TFM: Monitorización de Seguridad con Wazuh

Autor: **Jorge Tomás Guerra**

Tutor: **Pau del Canto Rodrigo**

Profesor: **Víctor Gracia Font**

Plan de Estudios: **Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones**

Fecha de entrega: **27 de diciembre de 2019**

Título del trabajo:	*Monitorización de Seguridad con Wazuh.*
Nombre del autor:	*Jorge Tomás Guerra*
Nombre del colaborador/a docente :	*Pau del Canto Rodrigo*
Nombre del PRA:	*Víctor Garcia Font*
Fecha de entrega (mm/aaaa):	*12/2019*
Titulación o programa:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*Seguridad empresarial*
Idioma del trabajo:	*Español*
Palabras clave	*Monitorización de la seguridad e integridad, detección de amenazas, respuesta a incidentes TI, cumplimiento normativo*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo*

El objetivo de este trabajo de fin de master es realizar una implantación de la herramienta de seguridad Wazuh. Durante este trabajo se ha planificado e implementado su instalación teniendo como premisas que la solución sea escalable y nos permita de forma ágil monitorizar la seguridad de los servidores de una institución.

La metodología seguida para la implementación de este sistema ha sido, analizar las posibilidades que da Wazuh a la hora de instalar, y seleccionar la óptima para un crecimiento de sistemas de monitorización. A su vez, se han analizado los diferentes módulos, tantos los obligatorios, como opciones y se han implementado ciertas características que dan riqueza a la instalación, por ejemplo, integración con VirusTotal, monitorización sin agente, análisis de reglas de bastionado de equipos según reglas de la institución, etc.

Una vez configurados estos sistemas, se han monitorizado una serie de equipos tanto servidores Linux como Windows, así como algún Switch router Cisco.

Finalmente, se han analizado los resultados obtenidos de estos módulos: el estado de seguridad teniendo cuenta parches de seguridad, vulnerabilidades, cambios detectados en ficheros de configuración, estado de políticas de seguridad basándonos en diferentes normativas, así como: NIST, GDPR.

Podemos concluir diciendo que Wazuh es una herramienta que nos da información de seguridad muy completa de los sistemas monitorizados y siempre desde el punto de vista del endpoint que es el que realmente queremos monitorizar, siendo muy

parametrizable todos los módulos permitiendo una gran granularidad en el análisis de seguridad y de alertas.

Abstract (in English, 250 words or less):

The objective of this Master's final dissertation is to implement the Wazuh security tool. During this work, this installation has been planned and implemented taking into account that the solution has to be scalable and allows us to quickly monitor the security of the servers of an institution.

The methodology followed for the implementation is the next. We have analysed the possibilities that Wazuh gives us to install, and we have selected the optimum for a growth of monitoring systems. Further, the different modules that we can find in Wazuh, have been analysed, both mandatory and optional and certain features that give richness to the installation, for example, integration with VirusTotal, monitoring without agent, analysis of equipment bastion rules according to rules of the institution, etc.

Once these systems have been configured, a series of computers have been monitored, both Linux and Windows servers, as well as a Cisco Switch router.

Finally, the results obtained from these modules have been analysed: the security status taking into account security patches, vulnerabilities, changes detected in configuration files, security policy status based on different regulations, as well as: NIST, GDPR.

We can conclude by saying that Wazuh is a tool that gives us very complete safety information of the monitored systems and always from the point of view of the endpoint, all modules being very configurable allowing a great granularity in the analysis of security and alerts.

ÍNDICE

1. Introducción. 8
1.1. Contexto y justificación del trabajo. 8
1.2. Objetivos del proyecto. 10
1.3. Metodología. 11
1.4. Planificación del Trabajo 12
1.4.1. Definición de las tareas. 12
1.4.2. Cronograma: 13
1.5. Estado del arte. 14
1.6. Recursos 16
2. Investigación del mercado y de la aplicación Wazuh. 17
1.1 Definición y características de los SIEMs, HIDS, NIDS,... 17
1.2 Justificación de la elección de Wazuh. 18
1.3 Módulos que integran Wazuh: descripción, Componentes y parametrización. 19
1.3.1 Módulos del Agente Wazuh 21
1.3.2 Módulos del Servidor Wazuh 22
1.3.3 Wazuh, sus módulos de análisis y ejemplos de parametrización 22
1.4 ¿Otras integraciones? 26
3. Implementación y puesta en marcha de Wazuh. 27
3.1. Dimensionamiento de un sistema en producción. 27
3.2. Planificación de la implementación, herramientas de despliegue, etc. 27
3.3. PoC de Wazuh. 28
3.3.1. Instalación de Wazuh. 28
3.3.1.1. Análisis de las opciones de Implementación de la solución Wazuh. 28
3.3.1.2. Instalación de Wazuh desde cero. 30
3.3.1.2.1. Instalación y configuración del Balanceador (Nginx). 31
3.3.1.2.2. Instalación y configuración de los servidores Maestro y Trabajador de Wazuh. 31
3.3.1.2.3. Instalación del servidor de Elastic Stack. 35
3.3.1.3. Cuestiones/problemáticas relacionadas con la instalación. 38
3.3.1.4. Configuraciones extras 38
3.3.1.4.1. Activar detección de vulnerabilidades 38
3.3.1.4.2. VirusTotal 39
3.3.1.4.3. Agentless 40

3.3.1.4.4. Alertas y Reportes vía Email 42
3.3.2. Instalación de los agentes 43
3.3.2.1. Linux 44
3.3.2.2. Windows 44
3.3.2.3. Herramientas de Automatización del despliegue del Agente 44
3.3.2.4. Configuraciones extras en los Agentes. 45
3.3.2.5. Problemas en las instalaciones 46
3.4. Wazuh en producción 47
3.4.1. Vulnerabilidades detectadas en los sistemas Monitorizados. 47
3.4.2. Alerta de Virus con el API de VirusTotal 48
3.4.3. Detección de ataque de fuerza bruta vía SSH 49
3.4.4. Detección de ataque Shellshock y su remediación 50
3.4.5. Ejemplo de análisis de la política de bastionado de la institución: OPENSCAP 52
4. Conclusiones 57
4.1 Resultados de la POC y sus conclusiones 57
4.2 Mejoras y trabajo futuro. 62
5. Trabajos citados 63
6. Glosario 63

IMÁGENES

Imagen 1 - SIEMs Gartner 18
Imagen 2 - Módulos del Agente Wazuh 20
Imagen 3 - Módulos del Servidor Wazuh 21
Imagen 4 - Esquema implementación Wazuh 29
Imagen 5 – Servidores usados para implementar Wazuh. 30
Imagen 6 - Estado del proceso Wazuh en el servidor 32
Imagen 7 - Servidores que forman parte del cluster de Wazuh 34
Imagen 8 - Pantalla de Wazuh en Kibana 37
Imagen 9 - Kibana Servidores de Wazuh 38
Imagen 10 - logs en Kibana de Wazuh del módulo agenless 41
Imagen 11 - Agentless cambio detectado Cisco 6500 41
Imagen 12 - Reporte diario de cambios en ficheros 43
Imagen 13 - Red Hat Satellite 45
Imagen 14 - Grupos de agentes 46
Imagen 15 - Kibaba-vulnerabilidades detectadas 47
Imagen 16 - Log de la detección de Virus (Virustotal) 48
Imagen 17 - Kibana-Wazuh-VirusTotal Alerta 48
Imagen 18 - Interface de las alertas de seguridad. 49

Imagen 19 - Alertas de seguridad- SSH 49
Imagen 20 - Tabla de alertas del endpoint - Shellshock..... 51
Imagen 21 - Imagen de ossec-logtest..... 51
Imagen 22 - Imagen del FW- remediación activa..... 52
Imagen 23 - Herramienta SCAP Workbench para parametrizar reglas..... 53
Imagen 24 - Política de severidad alta no cumplida. 55
Imagen 25 - Bastionado: Lista de las políticas no cumplidas 55
Imagen 26 - SSH accesos fallidos..... 58
Imagen 27 - Monitor de Integridad..... 59
Imagen 28 - Vulnerabilidad no parcheada 60
Imagen 29 - VirusTotal API estadísticas 60

1. INTRODUCCIÓN.

1.1. CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO.

La monitorización de la seguridad, sin duda alguna, ha sido desde siempre una asignatura pendiente en la mayoría de las empresas debido a que siempre se ha premiado más dar servicio al usuario que cerciorarse de que se cumplen ciertas medidas de seguridad. En los años 70 la seguridad se basaba en la confianza en los empleados, pero con la evolución de la tecnología aparecieron nuevos riesgos, que implican nuevas metodologías («La seguridad vista desde sus inicios», 2015).

Además, debemos tener en cuenta que hasta no hace mucho no se incluía dentro de la planificación docente de los profesionales TIC asignaturas relacionadas con la seguridad. Por suerte, hoy en día se ha tomado conciencia de los riegos que tenemos por el uso de las TIC, y que pueden ser muy peligrosas si no se toma conciencia sobre sus riegos y como evítalos o protegerse ante ellos, es por ellos que ha tomado un papel importante todas las disciplinas relacionadas con la seguridad, por ejemplo: programación segura, diseño seguro, sistemas de seguridad, etc. («Día Internacional de la Seguridad Informática», 2019).

Teniendo en cuenta que hasta ahora, no se tomaba tan enserio la seguridad en las TIC y si realizásemos una encuesta sobre qué tipo de herramientas usan para monitorizar la seguridad de sus sistemas, muchos de ellas contestarían que no disponen de este tipo de herramientas, ya son variadas las razones, pero destacaríamos las siguientes: por coste o por simple desconocimiento de cómo implementarlas o usarlas. Debemos entender que no es fácil comprender la información que suelen dar este tipo de herramientas y que normalmente se necesitan de expertos en seguridad TIC para poder analizar estos resultados.

Este tipo de herramientas, las cuales nos ayudan a monitorizar o gestionar la seguridad tic detectando amenazas, vulnerabilidades, malas configuraciones, agujeros de seguridad, etc, son comúnmente denominadas SIEM. Y como es de esperar, podemos encontrarnos todo un ecosistema de herramientas relacionadas con la seguridad en el mercado. De entre todas ellas, existe una que nos es de gran interés que es "Wazuh"[1] y nos aprovecharemos de este TFM para darle visibilidad. Sí que es verdad que, aunque se la podría considerar como un SIEM, está más centrada en la monitorización de seguridad del endpoint (en este caso servidores) mediante el uso de agentes ligeros.

Durante los siguientes puntos dentro de este proyecto entraremos en más profundidad dentro de esta herramienta analizándola y viendo sus características, pero a modo de introducción, podemos destacar sobre ella las siguientes características:

- Análisis de la seguridad: Recopila, agrega, indexa y analiza los datos de seguridad de los servidores monitorizados para ayudar en la detección de intrusiones, amenazas, etc.
- Detección de intrusión: El agente instalado en los servidores escanea estos en búsqueda de malware y anomalías, detectando archivos o procesos ocultos.

[1] https://Wazuh.com/

- Análisis de Log de Datos: Mediante el agente, se envían los logs del sistema operativo y aplicaciones al servidor central donde estos son analizados para realizar un análisis sobre ellos.
- Monitorización de la integridad de Ficheros: Wazuh monitoriza el sistema de archivos, identificando cambios en: el contenido, los permisos, el propietario y los atributos de los archivos.
- Detección de Vulnerabilidades: El agente de Wazuh instalado en el equipo a monitorizar extrae datos de inventario de software y envían esta información al servidor central, donde se correlaciona con bases de datos CVE (Common Vulnerabilities and Exposure)[2] para identificar software vulnerable conocido.

Como ya se ha comentado, el principal objetivo de este proyecto es realizar un análisis de esta herramienta, el cual será complementado con la puesta en marcha de esta herramienta en una institución con cientos de servidores. Sin duda alguna, no contamos con el suficiente tiempo como para acometer toda las instalación y parametrización en todos los servidores, es por ellos que se realizará un POC de unos 10-15 servidores a monitorizar, pero siempre sentando las bases para un sistema que permite acoger la monitorización de todos los servidores de la institución.

[2] http://cve.mitre.org/

1.2. OBJETIVOS DEL PROYECTO.

Los principales objetivos de este trabajo de fin de master los podemos agrupar en tres grandes bloques:

- Estudio de la herramienta Wazuh y descripción de las razones para la implementación de un sistema de seguridad basado en Wazuh con respecto a otras alternativas.
    - Investigar en el mercado herramientas con funcionalidades similares a Wazuh.
    - Analizar un grupo de estas herramientas (las más usadas, conocidas, ...), realizando una comparación con Wazuh, viendo las ventajas y desventajas de cada una de ellas.
    - Analizar los módulos que forman Wazuh y como parametrizarlos.
    - Analizar otros módulos integrables con Wazuh y sus funcionalidades, ampliando las funcionalidades de esta.

- Implantación del sistema Wazuh, en un sistema en producción.
    - Dimensionamiento de Wazhu para un sistema en producción de cientos de Servidores.
    - Estudio de técnicas de despliegue de configuraciones y agentes en diferentes plataformas.
    - Instalación y configuración de Wazhu.
    - Instalación de los agentes en diferentes plataformas.
    - Configuración de los agentes.
    - Analizar posibles integraciones con otro tipo de herramientas.

- Análisis de los resultados obtenidos por Wazuh
    - Análisis de los resultados obtenido para diferentes plataformas.
    - Ventajas e inconvenientes del sistema Wazuh en una implementación real.

1.3. METODOLOGÍA.

El método seguido para implementar este proyecto y como se menciona en el punto anterior consta de tres fases bien diferencias:

- Análisis de la herramienta de Wazuh en comparación con otras herramientas existentes en el mercado. Se compararán estas soluciones con la que se referencia en este proyecto, obtenido una tabla resumen de ventajas e inconvenientes de casa una de ellas. Y especificando las razones para ser Wazuh la elegida para la implementación.

- Se analizará para una solución en un entorno profesional, en este caso formado por unos 500 servidores, cuáles deberían ser las características de los servidores Wazuh, dimensionamiento del servidor o servidores (granjas de servidores), etc.

- En este proyecto se decide realizar la implementación para uno número limitado de servidores para una prueba de concepto (PoC) por parte de la dirección y ver el encaje dentro de la institución. Se instalarán los agentes en un total de 10 servidores, tanto Linux como Windows, así como sistemas sin agente (switches 6500 de Cisco). En esta fase se documentará la instalación, así como la parametrización del sistema. Se describirán todos los pasos dados para la instalación del sistema Wazuh, así como la instalación y configuración de los agentes en diferentes plataformas, tanto Linux como Windows.

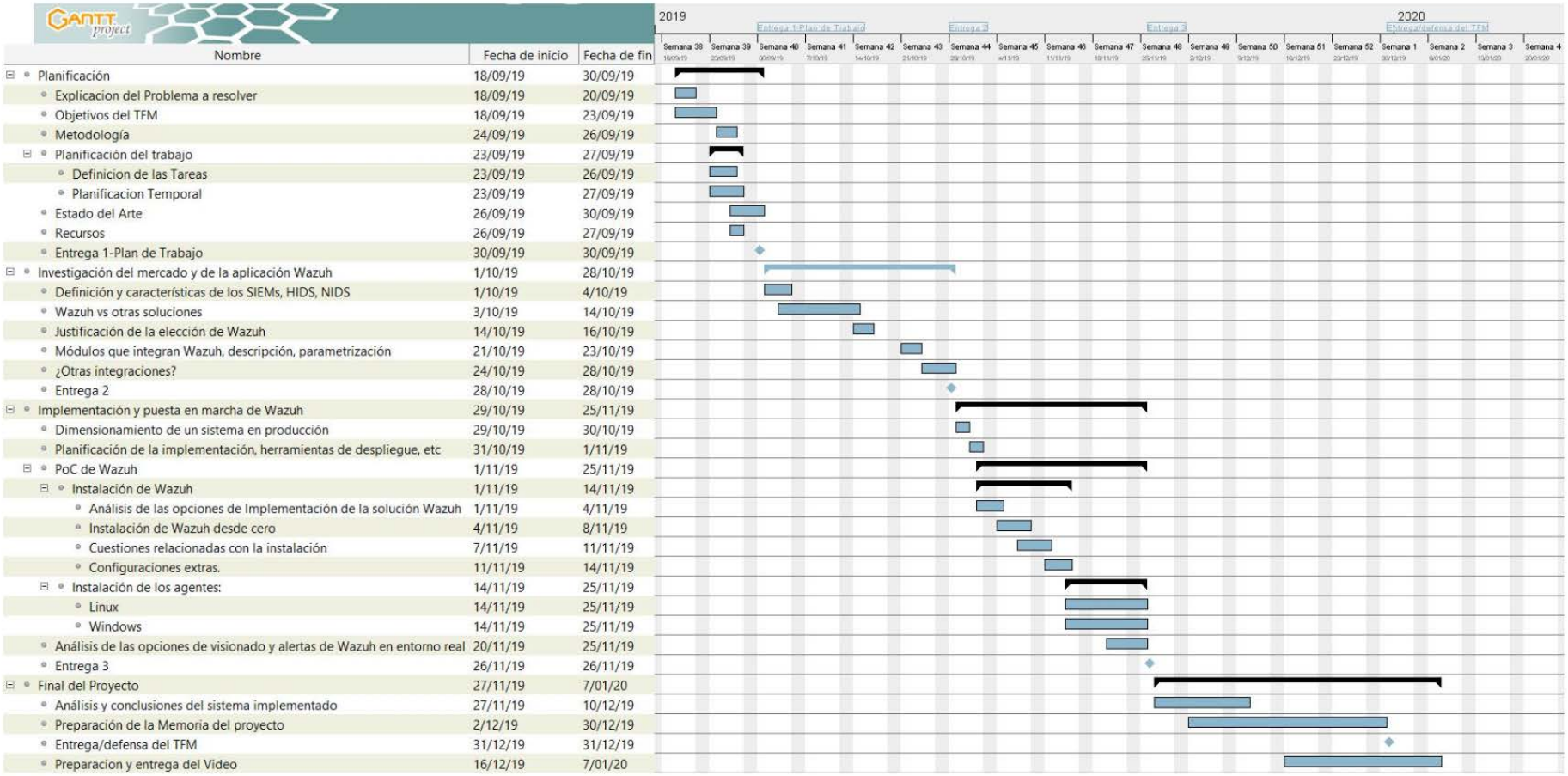
- Análisis de los resultados obtenidos en la implementación de Wazuh en un entorno de producción, donde se podrá analizar en un espacio de tiempo 1 a 2 meses, que es la duración de este proyecto, las vulnerabilidades detectadas, posibles ataques, etc.

1.4. Planificación del Trabajo

1.4.1. Definición de las tareas.

1. Planificación.
   1.1. Explicación del Problema a resolver
   1.2. Objetivos del TFM
   1.3. Metodología
   1.4. Planificación del trabajo
      1.4.1. Definición de las tareas.
      1.4.2. Planificación temporal
   1.5. Estado del arte.
   1.6. Recursos
   1.7. Entrega 1 – Plan de Trabajo
2. Investigación del mercado y de la aplicación Wazuh.
   2.1. Definición y características de los SIEMs, HIDS, NIDS.
   2.2. Wazuh vs otras soluciones.
   2.3. Justificación de la elección de Wazuh.
   2.4. Módulos que integran Wazuh, descripción, parametrización...
   2.5. ¿Otras integraciones?
   2.6. Entrega 2.
3. Implementación y puesta en marcha de Wazuh.
   3.1. Dimensionamiento de un sistema en producción.
   3.2. Planificación de la implementación, herramientas de despliegue, etc
   3.3. PoC de Wazuh
      3.3.1. Instalación de Wazuh.
         3.3.1.1. Análisis de las opciones de Implementación de la solución Wazuh.
         3.3.1.2. Instalación de Wazuh desde cero.
         3.3.1.3. Cuestiones relacionadas con la instalación.
         3.3.1.4. Configuraciones extras.
      3.3.2. Instalación de los agentes:
         3.3.2.1. Linux
         3.3.2.2. Windows
      3.3.3. Análisis de las opciones de visionado y alertas de Wazuh en entorno real
      3.3.4. Entrega 3.
4. Final del Proyecto
   4.1. Análisis y conclusiones del sistema implementado.
   4.2. Preparación de la Memoria del proyecto.
   4.3. Entrega/defensa del TFM

1.4.2. CRONOGRAMA:



Nombre	Fecha de inicio	Fecha de fin
Planificación	18/09/19	30/09/19
Explicacion del Problema a resolver	18/09/19	20/09/19
Objetivos del TFM	18/09/19	23/09/19
Metodología	24/09/19	26/09/19
Planificación del trabajo	23/09/19	27/09/19
Definicion de las Tareas	23/09/19	26/09/19
Planificacion Temporal	23/09/19	27/09/19
Estado del Arte	26/09/19	30/09/19
Recursos	26/09/19	27/09/19
Entrega 1-Plan de Trabajo	30/09/19	30/09/19
Investigación del mercado y de la aplicación Wazuh	1/10/19	28/10/19
Definición y características de los SIEMs, HIDS, NIDS	1/10/19	4/10/19
Wazuh vs otras soluciones	3/10/19	14/10/19
Justificación de la elección de Wazuh	14/10/19	16/10/19
Módulos que integran Wazuh, descripción, parametrización	21/10/19	23/10/19
¿Otras integraciones?	24/10/19	28/10/19
Entrega 2	28/10/19	28/10/19
Implementación y puesta en marcha de Wazuh	29/10/19	25/11/19
Dimensionamiento de un sistema en producción	29/10/19	30/10/19
Planificación de la implementación, herramientas de despliegue, etc	31/10/19	1/11/19
PoC de Wazuh	1/11/19	25/11/19
Instalación de Wazuh	1/11/19	14/11/19
Análisis de las opciones de Implementación de la solución Wazuh	1/11/19	4/11/19
Instalación de Wazuh desde cero	4/11/19	8/11/19
Cuestiones relacionadas con la instalación	7/11/19	11/11/19
Configuraciones extras.	11/11/19	14/11/19
Instalación de los agentes:	14/11/19	25/11/19
Linux	14/11/19	25/11/19
Windows	14/11/19	25/11/19
Análisis de las opciones de visionado y alertas de Wazuh en entorno real	20/11/19	25/11/19
Entrega 3	26/11/19	26/11/19
Final del Proyecto	27/11/19	7/01/20
Análisis y conclusiones del sistema implementado	27/11/19	10/12/19
Preparación de la Memoria del proyecto	2/12/19	30/12/19
Entrega/defensa del TFM	31/12/19	31/12/19
Preparacion y entrega del Video	16/12/19	7/01/20

1.5. ESTADO DEL ARTE.

Desde la aparición del PC y de los servidores, la seguridad de ellos ha sido un punto muy importante. Si nos referimos al pasado, hace 20 o 30 años, la seguridad TI se tomaba como algo secundario, sobre todo porque no había muchos servidores o PC en las empresas. Hoy en día, con el abaratamiento del hardware, así como con la virtualización, ha llevado a las empresas de tener unos pocos servidores, los cuales estaban muy controlados a tener cientos o miles de ellos, cada uno con una aplicación diferente, sistema operativo, versión de este etc.

Como consecuencia de esta proliferación de máquinas, la gestión de la seguridad de todo este parque de servidores se ha complicado exponencialmente, si además tenemos en cuenta que el número de personas dedicadas a mantener todo este parque de equipos es el mismo o ha crecido muy poco, no al ritmo de los servidores, se ve la necesidad de incorporar herramientas que ayuden a los técnicos a mantener esta seguridad y poder monitorizarlos («Los pros y los contras de la virtualización», 2009).

Sin duda alguna, es necesario incorporar herramientas que ayuden a los técnicos TI encargados de la seguridad para manejar toda esta cantidad de información que tienen. Estas herramientas deben darle la posibilidad de controlar o gestionar ciertos aspectos claves que son necesarios para mantener la seguridad, así como:

- versionado de los sistemas,
- versionado de las aplicaciones
- análisis de bugs reconocidos
- análisis de logs en busca de ataques
- etc

Este tipo de herramientas, en modo genérico, son más conocidas como herramientas SIEM (Security Information and Event Management), pero si entramos en más profundidad en el análisis de estas herramientas, podemos encontrarnos que nos aparecen más siglas para catalogar, así como HIDS, NIDS, etc. (Concha, 2019). Las herramientas que pertenezcan a un grupo u otro se caracterizarán por una serie de servicios que dan, así como que tipo de análisis de seguridad realizan, alcance, etc, a modo de ejemplo, las HIDS, se centran en el host, el equipo, las NIDS, en la red. Este tipo de catalogación será analizado en más profundidad en los siguientes puntos de este TFM.

Pero para situarnos y entender de qué tipo de herramienta estamos hablando, comenzaremos por explicar que es lo que entendemos por SIEM. Estas herramientas, los SIEMs, son un sistema que recopila los eventos de los diferentes elementos que tenemos en el área TIC en una organización, así como: servidores, firewalls, sondas, etc. Estos generan gran información que pude abrumar a cualquiera, y gracias a este tipo de herramientas, se puede recolectar toda esta información y analizarla, basándose en patrones, reglas etc., de esta forma, mediante estos patrones y los logs recibidos de los diferentes sistemas podemos detectar ataques, malas configuraciones,

agujeros de seguridad, etc. Resumiendo, sobre las características de estas herramientas, vemos un nexo de unión, características comunes a todas ellas, de las que destacamos las siguientes:

- Centralización de los logs
- Centralización de las vistas resumen (análisis de diferentes logs en búsqueda de patrones)
- Categorización de las amenazas en base a patrones
- Sirven para documentar los eventos o los ataques ya que recopilan todos los logs
- …

Pero estos sistemas (SIEM) no siempre y no todos cubren todas las funcionalidades deseadas, debemos tener en cuenta que según ha pasado el tiempo, estos sistemas han ido evolucionado. Inicialmente los sistemas SIEM, fueron desarrollados para reducir la cantidad de datos (no validos) que se recibían de los diferentes sistemas conectados a la red. Además, estos primeros sistemas tendían a ser muy complejos de parametrizar y configurar, por lo que no fueron bien acogidos por los responsables de las TIC, y no solo eso, la mayoría de ellos se utilizaban, una vez que se había sufrido un taque, para intentar recopilar la información que almacenaba en ellos para realizar el análisis de seguridad, pero siempre a posteriori, es decir se usaban más bien como un repositorio de logs para buscar indicios una vez sufrido un ataque.

Posteriormente se vio que con toda la información que recibían, podrían usarse para detectar “Cuando” uno estaba siendo atacado. De esta forma se añadieron motores de análisis en los sistemas SIEM para detectar si se está siendo atacado o no, analizando los logs enviados.

Finalmente, los SIEMs actuales se basan en el aprendizaje automático realizando análisis estadísticos y sobre todo usando el Big data que se comparte, o es propietario si es una herramienta de pago, para analizar el comportamiento que se entiende como “normal” y cuando algo sale de ese parámetro de normalidad, lanzar una alerta o incluso actuar con los sistemas de seguridad para proteger a todos los equipos (Ramiro, 2018).

En este caso nos basaremos en la herramienta Wazuh, un sistema de seguridad, más orientado al análisis del endpoint, mediante agente. En contraposición, tenemos los SIEMS que realizan análisis basándose en los que escuchan en la red o de los logs enviados por los sistemas de seguridad, así como Firewalls, IDSs, etc. Durante este trabajo, se irán reflejado las ventajas e inconvenientes que tiene esta solución y otras.

1.6. RECURSOS

Para llevar a cabo este trabajo de fin de mater, además del tiempo invertido el cual está planificado en los puntos anteriores, necesitaremos algunos recursos materiales los cuales tiene un coste asociado y una serie de servicios, estos son listamos a continuación:

- Ordenador personal: Será usado para redactar la memoria, acceder a las máquinas virtuales, acceso a internet para búsqueda de información, etc
- Servidor: será usado para implementar las diferentes máquinas virtuales que se necesiten en este proyecto, en este caso servidor con 2 CPUs 32 Gb de memoria RAM y con el hipervisor ESXi.
- Conexión a Internet: para descargar los paquetes necesarios para la instalación, búsqueda de información, etc
- Red Hat Satellite
- Directorio Activo
- Switch/Router Cisco 6500
- Servidores: Windows Server y Red Hat

2. INVESTIGACIÓN DEL MERCADO Y DE LA APLICACIÓN WAZUH.

1.1 DEFINICIÓN Y CARACTERÍSTICAS DE LOS SIEMS, HIDS, NIDS,...

Al adentrarnos dentro del mundo de la seguridad en los sistemas informáticos, búsquedas de vulnerabilidades, detección de ataques, siempre aparecen las siglas SIEM, pero ¿Qué es un SIEM?.

Comenzaremos este punto por definir exactamente qué es lo que conocemos como SIEM (Security Information and Event Management), es decir, la gestión de eventos e información de seguridad.

El origen de la creación de los SIEMs viene impulsado por el modo de trabajo que se tenía en el pasado con respecto al mantenimiento de la seguridad de los servidores y la detección de ataques que estos sufrían, ya que la mayoría de las veces esta tarea de seguridad se realizaba en cada máquina y muchas veces solo por el administrador de esa máquina (Mundhada, 2019).

Con el tiempo se ha visto la necesidad de centralizar toda esta información que tienen los servidores, elementos de red, firewalls, IDs etc. en una sola consola, donde se puede analizar toda la información que reciben y poder contrastar en la búsqueda de patrones fuera de lo común y tomar las acciones necesarias para mitigar un ataque, resolver un problema de configuración de seguridad, etc.

Es verdad que muchas veces es tanta la información que envían todos estos equipos que termina siendo un fracaso el montar un SIEM, sobre todo si no se parten de ciertas reglas de inteligencia que ayuden al analista de seguridad a obtener el dato importante de miles de datos recibidos. Es en este momento donde entran en juego los SIEMs actuales o herramientas de seguridad, que basándose en patrones predefinidos por forma de trabaja diferentes ataques, muestreos o experiencia de otros, son capaces de diferenciar entre todo los logs enviados patrones y con un alto porcentaje de acierto alertar de un ataque.

Así mismo, estas herramientas pueden ser usadas para comprobar que la política de una empresa, en cuanto medidas de seguridad tic, se están cumpliendo, por ejemplo, pensemos en política de passwords, parches de seguridad, cuentas de usuarios privilegiadas, etc.

Pero no todas las herramientas que encontremos en el mercado cumplen todas las funcionalidades, unas de ellas están más orientadas al análisis de eventos en la red, otras más relacionadas con el endpoint, etc. Es de aquí donde aparecen el resto de siglas, como HIDS (Host Intrusion Detections Systems), NIDS (Network Intrussion Detections Systems), IPS (Intrusion Prevention Systems), ... ya que cada uno a de ellas estaría más enfocada a una casuística dada.

Wazuh estaría enfocada o catalogada dentro del grupo HIDS, pero podemos encontrar referencias donde la enmarcan dentro de los SIEMs por la granularidad que tiene de recolectar logs de otros sistemas y mediante plugings poder tratarlos todos en conjunto aumentando el nivel de seguridad que se obtiene, convirtiendo esta herramienta en un SIEM.

Como ya se ha comentado podemos encontramos múltiples herramientas para realizar el análisis de la seguridad TI, a modo de ejemplo nombraremos las más destacadas, siempre basándonos en el cuadrante mágico de Gartner del 2018[3]:

Imagen 1 - SIEMs Gartner

Debemos mencionar que la mayoria de ellas son herraminetas de pago, aunque algunas de ellas tiene versiones gratuitas, eso si limitadas en caracteristicas, a modo de ejmplo: AlienVault deja de modo gratuita su versión llamada OSSIM.

Ejemplos de otras herraminetas, en este caso gratuitas tenemos las siguientes: SIEMonster, Apache Metron, AlienVault OSSIM, OSSEC, Wazuh, …

1.2 JUSTIFICACIÓN DE LA ELECCIÓN DE WAZUH.

Tras analizar diferentes soluciones que podemos encontrar en el mercado y teniendo en cuenta que se buscaba un sistema que no implicase ningún coste económico en esta primera fase, descartamos todos los sistemas de pago. Además, se buscaba un sistema que aparte de poder correlar logs o información dada por los sistemas de seguridad de la institución como firewalls o NGFW, IDSs, IPS, sondas etc., se buscaba un sistema que tuviese información de primera mano, es decir de los propios activos que se desean proteger. Es por ello que entre todas las soluciones se ha elegido implementar este sistema de seguridad con Wazuh + ELK.

A continuación, listaremos todos los puntos por lo que se ha decidido optar por esta herramienta para monitorizar la seguridad de los servidores de nuestra institución:

[3] https://www.gartner.com/reviews/customers-choice/security-information-event-management

- Herramienta gratuita, no debemos olvidarnos que la mayoría de las herramientas que más publicidad tiene, son herramientas caras y que tiene una dependencia del fabricante, tanto en el número de equipos a monitorizar como de funcionalidades nuevas que puedan aparecer o integraciones con terceros.
- API y módulos abiertos, es decir, desde la concepción de esta herramienta permite la integración con otros sistemas que ya existe en el mercado o nuevos que se deseen integrar para complementarla.
- Integración directa con la pila ELK. Debemos tener en cuenta que la solución ELK es uno de los sistemas que más se está implantado tanto para la regida de logs, como para en análisis de ellos. Además, si tenemos en cuenta que muchas empresas o instituciones ya tiene implementado algún servicio de ELK donde suelen enviarse los logs de los sistemas de seguridad, como firewalls, IDS, IPS, antivirus, etc., esta integración entre Wazuh y ELK es automática, consiguiendo tener un auténtico SIEM.
- Información detallada de los activos a proteger. Como se ha comentado es una solución basada en agente, con lo que estos agentes, recopilan información, analizan eventos chequean integridad de ficheros, etc. de los propios servidores, con lo que se obtiene una información 100% exacta de los que pasa en el equipo y no se queda en la simple correlación de ataques de red o intentos de ataques.
- Cumplimiento normativo. Al disponer de herramientas como OpenSCAP, nos permiten importar reglas de normativas externas, de obligado cumplimiento o internas de la propia empresa, ejp: políticas de passwords, política de puertos, etc. y de modo transparente podemos ver si los activos de la empresa cumplen con estas normativas.
- Gran comunidad de personas trabajando en Wazuh, añadiendo nuevas funcionalidades, nuevas versiones y corrigiendo parches continuamente.
- Posibilidad de en un futuro, una vez implantada en una institución, contratar servicio de mantenimiento de la aplicación para posibles bugs, nuevas funcionalidades, etc.
- Opción de análisis sin agente, sin duda alguna el gran potencial que tiene Wazuh es debido al agente que se instala, pero para otro tipo de equipamiento, por ejemplo, firewalls, routers, etc. tiene la posibilidad de conectarse sin agente, es verdad que las funcionalidades son muy limitadas, quedándose en cierto análisis básico, pero aún está en la versión 3.10.2, quien sabe lo que traerán futuras versiones.
- Instalación del sistema Wazuh en un equipo o máquina virtual ya preinstalada, te da la posibilidad de realizar un despliegue rápido de la herramienta usando la máquina virtual ya pre-configurada.

1.3 Módulos que integran Wazuh: descripción, Componentes y parametrización.

Como ya se ha comentado a lo largo de este proyecto Wazuh, según palabras de los responsables de este proyecto, es: una solución para la monitorización de la seguridad, es gratuita y de código abierto, además ayuda en la detección de amenazas, monitoriza la

integridad de ficheros, registros, etc., da respuesta a incidentes y ayuda para el cumplimiento de normativas.

Pero, ¿Cómo hace todo esto? Para entender como es capaz de realizar todo esto y muchas más cosas, debemos comprender el diseño de esta, que módulos la componen y de que se encarga cada uno de ellos.

Wazuh, está dividida en tres partes, siendo dos de ellas Wazuh (el agente y el servidor) y una tercera la el stack de Elastic para la representación y almacenamiento de los datos, debemos puntualizar que se podría usar Splunk para esa visualización y configuración de alertas, etc., pero como este proyecto lo estamos basando en software libre, se decide usar ELK.

En las siguientes imágenes podemos ver de modo esquemático que partes/módulos componen cada una de las partes de Wazuh.

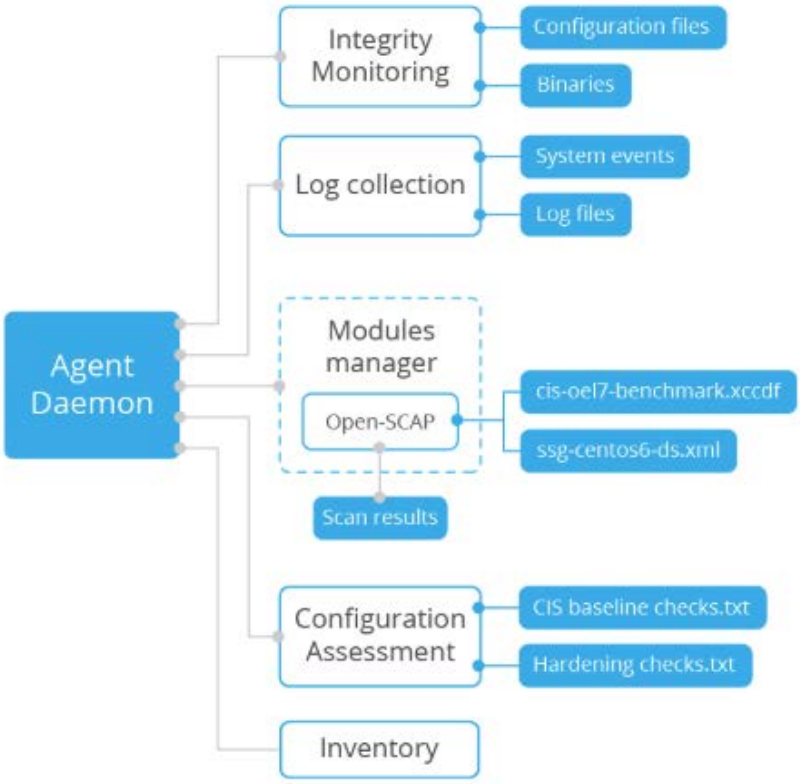


Imagen 2 - Módulos del Agente Wazuh

El servidor Wazuh, se compone a su vez de dos partes bien diferenciadas, una primera que podemos ver en la siguiente imagen, donde aparecen las diferentes partes que tratan la información recibida de los agentes, y una segunda que estaría formada por "Elastic Stack"[4], que se encargará de almacenar los logs, las alertas y demás información enviada por Wazuh, así mismo, este módulo se encarga de la representación en paneles gráficos, etc.

[4] https://www.elastic.co/es/what-is/elk-stack

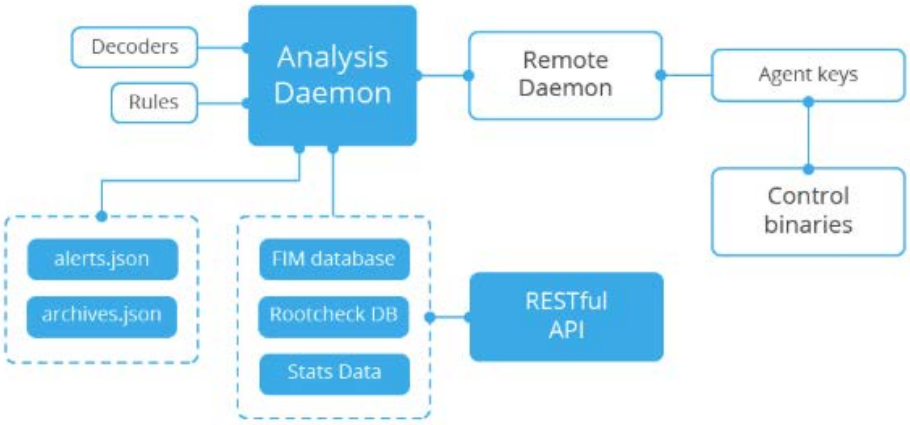


Imagen 3 - Módulos del Servidor Wazuh

A su vez estos dos grandes bloques (agente, servidor) están formados por diferentes módulos con diferentes funcionalidades que se integran para darnos esta herramienta de análisis de vulnerabilidades y seguridad. En los siguientes puntos analizaremos estos módulos y que módulos integran cada uno de los dos grandes bloques mencionados anteriormente.

1.3.1 Módulos del Agente Wazuh

El agente de Wazuh corre en los sistemas que queremos monitorizar recogiendo los logs y eventos que se producen en estos, además, se encarga de monitorizar y buscar malware y rootkits que puedan tener estos equipos, así como detectar el cambio de ficheros o claves del registro. Los módulos que hacen posible estas funcionalidades son los siguientes:

- **Integrity Monitorig (Monitorización de la Integridad):** Este módulo es el responsable de estar analizando los ficheros de configuración, binarios, claves del registro, etc. que se le digan para comprobar que no existen cambios no autorizados sobre ellos y en caso contrario producir una alerta.
- **Log Collection:** Este módulo se encarga de recolectar los logs del sistema o aplicaciones, así como eventos de estos para posteriormente ser tratados.
- **Inventory:** Realiza un inventario de los procesos que están corriendo en el sistema, así como de las aplicaciones instaladas.
- **Configuration Assessment:** Análisis de la configuración del sistema con respecto a reglas ya pactadas que deberían cumplir los equipos.
- **Modules Manager:** Módulo encargado de gestionar otros módulos externos, a modo de ejemplo OpenSCAP.

Toda la funcionalidad dada por los módulos anteriores es dada por los agentes que se instalan en los equipos finales, los endpoints. Estos agentes tienen varios componentes, que son los que realmente hacen el trabajo de buscar toda esta información, analizarla, etc. Estos componentes son los siguientes:

- **Rootcheck:** Componente encargado de analizar y buscar malware o rootkits.

- **Log monitoring/analysis:** Este componente se encarga de recolectar y analizar los logs del sistema monitorizado en búsqueda de actividad sospechosa.
- **Syscheck:** este componente se ejecuta periódicamente en búsqueda de cambios en los ficheros de configuración o el registro en el caso de los equipos Windows, en búsqueda de cambios.
- **OpenSCAP:** este componente nos permite comprobar configuraciones vulnerables, es decir, analizará el equipo para comprobar si se cumple con cierta política predefinida en la organización o que se deban cumplir por algún requisito como una ISO, u otro estándar que se quiera cumplir. Por ejemplo: reglas sobre los password, puertos abiertos, etc.

1.3.2 Módulos del Servidor Wazuh

El servidor Wazuh está formado por varios componentes que se encargan de toda la gestión de los agentes, distribución de parámetros, análisis de los resultados de los logs, etc. Dentro de estos componentes estacaríamos los siguientes:

- recolector de los logs o información enviada por los agentes y analizarlos.
- Procesamiento de eventos
- Búsqueda de IOCs (indicadores de compromiso) mediante inteligencia implementada en el módulo del servidor.
- Administración de los Agentes, configuración y actualización de forma remota.
- Envío de órdenes a los agentes para activar una respuesta ante un evento.

Además, debemos destacar que un servidor Wazuh puede analizar o soportar cientos o miles de agentes, con lo que con un solo servidor se podría dar respuesta a todos los agentes instalados en una institución. Pero como se detallará más adelante, es preferible crear una granja de ellos por redundancia, seguridad, carga, mantenimiento, etc.

En el siguiente punto, entraremos en más detalle en estos módulos y algunos ejemplos de parametrización.

1.3.3 Wazuh, sus módulos de análisis y ejemplos de parametrización

Los módulos anteriormente descritos permiten diferentes parametrizaciones, sobre todo a la hora de configurar que fichero o registros queremos verificar, o que reglas de seguridad se deben cumplir en los equipos finales. En este punto se describirán los módulos más importantes, los cuales serán implementados en el despliegue que se realizará en este TFM.

1.3.3.1 Log data collection

Este módulo permite dos tipos de recolección de logs:

- Mediante recolección de logs vía agente, el cual permite monitorizar un fichero log dado o por ejemplo que monitorice los logs de seguridad de windows.

  Ejemplos de configuración:

```
<localfile>
<location>/var/log/freeradius.log</location>
 <log_format>syslog</log_format>
</localfile>
```

- Envío de logs directamente al servidor vía syslog, ej.: un FW, envíe vía syslog estos logs al servidor para que estos sean tratados.

```
<ossec_config>
 <remote>
  <connection>syslog</connection>
  <port>513</port>
  <protocol>udp</protocol>
  <allowed-ips>10.0.3.0/24</allowed-ips>
 </remote>
<ossec_config>
```

(Este último, el envío de logs al propio servidor, no será usado en este TFM, nos centraremos en la recolección de información mediante agente y en ciertos casos el uso de "no agente" que tiene la aplicación.)

Existe múltiples modos para parametrizar la recogida de logs, por ejemplo, podemos configúralo para que recoja todos los logs de una carpeta, solos los de cierto formato, por tipo de fichero, etc.

1.3.3.2 File integrity monitoring

Mediante este módulo, conseguimos que el sistema esté en alerta y que, si se produce una modificación en un fichero el cual hemos configurado para que sea monitorizado, lance una alerta. Por defecto realiza un chequeo de integridad cada 12 horas, aunque se puede parametrizar para que el chequeo sea en tiempo real. En el análisis se tiene en cuenta diferentes parámetros como son: el tamaño del fichero, permisos, el dueño, la última modificación, hash del fichero. Ejemplos de cómo se configuraría tanto para la monitorización cada cierto tiempo como en tipo real:

```
<syscheck>
 <frequency>36000</frequency>
 <directories>/etc,/usr/bin,/usr/sbin</directories>
 <directories>/bin,/sbin</directories>
</syscheck>
```

```
<syscheck>
 <directories check_all="yes" realtime="yes">/var/log</directories>
</syscheck>
```

Este módulo nos permite múltiples parametrizaciones, pero entre ellas creemos que las más relevantes son las siguientes:

- Excepciones en la monitorización, nos permite configurar dentro de un directorio que fichero o ficheros no deben ser monitorizados.
- Informe de cambios, manda una alerta con los cambios ocurridos en el fichero.
- Definir el máximo nivel de recursión para monitorizar sub directorios.
- Modificar la severidad

1.3.3.3 Anomaly and malware detection

Este módulo se encarga de buscar anomalías o patrones de comportamiento anómalos o no esperados por las aplicaciones típicas instaladas en los equipos. Su funcionamiento se basa en la observación de: la integridad de los ficheros, los procesos activos, puertos ocultos, análisis de los ficheros inusuales y sus permisos, análisis de los ficheros ocultos.

Todo este análisis se realiza en base a unos ficheros con firmas, pero debemos puntualizar, que mientras se realiza esta documentación (22 de octubre de 2019) el fichero de firmas está desactualizado, según podemos informarnos en la web del producto.

1.3.3.4 Security Configuration Assessment

Como ya se ha comentado anteriormente, una de las grandes características de Wazuh es su versatilidad o facilidad para integrar nuevas funcionalidades de terceros con ella. Un ejemplo de esto es este módulo ya que en el integra: OpenSCAP, CIS-CAT oRootcheck, todas ellas relacionadas con la evaluación de la configuración de seguridad de los endpoints. Este módulo, SCA (Security Configuration Assessment), ha sido creado por los desarrolladores de Wazuh para poner fin a algunas limitaciones que tenían el resto, por ejemplo, OpenSCAP solo esté disponible para equipos LINUX.

Este módulo nos permite analizar los equipos finales comprobando si cumplen las reglas de seguridad establecidas por la organización, a modo de ejemplo, podríamos usar este módulo para:

- Comprobar que el tráfico ipv6 está prohibido.
- Que el firewall local, por ejemplo, que el firewall iptables está activado.

Todas estas reglas deben configurarse o definirse en ficheros de configuración los cuales deben ser desplegados en los equipos finales. Para realizar este despliegue, tenemos dos opciones: una primera que sería desplegarlo en cada equipo final, o usar la propiedad del servidor Wazuh de poder desplegar estos ficheros en los agentes finales. Aun así, si se tienen definidas reglas para los formatos OpenSCAP, CIS-CAT, pueden seguir usándose ya que como aparece en el siguiente punto (Monitoring security policies), estos módulos siguen integrados dentro de Wazuh.

1.3.3.5 Monitoring security policies

Este módulo nos permite analizar los hosts monitorizados verificando si cumplen una serie de reglas predefinidas por nosotros o por una institución para conocer el cumplimiento de una normativa, acuerdos internos, etc. Este módulo está compuesto por varias herramientas, las cuales son las encargadas de realizar este análisis:

Rootcheck: Este módulo ha sido sustituido desde las últimas versiones de Wazuh por el módulo SCA . A modo de ejemplo, en este módulo podemos crear reglas para comprobar que no se permite la conexión SSH de root.

OpenSCAP: Es usado para realizar análisis de configuración para comprobar el cumplimiento de seguridad dadas según la política de seguridad de la empresa, implica desarrollar en ficheros de configuración dicha política.

CIS-CAT: este sub-módulo se ha desarrollado con el propósito de integrar evoluciones CIS dentro de los agentes de Wazuh,. (CIS: Center for Intenet Security) Esta herramienta

implica la ejecución de ciertos scripts en el cliente, y necesita de java para su funcionamiento, con lo que sería necesario instalar Java en los clientes.

1.3.3.6 Command monitoring

Este módulo tiene una gran potencia ya que nos permite remotamente ejecutar comandos en los servidores que estamos monitorizando. Por ejemplo, se podría usar para detectar si cierto proceso critico está corriendo en el sistema y en caso contrario enviar una alerta, o detectar conexiones no permitidas vía netstat, etc. Debemos añadir que este módulo se puede utilizar tanto en los sistemas operativos Windows como Linux. Mencionar que, para activar esta opción, en el despliegue del agente se debe especificar expresamente mediante la siguiente instrucción en el fichero de configuración: local_internal_options.conf.

```
logcollector.remote_commands=1
```

1.3.3.7 Agentless monitoring

Wazuh, se caracteriza por el uso de agentes para monitorizar los hits. Gracias a este módulo, nos permite monitorizar ciertos aspectos, no todos, de los equipos a los que no se les puede instalar el agente, por ejemplo, switches, routers, equipamiento que aun teniendo un Linux no deja la instalación de agentes etc.

El módulo se conecta vía SSh y nos permite analizar la integridad de los sistemas, analizando configuraciones, ficheros etc, y posteriormente generar una alerta si algo no cumple con las reglas predefinidas. Esté módulo será usado para alertar los cambios de configuración en unos switch/routers 6500 de Cisco.

1.3.3.8 System inventory

Este módulo recoge información interesante sobre los equipos en los que está corriendo el agente, por defecto está habilitado y tiene una frecuencia de análisis de cada hora. Los datos que recoge son los siguientes: Datos del Hardware, datos del Sistema Operativo, Interfaces de Red, Paquetes instalados, Puertos abiertos, Procesos.

Este módulo es muy importante ya que en él se basa el siguiente módulo, el de búsqueda de Vulnerabilidades.

1.3.3.9 Vulnerability detection

Este módulo es de gran relevancia ya que analiza el sistema y aplicaciones en búsqueda de vulnerabilidades conocidas (CVE). Este sistema analiza el equipo y envía las versiones instaladas al nodo central, donde este lleva en su base de datos una recolección de las aplicaciones y versionado y en constante actualización con los repositorios de CVE, analiza y crea alertas si alguna de esas versiones tiene alguna vulnerabilidad conocida.

Actualmente **no** todos los sistemas operativos están soportados, a modo de ejemplo a día de esta redacción los Windows no estaba soportados.

Por defecto, el agente que se instala en los equipos recupera la versión del sistema operativo, así como las versiones de los paquetes y software instalado. Para Poder poner en marcha esta funcionalidad, se tiene que activar en el servidor de Wazuh esa característica, que realmente los que implica es bajarse los CVEs de seguridad y compararlos. Además, permite la parametrización de cada cuanto tiempo queremos que se realice esa validación, por defecto cada hora. Como se pone en marcha este módulo se explicarán en la parte del este TFM donde se pone en marcha la implementación de este sistema.

1.4 ¿OTRAS INTEGRACIONES?

En el punto anterior se han descrito partes o módulos que integran de base la solución Wazuh, pero como ya se ha mencionado anteriormente, la gran ventaja que tiene Wazuh frente a otras herramientas, es la posibilidad de realizar integraciones con otras utilidades, aplicaciones, etc., en este apartado mencionaremos alguna de ellas por tener gran importancia y relevancia para el uso de esta herramienta, y que se pondrán en producción en el sistema que se está implantando.

- Virus Total[5]: Existe la posibilidad de integrar Wazuh con VirusTotal para el análisis de virus de los ficheros alojados en el host, el mecanismo usado es mediante "hash" de los ficheros, los cuales son enviados, usando la API de VirusTotal, para compararlos con su base de datos y reportarnos si existe algún virus conocido. A la hora de configurar este módulo, debemos darnos de alta en la web de VirusTotal, y tenemos dos opciones para poder usar este servicio, la pública o privada, la diferencia principal es que el número de consultas de análisis de virus está limitado en la pública siendo estos las limitaciones:
    - 4 peticiones por minuto
    - 1000 peticiones al día
    - 30000 peticiones al mes

- Osquery: este módulo nos permite realizar consultas sobre el host/servidor como si fuese una base de datos relacional, es decir podríamos realizar consultas del estilo: "Select * from users;"
  (*Este módulo, a día de hoy no lo vemos de gran interés, por lo que no será implementado en este TFM*)

[5] https://www.virustotal.com/gui/sign-in

3. IMPLEMENTACIÓN Y PUESTA EN MARCHA DE WAZUH.

3.1. DIMENSIONAMIENTO DE UN SISTEMA EN PRODUCCIÓN.

Según podemos leer en la documentación encontrada en la página web de Wazuh, un solo servidor es capad de soportar miles de agentes. Debemos entender esta característica como algo teórico, ya que, dependiendo del análisis de esos agentes, por ejemplo, que esté analizando en tiempo real múltiples ficheros que tenga que analizar, envío de logs, etc. cargaría mucho a esos servidores, y no olvidemos que, por seguridad, redundancia, disponibilidad es aconsejable tener más de un servidor para llevar a cabo este servicio en cualquier organización.

Teniendo en cuenta lo comentado en el punto anterior y sabiendo que este proyecto será el embrión de un sistema formado por más de 500 equipos a analizar y desconociendo cuantos servidores Wazuh serán necesarios, debemos diseñar una solución que escale ante la posibilidad de crecer en servidores sin que se tenga que producir corte en el servicio. En este punto **no** nos adentraremos dentro del mundo de ELK, ya que la propia solución ELK, tendrá sus propios servidores y el número de ellos dependerá del número de logs que se envíen, el tiempo que se desee tener almacenado, etc., Es por ello que, en este POC nos centraremos solamente en la escalabilidad de los servidores Wazuh, dejando un solo servidor para la parte de ELK.

3.2. PLANIFICACIÓN DE LA IMPLEMENTACIÓN, HERRAMIENTAS DE DESPLIEGUE, ETC.

Una parte muy importante a la hora de desplegar un servicio como este en cientos de servidores es planificar como se realizará esta planificación, ya que llevaría mucho tiempo ir desplegándolo de uno en uno.

Una vez configurados los servidores de Wazuh, debemos ver que herramientas nos pueden facilitar el despliegue de los agentes en los diferentes servidores con diferentes sistemas operativos que poseemos.

En nuestro caso, la mayoría de los servidores a monitorizar son Windows o RedHat, es por ello que nos basaremos en ciertas herramientas que disponemos para el despliegue en ellos.

A continuación, se listarán dichas herramientas y ejemplos de cómo se configuran. Debemos tener en cuenta que solo es un ejemplo del despliegue ya que, en este POC, se desplegarán en pocos equipos, pero nos es muy válido para probar las herramientas de despliegue y analizar la parametrización de ellas para que enlace con el servidor de Wazuh, así como que debe usar conexiones TCP, etc. Mediante estos ejemplos se realiza una parametrización básica que servirá de base para los cambios que se deban realizar dependiendo de la naturaleza del servidor a auditar, por ejemplo, si tenemos varios servidores RedHat con apache, nos interesará analizar los ficheros de logs del servidor apache, carpetas donde se instale el servidor web, etc, que diferirán, de por ejemplo si son servidores mysql, etc.

Las herramientas que usaremos para el despliegue son:

- Red Hat Satellite
- Directorio Activo de Windows

Dentro de la documentación, en la fase de despliegue de los agentes, se mostrarán las configuraciones ejemplo y usadas para estos despliegues con estas herramientas.

3.3. PoC de Wazuh.

3.3.1. Instalación de Wazuh.

El sistema implementado en este proyecto consta de tres componentes bien diferenciados:

- los agentes.
- los servidores Wazuh
- el servidor ElastikSearch + Kibana

Estos tres puntos tienen funciones bien diferenciadas.

La parte de los agentes es la clave de este sistema, ya que es un software que se instala en el equipo que queremos monitorizar (puede ser Windows, Linux, Solaris, Mac) y recopila información del equipo, analiza los paquetes instalados, etc. y toda esta información es enviada a los servidores Wazuh para su análisis.

La parte del servidor/es de Wazuh, se encarga de analizar la información recibida de los agentes, lanzar alertas basada en reglas, etc.

La parte formada por ElastikSearch + Kibana, se encarga de indexar toda la información recolectada por los servidores Wazuh, además nos da una interface para poder explotar toda esta información y poder monitorizar los diferentes sistemas que estamos supervisando mediante los agentes.

En los siguientes puntos se detallará la instalación y parametrización de cada uno de los tres componentes mencionados anteriormente.

3.3.1.1. Análisis de las opciones de Implementación de la solución Wazuh.

En este PoC, y pensando a futuro, debemos realizar un diseño que nos permita crecer de un PoC para unos pocos servidores a uno en el que se pueda monitorizar la seguridad del total de servidores de la institución.

Podríamos estar pensando de partida, y basándonos en las especificaciones dadas por los creadores de Wazuh, que con un único servidor podríamos dar solución a nuestro reto. Pero preferimos analizar las diferentes soluciones dadas por la solución:

- Un único servidor.
- 2 servidores en HA, es decir en activo/pasivo.

- Granja de servidores.

A continuación, analizaremos las tres opciones.

Un único servidor: Como ventajas de esta solución podríamos decir que simplifica la instalación y administración ya que solo nos deberíamos preocupar de mantener un único equipo, por el contrario, esta solución no escalaria si se queda corto de recursos, o ante un fallo del servidor.

2 Servidores en HA: Está solución, aun complicando levemente la administración y la instalación inicial, tiene como ventaja que, en caso de fallo del servidor principal, tendríamos uno de respaldo con lo que no perderíamos el servicio en caso de fallo del servidor principal.

Granja de servidores: Sin duda esta opción sería la más compleja de administrar, ya que implicaría mantener varios servidores, un sistema de balanceo de carga que reparte el trabajo entre los servidores, etc, pero sin duda es la más ventajosa si estamos pensado en llevar a producción en una institución con cientos o miles de servidores, ya que facilita el crecimiento, simplemente añadiendo nuevos servidores a la granja.

Tras analizar los pros y contras anteriormente mencionados y teniendo en cuenta que este PoC se espera que sea el embrión de un sistema global dentro de la organización, nos hemos decantado por la tercera opción, la granja de servidores. Esta opción tiene algunas características que detallaremos a continuación y que en los siguientes puntos mostraremos como es esa implementación. En la siguiente imagen, podemos hacernos una idea de cómo será la implementación que se va a realizar en el TFM.

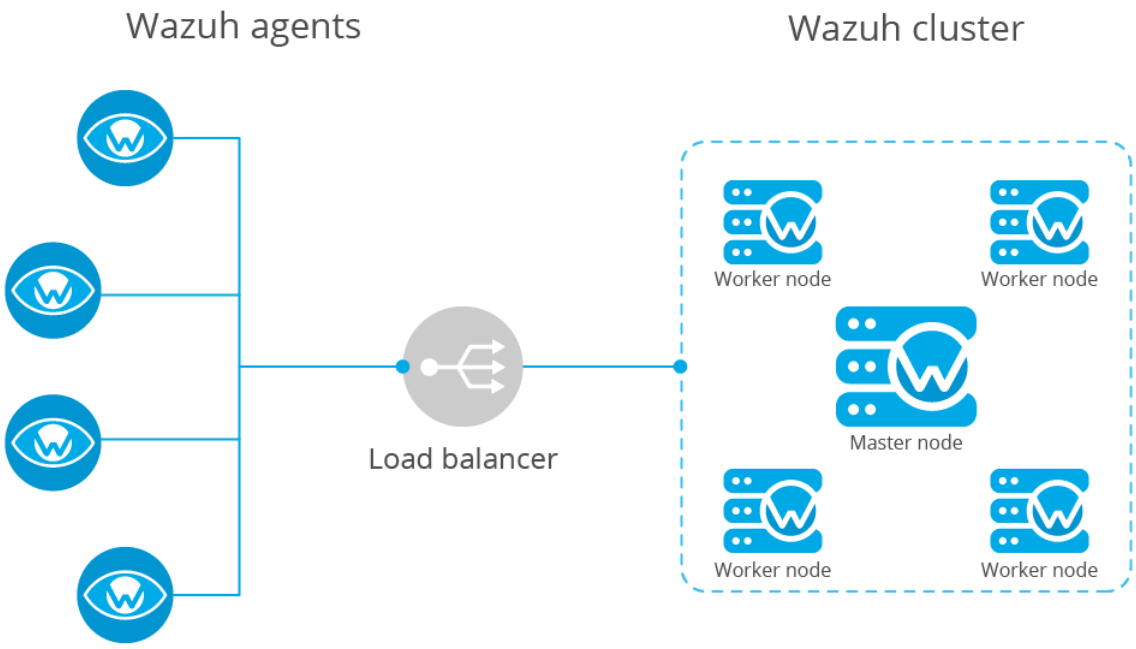


Imagen 4 - Esquema implementación Wazuh

Esta opción como característica principal y que debemos mencionar es que, aunque se balancee la carga entre todos los servidores, uno de ellos debe ser el maestro y el resto son los trabajadores, todos ellos tienen roles definidos:

El nodo maestro, sus funciones principales son:

- Registro de los agentes.
- Crear grupos de configuración compartidos.
- Actualización de reglas personalizadas, decodificadores y listas de CDB.
- Sincronizar toda esta información con los trabajadores.
- Los nodos maestros también pueden recibir y procesar eventos de agentes (como si fuesen un obrero más).

Los nodos Obrero, sus funciones principales son:

- Recibir actualizaciones del maestro
- Recepción y procesamiento de eventos de agentes.
- Enviando al maestro los últimos keepalives de agentes y asignaciones grupales remotas.

Como se muestra en la imagen anterior, es necesario implementar un sistema de balanceo que reparta la carga entre los servidores. En este caso tenemos varias opciones de balanceo, si la institución tiene ya un sistema para estos menesteres, ejemplo un balanceador F5[6] o A10[7], puede implementarse sin mayores complicaciones. En este PoC y aun teniendo en la institución un A10 para realizar estos trabajos, se ha decidido montar en una máquina virtual un sistema de balanceo Nginx[8]. En la siguiente imagen podemos ver el esquema que seguirá nuestro diseño en cuanto a los servidores que darán servicio a nuestra solución.

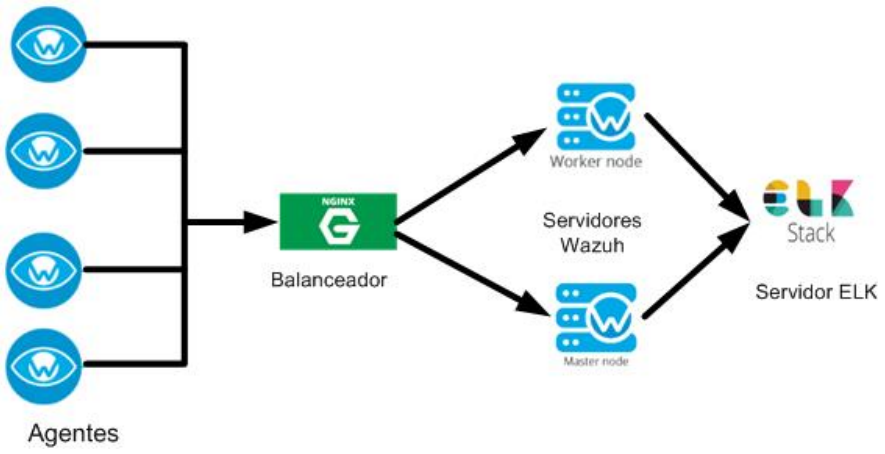


Imagen 5 – Servidores usados para implementar Wazuh.

3.3.1.2. INSTALACIÓN DE WAZUH DESDE CERO.

En esta instalación, usaremos un total de 4 servidores virtuales, los cuales realizaran las siguientes:

- Servidor de balanceo con Nginx (10.0.11.42/24)
- Servidor Maestro (10.0.11.8/24)
- Servidor Obrero (10.0.11.43/24)

[6] https://f5.com/es
[7] https://www.a10networks.com/
[8] https://www.nginx.com/

- Servidor de ELK (10.0.11.38/24)

Todos esto servidores tomarán como base el sistema operativo Ubuntu Server 18.04.1.

A continuación, detallaremos cada una de las instalaciones realizadas:

3.3.1.2.1. INSTALACIÓN Y CONFIGURACIÓN DEL BALANCEADOR (NGINX).

En este punto veremos las parametrizaciones realizadas para la instalación del servidor de balanceo, Nginx. Como ya se ha comentado, la principal funcionalidad de este servidor es la de balancear las conexiones de los Agentes con los servidores de Wazuh. Pasos dados para su instalación y configuración:

1. Instalar nginx
   # apt-get install nginx

2. Configurar nginx. Para ello editamos el fichero: /etc/nginx/nginx.conf añadiendo la siguiente configuración:

```
stream {
   upstream cluster {
      hash $remote_addr consistent;
      server 10.0.11.8:1514;
      server 10.0.11.43:1514;
   }
   upstream master {
      server 10.0.11.8:1515;
   }
   server {
      listen 1514;
      proxy_pass cluster;
   }
   server {
      listen 1515;
      proxy_pass master;
   }
}
```

   - En esta configuración el puerto 1514 está por defecto en TCP. Por defecto los servidores Wazuh funcionan con UDP, pero por necesidades y consejo de los desarrolladores de Wazuh, cuando se tiene una granja de servidores aconsejan usar TCP para mantener las conexiones entre los agentes y los servidores. Si se desease balancear por UDP, seria con la instrucción: listen 1514 udp;

3. Reiniciar nginx con el siguiente comando:

```
nginx -s reload
```

3.3.1.2.2. INSTALACIÓN Y CONFIGURACIÓN DE LOS SERVIDORES MAESTRO Y TRABAJADOR DE WAZUH.

(Los pasos dados en este punto son iguales para las dos versiones que vamos a configurar, posteriormente se añadirán las diferencias entre el maestro y el trabajador.)

La instalación en los servidores, se realizará desde los repositorios de Wazuh, con lo que primero debemos de ejecutar ciertos comandos para añadir en los servidores los

repositorios de Wazuh y de esta forma poder instalar o actualizar fácilmente el software. Los comandos utilizados son los siguientes:

```
# apt-get update
# apt-get install curl apt-transport-https lsb-release gnupg2
# curl -s https://packages.Wazuh.com/key/GPG-KEY-WAZUH | apt-key add -
# echo "deb https://packages.Wazuh.com/3.x/apt/ stable main" | tee -a
/etc/apt/sources.list.d/Wazuh.list
# apt-get update
```

A continuación, podemos proceder a instalar Wazuh con la siguiente instrucción:

```
# apt-get install Wazuh-manager
```

Instalación de Wazuh API:

```
# curl -sL https://deb.nodesource.com/setup_8.x | bash -
# apt-get install nodejs
# apt-get install Wazuh-api
# service Wazuh-api status
```

```
root@wazuh:~# service wazuh-api status
á wazuh-api.service - Wazuh API daemon
   Loaded: loaded (/etc/systemd/system/wazuh-api.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-10-09 15:07:06 UTC; 8s ago
     Docs: https://documentation.wazuh.com/current/user-manual/api/index.html
 Main PID: 20951 (nodejs)
    Tasks: 10 (limit: 4915)
   CGroup: /system.slice/wazuh-api.service
           áâ20951 /usr/bin/nodejs /var/ossec/api/app.js

oct 09 15:07:06 wazuh systemd[1]: Started Wazuh API daemon.
root@wazuh:~#
```

Imagen 6 - Estado del proceso Wazuh en el servidor

Instalación de Filebeat

Filebeat es la herramienta en el servidor Wazuh que reenvía alertas y eventos archivados de forma segura a Elasticsearch (servidor que guarda toda la información y posteriormente es explotada por Kibana para realizar una visualización de las alertas, etc.).

Para instalarlo debemos seguir los siguientes pasos:

```
# apt-get install curl apt-transport-https
# curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -
# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | tee
/etc/apt/sources.list.d/elastic-7.x.list
# apt-get update
# apt-get install filebeat=7.3.2
```

Descargamos el módulo de templates de alertas para ELK:

```
# curl -so /etc/filebeat/filebeat.yml
https://raw.githubusercontent.com/Wazuh/Wazuh/v3.10.2/extensions/filebeat/7.x/filebeat.yml
```

Descargamos el módulo de templates de alertas para Filebeat:

```
# curl -so /etc/filebeat/Wazuh-template.json
https://raw.githubusercontent.com/Wazuh/Wazuh/v3.10.2/extensions/elasticsearch/7.x/Wazuh-template.json
```

Descargamos el módulo de Wazuh para Filebeat:

```
# curl -s https://packages.Wazuh.com/3.x/filebeat/Wazuh-filebeat-0.1.tar.gz | sudo tar -xvz -C /usr/share/filebeat/module
```

Editar el fichero: /etc/filebeat/filebeat.yml y remplazamos YOUR_ELASTIC_SERVER_IP por la IP del servidor ELK

Ejemplo: output.elasticsearch.hosts: ['http://10.0.11.38:9200']

Habilitar y poner en marcha el servicio de Filebeat:

```
# systemctl daemon-reload
# systemctl enable filebeat.service
# systemctl start filebeat.service
```

- ***LA INSTALACIÓN EN EL WORKER(TRABAJADOR) ES IGUAL que en el maestro, por lo que no se detalla en esta documentación.***

Configuración Maestro Trabajador

Como se ha comentado anteriormente, un cluster de servidores Wazuh está formado por un servidor maestro y uno o varios servidores trabajadores (workers).

Para configurar estos servidores según su rol, se debe modificar el siguiente fichero:
/var/ossec/etc/ossec.conf
Se debe modificar la opcion de cluster: <cluster>...</cluster>
Además, debemos configurar una key en ellos que la podemos generar con el comando:

```
# openssl rand -hex 16
```

En nuestro caso debemos añadir las siguientes opciones en el maestro:

```
<cluster>
  <name>Wazuh</name>
  <node_name>master-node</node_name>
  <key>80c7691c96eddd199f8a607e27939ce7</key>
  <node_type>master</node_type>
```

```
  <port>1516</port>
  <bind_addr>0.0.0.0</bind_addr>
  <nodes>
    <node>10.0.11.8</node>
  </nodes>
  <hidden>no</hidden>
  <disabled>no</disabled>
</cluster>
```

En los nodos Workes:

```
<cluster>
  <name>Wazuh</name>
  <node_name>worker01-node</node_name>
  <key>80c7691c96eddd199f8a607e27939ce7</key>
  <node_type>worker</node_type>
  <port>1516</port>
  <bind_addr>0.0.0.0</bind_addr>
  <nodes>
    <node>10.0.11.8</node>
  </nodes>
  <hidden>no</hidden>
  <disabled>no</disabled>
</cluster>
```

Además, para que tome efecto las nuevas configuraciones debemos rearrancar los servicios de Wazuh:

```
# systemctl restart Wazuh-manager
```

Desde los equipos podemos comprobar el estado del cluster con el siguiente comando:

```
# /var/ossec/bin/cluster_control -l
```

```
root@wazuh:~# /var/ossec/bin/cluster_control -l
NAME      TYPE    VERSION  ADDRESS
master    master  3.10.2   10.0.11.8
worker-1  worker  3.10.2   10.0.11.43
root@wazuh:~#
```

Imagen 7 - Servidores que forman parte del cluster de Wazuh

Tenemos que desactivar la opción: use_source_ip

Cuando se usa un balanceador, los nodos del cluster no ven la ip de los agentes, solo la del balanceador. Como consecuencia los agentes no pueden conectarse con el cluster.
Para solucionarlo, debemos cambiar el parámetro “use_source_ip”, poniendo el valor: "no" en el servidor master (esta configuración se realiza sobre el fichero: ossec.conf).

```
<auth>
  <disabled>no</disabled>
  <port>1515</port>
  <use_source_ip>no</use_source_ip>
  ...
```

```
</auth>
```

Consideraciones de la instalación en cluster:

La comunicación por defecto entre los Agentes y el servidor Wazuh se realiza por el puerto 1514 en UDP, en el caso de tener un cluster con sistema de balanceo, se aconseja cambiar la conexión a TCP.

Par realizar este cambio se deben realizar ciertos cambios que se detalla a continuación:

Servidor Maestro y en los Workers:

Debemos cambiar en el fichero: "/var/ossec/etc/ossec.conf" el protocolo de "udp" a "tcp", quedando de la siguiente manera:

```
<remote>
  <connection>secure</connection>
  <port>1514</port>
  <protocol>tcp</protocol>
  <queue_size>131072</queue_size>
 </remote>
```

Servidor Balanceador con Nginx: Por defecto y con la configuración dada en el punto anterior ya estaría, pero sería modificar en el fichero: /etc/nginx/nginx.conf

UDP→listen 1514 udp;

TCP→listen 1514;

Agentes: Para modificar los agentes se debe cambiar el protocolo en el fichero:

Windows →C:\Program Files (x86)\ossec-agent\ossec.conf

Linux→/var/ossec/etc/ossec.conf

Modificaremos el protocolo según se muestra en la siguiente tabla:

UDP	TCP
<server> <address>10.0.11.42</address> <port>1514</port> <protocol>**udp**</protocol> </server>	<server> <address>10.0.11.42</address> <port>1514</port> <protocol>**tcp**</protocol> </server>

3.3.1.2.3. INSTALACIÓN DEL SERVIDOR DE ELASTIC STACK.

La principal funcionalidad de los servidores Wazuh es tener una comunicación con los agentes o equipos monitorizados sin agente. Estos servidores recogen la información enviada por los agentes y los analizan, pero toda esta información debe recopilarse ya almacenar en un servidor, además necesitamos una interface gráfica que nos permita de una forma clara, rápida y concisa explotar toda esta información. Es en este punto donde entra la pila ELK formada en este caso por:

- Elasticksearch para tener un motor de búsqueda.
- Kibana, que es la ventana a nuestros datos, es el GUI que se á usado para explotar todos los datos,

Para instalar ELK en el servidor Ubuntu, debemos realizar una serie de pasos, los cuales se describen a continuación:

1. Añadir el repositorio de Elastic y su clave para poder instalarlo.

```
# apt-get install curl apt-transport-https
# curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -
# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | tee
/etc/apt/sources.list.d/elastic-7.x.list
# apt-get update
```

2. Instalamos el paquete de Elasticsearch

```
# apt-get install elasticsearch=7.3.2
```

3. Por defecto Elasticsearch solo escucha localmente, y en nuestra instalación como los mensajes le llegaran desde el exterior, debemos configúralo para que escuche en la IP de servicio. Para ello debemos modificar:

   /etc/elasticsearch/elasticsearch.yml

   Debemos descomentar la línea “network.host” y añadir la ip de servicio, en nuestro caso quedaría:

```
network.host: 10.0.11.38
```

   Además, debemos descomentar o añadir las siguientes líneas:

```
node.name: elkWazuh
cluster.initial_master_nodes: ["elkWazuh"]
```

4. Debemos arrancar y activar Elasticsearch, para ello debemos usar los siguientes comandos:

```
# systemctl daemon-reload
# systemctl enable elasticsearch.service
# systemctl start elasticsearch.service
```

5. Una vez que tenemos instalado y arrancado Elasticsearch, es recomendable cargar el template de Filebeat, para ello usaremos el siguiente comando:

```
# apt-get install filebeat
```

   Editar: /etc/filebeat/filebeat.yml y modificar: hosts: ["localhost:9200"] por hosts: ["10.0.11.38:9200"]

```
# filebeat setup --index-management -E setup.template.json.enabled=false
```

Instalación de Kibana

6. Para instalar Kinaba, que es la interface gráfica vía web que tendremos para ver los logs, alertas, etc. debemos ejecutar el siguiente comando:

```
# apt-get install kibana=7.3.2
```

7. Instalamos el pluging de la aplicación Wazuh en Kibana con el siguiente comando:

```
# sudo -u kibana /usr/share/kibana/bin/kibana-plugin install
https://packages.Wazuh.com/Wazuhapp/Wazuhapp-3.10.2_7.3.2.zip
```

8. Por defecto el servicio de Kibana solo escucha localmente, es decir solo usa la ip de loopback. Para que este servicio sea usado desde equipos externos, debemos

configurar que el servidor web escuche en la ip de servicio externo, para ello debemos modificar el siguiente fichero: /etc/kibana/kibana.yml
Debemos descomentar: #server.host: "localhost" y cambiarlo por:

```
server.host: "10.0.11.38"
```

Además debemos decirle a Kibana donde escucha elasticsearch, para ello en el mismo fichero debemos cambiar: #elasticsearch.hosts: ["http://localhost:9200"] por:

```
elasticsearch.hosts: ["http://10.0.11.38:9200"]
```

9. Por último, debemos habilitar el servicio de Kibana con los siguientes comandos:

```
# systemctl daemon-reload
# systemctl enable kibana.service
# systemctl start kibana.service
```

En la siguiente imagen podemos ver el servidor Kibana con el pluging de Wazuh ya instalado:

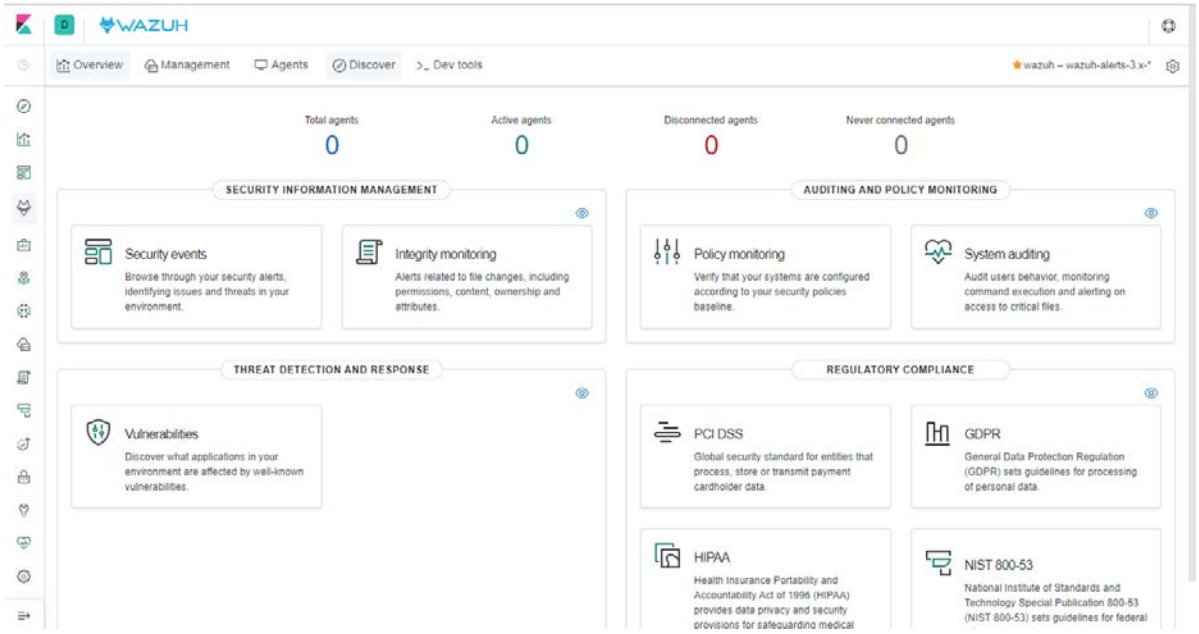


Imagen 8 - Pantalla de Wazuh en Kibana

Ahora, lo que debeos es añadirlos servidores Wazuh dentro de Kinaba, usando el API que se instaló anteriormente. Para ello, usando la interface gráfica de Kibana (http://10.0.11.38:5601) nos dirigimos al módulo de Wazuh y dentro de configuración añadimos los servidores, como podemos observar en la siguiente imagen: (añadimos en nuestro caso los servidores: 10.0.11.8 y el 10.0.11.43)

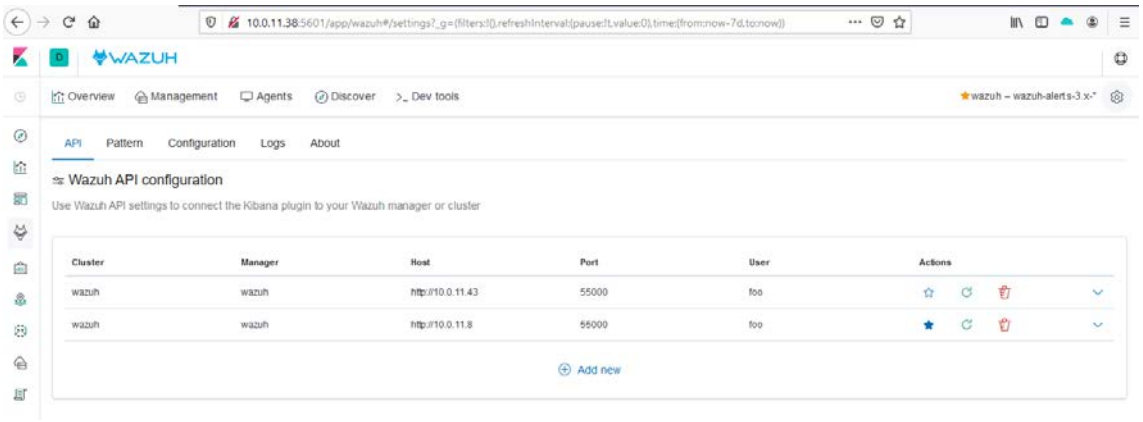


Imagen 9 - Kibana Servidores de Wazuh

3.3.1.3. CUESTIONES/PROBLEMÁTICAS RELACIONADAS CON LA INSTALACIÓN.

La versión instalada, la última estable: 3.10.2, en la versión de cluster, tiene un bug reconocido en los ficheros temporales donde dejan los servidores Wazuh información de los agentes. Dicho bug estará solucionado para la versión 3.11, de momento hemos realizado las correcciones directamente en código, según aparece reflejado en la web de mantenimiento de versiones del proyecto Wazuh.

Bug:

https://github.com/Wazuh/Wazuh/issues/4007

Solución:

https://github.com/Wazuh/Wazuh/pull/4025/files/bb1ea13917d7ad92de5f5c46f643eb77029bf744

3.3.1.4. CONFIGURACIONES EXTRAS

3.3.1.4.1. ACTIVAR DETECCIÓN DE VULNERABILIDADES

- *Este módulo en esta versión solo está presente para las versiones de Linux (Ubuntu, Red Hat, Centros, Debian y Amazon Linux)*

Este módulo es el encargado de analizar las versiones de los paquetes instalados en los equipos y compararlos con las CVEs publicadas en búsqueda de vulnerabilidades. Esta labor la realizan los servidores de Wazuh ya que ellos tienen en sus bases de datos los paquetes que tiene instalados los servidores, esta información es enviada por el agente.

La configuración para detección de vulnerabilidades, por defecto viene desactivado, para cambiar esta opción debemos editar: /var/ossec/etc/ossec.conf en el servidor Wazuh master.

En nuestro caso cambiamos la directiva “disabled” de “yes” a “no”, de forma genérica y en las versiones de Redhat, ya que de momento no tenemos Ubuntu en la infraestructura.

```
<wodle name="vulnerability-detector">
```

```
  <disabled>no</disabled>
  <interval>5m</interval>
  <ignore_time>6h</ignore_time>
  <run_on_start>yes</run_on_start>
  <feed name="ubuntu-18">
   <disabled>yes</disabled>
   <update_interval>1h</update_interval>
  </feed>
  <feed name="redhat">
   <disabled>no</disabled>
   <update_from_year>2010</update_from_year>
   <update_interval>1h</update_interval>
  </feed>
  <feed name="debian-9">
   <disabled>yes</disabled>
   <update_interval>1h</update_interval>
  </feed>
 </wodle>
```

3.3.1.4.2. VIRUSTOTAL

Este módulo es uno de los módulos externos que se deben configurar y que entendemos que es de gran interés ya que permite la integración entre Wazuh y VirusTotal para la búsqueda de virus en los ficheros que estén almacenados en los servidores. Pensemos en un servidor que se dedique al almacenamiento de archivos, estilo nube privada como Nextcloud.

Mediante este módulo podemos analizar en busca de virus todos los archivos que dejen los usuarios. Para implementar esta opción debemos realizar una serie de pasos:

- Darse de alta en la web de VirusTotal, para poder obtener la key que permitirá integrarse nuestro servidor Wazuh con VirusTotal.
- Configurar la integración de este API con Wazuh, para ello debemos editar: /var/ossec/etc/ ossec.conf y debemos añadir la siguiente parte de código dentro de la sección “<ossec_config>”

```
        <integration>
          <name>virustotal</name>
          <api_key>XXXXXXXXXXXXXXXXXXXX</api_key> <!-- Replace with your
          VirusTotal API key -->
          <group>syscheck</group>
          <alert_format>json</alert_format>
</integration>
```

- A continuación, debemos decidir que queremos chequear con este módulo, y si queremos que sea en tiempo real, para ello, y en nuestro caso se ha decidido monitorizar /home por si nuestros usuarios decidiesen introducir algún fichero con algún virus. Debemos añadir dentro de la sección: <syscheck> el siguiente código:

```
   <directories check_all="yes" realtime="yes">/home</directories>
```

En : /var/ossec/etc/ ossec.conf del cliente

```
#apt-get install python-requests
#systemctl restart Wazuh-manager
```

3.3.1.4.3. AGENTLESS

Una de las carencias que podría achacársele a este sistema es la opción de analizar otros sistemas en los cuales no se les puede instalar un agente, ejp de Firewalls, routers, etc. Esta merma se consigue solunar con el módulo de "Agentless" en el cual nos permite realizar ciertos análisis sobre el sistema que se quiere monitorizar. Estos análisis, en la versión actual, se basan en el análisis de ficheros o de configuraciones. Por ejemplo, si tenemos un sistema con un Linux embebido, podremos analizar la modificación de ciertos directorios o ficheros, o por el contrario si lo que tenemos es un Firewall o un router, ejemplo de Cisco ya que es la más integrada, podemos analizar cambios en la configuración.

A la hora de configurar el sistema "Agenless" tenemos varias opciones:

- Crear una ssh key en el servidor Wazuh, y esta instalarla en el equipo que queremos monitorizar, o
- Añadir usuario/password [enablepassword] en el propio servidor Wazuh para que este se conecte.

En nuestro caso desplegaremos este módulo para analizar los Switch/routers actuales de nuestra institución, unos Cisco 6500, y usaremos la segunda opción anteriormente mencionada.

A continuación, se listarán los pasos dados para configurar Wazuh para poder acceder a esos sistemas:

Versión de SSH nos soportada por el servidor, añadir esto en la configuración de /etc/ssh/ssh_config:

```
Host 10.0.10.31
   KexAlgorithms +diffie-hellman-group1-sha1
   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
```

Debemos añadir el equipo con la cuneta y password de acceso, para ello desde el servidor de Wazuh maestro ejecutamos la siguiente orden:

```
/var/ossec/agentless/register_host.sh add xxxx@10.0.10.31 password enablepass
```

Configura el fichero ossec.conf en el servidor de Wazuh para que use agenless, para ello hemos añadido el siguiente código en /var/ossec/etc/ossec.conf:

```
<agentless>
  <type>ssh_pixconfig_diff</type>
  <frequency>36000</frequency>
  <host>xxxxx@10.0.10.31</host>
  <state>periodic_diff</state>
</agentless>
```

Se debe reiniciar el proceso de Wazuh en el servidor:

```
# systemctl restart Wazuh-manager
```

Fallo al intentar recolectar la información, se necesitaba instalar el siguiente paquete:

```
#apt-get install expect
```

Información vista en los logs que proporciona Wazuh:

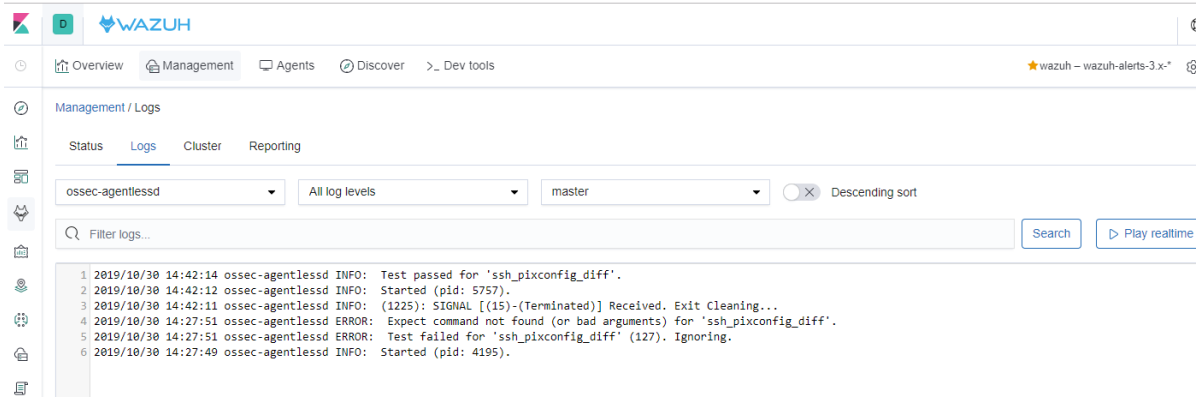


Imagen 10 - logs en Kibana de Wazuh del módulo agenless

Cambio detectado por el sistema sin agente en uno de los routers 6500 de la marca Cisco:

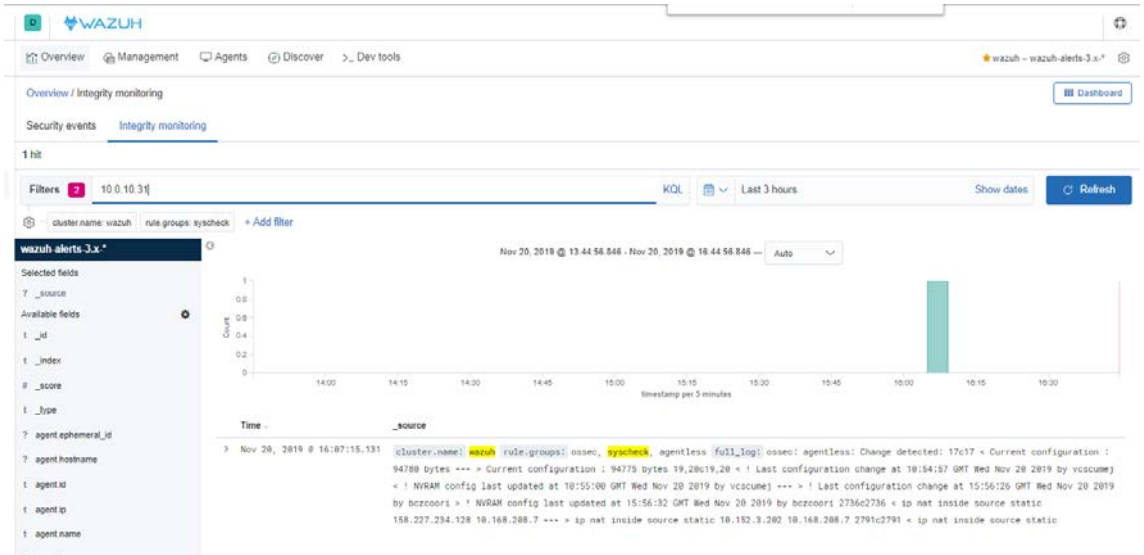


Imagen 11 - Agentless cambio detectado Cisco 6500

Información adicional sobre el módulo agentless y su puesta en marcha:

- Nos hemos encontrado con problemas para conectarnos vía SSH con el switch 6500 debido a cierta implementación que realiza el script: /var/ossec/agentless/ssh_pixconfig_diff, usa un parámetro que ha sido eliminado de la configuración: "-c des"
- Se puede probar la conexión con el switch, router, tec mediante el comando: /var/ossec/agentless/ssh_pixconfig_diff xxxxxx@10.0.10.31 pasword enablePassword
- Si se desea eliminar un elemento monitorizado sin agente, no existe el comando "`register_host.sh del`", se elimina editando el fichero: "/var/ossec/agentless/.passlist"

3.3.1.4.4. ALERTAS Y REPORTES VÍA EMAIL

Sin duda alguna, una opción indispensable es el envío de alertas en caso de detectar una vulnerabilidad, un acceso no autorizado, virus, etc. Esta utilidad se debe configurar para el envío de Emails para avisar al grupo administrados de los sistemas de alertas o de informes diarios para analizar el estado de seguridad de los sistemas monitorizados.

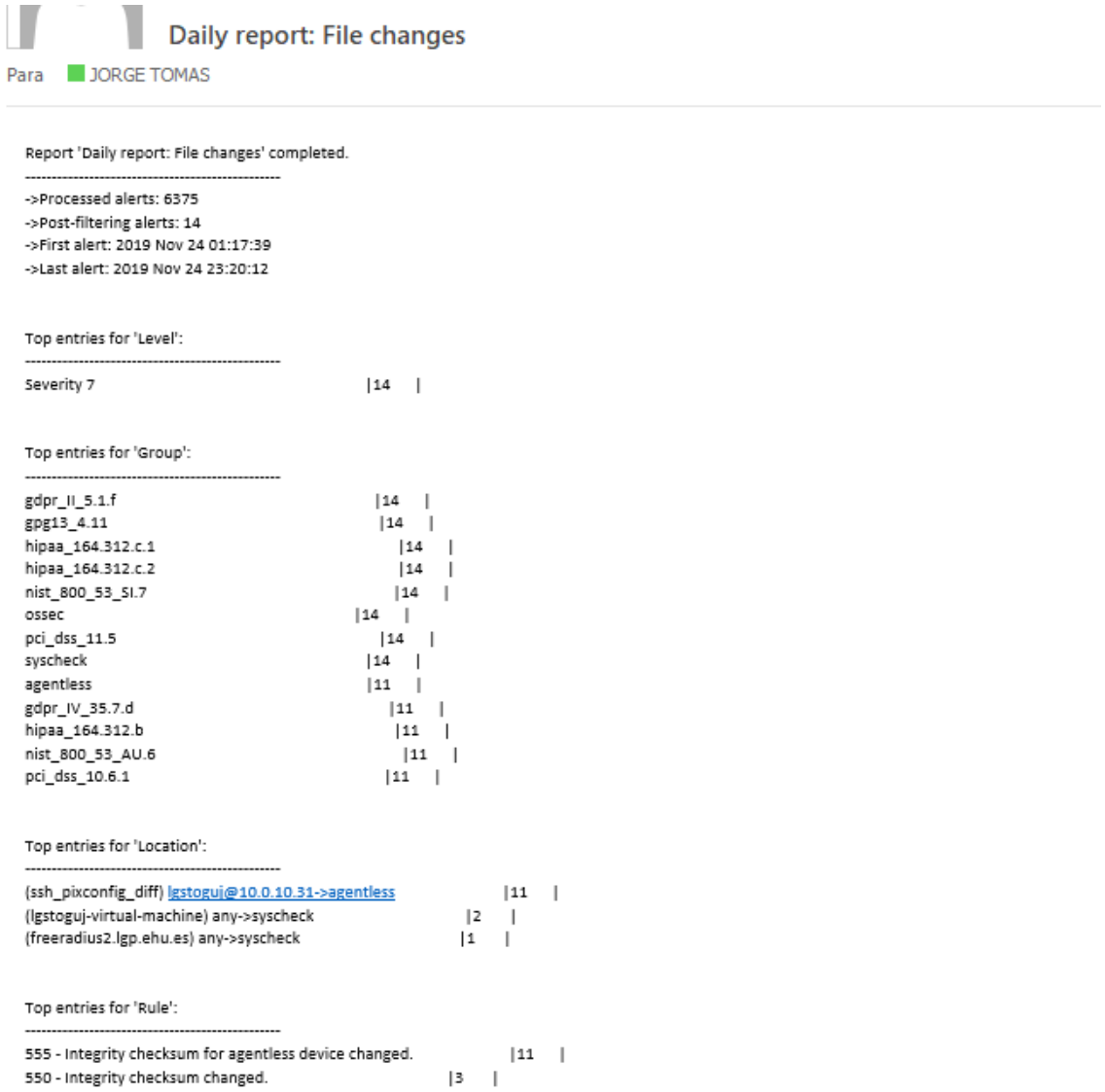
Para poder configurarlo, se debe realizar ciertos cambios en el fichero ossec.conf del servidor master de Wazuh, este fichero lo podemos encontrar en : /var/ossec/etc

Adjuntamos las configuraciones realizadas para el envío de los email y reports::

```
<ossec_config>
  <global>
……..
    <email_notification>yes</email_notification>
    <smtp_server>smtp.xxx.com</smtp_server>
    <email_from>Wazuh@ xxx.com </email_from>
    <email_to>cert@ xxx.com </email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
  </global>
```

```
<reports>
     <category>syscheck</category>
     <title>Daily report: File changes</title>
     <email_to>cert@xxx.com</email_to>
  </reports>
```

- *Sí que es verdad que en esta versión las alertas enviadas vía email, así como los informes diarios, son en texto y no muy detallados, pero son útiles para tener una ligera idea de por dónde investigar esos cambios, vulnerabilidades, o fallos de configuración.*



Daily report: File changes

Para JORGE TOMAS

Report 'Daily report: File changes' completed.

->Processed alerts: 6375
->Post-filtering alerts: 14
->First alert: 2019 Nov 24 01:17:39
->Last alert: 2019 Nov 24 23:20:12

Top entries for 'Level':

Severity 7 |14 |

Top entries for 'Group':
---
gdpr_II_5.1.f |14 |
gpg13_4.11 |14 |
hipaa_164.312.c.1 |14 |
hipaa_164.312.c.2 |14 |
nist_800_53_SI.7 |14 |
ossec |14 |
pci_dss_11.5 |14 |
syscheck |14 |
agentless |11 |
gdpr_IV_35.7.d |11 |
hipaa_164.312.b |11 |
nist_800_53_AU.6 |11 |
pci_dss_10.6.1 |11 |

Top entries for 'Location':

(ssh_pixconfig_diff) lgstoguj@10.0.10.31->agentless |11 |
(lgstoguj-virtual-machine) any->syscheck |2 |
(freeradius2.lgp.ehu.es) any->syscheck |1 |

Top entries for 'Rule':
---
555 - Integrity checksum for agentless device changed. |11 |
550 - Integrity checksum changed. |3 |

Imagen 12 - Reporte diario de cambios en ficheros

3.3.2. INSTALACIÓN DE LOS AGENTES

Para realizar la instalación de estos agentes se ha decidido en una primera fase realizar una instalación manual, tanto en un Linux como en un Windows para analizar las pegas o posibles problemas a la hora de pasarle configuraciones extras, así como, IP del servidor Wazuh, protocolo (TCP/UDP), etc.

Pero para el despliegue en todos los equipos usaremos herramientas que nos automaticen esta tarea, a modo de ejemplo serán listadas algunas de las herramientas de las que disponemos para realizar esa instalación y más adelante se mostrarán imágenes con ejemplos exactos con ellas:

Sobre Linux:

- Redhat Satélite, herramienta usada por la institución para el despliegue de paquetes, configuraciones etc.

Sobre Windows:

- Active Directory, mediante política de grupo, se puede desplegar aplicaciones y configuraciones.

3.3.2.1. LINUX

En Linux, nuestro caso son Red Hat usamos los siguientes comandos para introducir el repositorio y configurar el agente de Wazuh:

```
# rpm --import http://packages.Wazuh.com/key/GPG-KEY-WAZUH
# cat > /etc/yum.repos.d/Wazuh.repo <<\EOF
[Wazuh_repo]
gpgcheck=1
gpgkey=https://packages.Wazuh.com/key/GPG-KEY-WAZUH
enabled=1
name=Wazuh repository
baseurl=https://packages.Wazuh.com/3.x/yum/
protect=1
EOF


# WAZUH_MANAGER="10.0.11.42" WAZUH_PROTOCOL="tcp" yum install Wazuh-agent
```

3.3.2.2. WINDOWS

Para poder instalar el agente en los equipos Windows, la opción más rápida es bajar el agente desde la web de Wazuh. Este ejecutable instalará la aplicación y creará el servicio que arrancará automáticamente. En nuestro caso se ha cambiado la configuración inicial, añadiendo que la comunicación se debe realizar vía TCP. La instalación ha sido realizando la siguiente llamada desde el CMD:

```
Wazuh-agent-3.10.2-1.msi /q ADDRESS="10.0.11.42" AUTHD_SERVER="10.0.11.42"
WAZUH_PROTOCOL="tcp"
```

Por defecto, una vez instalado, los ficheros de configuración los encontramos en:

C:\Program Files (x86)\ossec-agent\

3.3.2.3. HERRAMIENTAS DE AUTOMATIZACIÓN DEL DESPLIEGUE DEL AGENTE

(Red HAT Satellite): En la siguiente imagen podemos ver como se parametriza y configura en Red Hat Satellite el despliegue del agente:

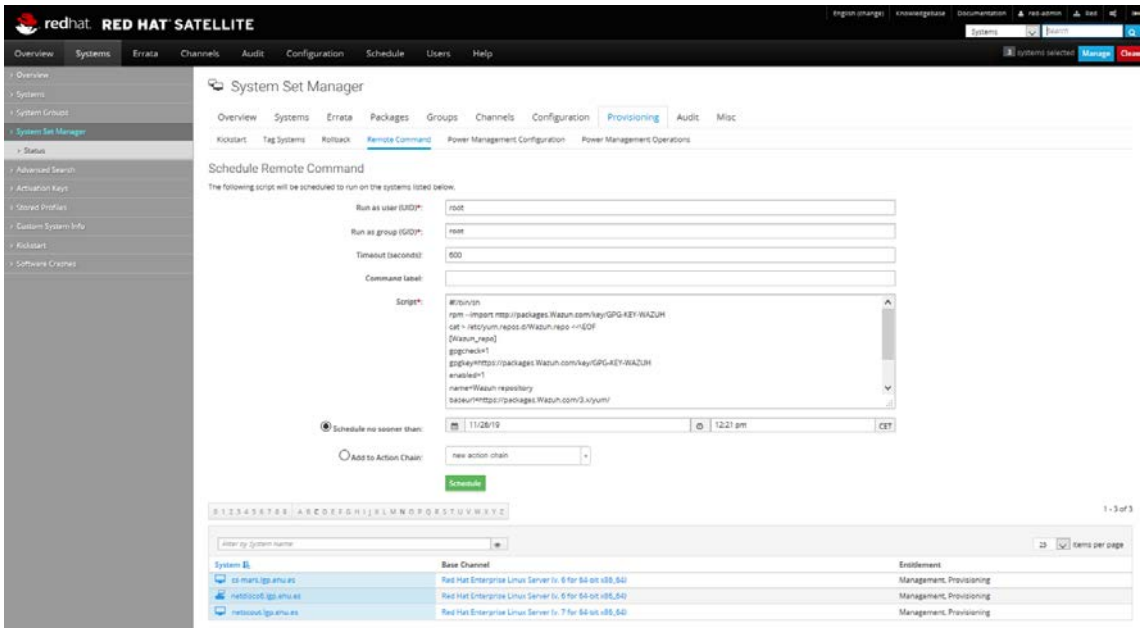


Imagen 13 - Red Hat Satellite

Esta herramienta nos permite desplegar en el número de servidores, que desde ella se gestionen, la instalación y configuración del agente. Además, nos permitiría desplegar ficheros extras si tenemos que añadir configuraciones especificas a las máquinas dependiendo del servicio que dan, la diferencia con el despliegue manual es que debemos decirle al instalador que será un despliegue desatendido, es decir en el caso de yum, añadir –y, quedando el comando de la siguiente manera:

WAZUH_MANAGER="10.0.11.42" WAZUH_PROTOCOL="tcp" yum -y install Wazuh-agent

(Directorio Activo): Para la instalación del agente en los equipos Windows, se ha usado la funcionalidad que nos dan los Directorio Activos de la empresa de mediante una directiva de grupo instalar software de forma remota. En este POE, solo se ha usado para dos servidores, aun así, se adjunta el link de Microsoft donde se puede seguir paso a paso como realizar esta configuración:

https://support.microsoft.com/es-es/help/816102/how-to-use-group-policy-to-remotely-install-software-in-windows-server

3.3.2.4. CONFIGURACIONES EXTRAS EN LOS AGENTES.

Como ya se ha comentado a lo largo de esta documentación, el sistema Wazuh se basa en la parametrización de los agentes finales mediante ficheros de configuración. Por ejemplo, en estos ficheros podemos decirle al agente que esté analizando cambios en la carpeta /etc/dhcpd/ o /home/uoc/. Para facilitarle esta o información al agente tenemos dos opciones:

- Una primera que sería configurar uno a uno los agentes finales
- Configurar en el servidor Wazuh y que sea este el que lo despliegue a los nodos.

Además, el sistema Wazuh, nos permite agrupar los agentes por grupos, pudiendo agruparlos por software instalado, sistema operativo etc. Una ventaja que tiene el sistema de grupos es que no es excluyente, es decir, un agente final puede pertenecer al grupo de servidores Web y al de Base de Datos, con lo que se le aplicaría las políticas y reglas de los dos.

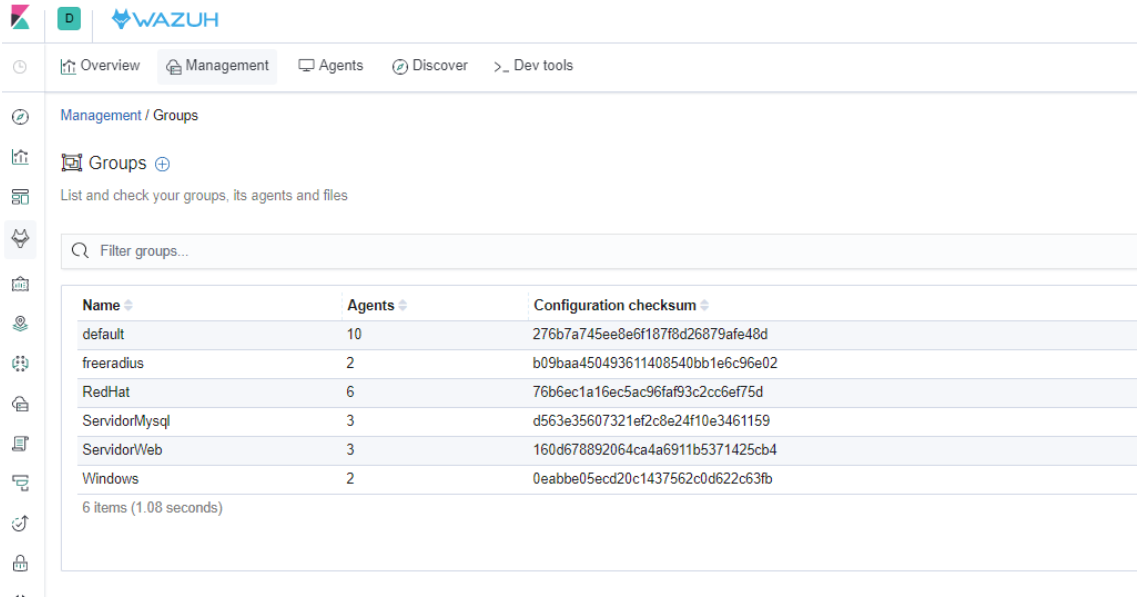


Imagen 14 - Grupos de agentes

3.3.2.5. PROBLEMAS EN LAS INSTALACIONES

Algunas instalaciones de agente, por razones desconocidas, no se conectan con el servidor para obtener la key, este identificador que es único, necesario para reconocer por parte de los servidores Wazuh al agente, es almacenado en el fichero: client.keys que lo podemos localizar en: /var/ossec/etc

Ejemplo del fichero client.keys

```
013 Wazuh-virtual-machine any a61ce5b0db98c8976ec681e3416b37c17c241676048c5b99
```

Para forzar que se conecte y obtener ese identificador único se debe ejecutar el comando:

```
/var/ossec/bin/agent-auth -m 10.0.11.42
```

De esta forma se consigue que tenga ese identificado único y ya se puede rearrancar el servicio en el equipo cliente:

```
systemctl restart Wazuh-agent.service
```

3.4. WAZUH EN PRODUCCIÓN

En este punto veremos Wazuh en producción, es decir, veremos casos prácticos donde utilizamos sus funcionalidades más relevantes en casos reales o simulados y veremos con actúa el sistema y cómo podemos ver las alertas generadas al detectarlos. Los casos usados en este punto son los siguientes:

- Vulnerabilidades detectadas en los servidores durante la duración del POC.
- Detección de virus mediante el módulo de VirusTotal.
- Detección de ataque de fuerza bruta de SSH.
- Detección del ataque shellshock y su remediación.
- Políticas de bastionado: OPENSCAP.

3.4.1. VULNERABILIDADES DETECTADAS EN LOS SISTEMAS MONITORIZADOS.

Como ya se ha comentado a lo largo de esta documentación, una gran ventaja que tiene esta herramienta es que podemos ver en un simple vistazo las alertas más significativas de vulnerabilidades detectadas.

En la siguiente imagen podemos ver un ejemplo de las vulnerabilidades reportadas por los agentes de los 13 equipos monitorizados durante las últimas 24 horas en la fecha que fue capturada la imagen. Como podemos apreciar, de un vistazo, tenemos toda la información relevante en cuanto a vulnerabilidades, marcándonos cuál de los servidores es el más expuesto a ellas. Datos recogidos a fecha 18/12/2019:

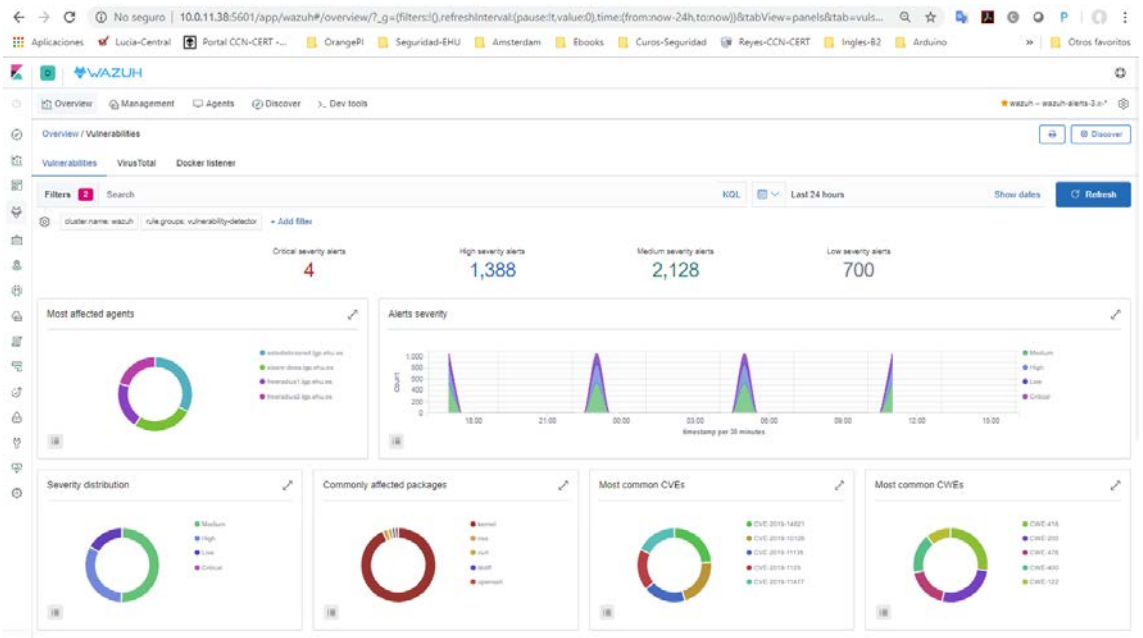


Imagen 15 - Kibaba-vulnerabilidades detectadas

Aun así, podemos tener este mismo análisis por cada endpoint que estamos monitorizando, teniendo una información más exacta de todas sus vulnerabilidades detectadas, sin duda, interface de gran interés para el responsable del servidor final para mantener la seguridad y tomar las medidas necesarias para evitarlas.

3.4.2. ALERTA DE VIRUS CON EL API DE VIRUSTOTAL

En muchos equipos, por política de empresa o por no tener contratado un sistema de antivirus, tenemos un agujero de seguridad en cuanto a los ficheros que los usuarios finales de los servidores pueden introducir en ellos.

Con este módulo podemos proteger este agujero de seguridad, ya que nos permite que todo fichero nuevo que sea copiado, movido o creado en ciertos directorios de interés, sean analizados contra el sistema de VirusTotal.

En la siguiente imagen podemos ver cómo ha reaccionado el servidor Wazuh cuando hemos descargado un virus de ejemplo de Eicar[9] en uno de los directorios que tenemos parametrizados para estar analizando continuamente.

```
** Alert 1571296458.7683499: - ossec,syscheck,pci_dss_11.5,gpg13_4.11,gdpr_II_5.1.f,hipaa_164.312.c.1,hipaa_164.312.c.2,nist_800_53_SI.7,
2019 Oct 17 07:14:18 (xixare-desa.lgp.ehu.es) any->syscheck
Rule: 554 (level 5) -> 'File added to the system.'
File '/home/eicarcom2.zip' was added.

Attributes:
 - Size: 308
 - Date: Thu Oct 17 07:09:26 2019
 - Inode: 14572
 - User: root (0)
 - Group: root (0)
 - MD5: e4968ef99266df7c9a1f0637d2389dab
 - SHA1: beclb52d350d721c7e22a6d4bb0a92909893a3ae
 - SHA256: e1105070ba828007508566e28a2b8d4c65d192e9eaf3b7868382b7cae747b397
 - Permissions: 100644

** Alert 1571296460.7684090: mail  - virustotal,gdpr_IV_35.7.d,
2019 Oct 17 07:14:20 (xixare-desa.lgp.ehu.es) 10.0.11.24->virustotal
Rule: 87105 (level 12) -> 'VirusTotal: Alert - /home/eicarcom2.zip - 53 engines detected this file'
{"virustotal": {"found": 1, "malicious": 1, "source": {"alert_id": "1571296458.7683499", "file": "/home/eicarcom2.zip", "md5": "e4968ef99266df7c9a1f06
37d2389dab", "sha1": "beclb52d350d721c7e22a6d4bb0a92909893a3ae"}, "sha1": "beclb52d350d721c7e22a6d4bb0a92909893a3ae", "scan_date": "2019-10-15 22:57:2
9", "positives": 53, "total": 60, "permalink": "https://www.virustotal.com/file/e1105070ba828007508566e28a2b8d4c65d192e9eaf3b7868382b7cae747b397/analy
sis/1571180249/"}, "integration": "virustotal"}
virustotal.found: 1
virustotal.malicious: 1
virustotal.source.alert_id: 1571296458.7683499
virustotal.source.file: /home/eicarcom2.zip
virustotal.source.md5: e4968ef99266df7c9a1f0637d2389dab
virustotal.source.sha1: beclb52d350d721c7e22a6d4bb0a92909893a3ae
virustotal.sha1: beclb52d350d721c7e22a6d4bb0a92909893a3ae
virustotal.scan_date: 2019-10-15 22:57:29
virustotal.positives: 53
virustotal.total: 60
virustotal.permalink: https://www.virustotal.com/file/e1105070ba828007508566e28a2b8d4c65d192e9eaf3b7868382b7cae747b397/analysis/1571180249/
integration: virustotal
```

Imagen 16 - Log de la detección de Virus (Virustotal)

Además de ver en los sistemas de logs del servidor Wazuh la alerta anterior, el sistema en la parte Web nos creará la alerta para una mejor visualización de ella, como podemos ver en la siguiente imagen:

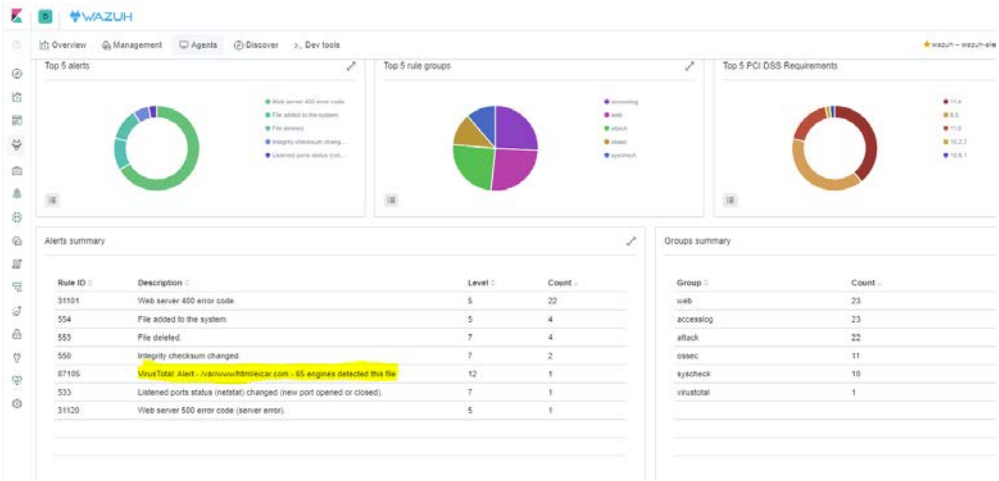


Imagen 17 - Kibana-Wazuh-VirusTotal Alerta

[9] https://www.eicar.org/?page_id=3950

3.4.3. DETECCIÓN DE ATAQUE DE FUERZA BRUTA VÍA SSH

Como framework de seguridad que este sistema, tiene un módulo encargado de detectar amenazas y comportamientos anómalos. En este ejemplo podemos observar como detecta un ataque SSH de fuerza bruta.

Para conseguir explotar esta alerta de seguridad realizamos un ataque ssh de fuerza bruta desde un Kali[10], con el siguiente comando:

```
hydra -l root -P /usr/share/wordlists/rockyou.txt 10.10.13.7 -t 4 ssh
```

Una vez que se ha realizado el ataque, podemos comprobar en la parte de Overview/Sedurity Alerts, que algo anómalo ha ocurrido ya que una de las máquinas está reportando muchas alertas, en este caso es el servidor "Xixare":

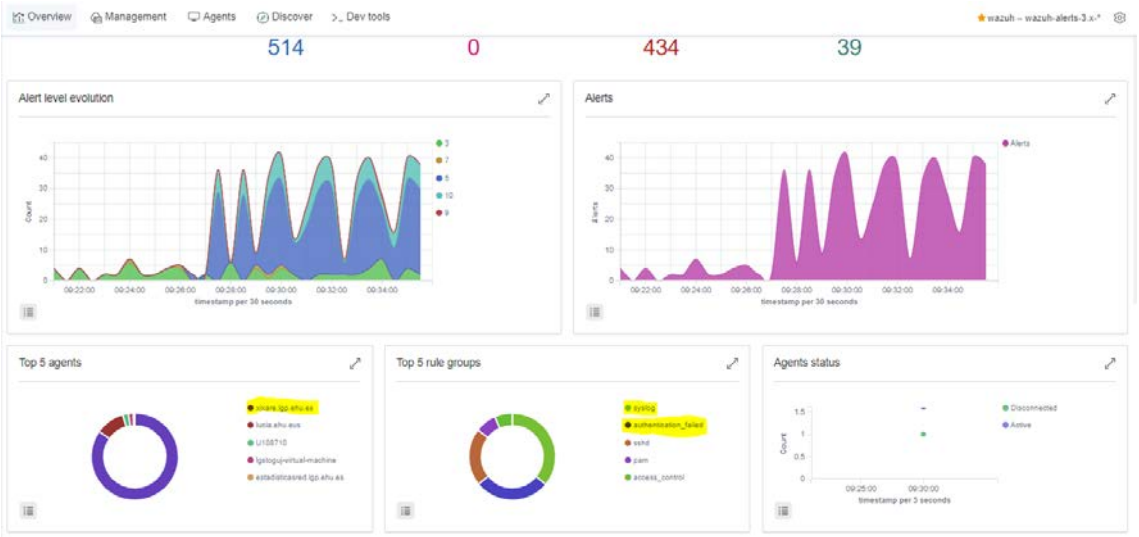


Imagen 18 - Interface de las alertas de seguridad.

Podemos ver en la misma página que la regla que más se está activando es la 5716: sshd: authentication failed. (en los últimos 15 minutos) como la que más veces se ha activado, lo que nos lleva a pensar que es un ataque vía ssh de fuerza bruta, ya que lleva contabilizados 253 intentos:

Imagen 19 - Alertas de seguridad- SSH

Este es un simple ejemplo, pero que, sin duda alguna, nos da mucha información de un solo vistazo y podemos ver la cantidad de ataques que podemos estar sufriendo de ssh, escaneo de puertos, etc.

[10] https://www.kali.org/

3.4.4. DETECCIÓN DE ATAQUE SHELLSHOCK Y SU REMEDIACIÓN

Últimamente y siguiendo las recomendaciones de seguridad en entornos Web, las comunicaciones con los servidores se realizan vía SSL, lo que dificulta en muchas ocasiones a los firewalls perimetrales analizar las conexiones entre los clientes y los servidores Web, ya que estas van cifradas y les es imposible detectar ataques a los servidores.

Este ejemplo que comentaremos a continuación, usa el poder de tener un agente dentro del equipos y la posibilidad de estar analizando los logs que genera el sistema Web en búsqueda de patrones que concuerden con ataques al servidor.

En este punto simulamos el ataque Shellshock (ataque de bash donde un atacante podría ejecutar comandos en el servidor remoto y robar información, manipularla, etc.) para comprobar que se detecta este tipo de ataques en servidores web.

Para que nuestro servidor sea capaz de detectar este tipo de ataque (en la documentación de Wazuh, podemos ver muchos más ataques que puede detectar y que se pueden parametrizar) debemos realizar una serie de adaptaciones en las configuraciones tanto del servidor Wazuh como en la del servidor endpoint. Los cambios que debemos realizar son los siguientes:

1. Debemos adaptar los logs que genera nuestro servidor Web apache donde tenemos unas de las aplicaciones internas de la institución. Este cambio consiste en dejar ciertos logs en un fichero que posteriormente serán leídos por el agente de Wazuh. En nuestra versión del redhat con apache, modificamos el fichero "/etc/httpd/conf.d/ssl.conf" para que deje el log en cierto fichero de logs y con cierto formato, ya que las reglas de Wazuh necesitan que el logs tengan cierto formato para poder ser analizados:

```
CustomLog logs/ssl_request_log  "%h - - %t \"%r\" %b \"%{Referer}i\" \"%{User-agent}i\""
```

2. Modificamos el fichero: /var/ossec/etc/ossec.conf para que el agente instalado en el servidor analice el fichero anterior (ssl_request_log) (por defecto en la configuración de los agentes, este fichero no es analizado). Para poder hacerlo debemos añadir las siguientes instrucciones en la sección "<ossec_config>":

```
<localfile>
  <log_format>apache</log_format>
  <location>/var/log/httpd/ssl_request_log</location>
</localfile>
```

   Y rearrancar el servicio: systemctl restart wazuh-agent
3. Para simular el ataque desde otro equipo Linux externo, ejecutamos el siguiente comando:

```
curl --insecure https://xixare.lgp.ehu.es -H "User-Agent: () { :; }; /bin/cat /etc/passwd"
```

4. En la interface de Wazuh en ELK, nos aparecen las alertas de este ataque, como podemos observar en la siguiente imagen:

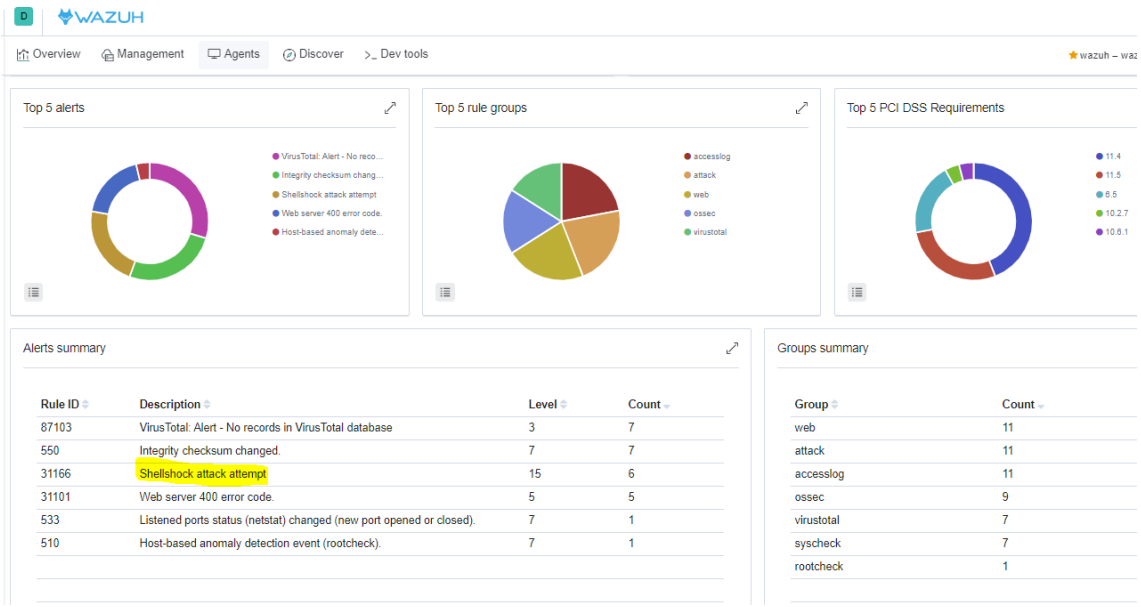


Imagen 20 - Tabla de alertas del endpoint - Shellshock

- Para poder parametrizar el formato del logs y que sea detectado, nos hemos basado en la utilizad que podemos encontrar en los servidores Wazuh "/var/ossec/bin/ossec-logtest" donde, una vez ejecutada, añadimos una línea de log, y nos muestra en que decodificador y reglas consigue hacer emparejamiento.

```
Croot@wazuh:/var/ossec/etc# /var/ossec/bin/ossec-logtest
2019/12/04 16:07:16 ossec-testrule: INFO: Started (pid: 29448).
ossec-testrule: Type one log per line.

10.0.3.102 - - [04/Dec/2019:16:52:06 +0100] "GET / HTTP/1.1" 4473 "-" "() { :; }; /bin/cat /etc/passwd"


**Phase 1: Completed pre-decoding.
       full event: '10.0.3.102 - - [04/Dec/2019:16:52:06 +0100] "GET / HTTP/1.1" 4473 "-" "() { :; }; /bin/cat /etc/passwd"'
       timestamp: '(null)'
       hostname: 'wazuh'
       program_name: '(null)'
       log: '10.0.3.102 - - [04/Dec/2019:16:52:06 +0100] "GET / HTTP/1.1" 4473 "-" "() { :; }; /bin/cat /etc/passwd"'

**Phase 2: Completed decoding.
       decoder: 'web-accesslog'
       srcip: '10.0.3.102'
       protocol: 'GET'
       url: '/'
       id: '4473'

**Phase 3: Completed filtering (rules).
       Rule id: '31166'
       Level: '15'
       Description: 'Shellshock attack attempt'
       Info - CVE: 'CVE-2014-6271'
       Info - Link: 'https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271'
**Alert to be generated.
```

Imagen 21 - Imagen de ossec-logtest

No solo podemos detectar los ataques y disparar la correspondiente alerta, podemos ir más lejos, con la capacidad de interactuar con el agente y la posibilidad que tiene este de ejecutar comando en el endpoint. En este caso usamos la opción que tiene que al ser disparada cierta alerta, esta interactúe con el Firewall del equipo y filtre en el Firewall local del endpoint la IP que está atacando.

Para implementar esta funcionalidad, debemos añadir cierta configuración en el fichero: /var/ossec/etc/ossec.conf de TODOS los servidores Wazuh (tonto Mestos como trabajadores), en nuestro caso es cuando se dispare la regla 31166, mandamos la orden para que filtre en el firewall por 5 minutos.

Con las siguientes instrucciones conseguiremos realizar dicha acción:

```
<active-response>
  <disabled>no</disabled>
  <command>firewall-drop</command>
  <location>local</location>
  <rules_id>31166</rules_id>
  <timeout>300</timeout>
</active-response>
```

Tras provocar el ataque usando el comando que fue usado anteriormente podemos ver en que ha sido filtrado en el FW del equipo:

```
[root@xixare etc]# iptables --list -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  10.0.3.102          0.0.0.0/0
```

Imagen 22 - Imagen del FW- remediación activa

3.4.5. EJEMPLO DE ANÁLISIS DE LA POLÍTICA DE BASTIONADO DE LA INSTITUCIÓN: OPENSCAP

Otras de las grandes funcionalidades que tiene esta herramienta, es poder analizar los servidores para comprobar si cumplen la política de bastionado de la organización. Por ejemplo, que no tenga activado IPV6, límite mínimo de la longitud de los passwords, activado el firewall local, etc. Este módulo se basa en openscap y un formato dado para generar todas estas reglas.

Para comprobar este módulo y ver cómo funciona, usaremos de ejemplo las reglas de bastionado usadas en nuestra institución para los servidores Linux. Las reglas usadas y que posteriormente deberemos crear en el formato apropiado para que las pueda interpretar el sistema openscap son las siguientes:

1. Prohibir el uso de validación como root.
2. Root único usuario con UID 0
3. Comprobar que nos existen cuentas sin password.
4. Comprobar que el password de las cunetas no está en /etc/passwd
5. Long passwod mínimo 10.
6. Máxima vigencia del password 90 días.
7. Máximo de sesiones por usuario=2
8. Firewalld debe estar habilitado.
9. Logs:
    a. Logs deben pertenecer al usuario apropiado
    b. Que se roten los logs
    c. Loggear todos los cambios en ficheros críticos como: passwd, shadow…
    d. Loggear información de comandos uso privilegiado: su, sudo, etc
10. Deshabilitar ciertos servicios como: DHCP server, DNS server
11. Habilitar servicios como NTP y configurarlo,

Estas reglas están basadas en gran parte por las recomendadas por el CCN-CERT para servidores catalogados de seguridad Media.

https://www.ccn-cert.cni.es/pdf/guias/series-ccn-stic/guias-de-acceso-publico-ccn-stic/3677-ccn-stic-619-implementacion-de-seguridad-sobre-centos7-anexoa/file.html

Como se ha comentado anteriormente, las reglas deben formatearse en un lenguaje que entienda este módulo, para realizar esta labor, nos basamos en la herramienta "SCAP Workbench" que podemos bajarnos de https://www.open-scap.org/tools/. EN la siguiente imagen podemos ver una captura de dicha implementación usando esta herramienta.

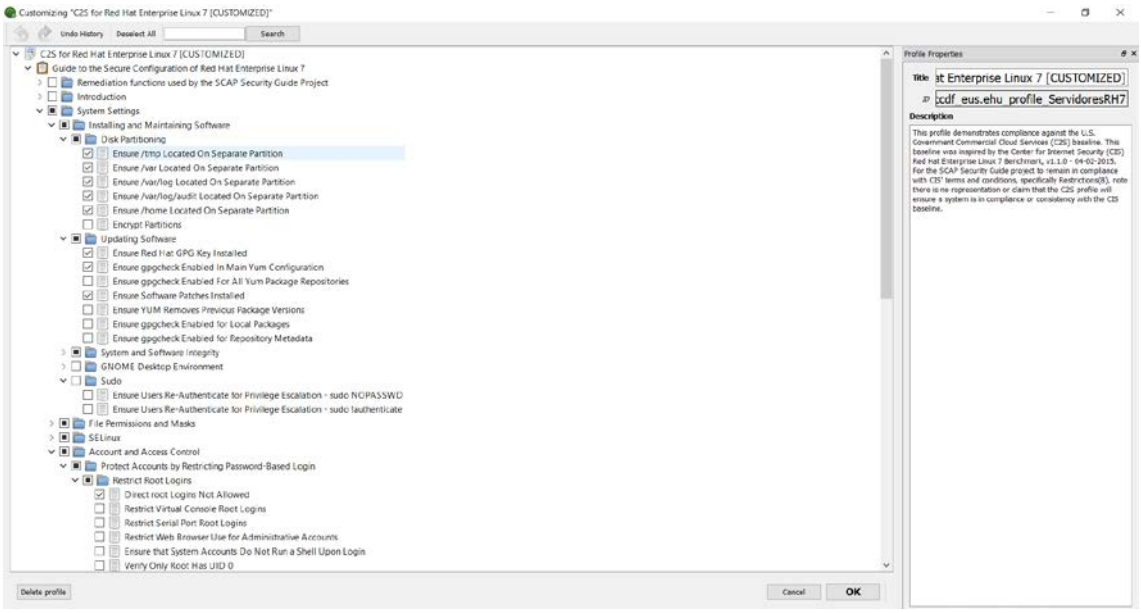


Imagen 23 - Herramienta SCAP Workbench para parametrizar reglas.

Usando esta herramienta, cargaremos las reglas básicas de RedHat 7 que nos da la propia tool, y la adaptamos a las necesidades descritas anteriormente. En nuestro caso hemos creado un nuevo "Profile" : "xccdf_eus.ehu_profile_servidores", el cual hemos añadido al fichero XML con todas las definiciones para redHat 7: "ssg-rhel-7-ds.xml".

Pero, para poder usar openscap el primer paso que debemos de dar es instalarlo en el servidor (endpoint), esta instalación se realiza, en nuestro caso en un Red Hat, con el siguiente comando:

```
yum install openscap-scanner
```

Una vez instalado el paquete anterior, y editada y dada un nombre a la política que usaremos de bastionado, debemos seguir ciertos pasos para activar en el servidor Wazuh y en el cliente esta funcionalidad. A continuación, describiremos los pasos que hemos dado para activarla.

Configuraciones en el servidor Wazuh:

1. En el Servidor Wazuh, configuramos para que se active open-scap:
   /var/ossec/etc/ossec.conf

```
<wodle name="open-scap">
  <disabled>no</disabled>
<timeout>1800</timeout>
```

```
        <interval>1d</interval>
        <scan-on-start>yes</scan-on-start>
      </wodle>
```

2. Para las pruebas, hemos creado un grupo dentro de Wazuh llamado: “PruebaSCAP” donde hemos añadido los servidores en los que queremos probar esta funcionalidad.
3. Automáticamente se crea una carpeta llamada: “/var/ossec/etc/shared/PruebaSCAP” donde tenemos un fichero llamado: “agent.conf” donde configuraremos que “Profile” queremos que nos chequee y dependiendo de ciertas etiquetas.”

```
<agent_config profile="EHUSCAP">
 <wodle name="open-scap">
  <content type="xccdf" path="ssg-rhel-7-ds.xml">
   <profile>xccdf_eus.ehu_profile_servidores</profile>
  </content>
 </wodle>
</agent_config>
```

Configuraciones en el Cliente:

1. Modificamos el fichero ossec.conf (ubicado en: /var/ossec/etc) donde le añadiremos la etiqueta “EHUSCAP” ya que solo queremos que los servidores que pertenezcan al grupo PuebaSCAP y que tengan la etiqueta “EHUSCAP” se les aplique el chequeo predefinido:

```
<ossec_config>
 <client>
  <server>
   <address>10.0.11.42</address>
   <port>1514</port>
   <protocol>tcp</protocol>
  </server>
  <config-profile>rhel, rhel7, rhel7.6,EHUSCAP</config-profile>
```

2. Añadimos el nuevo fichero XML con las políticas creadas sobre nuestra institución en el siguiente directorio: “/var/ossec/wodles/oscap/content” en nuestro caso el fichero se sigue llamando igual que antes, con lo que sobrescribimos el fichero anterior: “ssg-rhel-7-ds.xml”

Reiniciamos el proceso en el servidor Wazuh con el comando:

```
systemctl restart wazuh-manager
```

Reiniciamos los agentes desde el servidor Wazuh con el comando:

```
/var/ossec/bin/agent_control -R -a
```

Podemos probar el perfil que se ha creado desde el propio equipo monitorizado. Para poder realizarlos usaremos el siguiente comando:

```
oscap xccdf eval --profile xccdf_eus.ehu_profile_servidores --results Resultado.xml --report Reporte.html /var/ossec/wodles/oscap/content/ssg-rhel-7-ds.xml
```

Tras analizar el host final, podemos observar mediante la interface de Kibana, los resultados del análisis de que políticas marcadas por la institución, NO se están cumpliendo: a modo de ejemplo podemos ver en la siguiente imagen para la política definida cuál de las políticas catalogadas como de severidad Alta, no se está cumpliendo:

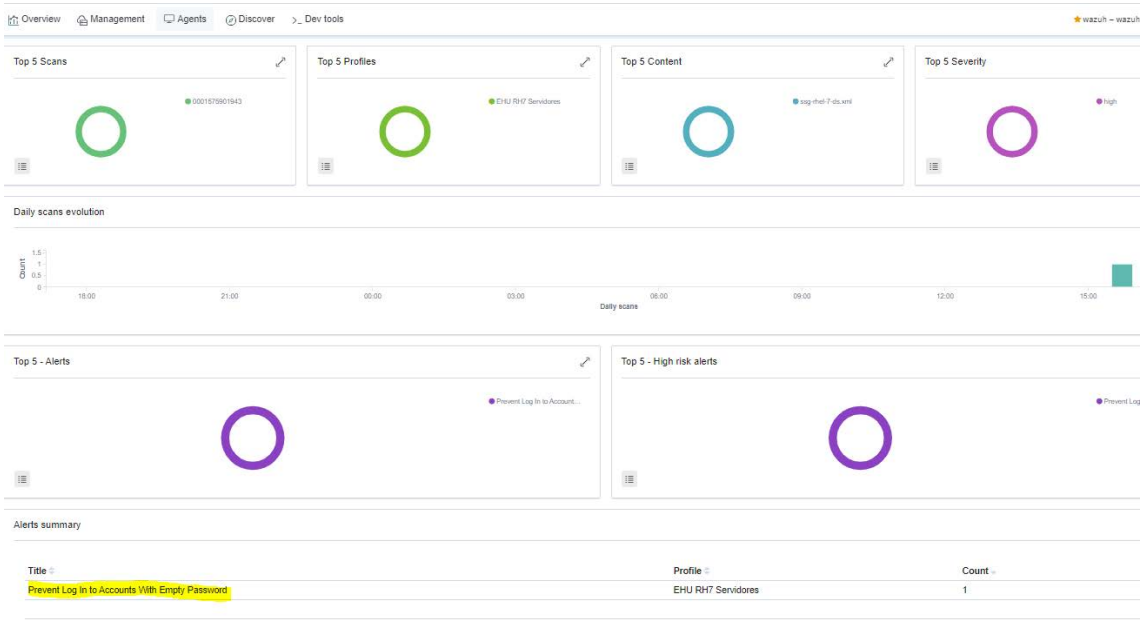


Imagen 24 - Política de severidad alta no cumplida.

En la siguiente imagen podemos ver el resultado total de las que este servidor en cuestión, no cumple con la política de la institución:

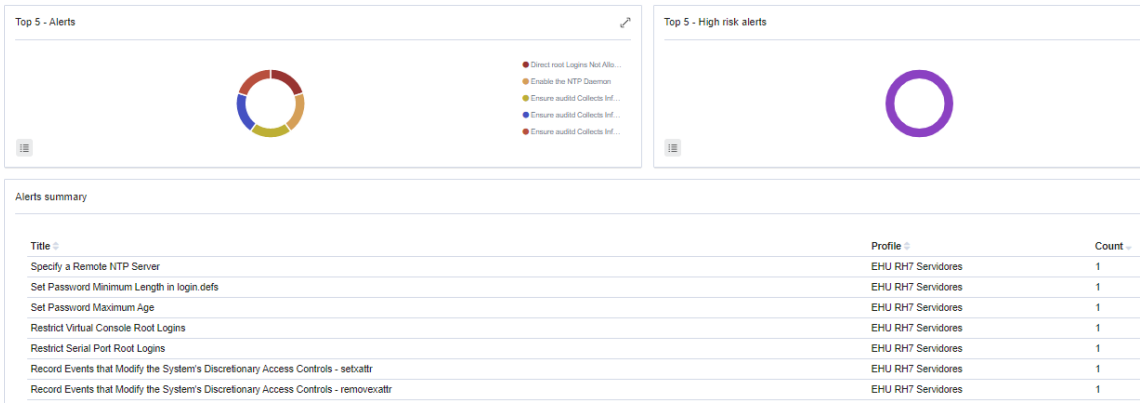


Imagen 25 - Bastionado: Lista de las políticas no cumplidas

4. CONCLUSIONES

En este TFM, se ha realizado la instalación y configuración de Wazuh teniendo como premisa que se deseaba realizar una instancia tolerante a fallos y escalable. A su vez, se han parametrizado/configurado ciertos módulos opciones que implementa Wazuh, dándole a la solución un punto más de seguridad sobre los endpoint. Estas funcionalidades extras que se han configurado y que se han visto a lo largo de esta documentación son las siguientes: VirusTotal, detector de Vulnerabilidades, análisis de cumplimiento de normativa o bastionado de los servidores (OpenSACAP), …

Para realizar estas conclusiones tomaremos usaremos los datos extraídos del uso de esta herramienta durante el periodo que ha durado este TFM, analizado las ventajas que nos ofrece esta, a las cuales añadiremos, según nuestro punto de vista, algunas mejoras que vemos que serían de gran importancia para tener un framework de seguridad completo.

4.1 RESULTADOS DE LA POC Y SUS CONCLUSIONES

La mejor forma de sintetizar la experiencia recibida de la herramienta, es analizarla desde el punto de vista de cada módulo, dando a conocer que ventajas o inconvenientes hemos tenido con ella. Es por ello, que analizaremos estos módulos de forma resumida en nuestra POC en los siguientes puntos:

Eventos de Seguridad

Como herramienta que se describe como “plataforma de seguridad”, debemos comentar por el módulo, sino el más importante uno de ellos que es el sistema de eventos de seguridad, debemos recordar que todo lo que digamos en este punto, está basado en los resultados obtenidos del análisis de los servidores usados en esta POC. En este módulo que visualizamos en la interface de Kibana, nos muestra los eventos de seguridad detectados, siempre basados en una serie de reglas predefinidas, las cuales podemos complementar con otro tipo de reglas que desarrollemos o que obtengamos de otras fuentes. Tenemos varias opciones, o visualizar los eventos de seguridad en general para todos los servidores monitorizados, o agente por agente que nos darán los eventos para un único equipo final. En cada evento, nos dará el número de veces que se produjo esa alarma, siempre referido al espacio de tiempo que estemos analizando, 15 minutos, 1 hora etc.

De este modo podremos detectar si realmente es un ataque, o simple mente un error del sistema cliente. Por ejemplo, podemos ver en la siguiente imagen una alerta de fallo de autenticación de SSH.

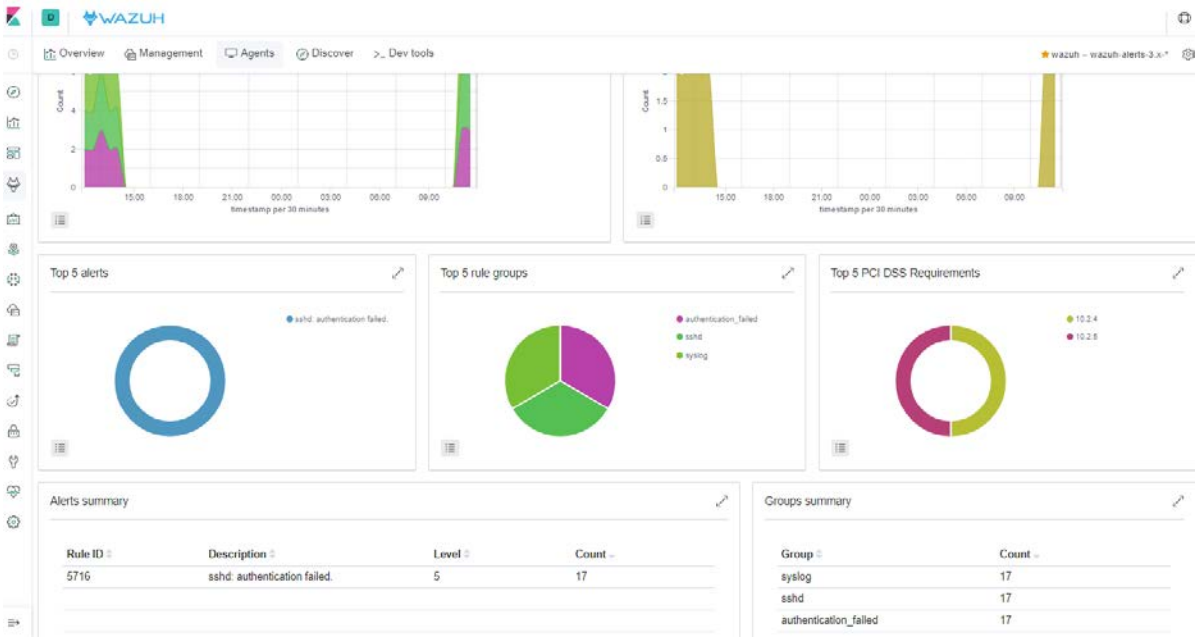


Imagen 26 - SSH accesos fallidos

Quedaría en nuestras manos analizar si realmente es un ataque o simplemente un fallo de validación de los usuarios. Para ello deberemos analizar los logs reportados, IP de origen, formato de la cuneta usada, etc., pero como podemos ver, nos marca ya la alerta para poder analizarla. Sí que es verdad que podemos parametrizarla, para que la alerta salte cada x intentos fallidos en cierto tiempo, etc.

Monitor de Integridad

Este monitor nos da información de los ficheros que se están modificando, borrando, etc., nos da una idea de que está pasando realmente dentro del servidor, lo cual nos puede ayudar a configurar mejor el sistema, por ejemplo, dentro de las pruebas realizadas en este POC, vemos que uno de los servidores Web, modifica continuamente un fichero que en vez de estar en el directorio de logs, está en un directorio accesible vía web y que es un log de lo que está haciendo la aplicación. En este caso, nos sirve para para decirles a los desarrolladores que los ficheros de logs de la propia aplicación web, deben estar ubicados en el director de logs.

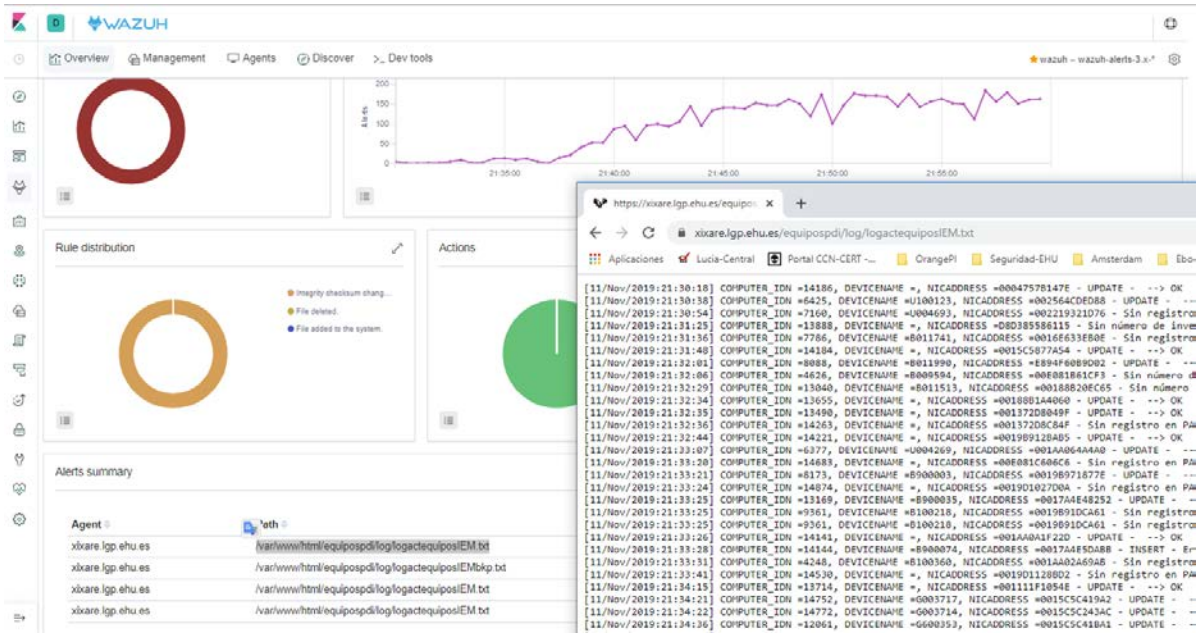


Imagen 27 - Monitor de Integridad

Vulnerabilidades.

El sistema usado para la detección de las vulnerabilidades, está basado en las CVEs publicadas sobre los diferentes sistemas operativos y sus versiones. Este sistema, nada más ponerlo en producción, ha sido de gran utilidad, ya que hemos encontrado errores en el parcheo de seguridad de algunos servidores.

Por ejemplo, nos ha pasado en uno de los servidores un RedHat 6, que, aunque la herramienta de RedHat (Satellite) así como vía CLI, nos decía que ya tenía puesto todos los parches de seguridad, en el servidor Wazuh seguía saltando una alerta de una vulnerabilidad, y analizándola, dentro de la misma plataforma de Wazuh, pudimos ver que, realmente Red Hat no había sacado un parche para esa versión. En la siguiente imagen podemos ver dicha vulnerabilidad. Lo cual nos recuerda, que se debe migrar ese servidor a una versión más reciente del sistema operativo.

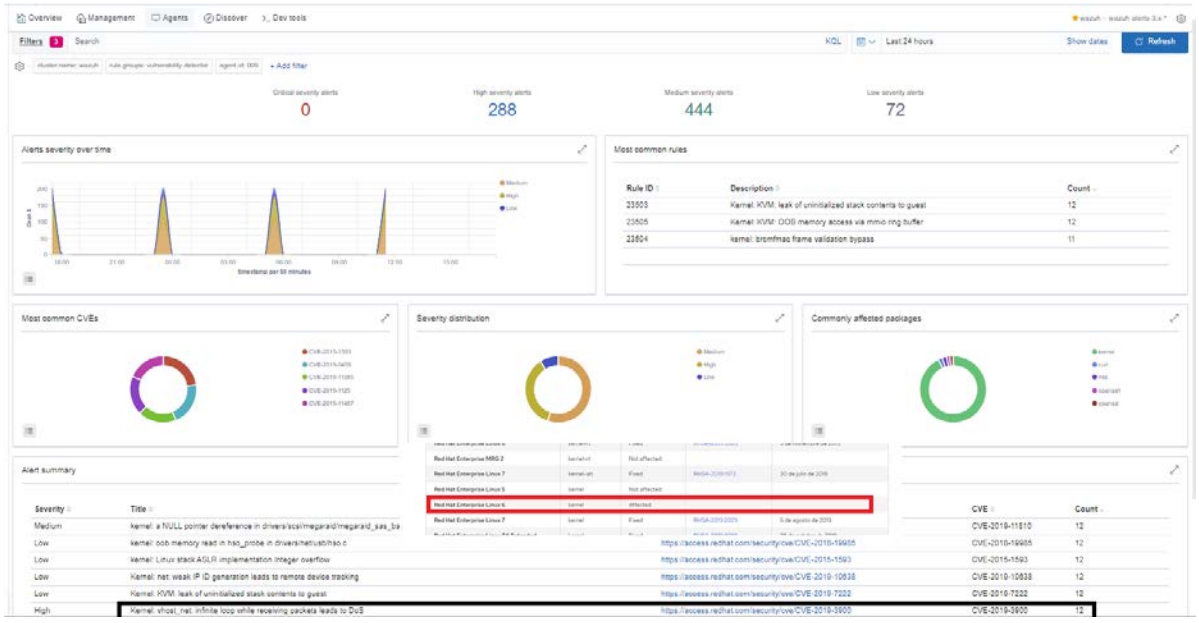


Imagen 28 - Vulnerabilidad no parcheada

Virus Total.

Este módulo, lo consideramos muy interesante ya que puede dar seguridad contra virus dentro de los servidores que no dispongan de antivirus, por recomendaciones del software instalado en el servidor, por no disponer de antivirus para esa plataforma, etc. Por ejemplo, podemos dar seguridad contra antivirus a un almacén de datos, que vía https o montando la unidad pueden subir ficheros de cualquier índole. Sí que es verdad que para un pleno funcionamiento de este módulo deberemos contratar con VirusTotal la posibilidad de quitar el límite de ficheros que se pueden analizar. En este Poc, usando la versión gratuita del API y con pocos ficheros cambiantes en los servidores, hemos comprobado que llegábamos al límite que nos ofrece VirusTotal de análisis por minutos:

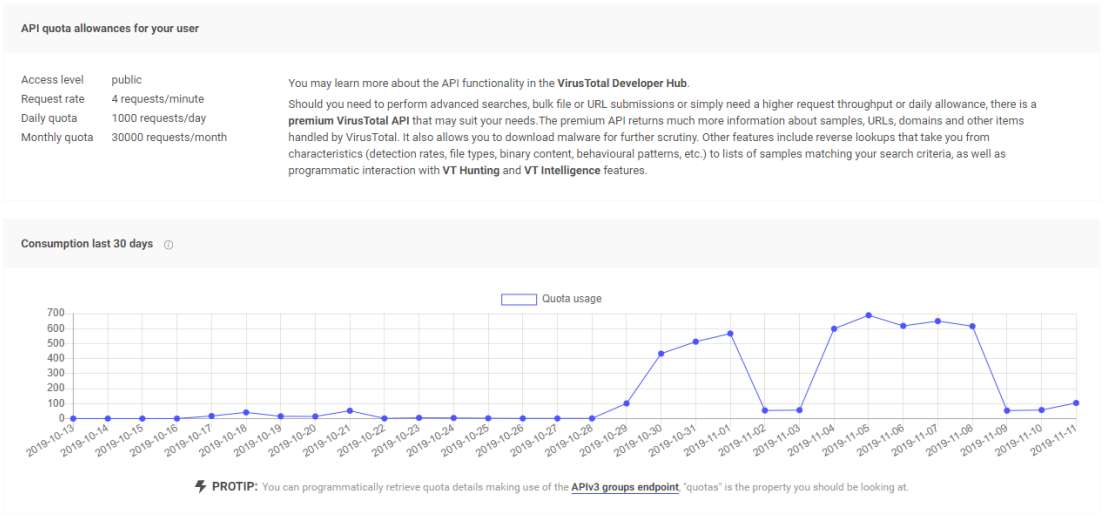


Imagen 29 - VirusTotal API estadísticas

Aun así y pensando a futuro, no sería mala idea contratar el API que permite el análisis de todos los ficheros que se instalen en ciertas plataformas a las cuales no se les podría dar seguridad ante virus y teniendo en cuneta que VirusTotal, no usa solouna base de datos de

virus de un solo fabricante, por el contrario, usa múltiples fabricantes, con lo que la seguridad, estará más asegurada.

Algunas desventajas o posibles mejoras para futuras versiones.

Tras el uso intensivo de la herramienta durante estos meses encontramos algunas carencias o desventajas a la hora de utilizarla diariamente, las cuales entendemos que se solventarán en futuras versiones, pero creo que es bueno puntualizarlas y tenerlas en cuenta.

Comenzaremos por mencionar que muchas de las parametrizaciones que se realizan se deben realizar directamente en los servidores Wazuh, vía comando, sería deseable que, desde la interface Web, desde Kibana, se pudiese parametrizar sin necesidad de entrar en los servidores.

El sistema de vulnerabilidades en la versión actual 3.10, solo soporta las versiones Linux, y sería de gran ayuda que también se integrase con las Versiones de Windows, de este modo podríamos tener en una única herramienta el estado de todos los servidores de la institución, e incluso añadir sistemas externos como Switches, routers, Firewalls, etc. En la web comentan que están trabajando en añadir nuevos sistemas. (**Durante las últimas revisiones de esta documentación a fecha de 27/12/2019 han liberado la versión 3.11, la cual da soporte a la detección de vulnerabilidades en Windows.**)

Sistemas monitorizados sin agentes (agentless), sería deseable que apareciesen en la interface en el Kibana, para tener un repositorio de ellos y tener monitorizado cuáles de ellos tenemos. Ahora mismo la única forma es vía CLI desde el propio servidor Wazuh, debiendo de buscar las alertas de ellos de forma muy manual y no pudiendo desde la opción de Agentes ir directamente a ellos y ver sus posibles alertas.

Conclusión.

Para finalizar, podemos concluir que esta herramienta, Wazuh, es una herramienta muy potente que ofrece muchas funcionalidades para tener un framework completo de seguridad y sobre todo seguridad desde el punto de vista del endpoint, que a fin de cuentas es el que se desea proteger y es teste el que nos dará las pistas de si un ataque ha tenido efecto, si han conseguido entrar en el sistema, que cambios se han realizado etc. Y no debemos olvidar que es una herramienta modulable que se puede complementar con otros módulos que se desarrollen como se ha visto en este TFM con el módulo de VirusTotal.

Además, integra nativamente la capacidad de analizar los endpoint en base al cumplimiento de normativas como la RGPD o NIST, pudiendo ser complementado con normativas propias que tenga la institución y como se ha visto en los ejemplos de configuración. Y como no, no podemos olvidar que esta herramienta es gratuita, que suele ser uno de los impedimentos para montar cualquier tipo de herramienta de seguridad, ya que suelen ser caras.

Por otra parte, no creamos que el uso de esta herramienta es posible por cualquier persona. Como toda herramienta de seguridad, implica del uso de expertos en seguridad y sistemas operativos para poderla parametrizar y poderle sacar el máximo jugo de ella. Es por ello que, aunque es gratuita, fácil de instalar y existir guías para configurar, son guías de mínimos, y dependerá del analista de seguridad el hilar más fino en todas las alertas, parametrizaciones, configuraciones, niveles de los disparadores de las alertas, etc.

4.2 MEJORAS Y TRABAJO FUTURO.

Como Mejoras o trabajo futuro, se tomarán ciertas medidas para darle más importancia dentro de la organización a esta herramienta. Estas medidas serán las siguientes:

Asegurar las comunicaciones.

Una vez que ya tenemos una implementación estable del sistema y además escalable, toca planificar los siguientes pasos para mejorar la instalación realizada. Analizando el sistema montado, vemos algunas deficiencias en cuanto a seguridad, es decir, vemos la falta de seguridad de acceso al sistema web que gestiona todo el sistema, así como las comunicaciones entre los servidores que dan esta solución. Para solventar estos agujeros de seguridad en los siguientes meses se tomarán las siguientes medidas:

- Asegurar los accesos a ELK con cunetas de usuario, ahora mismo solo está filtrado en el Firewall vía IP.
- Cifrar vía TLS el acceso a ELK usando certificados.

(Para realizar este trabajo seguiremos la guía que podemos encontrar en: https://documentation.Wazuh.com/3.10/installation-guide/installing-elastic-stack/protect-installation/index.html)

Despliegue en el resto de servidores.

Despliegue en el resto de servidores de la organización. Una vez que se presente este trabajo, así mismo será presentado internamente para su aprobación y despliegue en el resto de servidores de la organización. Ahora mismo está desplegado en los servidores del área de Red.

Actualización a la nueva versión 3.11.

Mientras se finaliza esta documentación y se estaba preparando para la entrega a fecha 27 de diciembre de 2019, se ha liberado la versión 3.11, la cual da soporte a las vulnerabilidades en sistemas Operativos Windows. Como trabajo a muy corto plazo, comenzaremos con la actualización de los servidores y de los agentes a la nueva versión.

Dicha actualización se realizará manualmente en los servidores Wazuh, pero en los endpoints, usaremos las herramientas vista en esta documentación y anteriormente usadas para el despliegue de los agentes.

5. TRABAJOS CITADOS

Concha, Felipe. (12/02/2019). Sistemas IDS, IPS, HIDS, NIPS, SIEM ¿Qué son? Recuperado de https://www.a2secure.com/blog/ids-ips-hids-nips-siem-que-es-esto/

Mundhada, Chetan. (23/01/2019) The Evolution of SIEM. Recueprado de https://www.darkreading.com/vulnerabilities---threats/the-evolution-of-siem/a/d-id/1333675

Ramiro, Ruben (28/02/2018). Métodos de búsqueda de amenazas de ciberseguridad. Recuperado de https://ciberseguridad.blog/metodos-de-busqueda-de-amenazas-de-ciberseguridad/

Día Internacional de la Seguridad Informática. (2019) Recuperado de https://www.internautas.org/seguridad/html/10147.html

La seguridad vista desde sus inicios. (11/03/2015) Recuperado de https://www.incibe.es/protege-tu-empresa/blog/seguridad-desde-inicio

Los pros y los contras de la virtualización. (26/05/2009) Recuperado de https://www.networkworld.es/actualidad/los-pros-y-los-contras-de-la-virtualizacion

6. GLOSARIO

CIS: Center for Intenet Security
CVE. Common Vulnerabilities and Exposures
ELK o ELK Stack: Elasticsearch, Logstash, Kibana
GDPR o RGPD: Reglamento General de Protección de Datos
HIDS: Host Intrusion Detections Systems
IOCs: Indicadores de compromiso
IPS: Intrusion Prevention Systems
NIDS: Network Intrussion Detections Systems
NIST: National Institute of Standards and Technology
POC. Una prueba de concepto o PoC (del inglés proof of concept)
SCA: Security Configuration Assessment
SIEM: Security Information and Event Management
TFM. Trabajo de Fin de Master.
TIC: Tecnologías de la Información y la Comunicación