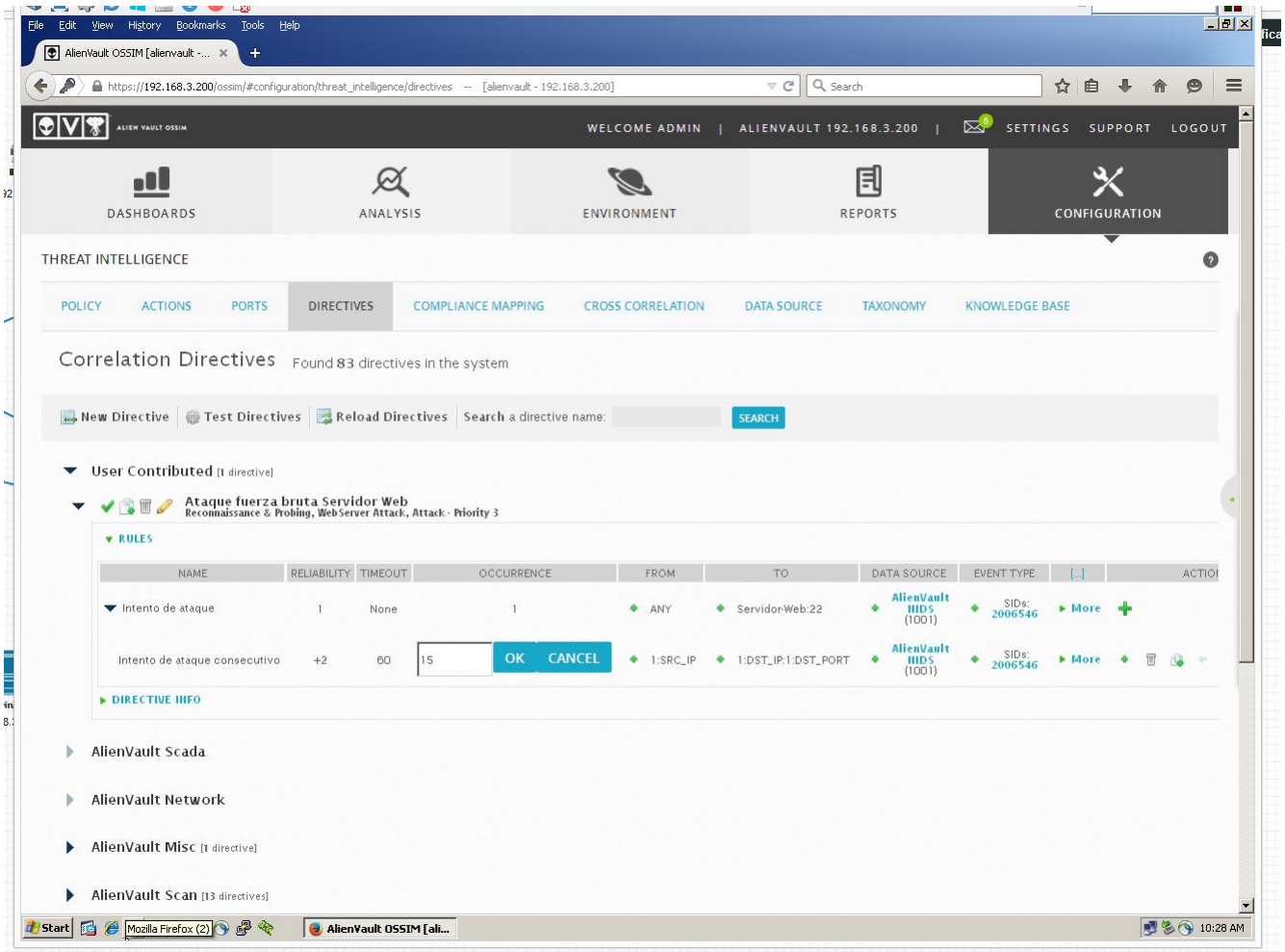


Fig. 10.80: Configuración del número de ocurrencias de la directiva de correlación de segundo nivel.

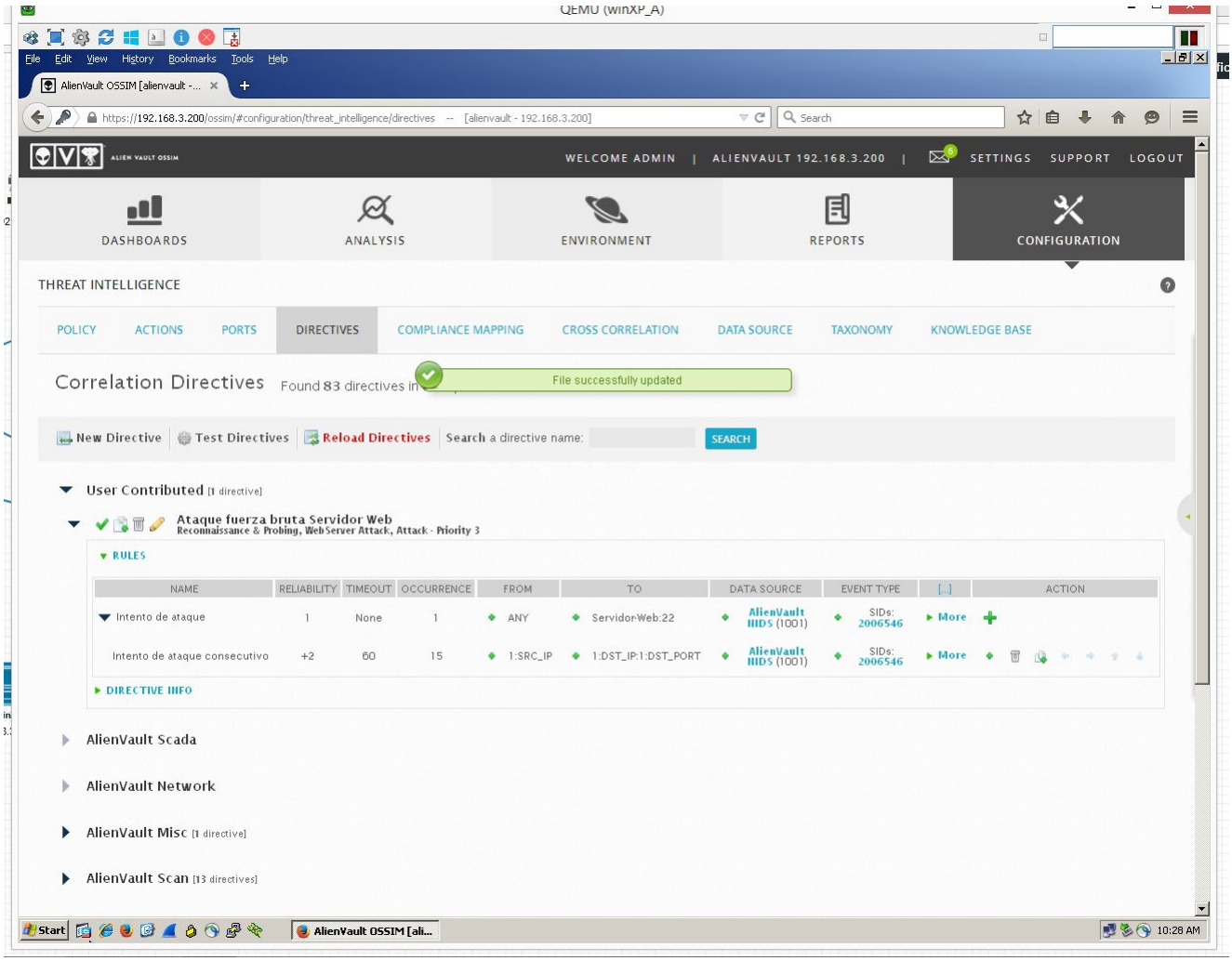


Fig. 10.81: Lista de las directivas de correlación.