UNIVERSIDAD INTERNACIONAL SEK

DIGITAL SCHOOL

TRABAJO DE FIN DE CARRERA

TITULADO:

SISTEMA DE GESTIÓN DE EVENTOS E INFORMACIÓN DE SEGURIDAD (SIEM) DE LA INFRAESTRUCTURA TECNOLÓGICA DE LA UNIVERSIDAD INTERNACIONAL SEK DEL ECUADOR

Realizado por:

Ing. Jorge Danilo Añazco Bedón

Directora del proyecto:

Ing. Verónica Rodríguez Arboleda, MBA.

Como requisito para la obtención del título de:

MAGISTER EN CIBERSEGURIDAD

QUITO, septiembre 2021

DECLARACION JURAMENTADA

Por la presente, yo, JORGE DANILO AÑAZCO BEDÓN, con cédula de ciudadanía Nº. 1724515596, declaro bajo juramento, que el trabajo aquí desarrollado es de mi autoría, que no ha sido previamente presentado para ningún grado a calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento. A través de esta declaración cedo mis derechos de propiedad intelectual de autora a la UNIVERSIDAD INTERNACIONAL SEK UISEK, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente

_____

JORGE DANILO AÑAZCO BEDÓN

CC: 1724515596

DECLARACIÓN DE DIRECTOR DE TESIS

El presente que el presente trabajo de investigación titulado:

"SISTEMA DE GESTIÓN DE EVENTOS E INFORMACIÓN DE SEGURIDAD (SIEM) DE LA INFRAESTRUCTURA TECNOLÓGICA DE LA UNIVERSIDAD INTERNACIONAL SEK DEL ECUADOR"

Realizado por:

Ing. Jorge Danilo Añazco Bedón

Como requisito para la obtención del título de

MÁSTER EN CIBERSEGURIDAD

Ha sido dirigido por mi persona a través de reuniones periódicas con la estudiante y cumple con todas las disposiciones que rigen los trabajos de titulación.

_____

Ing. Verónica Rodríguez Arboleda,

MBA. DIRECTORA DEL PROYECTO

CC: 1707522312

LOS PROFESORES INFORMANTES

Los profesores informantes:

Ing, José Vinicio Freire Rumazo, Mgtr.

MSc Ing. Joe Carrión Jumbo, PhD

Después de revisar el trabajo, lo han calificado como apto para su defensa oral ante el tribunal examinador

El profesor informante:

_____ _____

Ing. Joe Carrión Jumbo, PhD Ing, José Vinicio Freire Rumazo, Mgtr.

Quito, septiembre de 2021

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es de mi autoría, que se han citado las fuentes correspondientes y que en su desarrollo se respetaron las disposiciones legales vigente, que protegen los derechos de autor.

Jorge Danilo Añazco Bedón

CC: 1724515596

AGRADECIMIENTO

A mi esposa Priscila por estar siempre a mi lado y ser un pilar fundamental para mi formación profesional, y a mis hijas Daniela y Alejandra les dedico este trabajo ya que por ellas tengo la determinación para cursar la maestría.

A mis padres Jorge y María que por su esfuerzo, sacrificio y paciencia pude estudiar la carrera que siempre me interesó, contando siempre con su apoyo incondicional en los momentos que más lo necesité.

A mi tía Enmita ya que fue una de las personas que ha confiado en mí dándome su cariño y apoyo para lograr mis objetivos.

A los docentes de la facultad Digital School de la Universidad Internacional Sek, por haber impartido sus conocimientos a lo largo de la preparación de esta maestría, de manera especial, a la Ing. Verónica Rodríguez Arboleda tutora de este proyecto de investigación quien me ha guiado con sus conocimientos, paciencia y rectitud como docente.

RESUMEN

In the following project, the implementation of a Security Information and Event Management System (SIEM) was carried out, due to the need to optimize and improve the cybersecurity of the technology infrastructure of the SEK International University, in addition control and monitoring procedures were defined of computer incidents that are detected through customized developments for monitoring with the availability module of the OSSIM AlienVault tool. Previously, with the 20 Center for Internet Security, an analysis was carried out to obtain indicators of the current state of the institution's information security, with which it was possible to determine the systems and unprotected areas that needed the immediate intervention of the module of availability or monitoring that the OSSIM AlienVault tool has integrated.

Finally, this document explains a guide for the development of a server and services monitoring script, which facilitates obtaining the necessary information to define alert states and notify when there is a risk in computer security.

Palabras claves: SIEM, OSSIM AlienVault, 20 controles de la CIS

ABSTRACT

In the following project, the implementation of a Security Information and Event Management System (SIEM) was carried out, due to the need to optimize and improve the cybersecurity of the technology infrastructure of the SEK International University, in addition control and monitoring procedures were defined of computer incidents that are detected through customized developments for monitoring with the availability module of the OSSIM AlienVault tool. Previously, with the 20 controls of the CIS Center for Internet Security, an analysis was carried out to obtain indicators of the current state of the institution's information security, with which it was possible to determine the systems and unprotected areas and which they needed the immediate intervention of the availability or monitoring module that has integrated the OSSIM AlienVault tool.

Finally, this document explains a guide for the development of a server and services monitoring script, which facilitates obtaining the necessary information to define alert states and notify when there is a risk in computer security.

Keywords: SIEM, OSSIM AlienVault, 20 CIS controls.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I

INTRODUCCIÓN

1.1 Planteamiento del Problema

Ecuador se encuentra en una situación compleja en relación a temas de ciberseguridad, debido a que se ha convertido en un objetivo de ataques cibernéticos, el cual ha incrementado a partir del retiró de asilo político a Julián Assange, programador, periodista y activista de internet australiano, conocido por ser el fundador, editor y portavoz del sitio web WikiLeaks. En el año 2019 según el Ministerio de Telecomunicaciones y de la Sociedad de la Información se confirmaron alrededor de 40 millones de ataques cibernéticos a entidades públicas y privadas (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2019), así como también se ha detectado un incremento anual de infecciones de malware en empresas ecuatorianas comparado a otros países de la región (El Comercio, 2021).

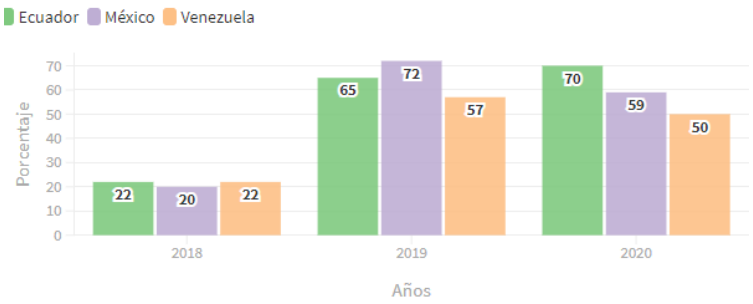


Figura 1: Infecciones de 'malware' en empresas de tres países latinoamericanos
Fuente: El Comercio (2021)

Por este motivo, Ecuador actualmente se encuentra entre los 32 países más atacados a nivel mundial según Kaspersky (Kaspersky, 2021), lo cual demuestra no estar preparado para

afrontar este volumen de ataques. Como antecedente, el pasado 23 de julio del 2021 CNT (Corporación Nacional de Telecomunicaciones) confirmó un ataque informático a sus sistemas con el virus de la familia RansomEXX, con el cual se sustrajeron y secuestraron información de los servidores (El Comercio, 2021).

Las Instituciones de Educación Superior (IES) no están libres de ataques a sus sistemas de información y manejan datos sensibles relacionados con aspectos económicos, información académica y del personal en general.

Este es el caso de la Universidad Internacional SEK, que desde la pandemia y el confinamiento el personal docente y administrativo ha tenido que adaptarse a trabajar desde casa, lo que ha conllevado a realizar la mayor parte de sus actividades de forma remota, de tal manera que ha aumentado considerablemente los riesgos de sufrir ataques informáticos, por la falta de familiaridad que existe con el teletrabajo y el bajo nivel de medidas de seguridad de la información.

El departamento de Recursos Tecnológicos de la Universidad Internacional SEK, no cuenta con un sistema que permita verificar el estado de la infraestructura tecnológica y a definir la situación actual de la mayoría de estos sistemas, por lo que, se ha presentado varios inconvenientes, como incidentes de seguridad de la información.

La falta de un sistema de análisis y monitoreo a las incidencias de seguridad que incluya el registro, detección, clasificación, priorización, resolución, seguimiento y cierre de la incidencia, repercute en problemas de funcionamiento de los sistemas informáticos, exponiendo la información sensible de la institución, además se desconoce la interacción de los sistemas tecnológicos con las actividades de la Universidad, ya sea de los estudiantes como del personal administrativo, por este motivo no es posible detectar fallos de rendimiento o mal funcionamiento, errores e intrusiones.

1.2 Formulación del Problema.

El departamento de Recursos Tecnológicos de la Universidad Internacional SEK, carece de un sistema de análisis y monitoreo de ficheros de registro (logs), que permita la atención a los posibles riesgos, incidencias o requerimientos a nivel de seguridad informática de la comunidad universitaria de una manera ordenada, oportuna y formal.

1.3 Objetivo general

Integrar un sistema de Gestión de Eventos e Información de Seguridad (SIEM), a través de la herramienta libre OSSIM AlienVault, para la detección de fallos de rendimiento y posibles ataques de seguridad a la infraestructura tecnológica de la Universidad Internacional SEK del Ecuador.

1.4 Objetivos específicos

- Analizar el estado actual de la infraestructura tecnológica de la Universidad Internacional SEK mediante *CIS Critical Security Control v6.1 Assessment Tool* para la determinación de los riesgos de seguridad que posee la entidad.

- Implementar un servidor Linux con la herramienta OSSIM AlienVault y las respectivas configuraciones de seguridad, acceso para monitoreo y control de riesgos de seguridad.

- Establecer los métodos para el comportamiento de los scripts, utilizando herramientas y tecnologías de monitoreo y obtención de datos, que permitan su eficiente desarrollo.

- Desarrollar scripts personalizados en lenguaje SHELL, a través del módulo de disponibilidad de la herramienta OSSIM AlienVault para la obtención de información de los posibles fallos de rendimiento, mal funcionamiento y

detección de posibles ataques de seguridad a la infraestructura tecnológica de la Universidad Internacional SEK del Ecuador.

1.5 Justificación

Técnica

En la actualidad, la mayoría de los sistemas tecnológicos generan registros, bitácoras relacionadas con su funcionamiento, toda esta información brinda indicios de posibles eventos que podrían ser catalogados como un riesgo para la entidad donde funciona, por esta razón, se han desarrollado y liberado herramientas que pueden ser personalizadas y adaptadas a las necesidades de la institución, lo cual permite analizar en qué estado de operatividad y seguridad se encuentra el sistema tecnológico, y así tomar acciones oportunas. En base a estas consideraciones se realiza el presente trabajo, en el que se implementa la herramienta OSSIM AlienVault de acceso libre, que obtiene y genera información para el análisis y control de los eventos e incidentes de seguridad, según los requerimientos de los sistemas tecnológicos de la Universidad Internacional SEK del Ecuador.

Social

De acuerdo a los objetivos de estudio, esta investigación se realiza debido a que es necesario conocer el estado de los sistemas tecnológicos de la Universidad Internacional SEK del Ecuador y su comportamiento de manera detallada, ya que podrían presentar riesgos en seguridad informática, que impacten al desarrollo de la información académica que manejan los estudiantes y personal administrativo, toda vez que dificultaría el cumplimiento de los objetivos institucionales, lo cual no es favorable para su prestigio.

Por esta razón, se desarrollan scripts personalizados para los diferentes servicios de la institución, lo que permite monitorear y alertar los posibles riesgos de seguridad y resguardar la información que se genera.

1.6 Estado del arte

Existen varias investigaciones con diferentes perspectivas de monitoreo de servicios y Gestión de Eventos de seguridad, a continuación, se indica algunos ejemplos:

En la investigación realizada en el año 2015, por Diana Joselyn Espinoza Villón, en su documento "Estudio de la herramienta de seguridad Open Source Security Information Management (OSSIM) en la Universidad Tecnológica Empresarial de Guayaquil (UTEG)" se manifiesta:

- El software OSSIM AlienVault puede indicar posibles riesgos de seguridad así como también la fuente de origen dentro de la infraestructura y parque tecnológico de la institución.

- Gracias a la herramienta se pudo gestionar de mejor manera los incidentes de seguridad y especialmente en lo que corresponde a la fase reactiva.

- Los reportes de OSSIM AlienVault dan claramente información de lo que está ocurriendo en la infraestructura de la organización.

En el artículo científico del año 2015, de Ángel Heraldo Bravo Bravo, Álvaro Luis Villafuerte Quiroz, Ing. José Patiño S. "Implantación de una herramienta OSSIM para el monitoreo y gestión de la seguridad de la red y plataformas Windows y Linux aplicado a empresas medianas" se indica:

- Que al implementar una herramienta OSSIM se redujo el tiempo necesario para la gestión y resolución de incidentes.

- Que las herramientas OSSIM otorgan un aporte invaluable al administrador de red, brindándole información útil para la toma de decisiones en el campo de la

seguridad y que enfocados en la visión principal hemos logrado integrar varios dispositivos de red.

- Que este tipo de herramientas ayudan sobre todo a empresas medianas para optimizar su gestión y el control de incidentes.

En el estudio de Alexis Fernando Balarezo Chávez y Diego Xavier Poveda Pilatasig "Propuesta de mejoramiento de la herramienta OSSIM SIEM (open source), para obtener los niveles óptimos de gestión en la administración de la seguridad, en una red implementada en cloud computing" se señala:

- Que, el monitoreo de incidentes a través de una herramienta OSSIM es una excelente respuesta para robustecer la seguridad en la infraestructura haciendo uso del control del hardware o software que garanticen medidas efectivas ante los ataque o daños.

- Además, la herramienta OSSIM permite verificar las amenazas en tiempo real y trabaja de forma inteligente ya que es capaz de correlacionar eventos en busca de patrones.

- Para implementar este tipo de herramientas en una empresa se requiere conocer detalles sobre aplicaciones o servicios de la organización con el fin de aplicar al máximo la seguridad de acuerdo a las necesidades del negocio.

Por lo manifestado en las diferentes investigaciones se puede concluir que el implementar la herramienta OSSIM AlienVault permitirá minimizar el tiempo de respuesta a los incidentes presentados a la infraestructura tecnológica de la Universidad, así como también contribuye a mantener un adecuado control y monitoreo de los sistemas.

Adicionalmente el desarrollo e implementación de scripts de monitoreo que cumplan con las necesidades de la institución, permitirá robustecer la seguridad en la infraestructura tecnológica de la UISEK.

CAPÍTULO II

MARCO TEÓRICO

A continuación, se describe la teoría que sustenta el trabajo realizado:

2.1 Seguridad informática

> Podemos definir la Seguridad Informática como cualquier medida que impida la ejecución de operaciones no autorizadas sobre un sistema o red informática, cuyos efectos puedan conllevar daños sobre la información, comprometer su confidencialidad, autenticidad o integrada, disminuir el rendimiento de los equipos o bloquear el acceso de usuarios no autorizados al sistema. (Gómez, 2011, p. 38)

Amenaza informática

Se puede considerar como amenaza informática a cualquier evento accidental, intencionado que pueda ocasionar algún daño a la infraestructura informática provocando pérdidas materiales, financieras o de otro tipo a la organización (Gómez, 2011; Noguera, 2011).

Vulnerabilidad

Una vulnerabilidad es una debilidad del sistema informático que puede ser explotada por un atacante para obtener acceso no autorizado o realizar acciones no autorizadas que pueden causar daños y pérdidas en una organización (Gómez, 2011).

Impacto

El impacto es la medición y valoración del daño tangible como el daño intangible (Información) que podría producir a la organización un incidente de seguridad, el impacto se categorizar como: bajo, medio y alto (Gómez, 2011).

Riesgo

El riesgo es la probabilidad de que una amenaza se materialice sobre una vulnerabilidad previo a un análisis al sistema informático, causando un determinado impacto en la organización, el riego se categorizar como: bajo, medio y alto (Gómez, 2011).

Políticas de Seguridad

> Las políticas de seguridad son reglas establecidas para el dominio de la empresa. Existen políticas de usuarios y de equipos. Las primeras restringen las acciones de los usuarios una vez que ingresan en la red, los equipos y a los servicios; por ejemplo, se puede evitar que se ejecuten ciertos programas en los equipos y realizar muchas otras configuraciones como acciones. Al aplicar políticas de máquinas, permite la opción de estandarizar las propiedades de las PCs de escritorio y los servidores para que tengan una configuración general única; es decir que cualquier usuario que use la máquina tendrá las mismas configuraciones (Marchionni, 2011, p. 72).

Las políticas son reglas, normas y directrices que permiten garantizar la confidencialidad, integridad y disponibilidad de los datos y minimizar los posibles riesgos que puedan presentar.

2.2 SIEM

> Gestión de Eventos e Información de Seguridad o por sus siglas en *ingles Security Information and Event Management e*s un tipo de software que tiene como objetivo centralizar y brindar a las organizaciones información útil sobre potenciales amenazas de seguridad de sus redes críticas, a través de la estandarización de datos y priorización

de amenazas. Esto es posible mediante un análisis centralizado de datos de seguridad, obtenidos desde múltiples sistemas, que incluyen soluciones de prevención de intrusiones (helpsystem, 2020).

Este aplicativo es capaz de detectar rápidamente, responder y neutralizar las amenazas informáticas, protegiendo a la empresa de posibles riegos y disminuyendo el impacto en los datos y enfocar los esfuerzos del equipo hacia donde puedan tener mayor nivel de impacto.

OSSIM AlienVault

Open Source Security Information Management en sus siglas en inglés, es una herramienta de código abierto y gestión de eventos (SIEM), con funciones de recopilación, normalización y correlación de eventos. Lanzado por la falta de productos de código abierto especializados en esta rama, OSSIM AlienVault se creó específicamente para abordar la realidad que enfrentan muchos profesionales de la seguridad hoy en día (AT&T, 2021, p. 1).

OSSIM AlienVault es un conjunto de herramientas centralizada que ayuda la administración de eventos de seguridad, las herramientas integradas son:

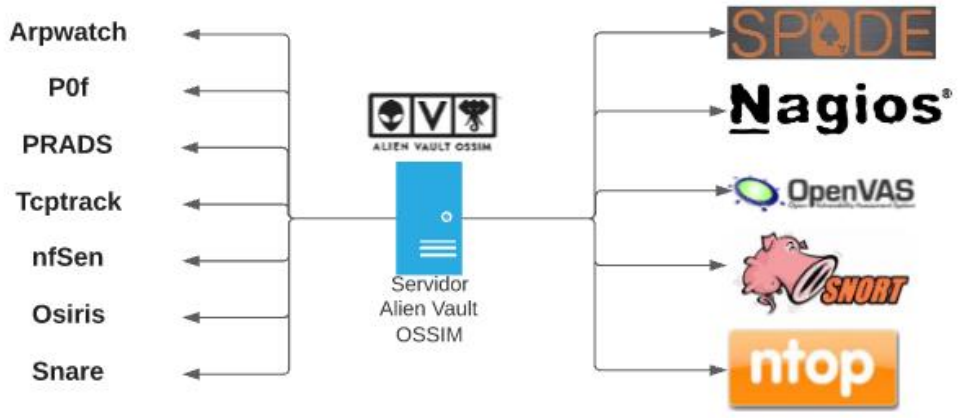


Figura 2: Herramientas de OSSIM AlienVault
Elaborado por: Jorge Añazco

Esta herramienta trae incorporada cerca de 22 Funciones, todas estas son Open Source capaces de correlacionarse y así poder tener el control centralizado (Heraldo, Villafuerte, & Patiño, 2015) se lo puede representar en el siguiente diagrama:

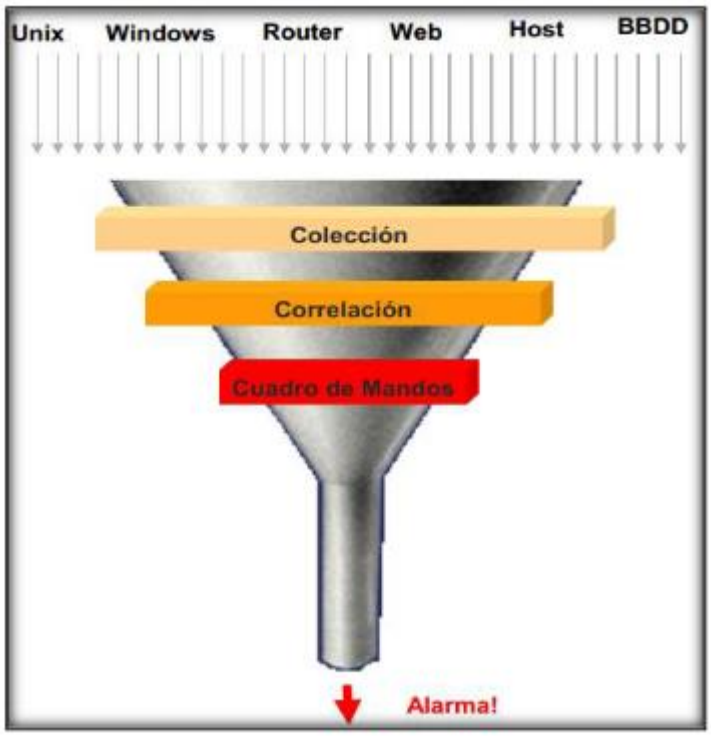


Figura 3: Modelo de relación OSSIM
Fuente: Implantación De Una Herramienta Ossim Para El Monitoreo Y Gestión De La Seguridad De La Red Y Plataformas Windows Y Linux Aplicado A Empresas Medianas (Bravo, Villafuerte, & Patiño, 2015)

2.3 20 Controles de la CIS

CIS ControlsTM es un conjunto de acciones priorizadas que colectivamente forman un conjunto de mejores prácticas de defensa que mitigan los ataques más comunes contra sistemas y redes. Los Controles CIS son desarrollados por una comunidad de expertos en TI que aplican su experiencia de primera mano cómo defensores cibernéticos para crear estas mejores prácticas de seguridad aceptadas globalmente. Los expertos que desarrollan los Controles CIS provienen de una amplia gama de sectores que incluyen *retail*, fabricación, salud, educación, gobierno, defensa y otros (CIS Controls, 2018, p. 5).

Los controles son actividades que aseguran buenas prácticas ya que son un conjunto de acciones priorizadas y altamente focalizadas compatibles con todos los requerimientos de seguridad gubernamental o industrial (CIS Controls, 2018). los controles son los siguientes:

- Control 1: Inventario y control de activos de hardware.
- Control 2: Inventario y control de activos de software.
- Control 3: Gestión continua de vulnerabilidades.
- Control 4: Uso controlado de los privilegios administrativos.
- Control 5: Configuración segura para el hardware y el software de los dispositivos móviles, laptops, estaciones de trabajo y servidores.
- Control 6: Mantenimiento, monitoreo y análisis de logs de auditoría.
- Control 7: Protección de correo electrónico y navegador web.
- Control 8: Defensas contra malware.
- Control 9: Limitación y control de puertos de red, protocolos y servicios.
- Control 10: Funciones de recuperación de datos.
- Control 11: Configuración segura para dispositivos de red, tales como firewalls, routers y switches.
- Control 12: Protección perimetral.
- Control 13: Protección de datos.
- Control 14: Control de acceso basado en la necesidad de saber.

- Control 15: Control de acceso inalámbrico.
- Control 16: Monitoreo y control de cuentas.
- Control 17: Implementar un programa de concienciación y capacitación en seguridad.
- Control 18: Seguridad del software de aplicación.
- Control 19: Respuesta y gestión de incidentes.
- Control 20: Pruebas de penetración y ejercicios de equipo rojo.

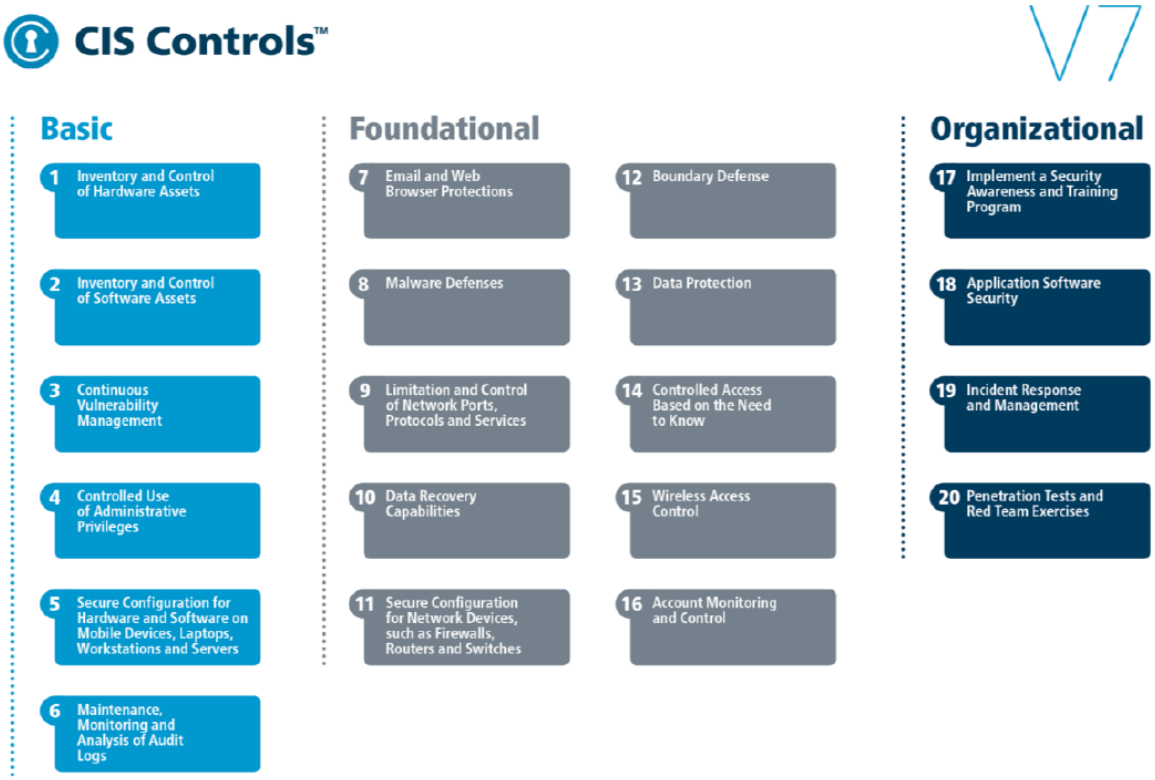


Figura 4: 20 Controles de la CIS v7
Fuente: CIS Controls (2018)

Los Controles CIS v7 se nutren de información de ataques reales y defensas efectivas y reflejan el conocimiento combinado de expertos de cada parte del ecosistema. Esto asegura que los Controles CIS v7 son el conjunto más efectivo y específico de medidas

técnicas disponibles para detectar, prevenir, responder y mitigar el daño desde el más común al más avanzado de esos ataques (CIS Controls, 2018, p. 6).

2.4 Script

Un script es un documento con instrucciones, ordenes o sentencias que tienen un objetivo específico, se considera pequeños programas que son parte de un sistema más grande (Noguera, 2011).

Características

- Cada script tiene una función a cumplir.
- Combinación de elementos, tecnologías y lenguajes.
- Interacción especifica con el usuario o en otro caso con el propio sistema operativo.
- Controla una decretada aplicación o programa en específico.
- Configuración como al mismo tiempo instalación de distintos sistemas de operación.

Los scripts son ampliamente utilizados dentro del diseño y desarrollo de páginas web, para el monitoreo de redes y servidores, manipulación de datos como generación de informes, para cumplir procesos y servicios *serverless*, *haking*, para inteligencia artificial y *BigData*, y por su versatilidad se puede utilizar en varios lenguajes de programación como de tecnologías.

Lenguaje Shell

El intérprete de mandatos o "shell" es la interfaz principal entre el usuario y el sistema, permitiéndole a aquél interactuar con los recursos de éste. El usuario introduce sus órdenes, el intérprete las procesa y genera la salida correspondiente.

Por lo tanto, un intérprete de mandatos de Unix es tanto una interfaz de ejecución de órdenes y utilidades, como un lenguaje de programación, que admite crear nuevas órdenes – denominadas guiones o "shellscripts"–, utilizando combinaciones de mandatos y estructuras lógicas de control, que cuentan con características similares a las del sistema y que permiten que los usuarios y grupos de la máquina cuenten con un entorno personalizado.

En Unix existen 2 familias principales de intérpretes de mandatos: los basados en el intérprete de Bourne (BSH, KSH o BASH) y los basados en el intérprete C (CSH o TCSH) (Labrador, 2003, p. 4).

CAPÍTULO III

ANÁLISIS Y SITUACIÓN ACTUAL

3.1 Universidad Internacional Sek (UISEK)

La Universidad Internacional SEK es un centro de educación superior laica y privada con 30 años de trayectoria en Ecuador y como Institución SEK con 125 años en experiencia educativa, actualmente sus 3 campus funcionan en la ciudad de Quito, en las siguientes direcciones, Miguel de Cervantes, al norte de la ciudad sector Carcelén, el Campus Juan Montalvo, se encuentra en el Monasterio de Guápulo, zona céntrica de la ciudad además de que es considerada como Patrimonio de la Humanidad y el Campus Felipe Segovia Olmo que se encuentra ubicada en la parte centro norte de la ciudad, actualmente la universidad por su función académica ofrece 21 carreras de pregrado y 24 programas de posgrado tratando de ampliar las propuestas académicas según las necesidades del país.

Figura 5: Logo Universidad Internacional Sek
Fuente: Universidad Internacional Sek (2021)

"La Universidad Internacional SEK (UISEK), es una institución de Educación Superior que se encuentra encaminada en alcanzar la excelencia educativa y que viene paso a paso consolidando la mejora continua" (Universida Internacional SEK, 2021).

“La Planificación Estratégica, como uno de sus procesos clave, ha sido instituida bajo parámetros de reflexión y respaldo institucional” (Universida Internacional SEK, 2021).

“Su formulación está diseñada con la convicción, el interés y el absoluto compromiso de cada autoridad, directivo, decano, docente, estudiante y personal administrativo, en pro de la sociedad ecuatoriana y de la UISEK” (Universida Internacional SEK, 2021).

Misión:

“Desarrollar sus actividades de docencia, investigación y vinculación, con los más altos estándares de calidad y pertinentes para la solución de problemas de la sociedad de manera sostenible” (Universida Internacional SEK, 2021).

Visión:

“Ser una universidad referente de calidad acreditada, apoyada en políticas de mejora continua, comprometida con la generación y transferencia del conocimiento en beneficio de la sociedad” (Universida Internacional SEK, 2021).

Historia

En 1892, nace en Madrid la Institución SEK, oficialmente reconocida en 1905, y al finalizar la Guerra Española en 1939 inicia su expansión educativa y en 1960 las innovaciones pedagógicas guían el proyecto educativo, a partir de 1982 adquiere proyección internacional mediante la creación de centros educativos en diversos países: SEK-Chile (Santiago y Viña del Mar) y SEK- Ecuador (Quito y Guayaquil), a partir de 1986 se vinculan otros países a la red: SEK-Panamá, SEK-Paraguay, SEK-Costa Rica, en España se amplía a Coruña y Valencia, en 1996 se inauguró SEK-República Dominicana, en 1997 SEK- Hungría, en 1999 SEK- EE.UU, en 2001 SEK-Guatemala, en 2003 SEK-Sud África y en 2005 SEK- México.

La Institución Internacional SEK concreta su proyecto educativo universitario a partir de 1990 con la creación de la UISEK- Santiago de Chile, en 1993 la UISEK-Quito y, en 1997 la UISEK- Segovia. La afinidad cultural y lingüística ha propiciado el afianzamiento de lazos educativos en 15 países sobre la base del aprendizaje de calidad, en busca de la excelencia académica (Universidad Internacional Sek, 2021).

3.2 Situación Tecnológica de la Universidad Internacional SEK

La Universidad Internacional SEK para sus actividades necesita sistemas o módulos informáticos que den soporte a los procesos diarios de la universidad, actualmente no existe un rol de Oficial de Seguridad de la Información responsable de aplicar todas las normas y buenas prácticas de seguridad informática, así como también no se encuentra implementado un sistema de gestión de eventos e información de seguridad (SIEM), que permita la detección de fallos de rendimiento y posibles ataques de seguridad a la infraestructura tecnológica.

No cuenta con un comité de seguridad de la información, integrado por los responsables de las áreas administrativas de la Universidad, con el fin de diseñar los requerimientos y procesos que cada área necesita, con el fin de precautelar la información y el funcionamiento de la Universidad, además no se ha definido los procesos de los diferentes módulos y subsistemas que cuenta la entidad para su funcionamiento, por ejemplo, el Portal Alumnos (https://portalalumnos.uisek.edu.ec/Account/Login.aspx), el Portal de Docentes (https://portaldocentes.uisek.edu.ec), Aulas Virtuales Canvas (https://canvas.uisek.edu.ec), portales de la biblioteca (https://repositorio.uisek.edu.ec/, http://biblioteca.uisek.edu.ec/home/index.php) y otros aplicativos internos como externos que maneja la institución. Al no contar con una estructura organizada, coherente y optimizada no se ha logrado expandir el grado de seguridad de los aplicativos como de la infraestructura de la universidad y la de sus proveedores de servicios.

No se ha implementado una norma estándar de seguridad que permita unificar los criterios de evaluación de los riesgos asociados al manejo de la información de la Universidad, el objetivo principal del Sistema de Gestión de Seguridad de la Información es preservar la confidencialidad, integridad y disponibilidad de la información.

3.3 Descripción de la infraestructura

La UISEK funciona en tres campus:

1. Campus Felipe Segovia Olmo: Calle Italia N31-125 y Av. Mariana de Jesús.
2. Campus Miguel de Cervantes: Calle Alberto Einstein s/n y 5ta. transversal (Carcelén).
3. Campus Juan Montalvo: El Calvario s/n y Fray Francisco Compte.

Tabla 1: Descripción campus UISEK

Campus	Edificio	Aulas
Campus Juan Montalvo	1	3
	2	18
	3	2
	4	6
	5	9
Campus Felipe Segovia		7
Campus Juan Montalvo		19

Elaborado por: Jorge Añazco

Cada campus dispone de su propia infraestructura tecnológica y su respectivo proveedor de internet, las 64 aulas de los 3 campus cuentan con su proyector, computador para el profesor y un dispositivo wifi, con respecto al ingreso a la red es abierta para los estudiantes como para los profesores.

Es importante mencionar que no se ha implementado una herramienta que detecte riesgos de seguridad en tiempo real, esto es un punto importante, ya que, un ataque que no puede ser detectado de manera inmediata, puede afectar el desarrollo normal de las actividades académicas.

3.4 Usuarios

Es importante definir el tipo de usuario que tiene la institución, ya que, como entidad de educación existe un considerable flujo de personas, las cuales pueden ser un potencial riesgo de seguridad, la UISEK cuenta con un flujo aproximado de usuarios de acuerdo al siguiente detalle:

Tabla 2: Cantidad de usuarios UISEK

Descripción	Cantidad
Personal Administrativo	80
Estudiantes	2.000
Profesores	164
Total	2.244

Elaborado por: Jorge Añazco

Con estas estadísticas, se puede obtener un alcance de la población que sería afectada en un ataque informático, así como también, puede ser considerada como posibles fuentes de eventos e incidentes de seguridad dentro de la organización.

3.5 Operatividad de los portales o aplicativos web

A nivel de infraestructura tecnológica, los aplicativos se encuentran en una plataforma virtual en la nube, el proveedor es el encargado de la operatividad y seguridad informática, el riesgo del funcionamiento de los aplicativos está dado por la calidad del código de desarrollo.

Los portales o aplicativos webs que maneja la Universidad tienen como objetivo principal la visualización y accesibilidad de la información requerida por los estudiantes, docentes y personal administrativo, así como también ayudar con el proceso de aprendizaje de los alumnos. También se ofrece servicios adicionales como la matriculación de los estudiantes, compra de certificados, bienestar estudiantil, secretaria académica y secretaria general.

Los aplicativos web y herramientas informáticas que maneja la institución no se encuentran en un nivel óptimo de seguridad, por esta razón determinan un riesgo inherente como residual,

y al no contar con procesos que permitan disminuir la susceptibilidad a los ataques, siempre existirá un riesgo alto de seguridad en los diferentes sistemas que tiene la institución.

3.6 Módulos críticos

A continuación, se detallan los módulos que se consideran críticos para la operatividad de la universidad:

Módulo de inicio de sesión

Este módulo tiene un riesgo alto, ya que, si el atacante obtiene las credenciales del estudiante, puede ingresar y visualizar toda la información del mismo, pero hay que tomar en cuenta que la Universidad ha invertido recursos sobre este módulo en criptografía de los parámetros importantes para el inicio de sesión, además, cuenta con el respaldo del ingreso al portal, a través del correo institucional con Google.

Figura 6: Inicio de sesión Portal Alumnos UISEK
Fuente: Portal Alumnos UISEK

Módulo de hoja de vida e información del estudiante

Este módulo tiene un nivel de riesgo alto, debido a que se encarga de guardar la información del estudiante realizando un histórico para obtener estadísticas según las necesidades de la universidad, por último, si el atacante logra obtener toda la información del estudiante, puede utilizarlo para diferentes fines.

Figura 7: Módulo de hoja de vida UISEK
Fuente: Portal Alumnos UISEK

Módulo de certificados

Este módulo tiene un nivel de riesgo bajo, ya que son solicitudes del estudiante hacia la Universidad.

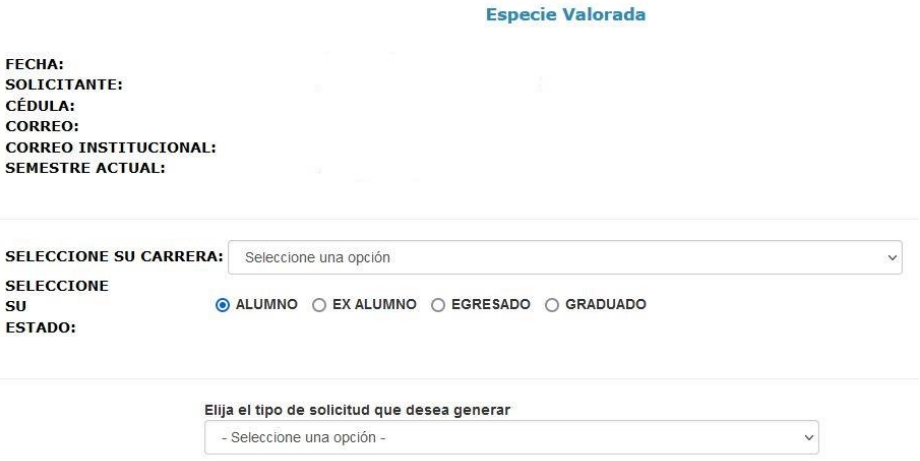


Figura 8: Módulo de certificados UISEK
Fuente: Portal Alumnos UISEK

Módulo de Pagos

Este módulo permite realizar al estudiante los pagos de los servicios ofertados en el Portal Alumnos, tienda virtual de la universidad y en el proceso de pago de inscripciones, matrículas y créditos, por lo cual, tiene un nivel de riesgo alto, ya que es un punto en que el atacante puede obtener información sensible del estudiante y beneficiarse de manera económica.

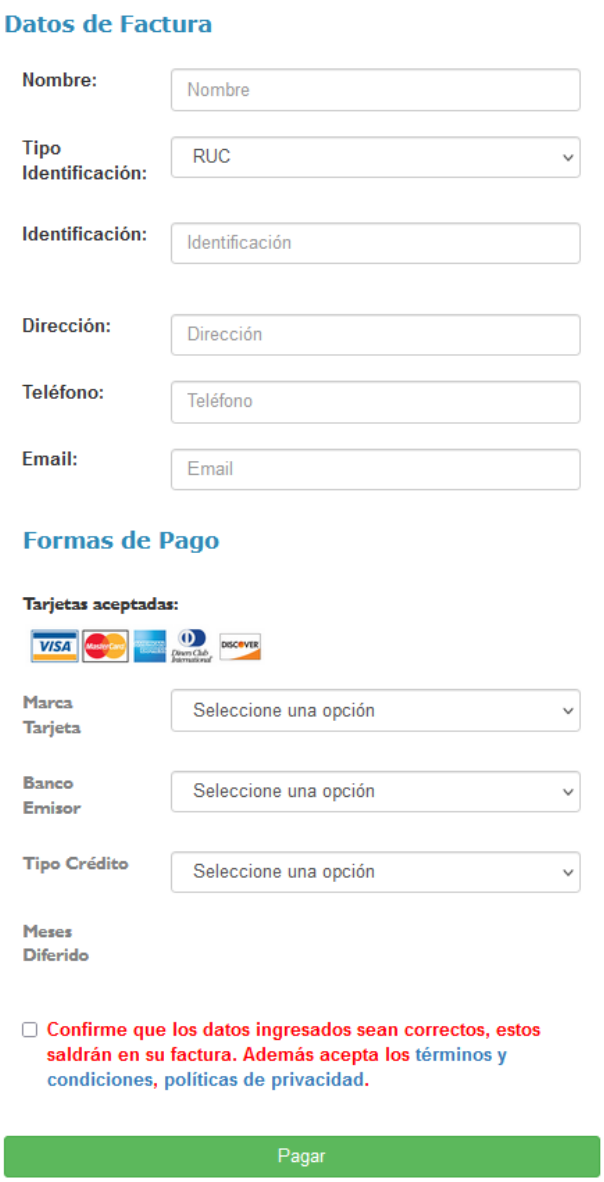


Figura 9: Módulo de pagos UISEK
Fuente: Portal Alumnos UISEK

Módulo de notas y asistencia

Este módulo es un nivel de riesgo medio, ya que el atacante podría modificar las notas de los estudiantes, sin embargo, la Universidad puede detectar estos cambios y rectificarlos.

Nivel	TIPO	Código Asignatura	Nombre Asignatura	Paralelo	P1	P2	N.Ex	NS	NES	Asis %	ASISTENCIA	Nota Final	Estado
3	MAT1	MCIBDIFP3SO	SEGURIDAD ORGANIZACIONAL Y PERSONAL						--	100	4/4 (100%)	7,8	APROBADO
3	MAT1	MCIBDIFP3SS	SEGURIDAD SOCIAL EN LA ERA DIGITAL						--	100	8/8 (100%)	9,4	APROBADO
3	MAT1	MCIBDIIA3SE	SEGURIDAD EN HARDWARE						--	100	7/7 (100%)	9,7	APROBADO
3	MAT1	MCIBTIIA3TA	TALLER INTEGRAL PARA LA IMPLEMENTACION DEL TRABAJO DE TITULACION						--	100	3/3 (100%)	9,1	APROBADO

Figura 10: Módulo de notas y asistencia UISEK
Fuente: Portal Alumnos UISEK

3.7 Riesgos indirectos

Como cualquier entidad o empresa existe un riesgo importante de amenazas que pueden llegar a través de métodos o medios muy diversos, que pueden ser, desde negligencia del personal interno como de proveedores de servicios o ataques maliciosos cuidadosamente planificados con un objetivo específico, según PandaSecurity (2018) en una encuesta realizada por Crowd Research el 90% de las empresas consultadas se considera vulnerable a este tipo de problemas de seguridad de hecho, el 53% de las compañías asegura haber sufrido ataques internos en los últimos doce meses.

3.8 20 controles de la CIS

El presente análisis se realiza con el documento *CIS Critical Security Control v6.1 Assessment Tool*, proporcionado por el proyecto (AuditScripts, 2018), el cual brinda una estructura de calificación del 0 al 5 del estado de la seguridad informática de la empresa respecto a los 20 controles de las CIS, que se detallan a continuación:

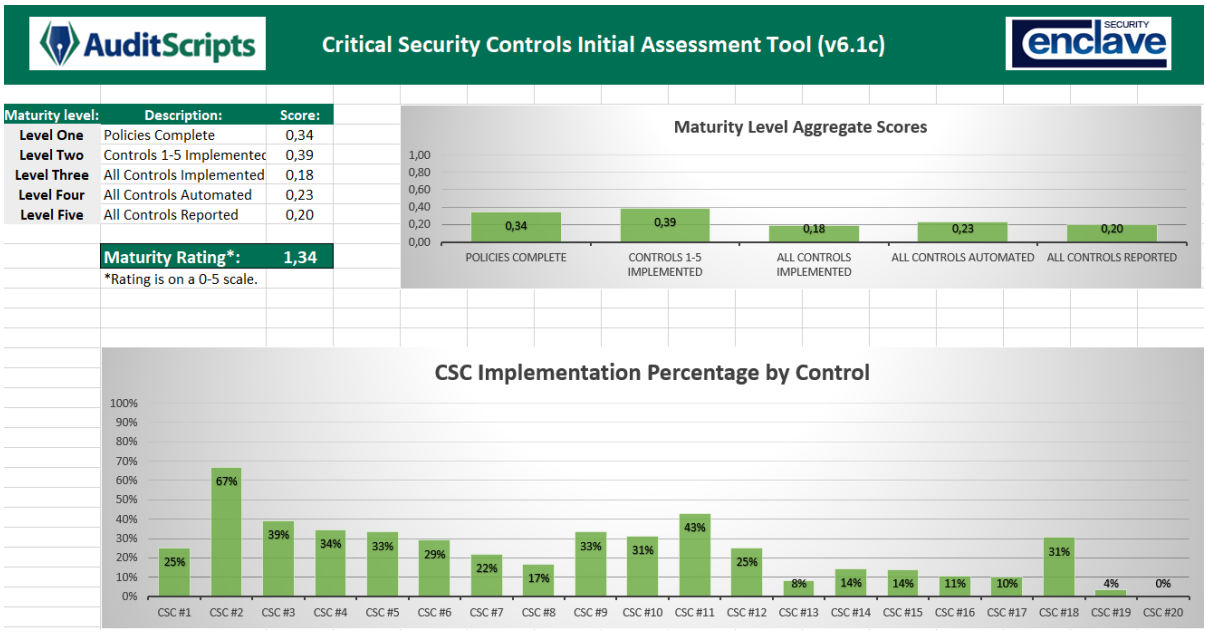


Figura 11: Análisis de los 20 controles de las CIS.
Elaborado por: Jorge Añazco

Con los resultados de la figura anterior, análisis de los 20 controles de las CIS v7, se puede concluir que las políticas de seguridad no están completas y existe fallas en su ejecución, que en general los controles de las CIS v7 no se han implementado de manera óptima y existe problemas de inventarios de hardware como de software, ningún control de la CIS v7 se ha implementado más del 70% y en otros controles es nula su implementación, no existen procesos completos ni automatizados de seguridad informática, lo cual se deriva por la falta de un programa de seguridad de la información, por último, con el puntaje obtenido de 1.34 se establece que el nivel de seguridad de la institución es baja y es necesario establecer acciones y procesos para mejorar la seguridad.

En las siguientes matrices, se detalla por cada control de la CIS v7 los problemas encontrados y su puntaje de cumplimiento:

Tabla 3: Descripción análisis de los controles de las CIS v7 (1-5)

Control	Descripción	Observación	Puntaje
1	Inventario y control de activos de hardware	Se puede observar falta de inventarios completos de los equipos de red y equipos tecnológicos, además, no existe un control de los equipos que se conectan y se desconectan de la red, lo que implica que no se puede identificar, registrar, responder y proteger a las acciones de estos equipos de manera inmediata y segura, se ha verificado que no existe ninguna implementación de certificados para que los equipos de la red de confianzas se puedan identificar.	25%
2	Inventario y control de activos de software	No existe ningún control de inventarios de software autorizado de manera óptima ya que no existe herramientas ni políticas (listas blancas, Active Directory) que estandaricen la utilización del software y actualizaciones de las mismas para evitar brechas de seguridad.	67%
3	Gestión continua de vulnerabilidades	No existen herramientas que ayuden a detectar o escanear posibles vulnerabilidades y evitar los riesgos que se genera, actualmente existe herramientas que solo ayudan en proteger a nivel de red posibles intrusiones. Y no están definidos procesos de calificación de riesgo para priorizar la corrección de vulnerabilidades descubiertas.	39%
4	Uso controlado de los privilegios administrativos	No existe procesos o herramientas que ayuden a controlar, rastrear, prevenir, corregir el uso, la asignación y la configuración de privilegios administrativos en computadoras, redes. Las aplicaciones web desarrolladas cuentan con un control de privilegios por medio de roles asignados a cada usuario, pero no se cuenta con un inventario general de cuentas administrativas que consten todos los accesos y cuentas de todos los sistemas de la universidad	34%
5	Configuración segura para el hardware y el software de los dispositivos móviles, laptops, estaciones de trabajo y servidores	Se ha detectado que no todas las estaciones de trabajo y servidores cuentan con una configuración con buenas propiedades de seguridad, además si incluso existe una configuración solida de seguridad no hay una gestión para evitar la degradación de la seguridad a medida que se actualiza, se repare o exista algún cambio en el software. Por último, no existe respaldo de las configuraciones de todos los dispositivos y las mismas no son actualizadas frecuentemente.	33%

Elaborado por: Jorge Añazco

Tabla 4: Descripción análisis de los controles de las CIS v7 (6-12)

Control	Descripción	Observación	Puntaje
6	Mantenimiento, monitoreo y análisis de logs de auditoría	No existe un manejo y análisis total de los registros que generan todos los dispositivos informáticos por las actividades y eventos que genera la universidad, por esta razón no se puede detectar, comprender o recuperase de un ataque o fallo en los sistemas. Por último, no se tiene implementado un nivel de detalle de los registros para tener más información de los posibles eventos de seguridad que puedan existir.	29%
7	Protección de correo electrónico y navegador web	No existe configuraciones que ayuden asegurar y precautelar el uso de navegadores y clientes de correo electrónico, además no hay un control de soporte y que estén en la última versión que libera el proveedor. No hay control de los plugins o aplicaciones add-on instalados en los navegadores o cliente de correo electrónico y limitar solo lenguajes scripting autorizados en los navegadores.	22%
8	Defensas contra malware	No esta implementado un software antimalware gestionado centralmente para monitorear y defender continuamente cada una de las estaciones de trabajo y servidores. No existe la limitación para no ejecutar el contenido de los medios extraíbles. No existe información necesaria para realizar auditorías para detectar posibles ataques de los equipos de la institución.	17%
9	Configuración segura para dispositivos de red, tales como firewalls, routers y switches	No hay un control completo en la administración del uso operacional continuo de puertos, protocolos y servicios en dispositivos en red para poder rastrear, controlar, corregir y minimizar los posibles ataques.	33%
10	Funciones de recuperación de datos	No se han realizado pruebas que aseguren la integridad que los datos respaldados y de forma periódica mediante procesos automatizados.	31%
11	Protección perimetral	No hay herramientas que automaticen la verificación de configuraciones de equipos y detectar cambios, además que alerte cuando exista algún cambio. No existe maquinas dedicadas para las tareas administrativas o tareas que necesiten un acceso elevado.	43%
12	Defensa de borde	No existe equipos IDS/IPS para registrar, monitorear y alertar el flujo de paquetes en el límite de cada una de los bordes de la organización. No existe colectores NetFlow para registrar todos los datos de los equipos de borde de la red.	25%

Elaborado por: Jorge Añazco

Tabla 5: Descripción análisis de los controles de las CIS v7 (13-20)

Control	Descripción	Observación	Puntaje
13	Protección de datos	No hay cifrado de los discos duros internos, externos, medios extraíbles como flash, tarjetas de memorias y equipos móviles y no existe configuraciones para que la información no sea copiada en cualquier dispositivo de almacenamiento.	8%
14	Control de acceso basado en la necesidad de saber	No se ha implementados procesos y herramientas utilizados para rastrear, controlar, prevenir, corregir el acceso seguro a activos críticos como la información y recursos que genera la institución, además no hay una jerarquización a la necesidad de acceder a estos activos críticos basado en una clasificación aprobada.	14%
15	Control de acceso inalámbrico	No hay autenticación inalámbrica con protocolos que requieran autenticación mutua de múltiples factores, tampoco hay configuraciones que deshabiliten conexiones con equipos o periféricos a través Bluetooth y NFC.	14%
16	Monitoreo y control de cuentas	No hay una gestión activamente del ciclo de vida de las cuentas del sistema y de aplicaciones, tampoco hay un inventario de las cuentas y crear un punto de autenticación centralizado.	11%
17	Implementar un programa de concienciación y capacitación en seguridad	No se ha realizado un análisis de brecha de habilidades por ende no se puede realizar capacitaciones que ayuden a suplir la deficiencia de los empleados. No se han realizado capacitaciones, talleres o cursos que actualicen, eduquen o informe sobre temas de ciberseguridad al personal de la institución.	10%
18	Seguridad del software de aplicación	No se ha implementado prácticas seguras de codificación para el desarrollo interno y la adquisición de nuevo software, no se ha definido procesos ni herramientas para receptar y tratar reportes de seguridad (Errores, logs, bugs) del software adquirido por la institución.	31%
19	Respuesta y gestión de incidentes	No existe documentación de los procedimientos de respuesta a incidentes, tampoco existe asignación de roles según el tipo de incidencia y por último, no hay un esquema de priorización y puntuación de incidentes.	4%
20	Pruebas de penetración y ejercicios de equipo rojo	No se cuenta con personal, herramientas ni se ha contratado equipos externos para realizar pruebas de penetración de servicio por ende no hay auditorias tanto internas como externas que ayuden con las pruebas de penetración ni ejercicios de equipo rojo.	0%

Elaborado por: Jorge Añazco

En virtud de lo expuesto, en la calificación de los 20 controles de la CIS v7 y de acuerdo al gráfico de la figura 9, se puede identificar que la Universidad cuenta con un bajo nivel de seguridad informática, al no disponer de una estructura organizada ni estándares de calidad.

3.8.1 Posible escenario de mejora

Para mejorar la seguridad informática de la institución, se debería:

- Implementar un área administrativa que gestione transversalmente la seguridad de la información, mediante programas, políticas de seguridad según las necesidades de la institución, basándose en los controles de la CIS v7, ya que, en la actualidad no se realiza monitoreo, control, pruebas de calidad, pruebas de penetración de servicios, ni auditorias de seguridad informática.

- Desarrollar e implementar procesos de observación y monitorio de los sistemas como del comportamiento de los trabajadores dentro de las actividades de la universidad en tiempo real, realizar revisiones periódicas de los registros de los servidores, con el objetivo de comprobar algún comportamiento sospechoso, obtener datos específicos que permitan realizar análisis con el fin de detectar o prever de una posible amenaza interna.

- Planificar estrategias de prevención de pérdida y manipulación de datos, además de proteger información, ya sea encriptar o limitar su acceso según la criticidad de la misma, por último, desarrollar un control e identificación de acceso, así como restringir y registrar el uso del software como del hardware.

CAPÍTULO IV

PROPUESTA

De acuerdo a los objetivos planteados, a continuación, se detalla los pasos realizados para la instalación de la herramienta OSSIM AlienVault y la configuración de sus módulos.

Se define los tipos de scripts para el monitoreo y el proceso de implementación de los scripts en el Módulo de Disponibilidad.

4.1 Instalación/configuración de la herramienta OSSIM

4.1.1 Instalación

Herramientas

- Herramienta SIEM: OSSIM AlienVault versión 5.8.1
- Página Oficial: https://cybersecurity.att.com/products/ossim
- Link: https://dlcdn.alienvault.com/AlienVault_OSSIM_64bits.iso

Herramientas para ambiente de pruebas

- Herramienta de virtualización: VirtualBox 6.1.10 64bits
- Página Oficial: https://www.oracle.com/virtualization/virtualbox/
- Link: https://download.virtualbox.org/virtualbox/6.1.10/VirtualBox-6.1.10-138449-Win.exe

Para la instalación OSSIM AlienVault se debe tener conocimiento de la infraestructura de la organización donde se va a instalar es importante saber las configuraciones de las redes, subredes, los DNS, DHCP, puertas de enlaces, proxy y configuración de los equipos perimetrales para que la herramienta funcione correctamente, a continuación, la instalación:

La configuración de la máquina virtual será la siguiente:

- Sistema Operativo: Linux Debian 64Bits

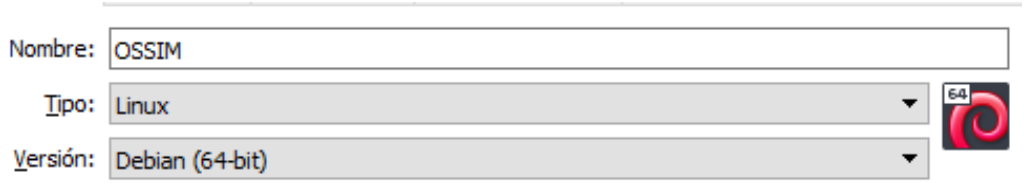


Figura 12: Configuración OSSIM AlienVault: Versión Sistema Operativo
Fuente: OSSIM AlienVault

- Memoria RAM: 2048MB

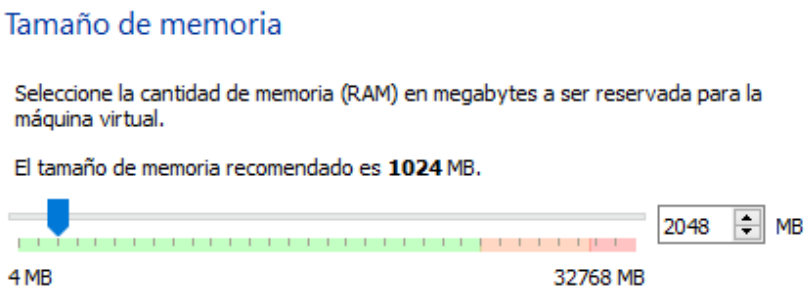


Figura 13: Configuración OSSIM AlienVault: Configuración memoria RAM
Fuente: OSSIM AlienVault

- Tamaño de disco: 20GB

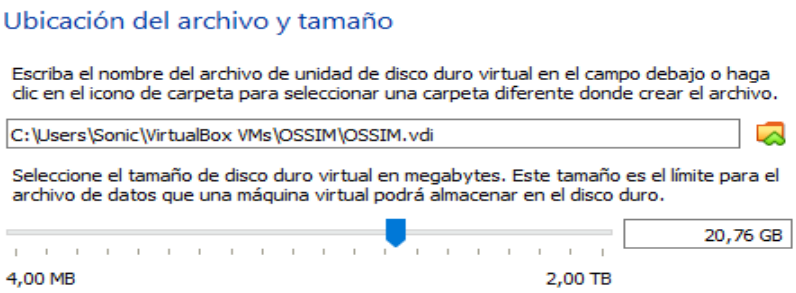


Figura 14: Configuración OSSIM AlienVault: Tamaño de disco duro
Fuente: OSSIM AlienVault

Agregar la imagen ISO para la instalación del OSSIM AlienVault, en el nombre de la máquina virtual, presionar clic derecho y seleccionar Configuración:

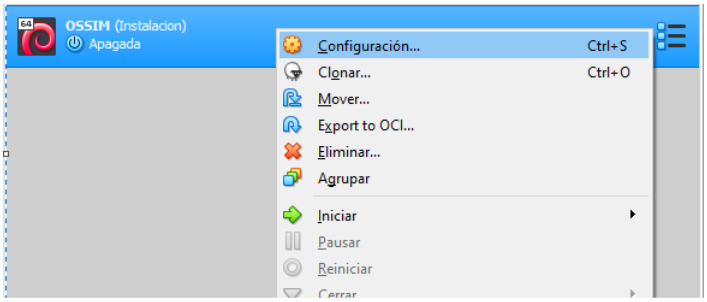


Figura 15: Configuración OSSIM AlienVault: Configuración máquina virtual
Fuente: OSSIM AlienVault

Después de ingresar a la opción de almacenamiento, en la parte de Controlador: IDE en la opción de Vacío seleccionar la ISO de OSSIM AlienVault.

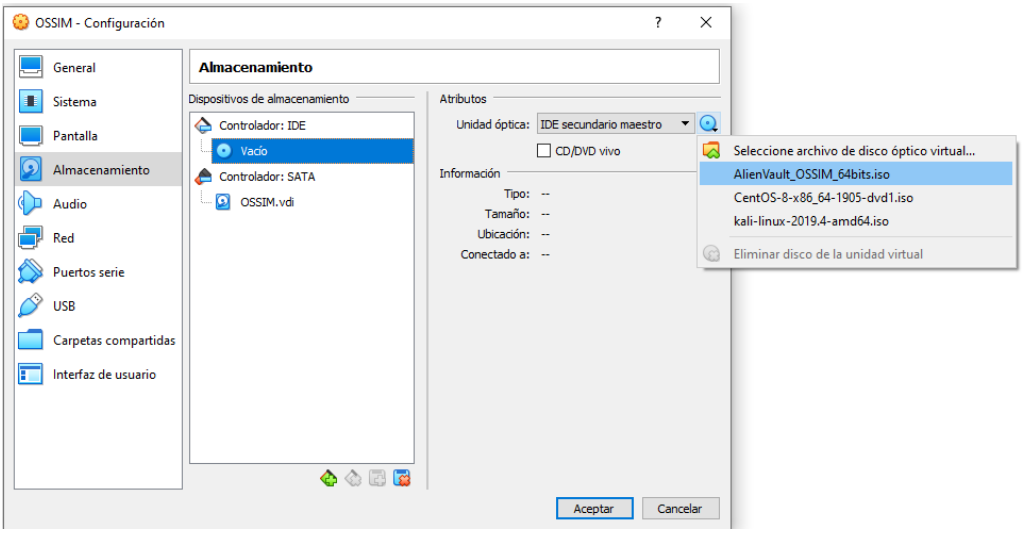


Figura 16: Configuración OSSIM AlienVault: Selección de la imagen OSSIM AlienVault
Fuente: OSSIM AlienVault

Al iniciar la máquina virtual se muestra las opciones de instalación en este proyecto vamos a elegir la opción de *Install OSSIM AlienVault*.

Figura 17: Configuración OSSIM AlienVault: Pantalla de instalación OSSIM AlienVault
Fuente: OSSIM AlienVault

Luego de Seleccionar la primera opción se desplegará la siguiente pantalla para seleccionar el idioma que desee.

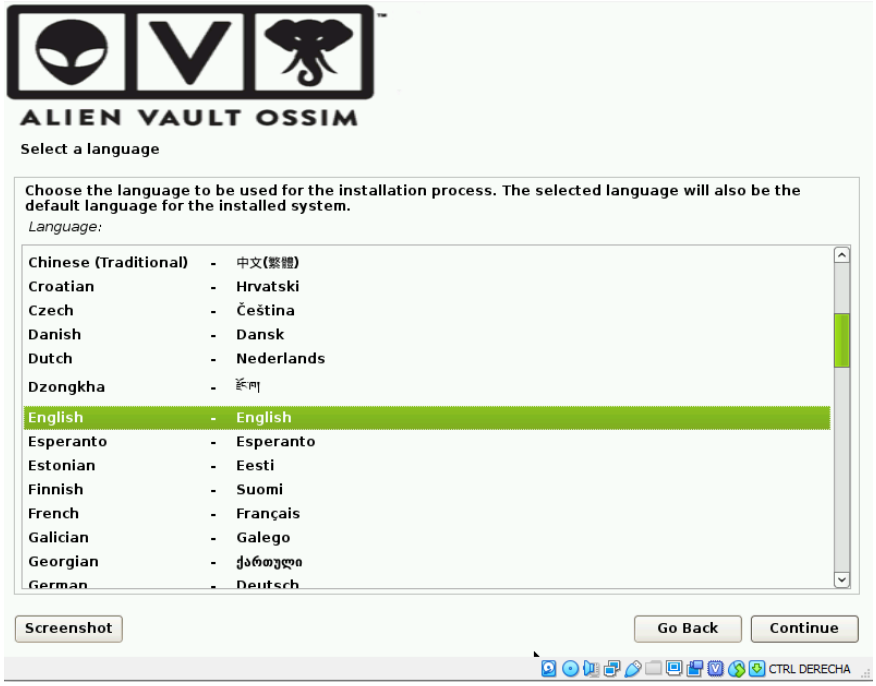


Figura 18: Configuración OSSIM AlienVault: Selección idioma.
Fuente: OSSIM AlienVault

Se debe seleccionar la zona horaria, en este caso Ecuador – Guayaquil

Figura 19: Configuración OSSIM AlienVault: Selección país.
Fuente: OSSIM AlienVault

Figura 20: Configuración OSSIM AlienVault: Selección zona horaria.
Fuente: OSSIM AlienVault

Se debe elegir la distribución del teclado a utilizar.

Figura 21: Configuración OSSIM AlienVault: Distribución teclado.
Fuente: OSSIM AlienVault

Luego el proceso de instalación cargará los componentes y comenzará la instalación.

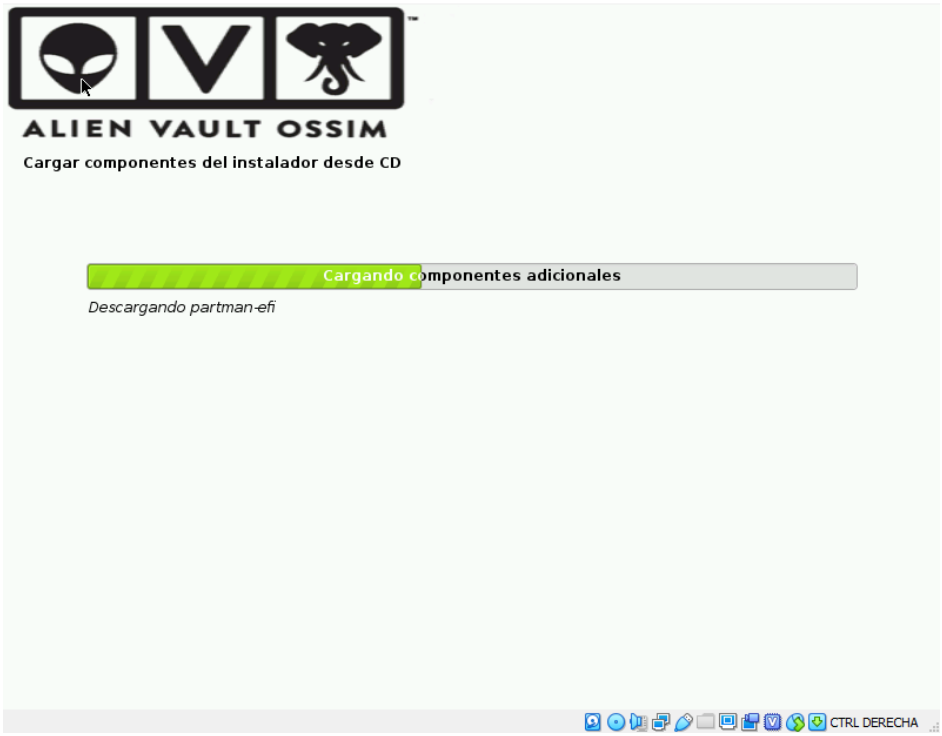


Figura 22: Configuración OSSIM AlienVault: Instalación componentes.
Fuente: OSSIM AlienVault

Inmediatamente se procede a configurar la red, para esta parte es necesario poner en modo puente/bridge la interface de red.

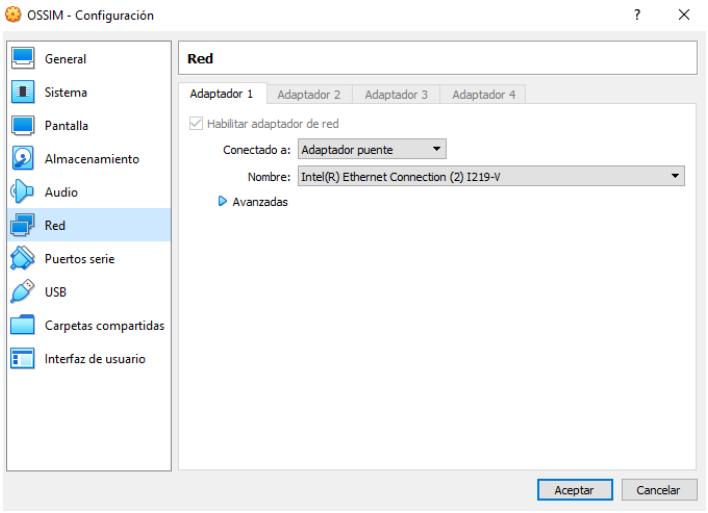


Figura 23: Configuración OSSIM AlienVault: Configuración de tarjeta de red (Virtual Box).
Fuente: OSSIM AlienVault

La configuración de red en el ambiente de pruebas va a ser la siguiente:

- IP: 192.168.0.161

Figura 24: Configuración OSSIM AlienVault: Configuración IP (OSSIM AlienVault)
Fuente: OSSIM AlienVault

- Máscara de red: 255.255.255.0

Figura 25: Configuración OSSIM AlienVault: Configuración mascara de red.
Fuente: OSSIM AlienVault

- Gateway: 192.168.0.1

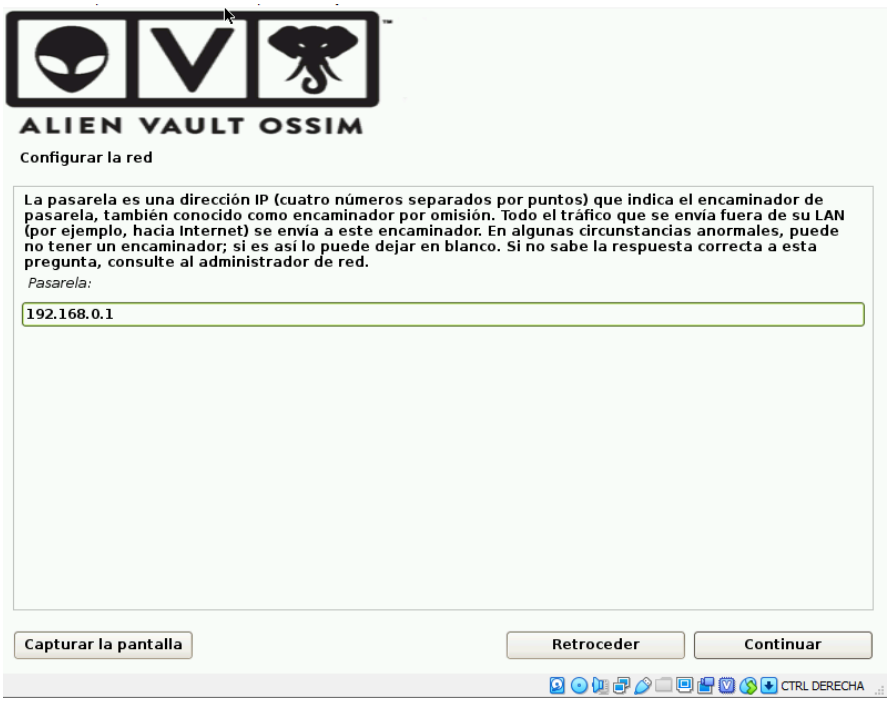


Figura 26: Configuración OSSIM AlienVault: Configuración *Gateway*.
Fuente: OSSIM AlienVault

- DNS: 192.168.0.1

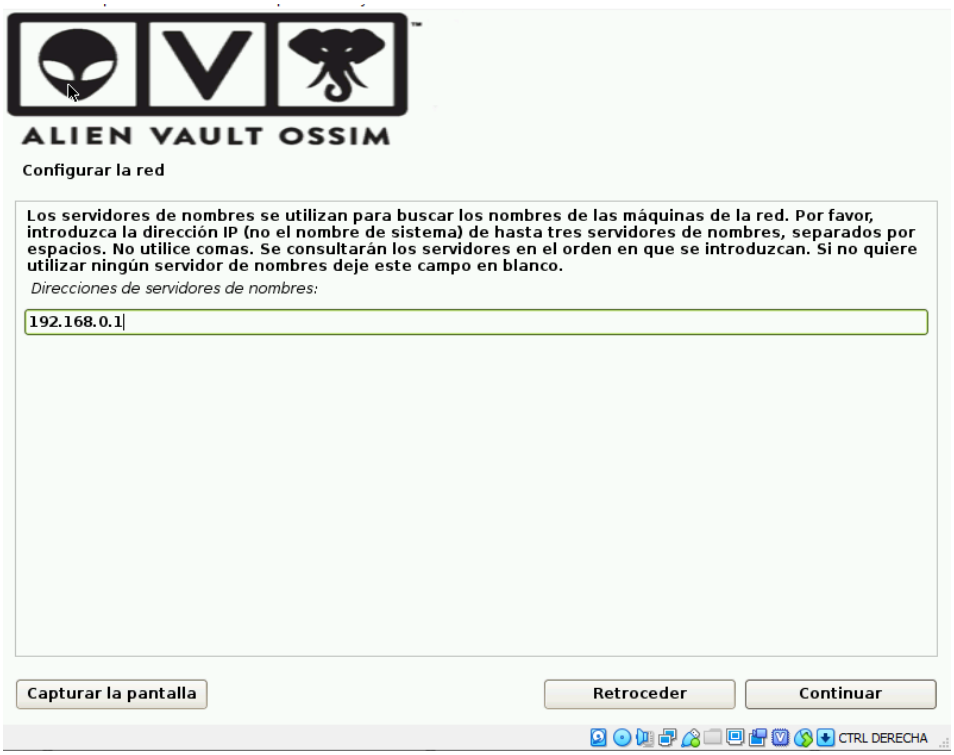


Figura 27: Configuración OSSIM AlienVault: Configuración DNS.
Fuente: OSSIM AlienVault

Luego se debe generar la contraseña del usuario root.

Figura 28: Configuración OSSIM AlienVault: Configuración de usuarios
Fuente: OSSIM AlienVault

Luego comenzará la instalación del sistema en el disco duro.

Figura 29: Configuración OSSIM AlienVault: Instalación
Fuente: OSSIM AlienVault

Una vez que la instalación finalice, se reinicia el servidor para ingresar a la pantalla de inicio de la herramienta OSSIM AlienVault.

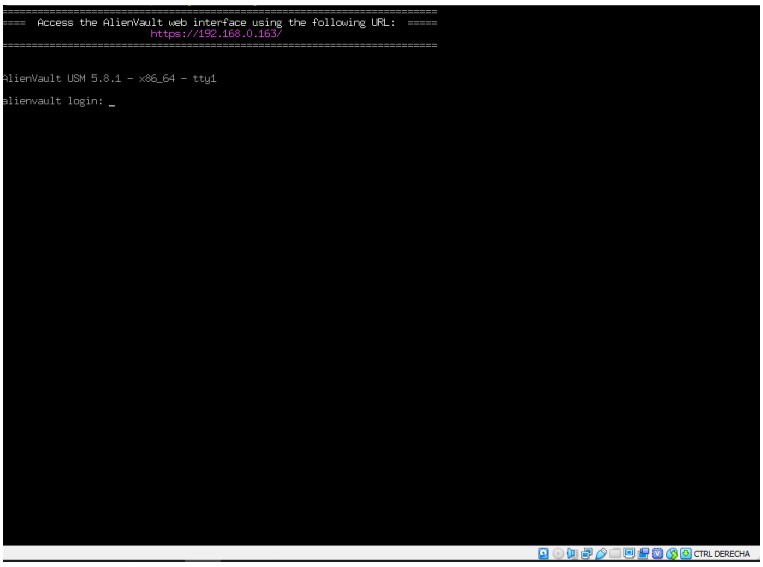


Figura 30: Configuración OSSIM AlienVault: Ingreso al sistema (Pantalla de inicio).
Fuente: OSSIM AlienVault

Al ingresar el usuario root con su respectiva contraseña se despliega la siguiente pantalla:

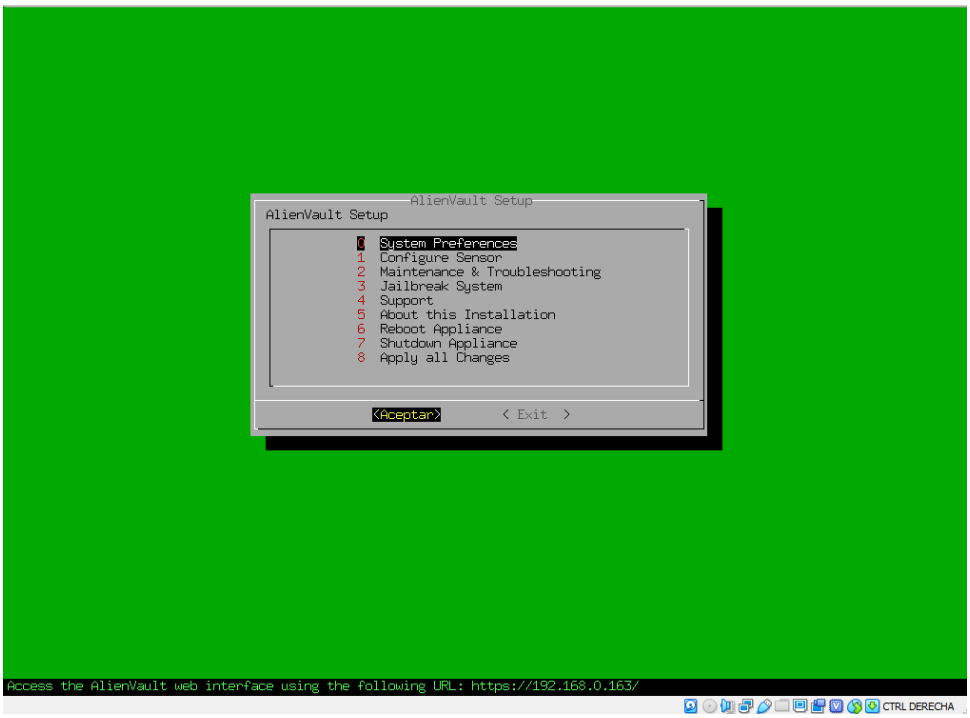


Figura 31: Configuración OSSIM AlienVault: Levantamiento de servicios.
Fuente: OSSIM AlienVault

Para validar si el servidor se encuentra con acceso local, ingresar en un navegador la IP y si se refleja un mensaje de sitio no seguro, ingresar a Configuración Avanzada y luego acceder a 192.168.0.163 (sitio no seguro)

Figura 32: Configuración OSSIM AlienVault: Entorno web.
Fuente: OSSIM AlienVault

Una vez que se ha ingresado a las opciones del punto anterior, se puede visualizar la siguiente página, para lo cual se debe detallar la información solicitada:

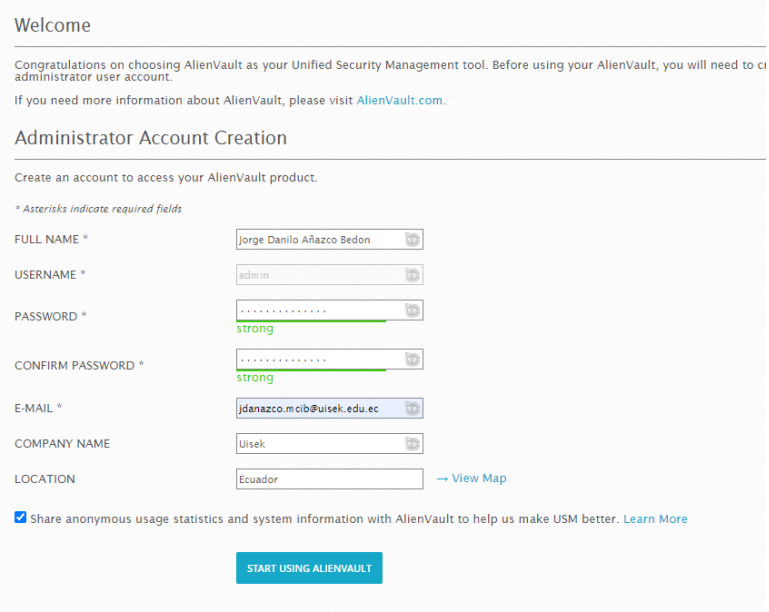


Figura 33: Configuración OSSIM AlienVault: Creación de cuenta.
Fuente: OSSIM AlienVault

Luego de ingresar los datos, damos clic en *Start Using AlienVault*, lo que permite iniciar sesión.

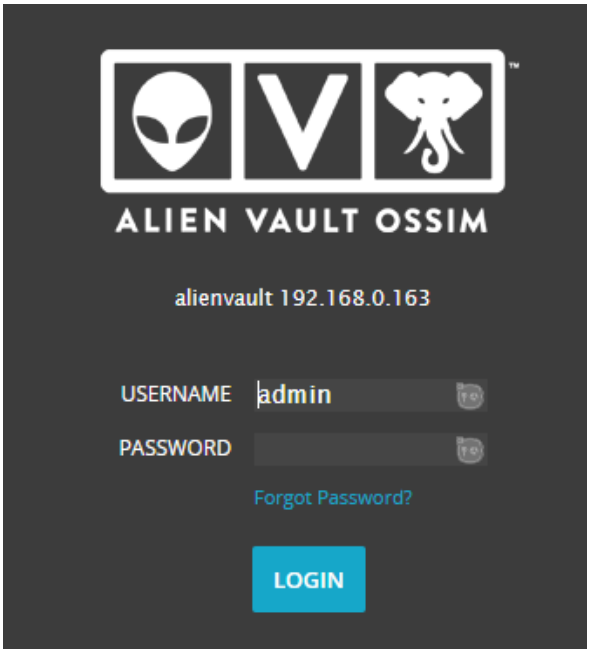


Figura 34: OSSIM AlienVault : Ingreso AlienVault.
Fuente: OSSIM AlienVault

Al ingresar por primera vez por la página web, se debe configurar el servidor de Alienvault.

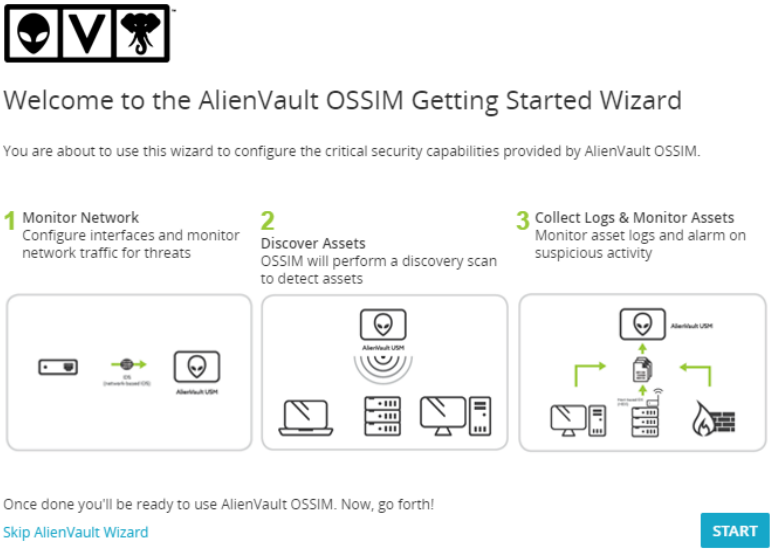


Figura 35: OSSIM AlienVault : Pantalla de inicio.
Fuente: OSSIM AlienVault

Para la configuración del servidor, primero la interface de red.

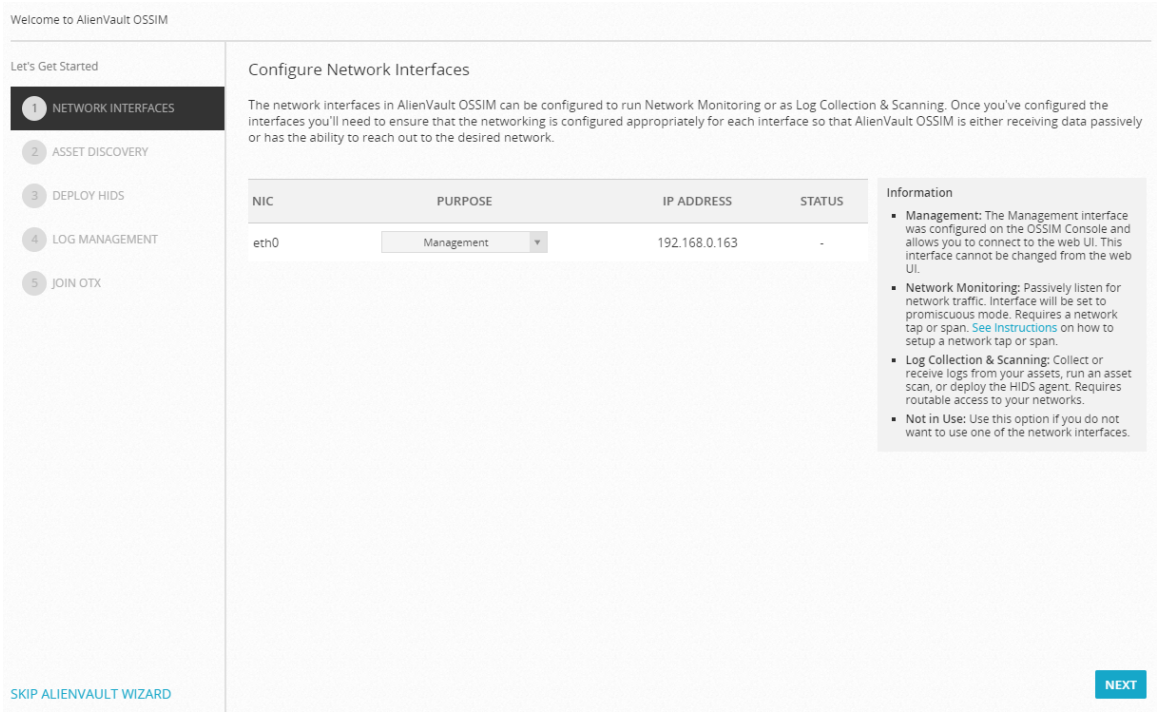


Figura 36: OSSIM AlienVault: Configuración interface de red.
Fuente: OSSIM AlienVault

La siguiente pantalla, permite agregar los dispositivos de la red y catalogarlos, esto ayudará con un inventario de equipos en el sistema.

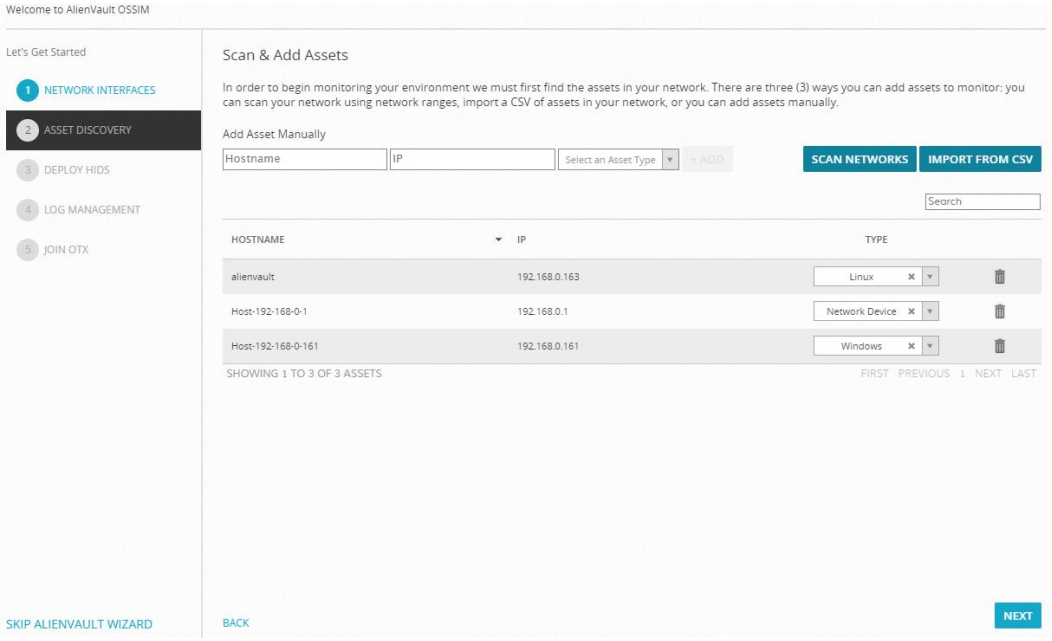


Figura 37: OSSIM AlienVault: Configuración de *asset.*
Fuente: OSSIM AlienVault

En la pantalla que se presenta a continuación, se requiere el usuario y contraseña del administrador de los equipos, para desplegar agente HIDS (Sistema de detección de intrusos en un Host) para monitorear los equipos.

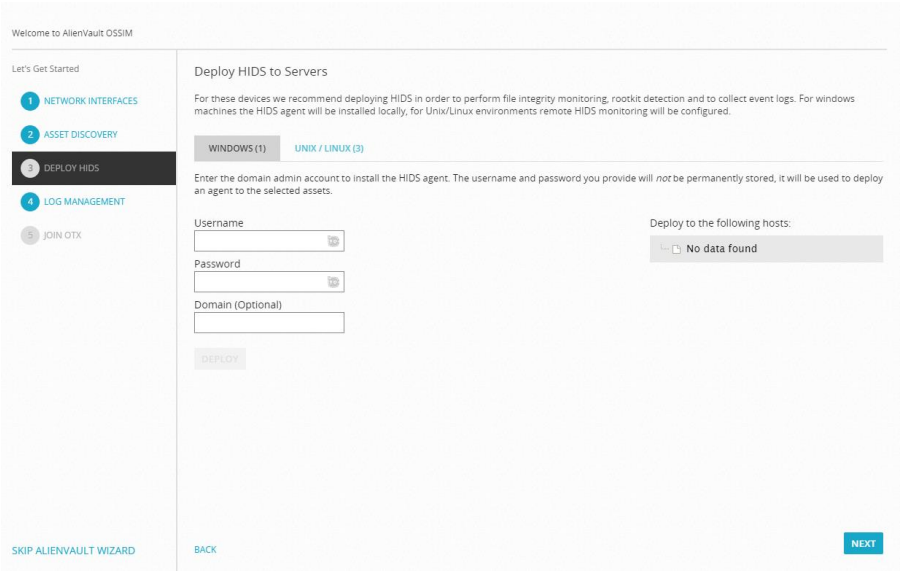


Figura 38: OSSIM AlienVault: Desplegar HIDS
Fuente: OSSIM AlienVault

En la siguiente pantalla se solicita las características del dispositivo de red para el manejo de los logs.

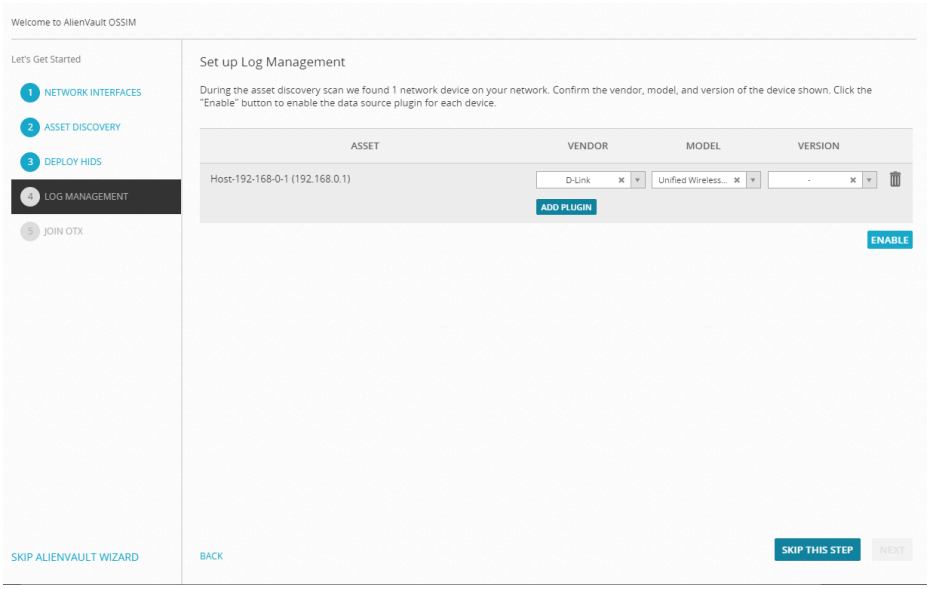


Figura 39: OSSIM AlienVault: Manejo y configuración de los logs.
Fuente: OSSIM AlienVault

A continuación, se ingresa el código de OTX que permite la integración del sistema inteligente contra amenazas creado por la comunidad.

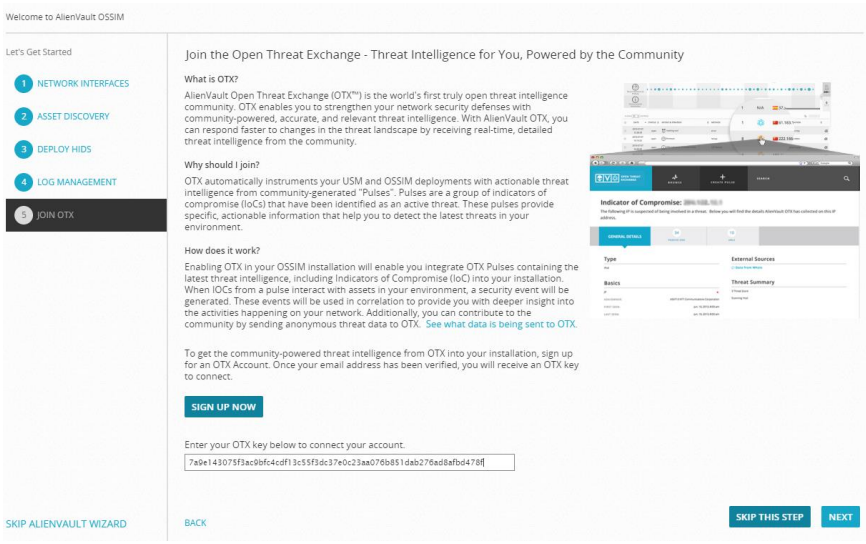


Figura 40: OSSIM AlienVault: Configuración OTX.
Fuente: OSSIM AlienVault

4.1.2 Configuración e instalación del OSSIM AlienVault

Para realizar ciertas configuraciones del AlienVault se debe ingresar a través del servicio ssh con la IP del servidor, el usuario administrador en este caso root y su respectiva contraseña, al momento de ingresar correctamente se despliega la siguiente pantalla:

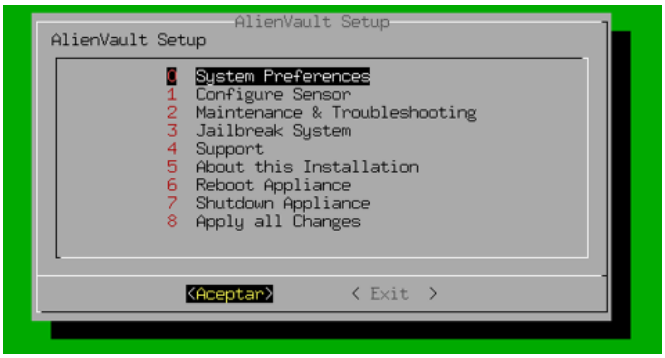


Figura 41: OSSIM AlienVault: Configuraciones de preferencias del sistema (AlienVault *Setup*).
Fuente: OSSIM AlienVault

Se procede a configurar el sensor del equipo para que detecte las IPs y realice el monitoreo respectivo, para lo cual, se ingresa a *Configure Sensor* y después se agrega la red, en este caso es la 192.168.0.0/24, y en el caso que se requiera que la herramienta monitoree otras redes, simplemente se ingresa cada red separado por comas.

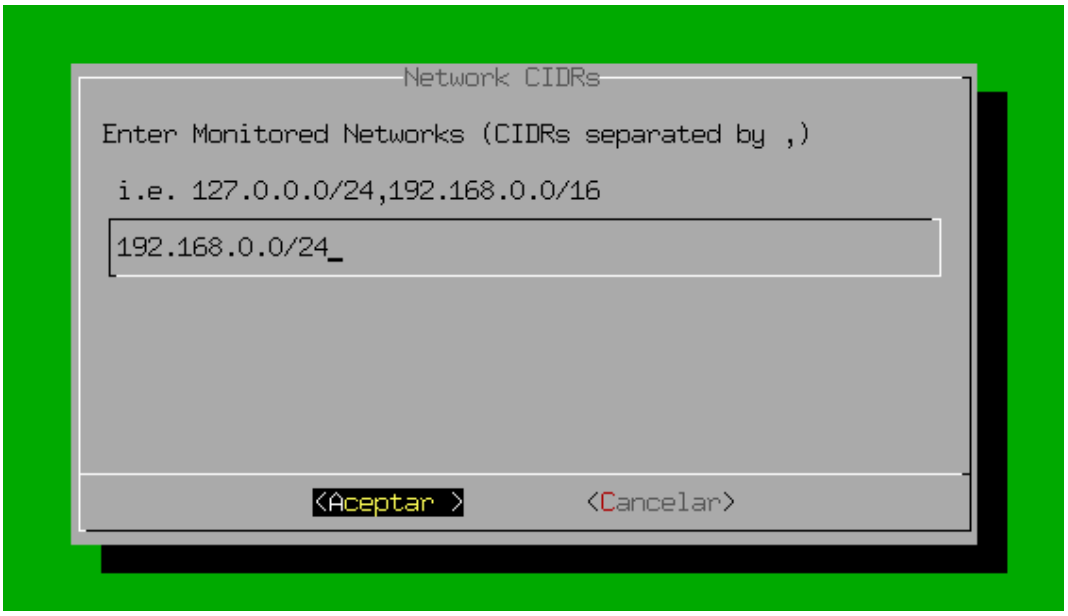


Figura 42: OSSIM AlienVault : Configuración de los sensores.
Fuente: OSSIM AlienVault

Para guardar los cambios realizados se debe ir al menú principal y seleccionar *Apply all Changes*

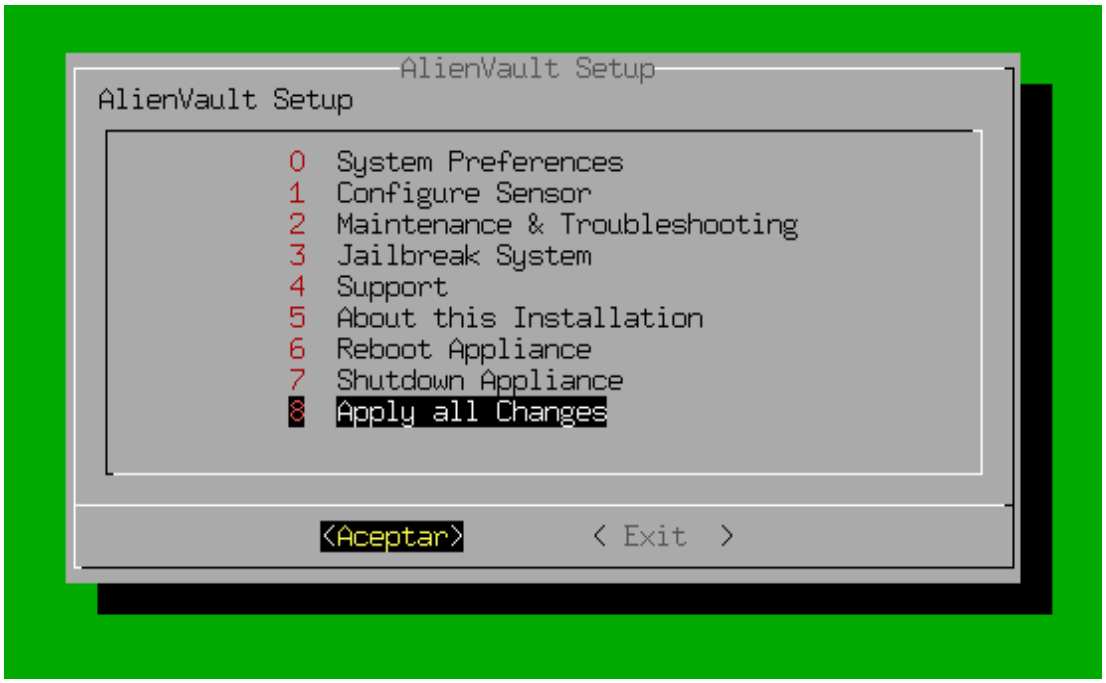


Figura 43: OSSIM AlienVault : Guardado de configuraciones.
Fuente: OSSIM AlienVault

Para ingresar a la administración del servidor a través de consola se debe dirigir a la opción *Jailbreak System* y seleccionar aceptar.

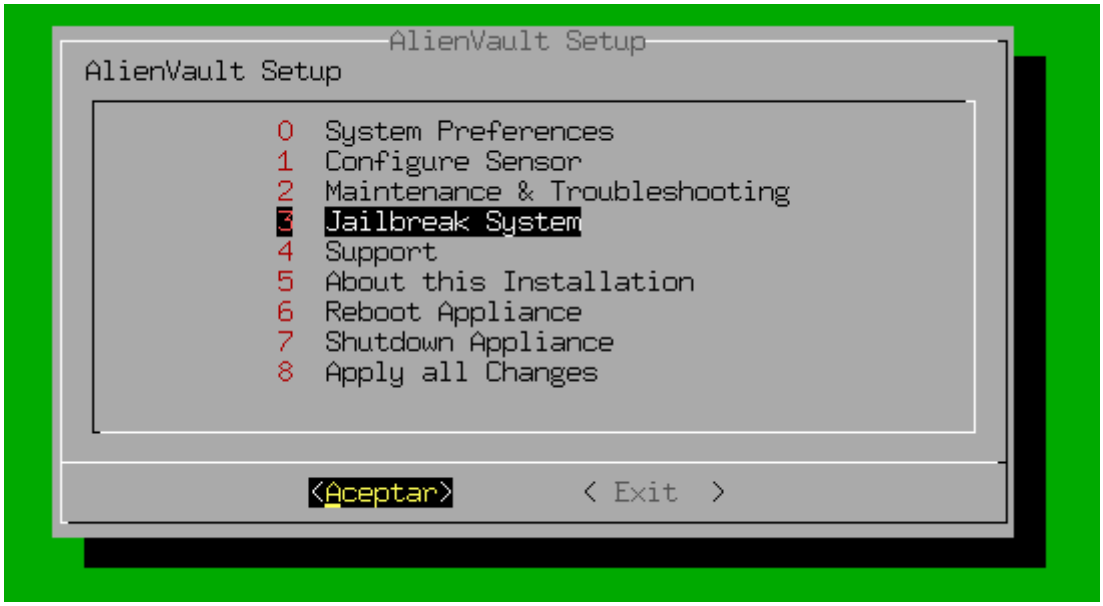


Figura 44: OSSIM AlienVault : Configuración *Jailbreak System*.
Fuente: OSSIM AlienVault

Al ingresar a la consola se puede ingresar los comandos de cualquier servidor basado en Debian.

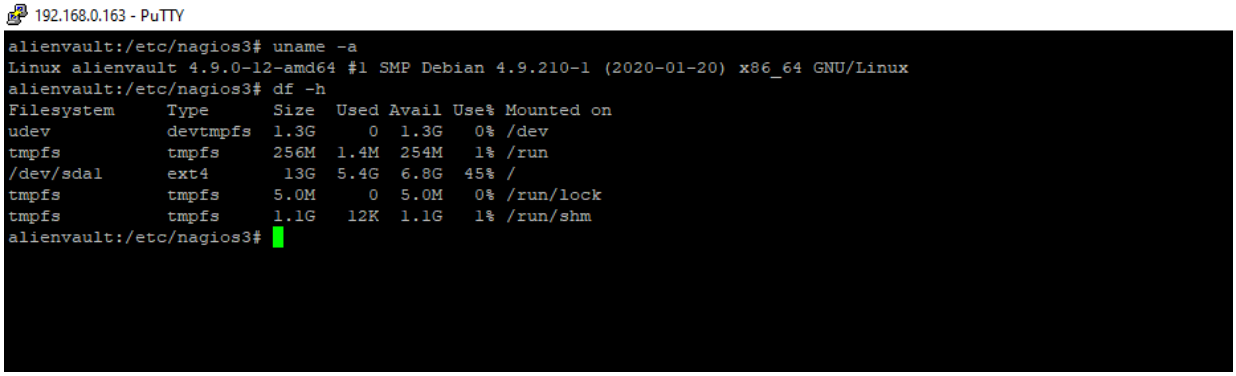


Figura 45: OSSIM AlienVault : Conexión por ssh.
Fuente: OSSIM AlienVault

4.2 Gestión de activos

Se debe ingresar por la interfaz web, se dirige a Entorno y elegir Activos y Grupos

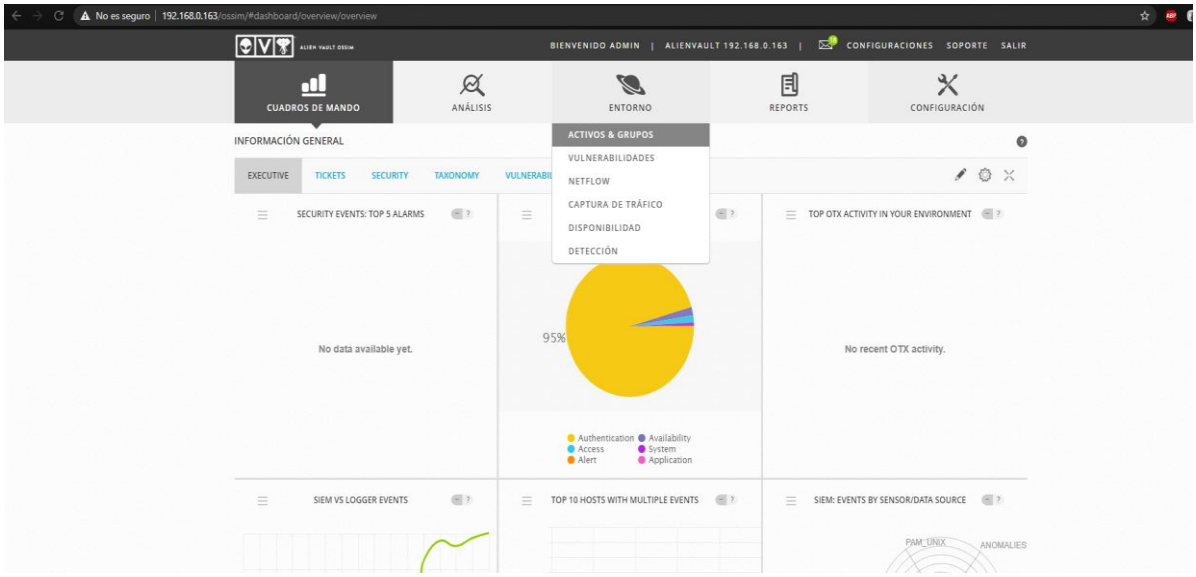


Figura 46: Gestión de activos: Consola.
Fuente: OSSIM AlienVault

En esta pantalla se despliega los dispositivos que ha detectado OSSIM, para agregar nuevos se dirige a Añadir Activos y seleccionar Scan for new assets

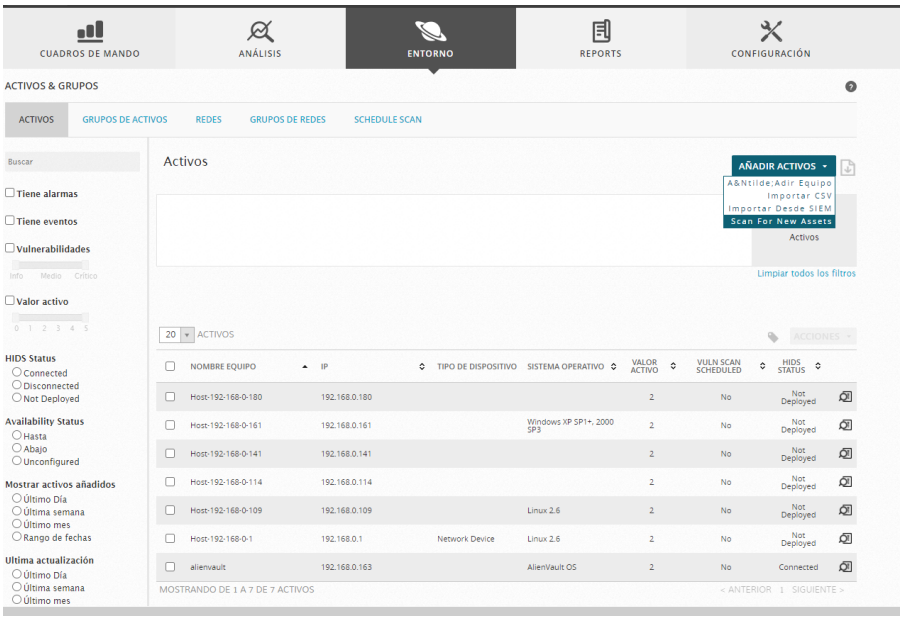


Figura 47: Gestión de activos: Escáner.
Fuente: OSSIM AlienVault

En esta pantalla se elige los objetos de red que se desea añadir, en este caso se procede a revisar los dispositivos de la red 192.16.0.0/24, con una búsqueda rápida y con el sensor que viene en el OSSIM AlienVault se escoge las opciones de Resolución de DNS y autodetectar servicios para el escaneo.

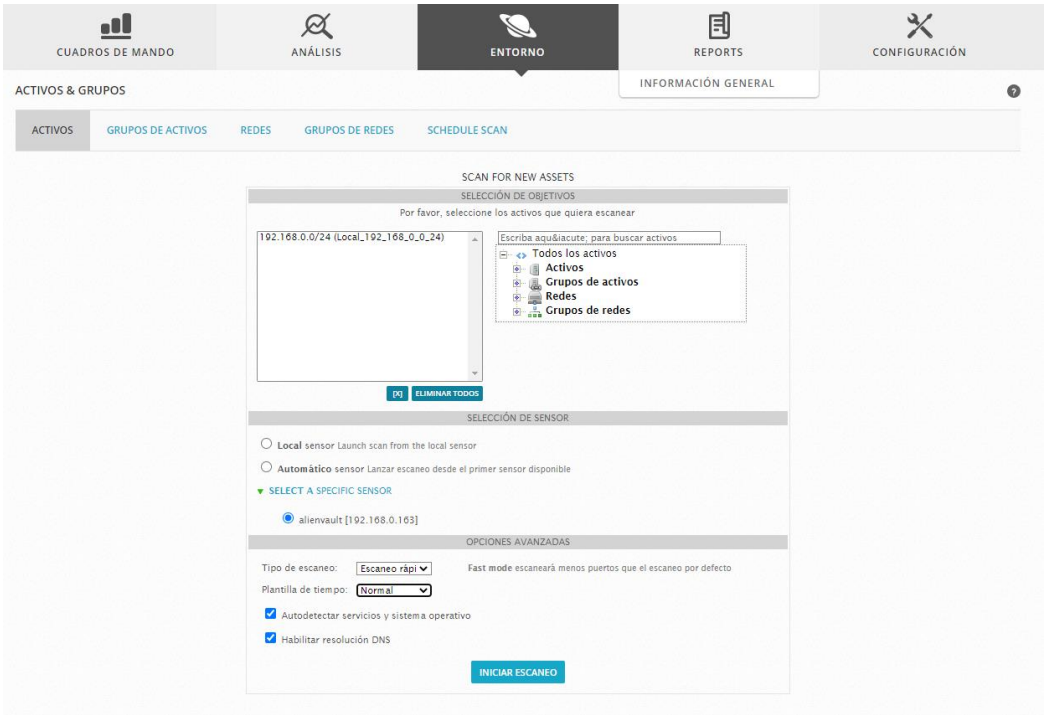


Figura 48: Gestión de activos: Búsqueda rápida
Fuente: OSSIM AlienVault

Al iniciar el escaneo se puede demorar según el alcance y el número de dispositivos que se encuentran en la red, al momento de encontrarlos se despliega una lista de los equipos, a continuación, se debe guardar con el botón *Update Manager Assets*

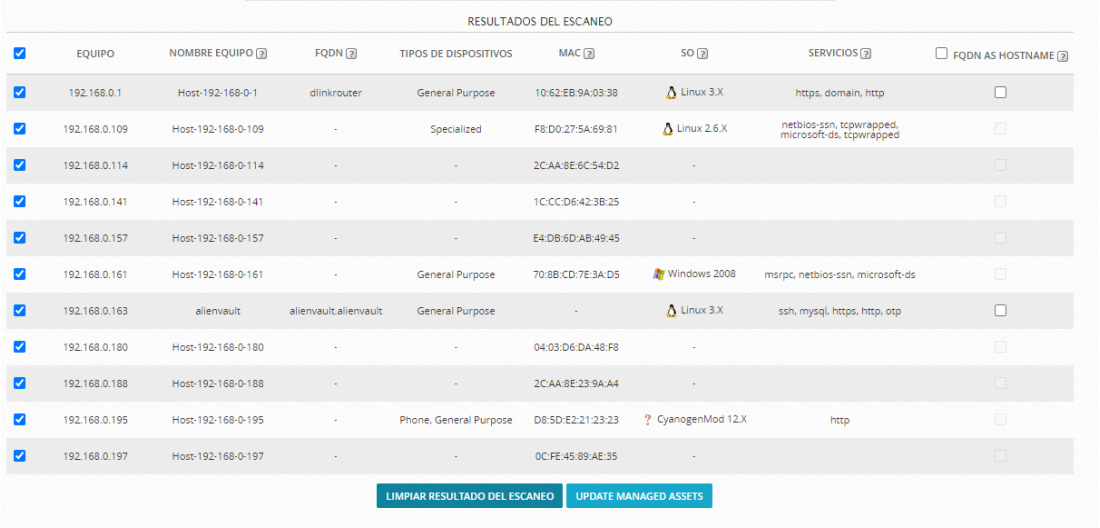


Figura 49: Gestión de activos: Update Maneger Assets.
Fuente: OSSIM AlienVault

Antes de guardar se procede a ingresar la información de la lista, por ejemplo, a qué grupo pertenece, el valor del activo, a qué sensor pertenece y la descripción.

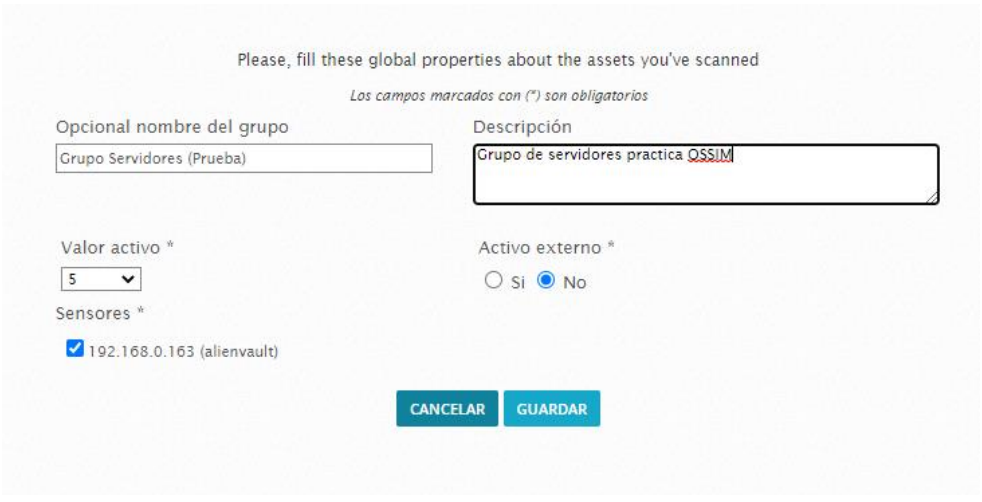


Figura 50: Gestión de activos: Creación de grupos activos.
Fuente: OSSIM AlienVault

Luego se regresa a Entorno y se elige Activos y Grupos, en la opción Grupos de Activos se visualiza las diferentes estadísticas del grupo que se acaba de crear, pero hay que mencionar

que se debe activar los módulos *Nagios, Snort y OpenVas*, para que se refleje las estadísticas de cada opción.

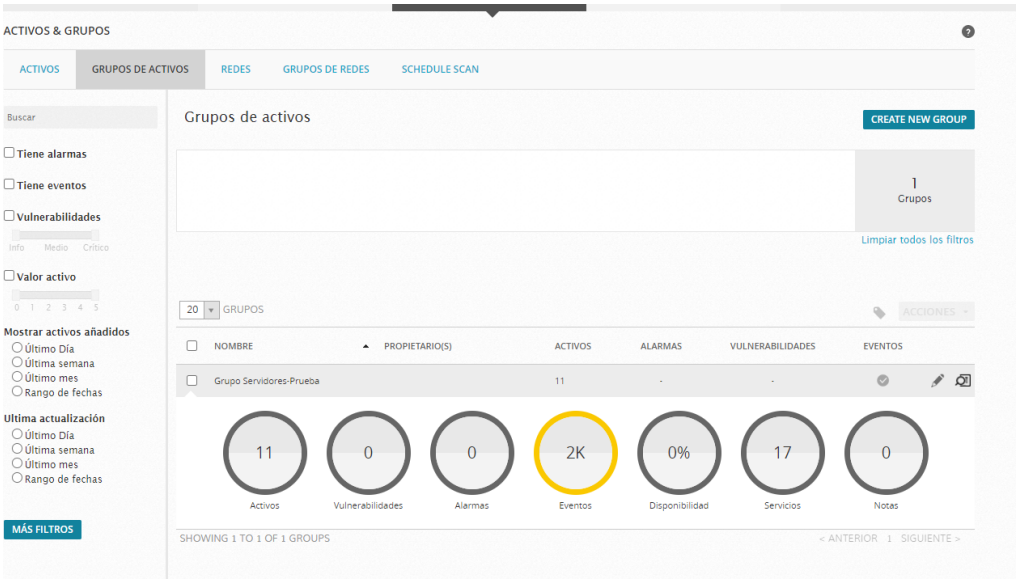


Figura 51: Gestión de activos: Módulos.
Fuente: OSSIM AlienVault

En la opción de Redes, se muestran las redes que se puede escanear con la herramienta.

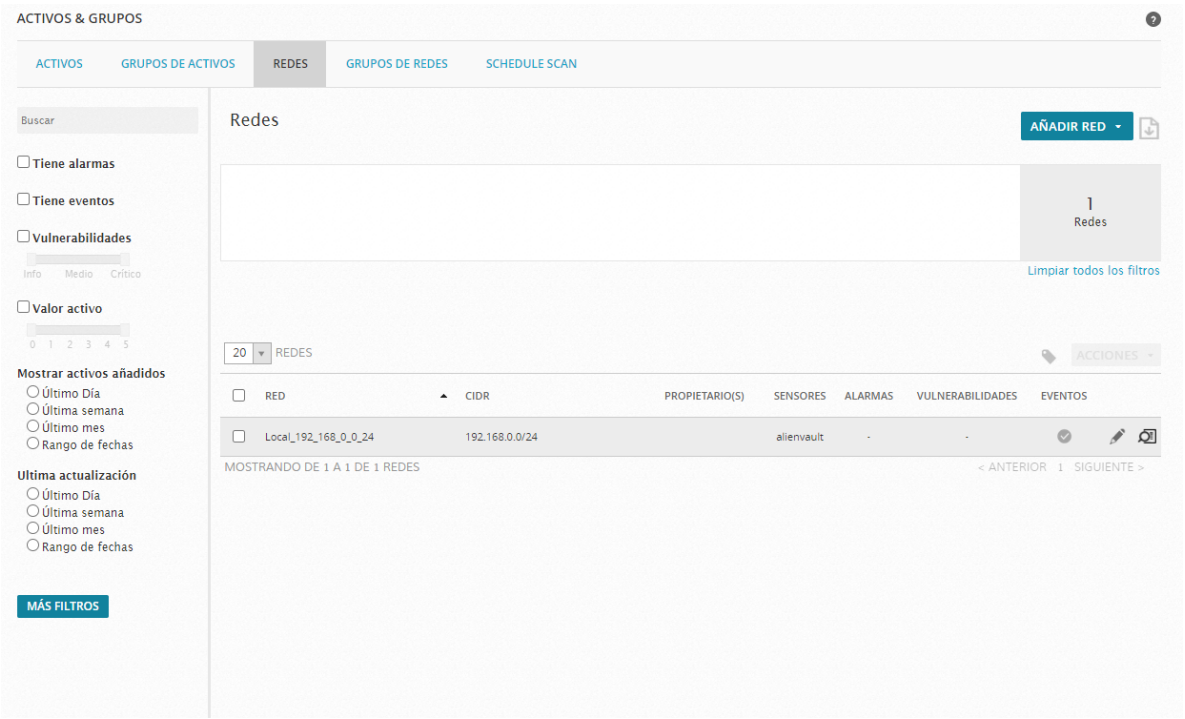


Figura 52: Gestión de activos: Redes
Fuente: OSSIM AlienVault

Para visualizar las estadísticas de cada dispositivo se dirige a Entorno y se elige la opción de Activos, y en la lista que se despliega se debe ingresar en la opción lupa, que se encuentra en la parte derecha de cada dispositivo.

Figura 53: Gestión de activos: Estadísticas.
Fuente: OSSIM AlienVault

Al ingresar se despliega toda la información del dispositivo, desde los servicios hasta las vulnerabilidades que tiene, también se puede editar algunas características de cada dispositivo.

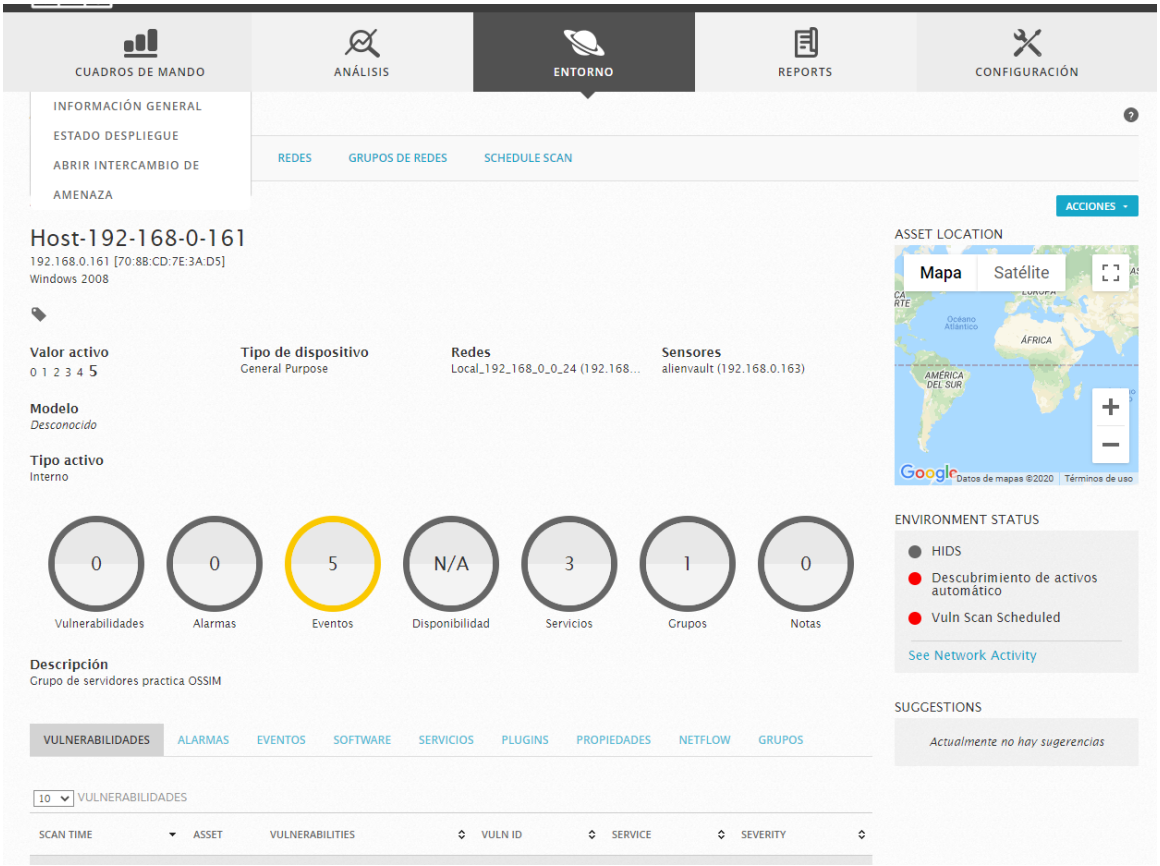


Figura 54: Gestión de activos: Dashboard.
Fuente: OSSIM AlienVault

Para editar la información del dispositivo, se ingresa a Acciones y luego Editar.

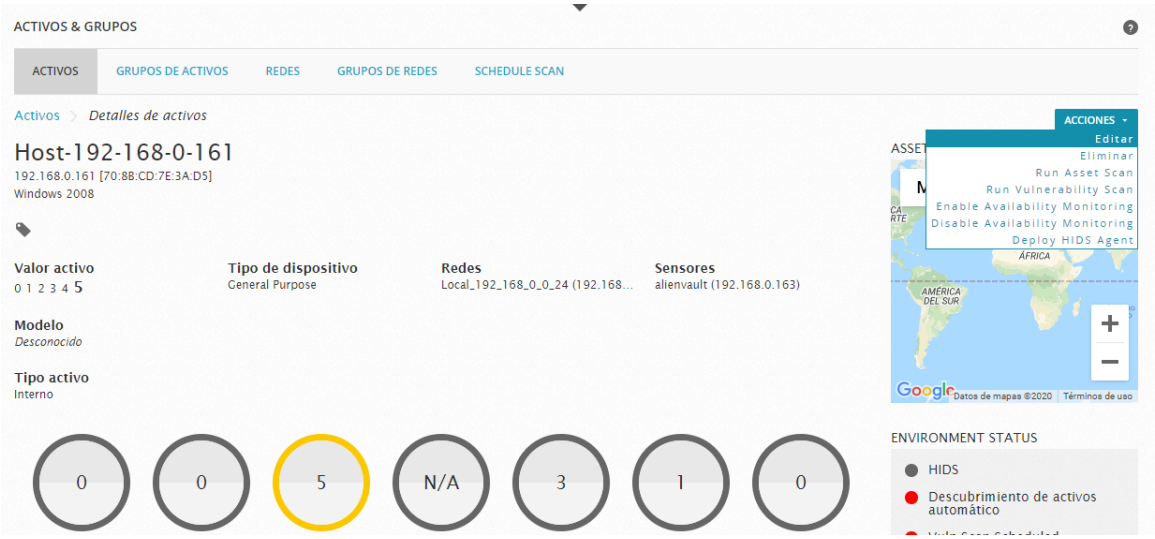


Figura 55: Gestión de activos: Editar
Fuente: OSSIM AlienVault

Al ingresar a la opción Editar se despliega la siguiente pantalla, donde se configura y actualiza la información del dispositivo.

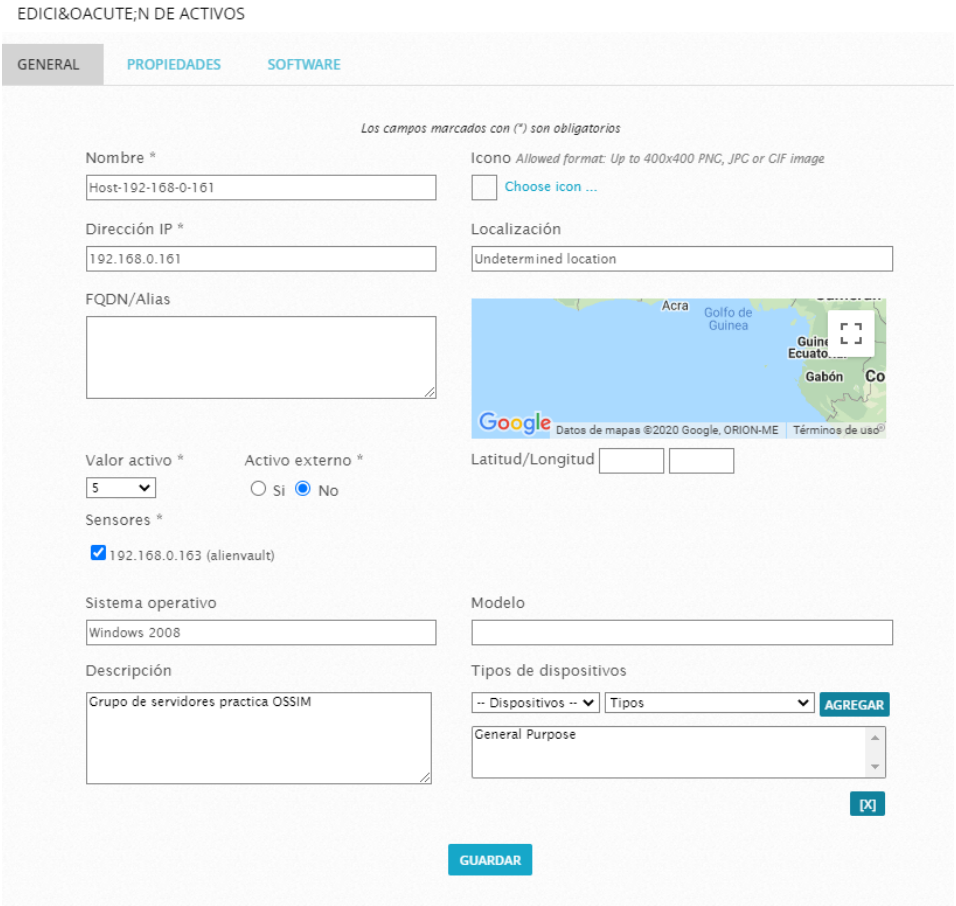


Figura 56: Gestión de activos: Editar y actualizar.
Fuente: OSSIM AlienVault

4.3 Gestión de disponibilidad

Para habilitar el monitoreo en la herramienta hay que ingresar al Módulo de Disponibilidad y dirigirse a Entorno y se elige opción de Activos, en la lista que se despliega, se ingresa en la opción de lupa que se encuentra en la parte derecha de la lista por cada dispositivo, al ingresar se visualiza la información del dispositivo y en Acciones se encuentra la opción para habilitar el monitoreo la opción es *Enable Availability Monitoring*, al hacer clic en esta opción se refleja el mensaje que ya está habilitado el monitoreo.

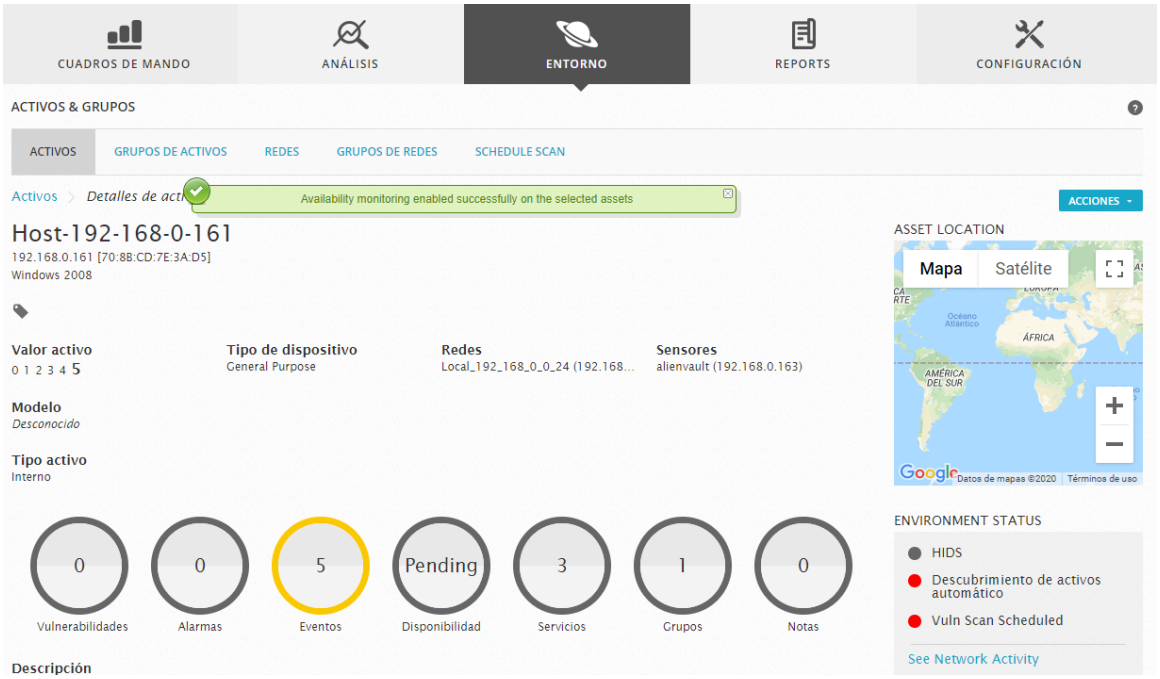


Figura 57: Gestión de disponibilidad: Activación.
Fuente: OSSIM AlienVault

Para habilitar el monitoreo de los servicios se dirige en la parte inferior de la pantalla anterior y se ingresa a *Edit Services.*

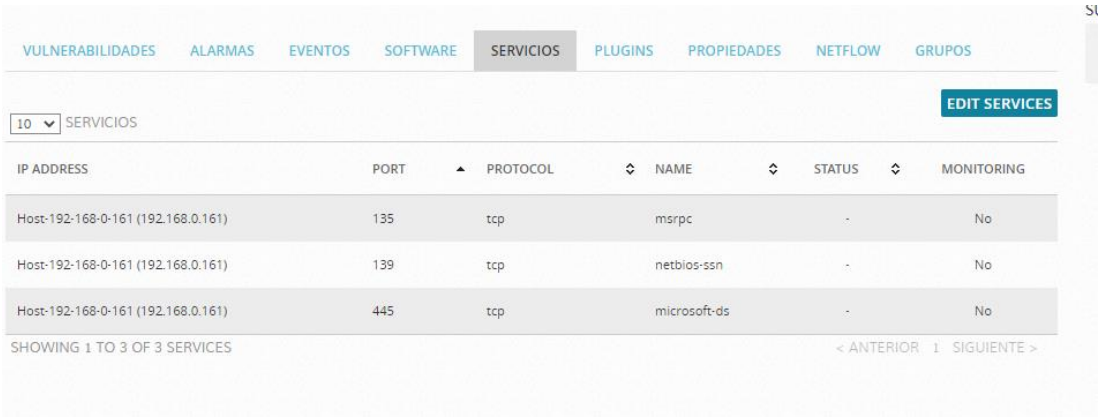


Figura 58: Gestión de disponibilidad: Editar servicios.
Fuente: OSSIM AlienVault

Se despliega la lista de servicios que tiene el servidor, y en la columna Monitorizando, se tiene la opción para habilitar el monitoreo de servicios.

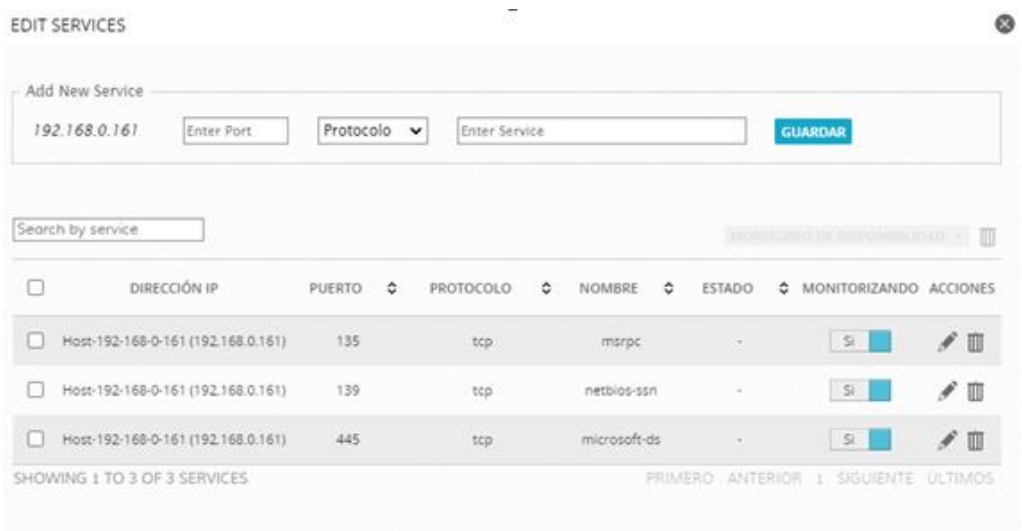


Figura 59: Gestión de disponibilidad: Servicios.
Fuente: OSSIM AlienVault

Al habilitar el monitoreo de los servicios se actualiza la información de los paneles, como se muestra a continuación:

Figura 60: Gestión de disponibilidad: Paneles de información.
Fuente: OSSIM AlienVault

Para visualizar las estadísticas globales de monitoreo se dirige a Entorno y se elige la opción de disponibilidad.

Figura 61: Gestión de disponibilidad: Estadísticas globales.
Fuente: OSSIM AlienVault

Al ingresar se despliega la información de todos los dispositivos que se encuentran en el monitoreo sus respectivos servicios.

Figura 62: Gestión de disponibilidad: *Dashboard*.
Fuente: OSSIM AlienVault

Los servicios que se encuentran disponibles están en verde, los que presenten algún problema cambiarán de color a rojo o a naranja, según el estado del servicio o del dispositivo.

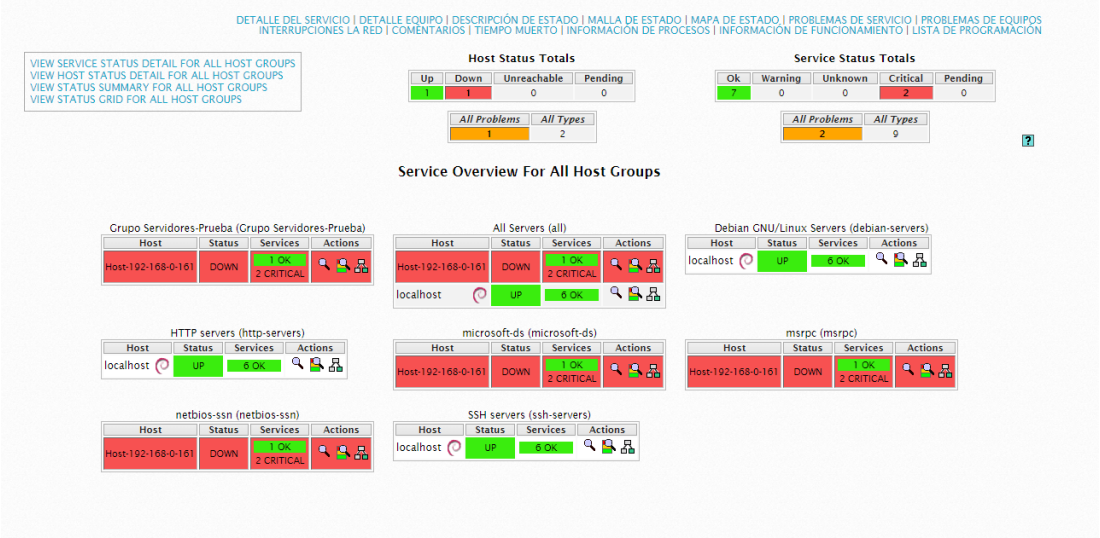


Figura 63: Gestión de disponibilidad: Verificación de servicios.
Fuente: OSSIM AlienVault

Una de las funcionalidades de este módulo es la creación de informes, ya sea por dispositivos o grupos de dispositivos, por servicio o grupo de servicios y por el estado, para esta función se ingresa a informes tendencias, se elige el dispositivo, el lapso de tiempo y las características del informe, cabe señalar que existe varios tipos de gráficos para la generación de informes, según las necesidades del administrador.

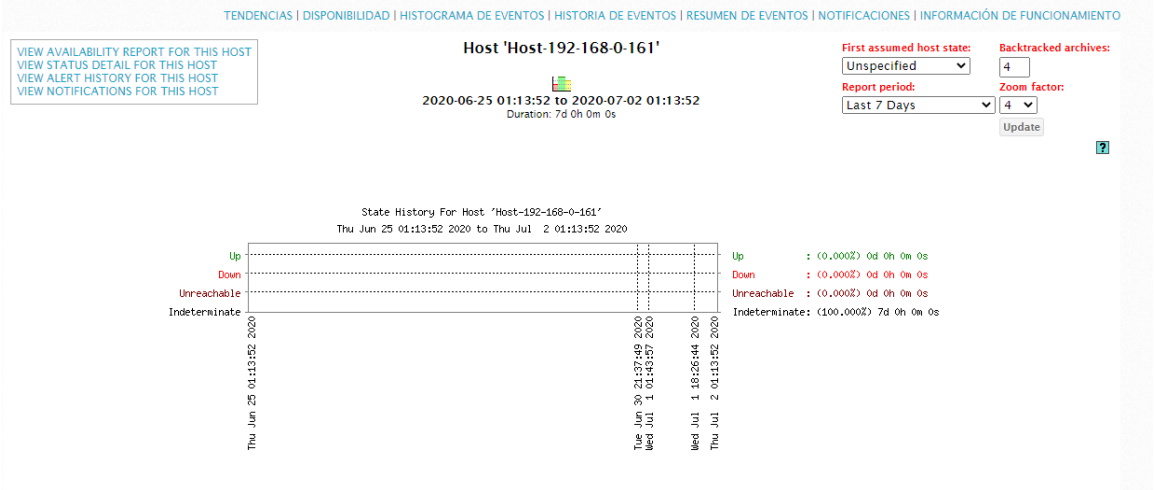


Figura 64: Gestión de disponibilidad: Informes.
Elaborado por: Jorge Añazco

4.4 Gestión de notificaciones

Para configurar el envío de notificaciones de Nagios, se ingresa al servidor por ssh y como primer requisito se establece los contactos que se enviara las notificaciones, para editar los contactos se ingresa a /etc/nagios3/conf.d/contacts_nagios2.cfg.

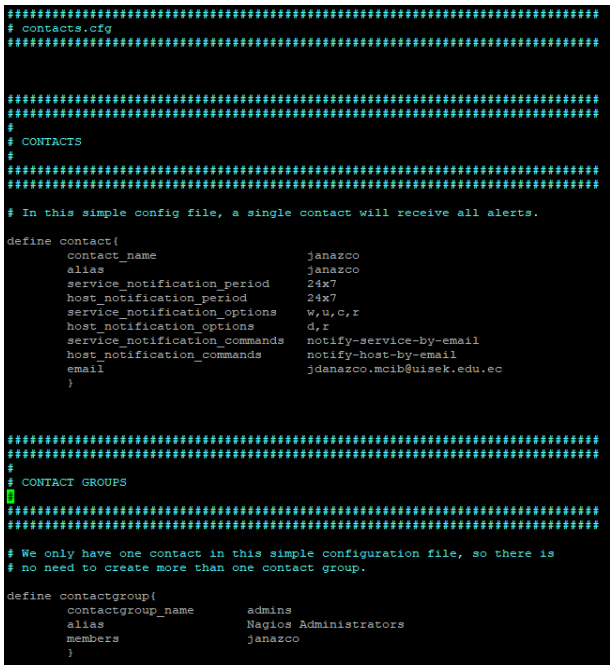


Figura 65: Gestión de notificaciones: Configuración.
Fuente: OSSIM AlienVault (Nagios)

Para aplicara los cambios se reinicia el servicio de Nagios.

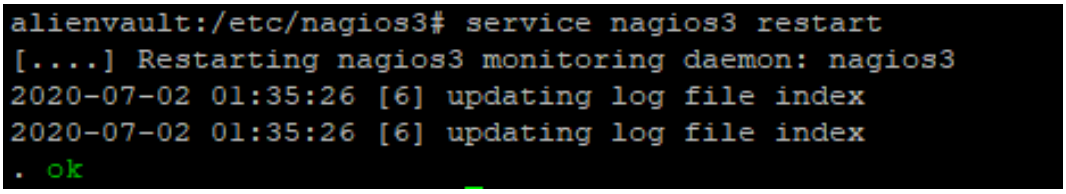


Figura 66: Gestión de notificaciones: Configuración nagios.
Elaborado por: Jorge Añazco

Según las necesidades del administrador se puede configurar el servicio de mail saliente ya que puede ser sendmail o Postfix, en la instalación por defecto de la herramienta OSSIM AlienVault se encuentra por defecto Postfix como servidor de correo, pero con el comando dpkg-reconfigure postfix o dpkg-reconfigure sendmail se puede establecer y configurar el servidor de correo.

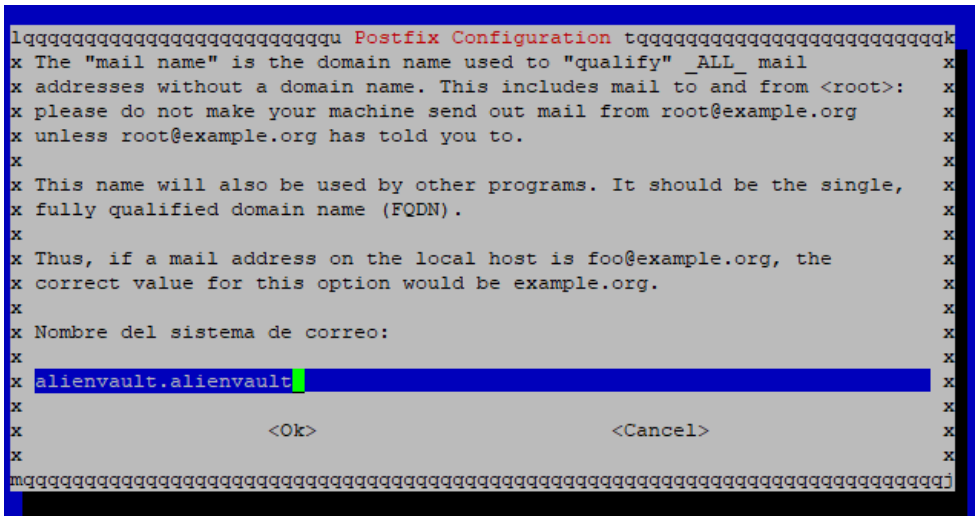


Figura 67: Gestión de notificaciones: Configuración correo nombre del sistema (1).
Fuente: OSSIM AlienVault

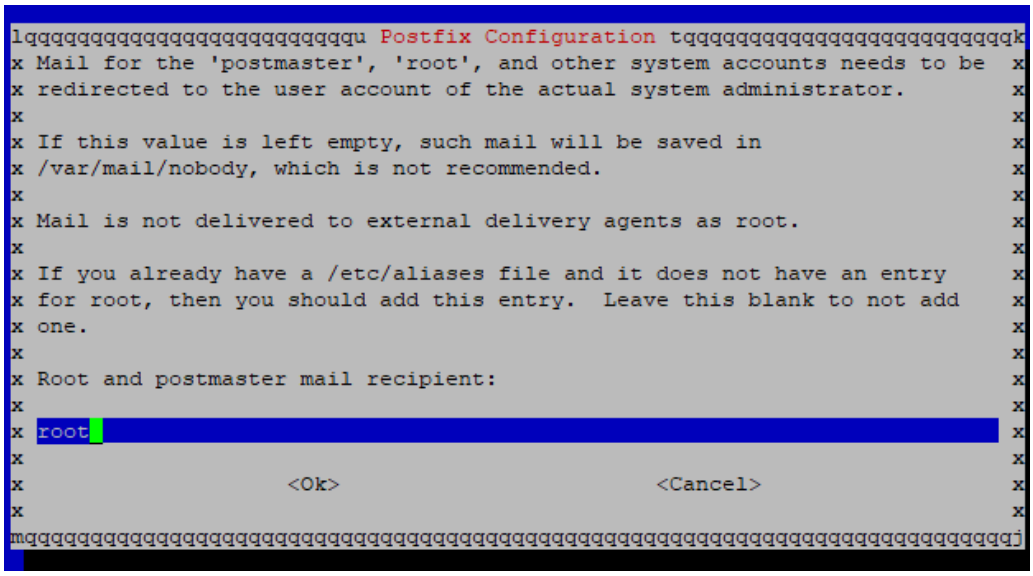


Figura 68: Gestión de notificaciones: Configuración correo remitente (2).
Fuente: OSSIM AlienVault

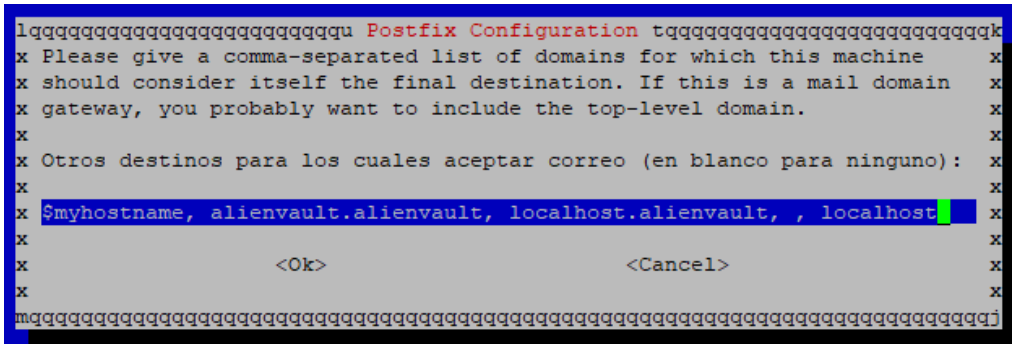


Figura 69: Gestión de notificaciones: Configuración correo dominio (3).
Fuente: OSSIM AlienVault

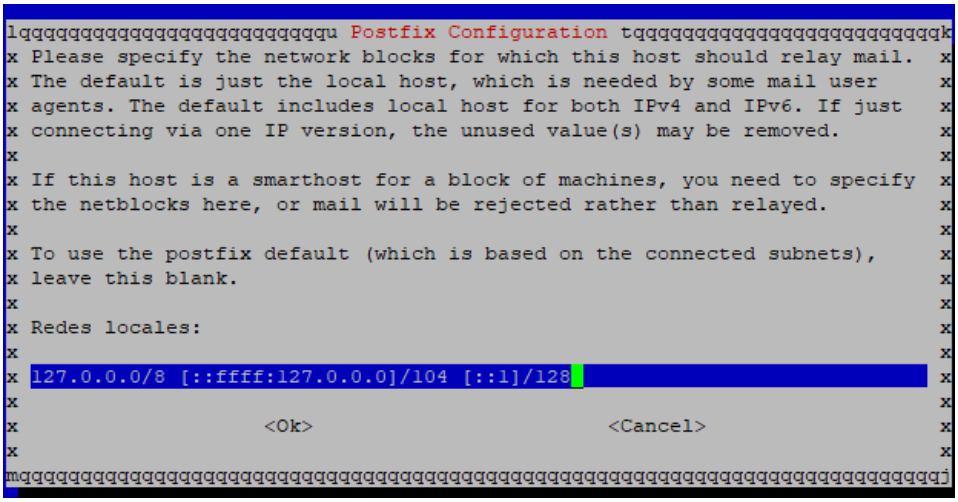


Figura 70: Gestión de notificaciones: Configuración correo redes (4).
Fuente: OSSIM AlienVault

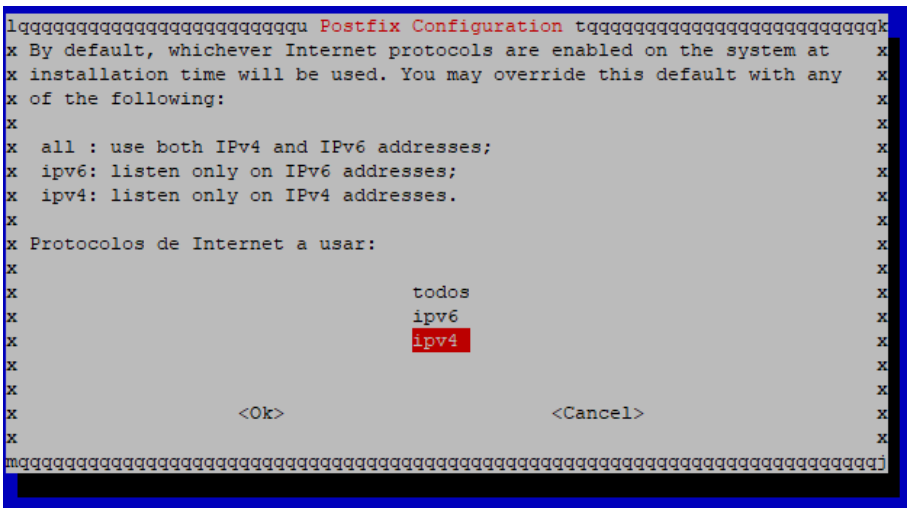


Figura 71: Gestión de notificaciones: Configuración correo protocolo de internet (5).
Fuente: OSSIM AlienVault

Después de la configuración del servidor de correo, se realiza la prueba de envío del mail con el comando mail.

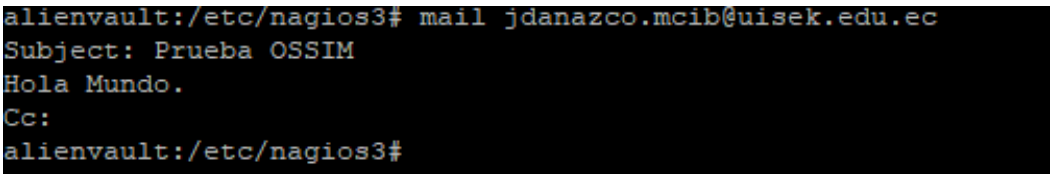


Figura 72: Gestión de notificaciones: Envió de prueba de correo.
Elaborado por: Jorge Añazco

Al realizar pruebas de disponibilidad del servicio, al momento que exista un problema el nagios notificará al mail.

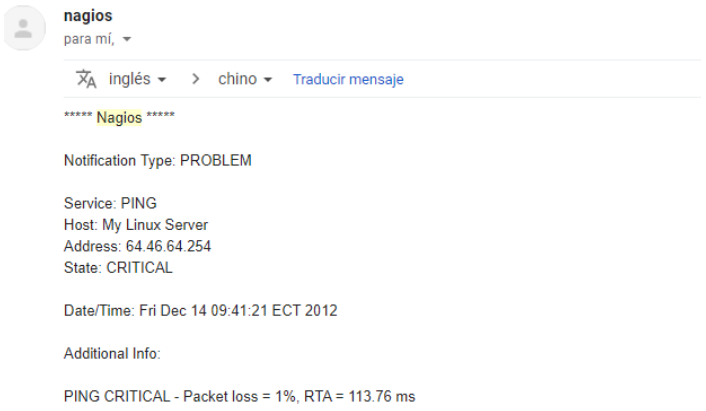


Figura 73: Gestión de notificaciones: Correo de prueba.
Elaborado por: Jorge Añazco

Para editar el contenido de las notificaciones vía mail, se puede hacer el cambio en el archivo /etc/nagios3/commands.cfg, se agrega más métodos de notificación por vía SMS y por whatsapp.

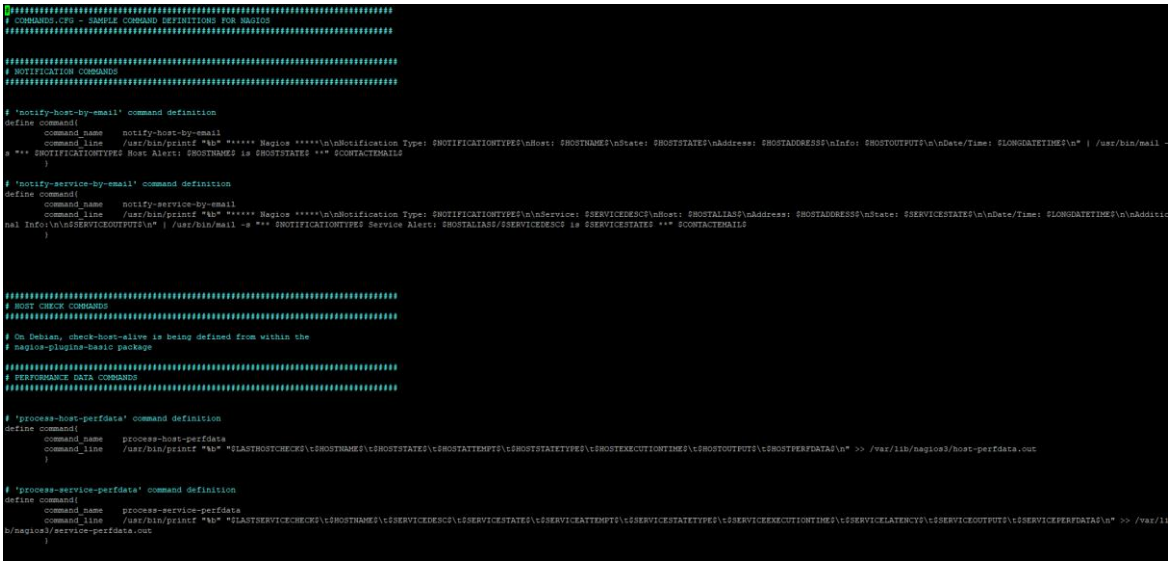


Figura 74: Gestión de notificaciones: Configuración correo reglas nagios.
Elaborado por: Jorge Añazco

4.5 Gestión de vulnerabilidades

Para ingresar a la gestión de vulnerabilidades se dirige a Entorno y se selecciona Vulnerabilidades.

Figura 75: Gestión de vulnerabilidades: Ingreso.
Fuente: OSSIM AlienVault

Al ingresar en el panel vulnerabilidades, se dirige a trabajos de escaneo para comenzar con la programación del escaneo de vulnerabilidades.

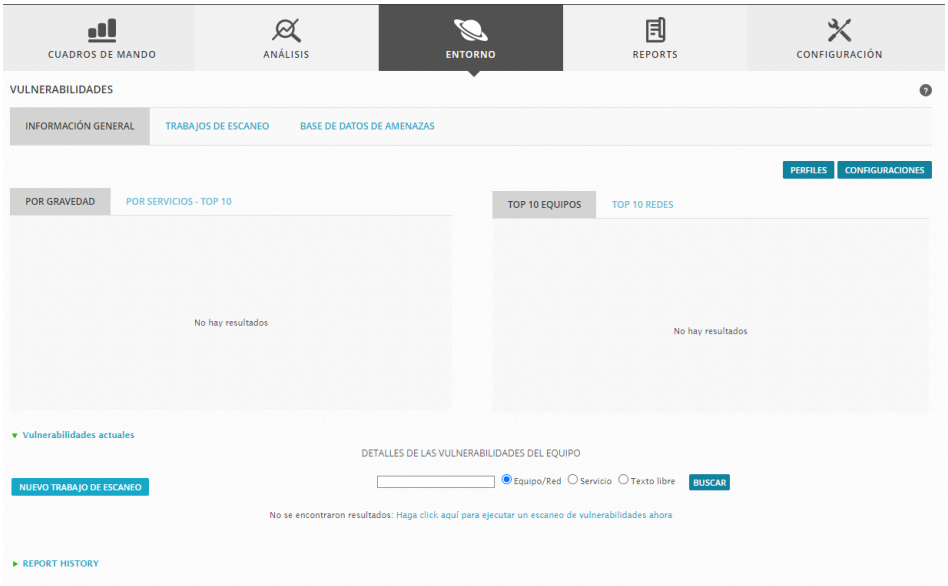


Figura 76: Gestión de vulnerabilidades: *Dashboard*.
Fuente: OSSIM AlienVault

En esta pantalla hay que ingresar a nuevo trabajo de escaneo, para programar un escaneo de vulnerabilidades a un activo o dispositivo.

Figura 77: Gestión de vulnerabilidades: Escaneo de vulnerabilidades.
Fuente: OSSIM AlienVault

En la pantalla *CREATE SCAN JOB*, se puede realizar varios tipos de escaneo, desde un escaneo exhaustivo, personalizado y con niveles profundidad hasta un escaneo rápido, también se puede dejar tareas programadas para que se ejecute en un periodo de tiempo, por último, se puede escanear a varios activos a la vez.

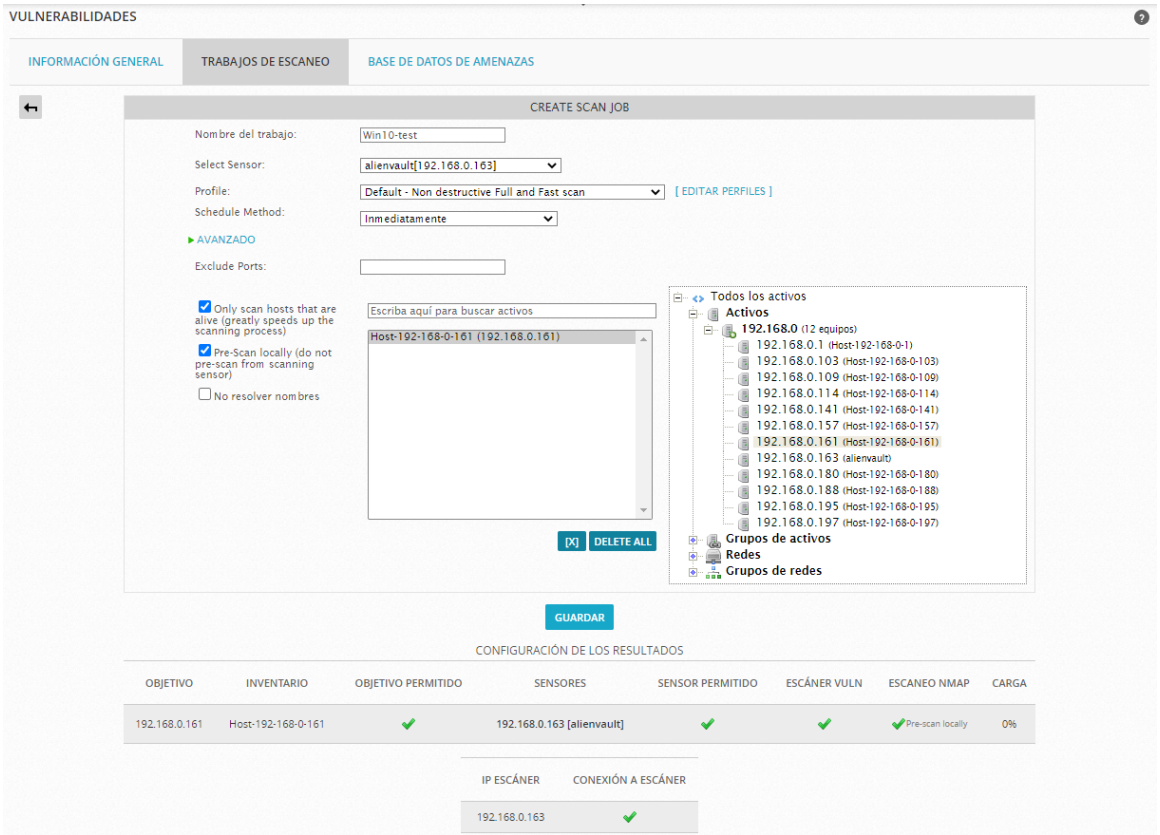


Figura 78: Gestión de vulnerabilidades: Programar un trabajo de escaneo.
Fuente: OSSIM AlienVault

Para guardar el informe o reporte de escaneo se debe esperar hasta que termine el proceso para que aparezca la opción de guardado.

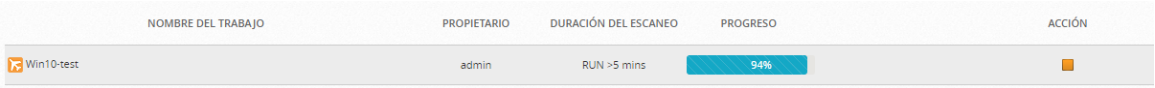


Figura 79: Gestión de vulnerabilidades: Progreso de escaneo.
Fuente: OSSIM AlienVault

Al momento de terminar el escaneo, se genera el reporte de vulnerabilidades, también se visualiza un resumen del proceso de escaneo, como puede ser el tiempo de duración y las fechas.

Figura 80: Gestión de vulnerabilidades: Escaneo completado.
Fuente: OSSIM AlienVault

Al momento que se genera el reporte se exportar o guardar en formato de tipo web, pdf, Excel, a continuación, se visualiza el reporte web que se genera:

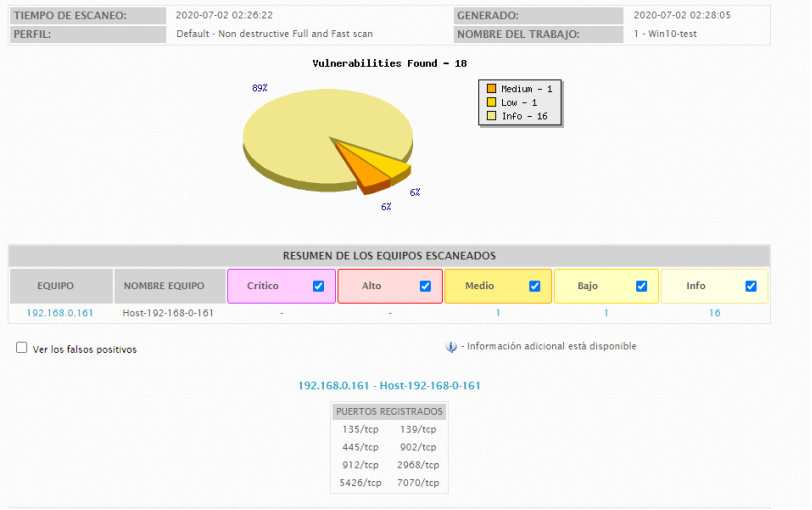


Figura 81: Gestión de vulnerabilidades: Informe de escaneo.
Fuente: OSSIM AlienVault

En el reporte indica un estado de las vulnerabilidades en un diagrama de pastel, cada tipo de vulnerabilidad es representado por un color, por ejemplo, las vulnerabilidades serias de color morado, las vulnerabilidades altas de color rojo, las vulnerabilidades medias de color amarillo, las vulnerabilidades bajas de color azul y las vulnerabilidades que solo son informativas o de advertencia de color verde, esta información también se puede visualizar en la opción de Información General, además de ver los escaneos anteriores de otros activos.

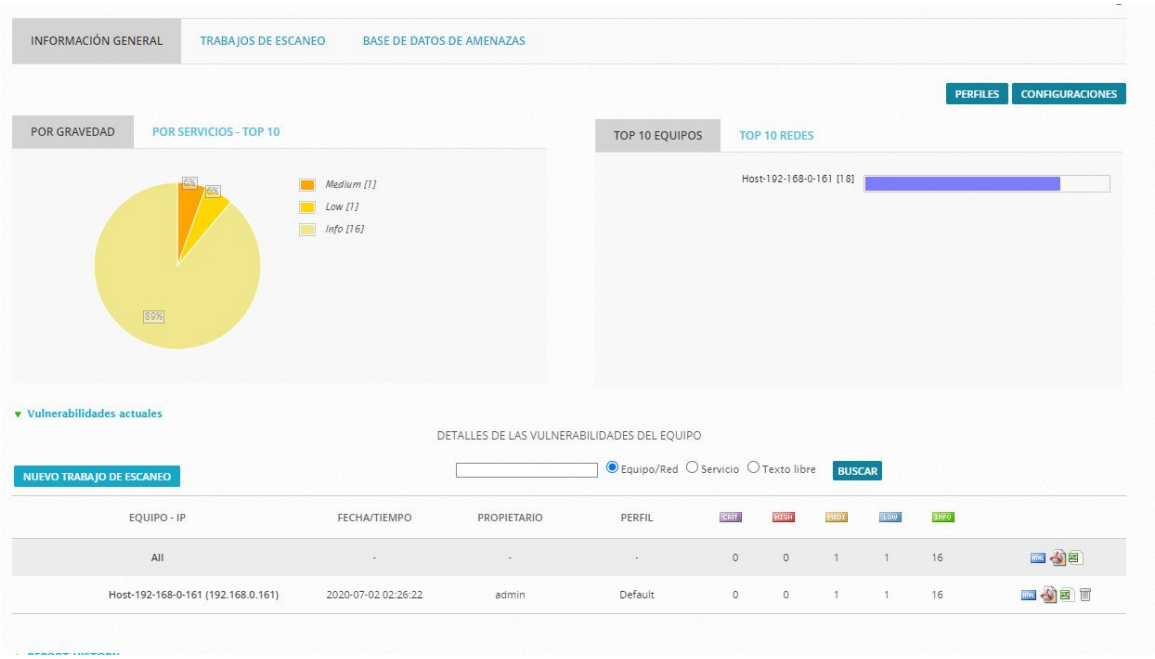


Figura 82: Gestión de vulnerabilidades: Representación de las vulnerabilidades.
Fuente: OSSIM AlienVault

4.6 Gestión de riesgos

Para ingresar al módulo de gestión de riesgos, se dirige a la parte de Análisis, a la opción de Eventos SIEM.

Figura 83: Gestión de riesgos: Ingreso.
Fuente: OSSIM AlienVault

Al ingresar se despliega una lista de los diferentes eventos de seguridad detectados por los diferentes módulos de OSSIM AlienVault, en esta pantalla se visualiza el riesgo, la descripción, el origen, el destino y el grado del riesgo detectado.

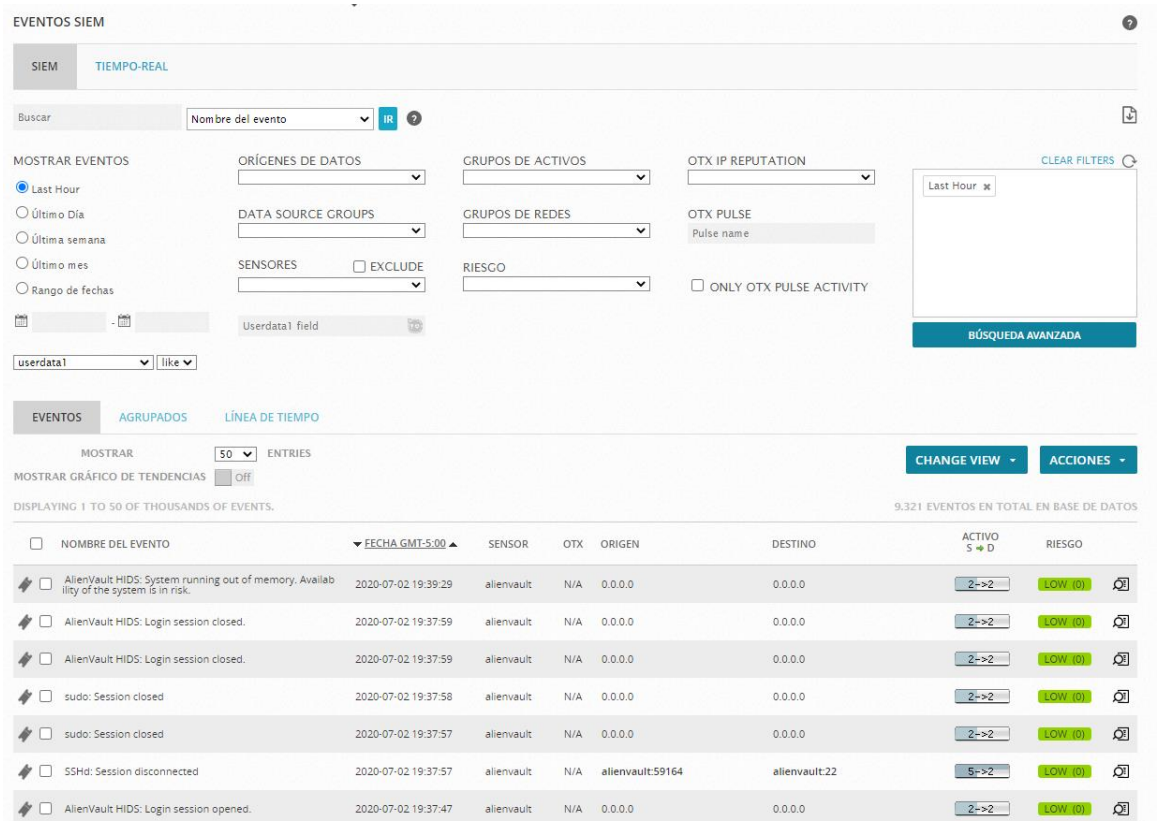


Figura 84: Gestión de riesgos: *Dashboard*.
Fuente: OSSIM AlienVault

Para ver su funcionamiento se realiza una prueba para la detección de eventos de seguridad simplemente tratando de ingresar al servidor con un usuario que no existe.

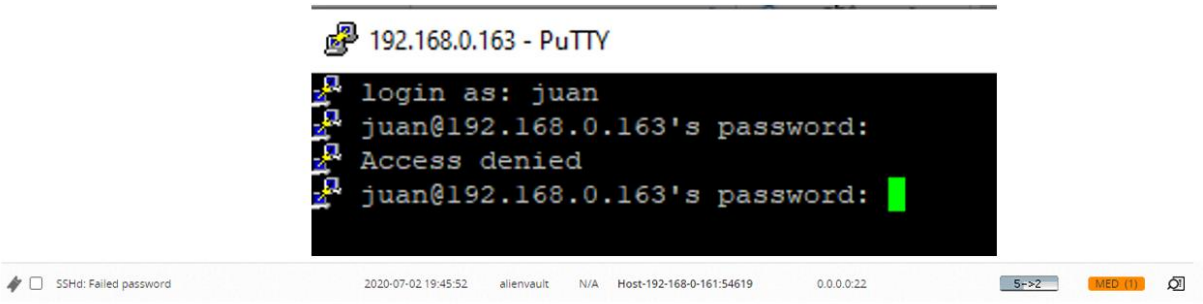


Figura 85: Gestión de riesgos: Prueba de detección.
Elaborado por: Jorge Añazco

Al dar clic en el evento se reflejará una pantalla con los detalles del evento de seguridad.

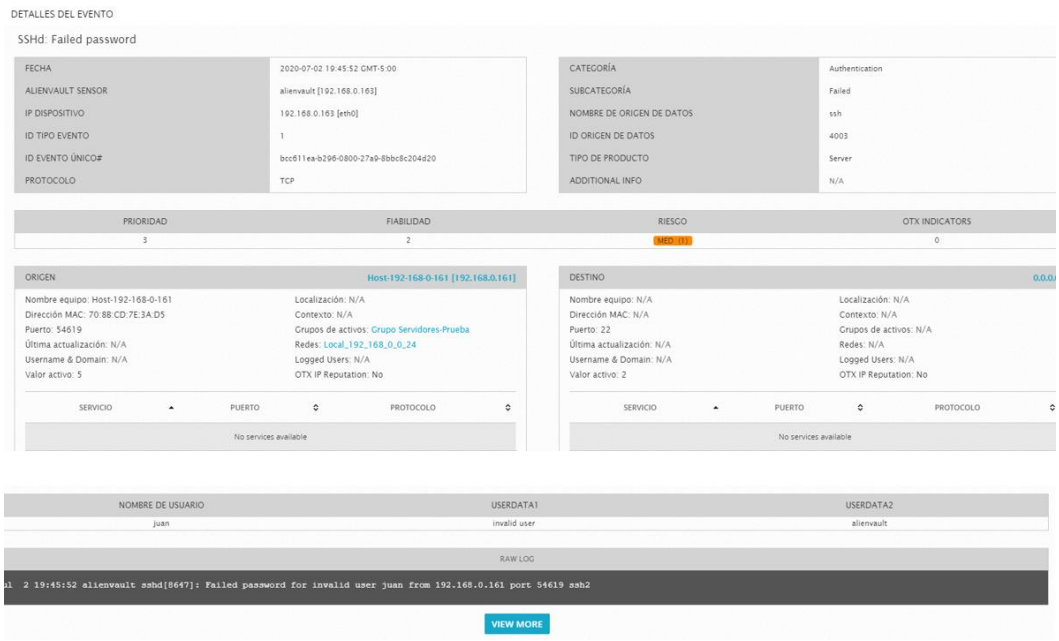


Figura 86: Gestión de riesgos: Evento de seguridad.
Fuente: OSSIM AlienVault

En la pantalla principal de los Eventos SIEM, también existe la opción de filtrar los eventos según las necesidades del administrador, también pueden ser agrupados por el tipo de evento.

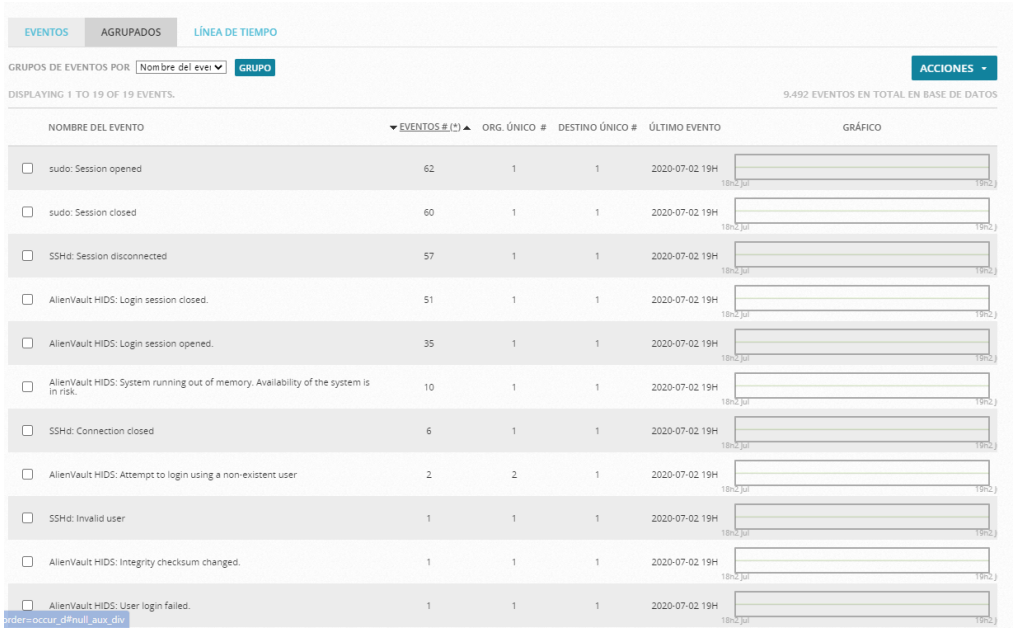


Figura 87: Gestión de riesgos: Filtrar eventos.
Fuente: OSSIM AlienVault

En la parte de la Línea de tiempo se observa de manera más exacta los eventos de seguridad y cuándo se detectaron.

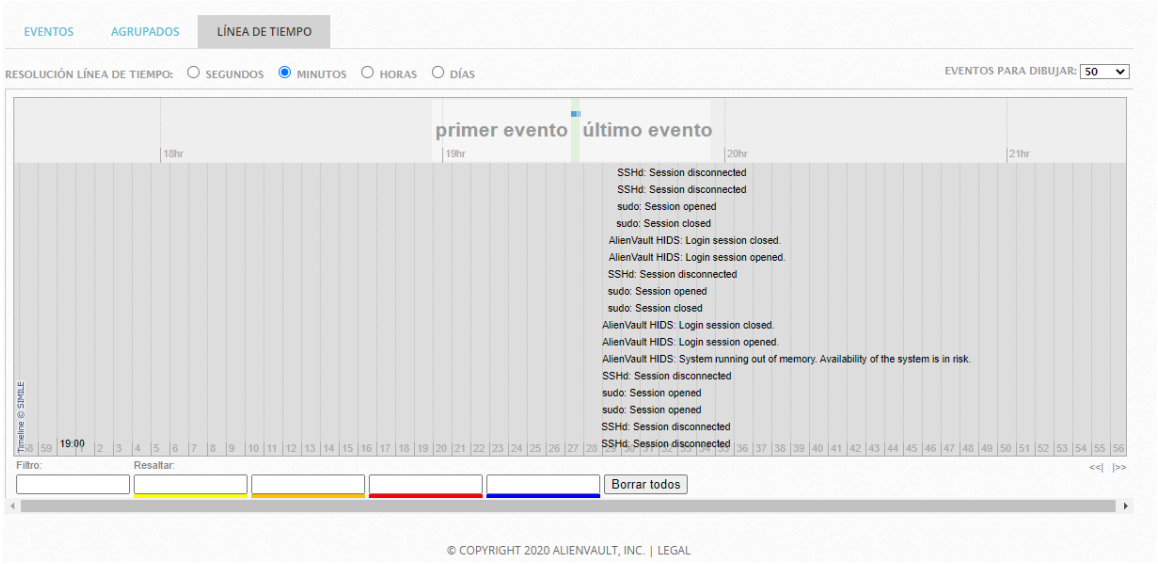


Figura 88: Gestión de riesgos: Línea de tiempo.
Fuente: OSSIM AlienVault

4.7 Definición, desarrollo e implementación de scripts en el Módulo de Disponibilidad

Uno de los objetivos principales de este proyecto es la implementación de scripts personalizados para el monitoreo, pero como un paso previo se definió tipos scripts según la obtención de datos, ya que con esta definición se podrá homologar procedimientos para la obtención de los datos ya que este aspecto se le puede considerar el más complejo y genera malestar en el desarrollo.

4.7.1 Tipos de script según su funcionamiento

Un script de programación es un conjunto de intrusiones u órdenes con un objetivo específico que generalmente es ejecutado por otras entidades ya sea de software o hardware.

Un script es un texto plano que está definido por el lenguaje de programación, por ende, tiene extensión un ambiente donde se pueda ejecutar una entrada y una salida, ya que esta estructura es necesaria para que el sistema Nagios funcione correctamente.

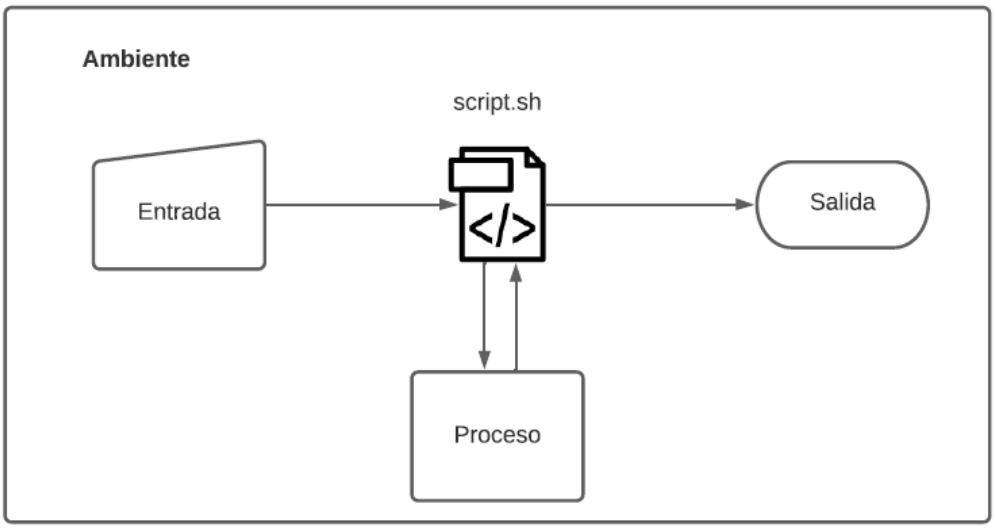


Figura 89: Funcionamiento de un script
Elaborado por: Jorge Añazco

Como muestra la imagen anterior un script necesita una entrada de datos, para realizar el análisis o procesamiento de datos y mostrar una salida y por ultimo necesita ejecutarse en un ambiente también conocido como framework que define el tipo de estructura y lenguaje de programación la cual debe estar construido el script.

4.7.2 Scripts de monitoreo por servicio

Este tipo de script tiene como principal funcionalidad hacer peticiones nativas de los diferentes servicios que existen, por ejemplo, el ping del protocolo ICMP para verificar si un servicio o equipo está activo en una red, peticiones DNS para detectar si el servicio de DNS se encuentra activo o hacer una petición Telnet para ver si un puerto se encuentra activo.

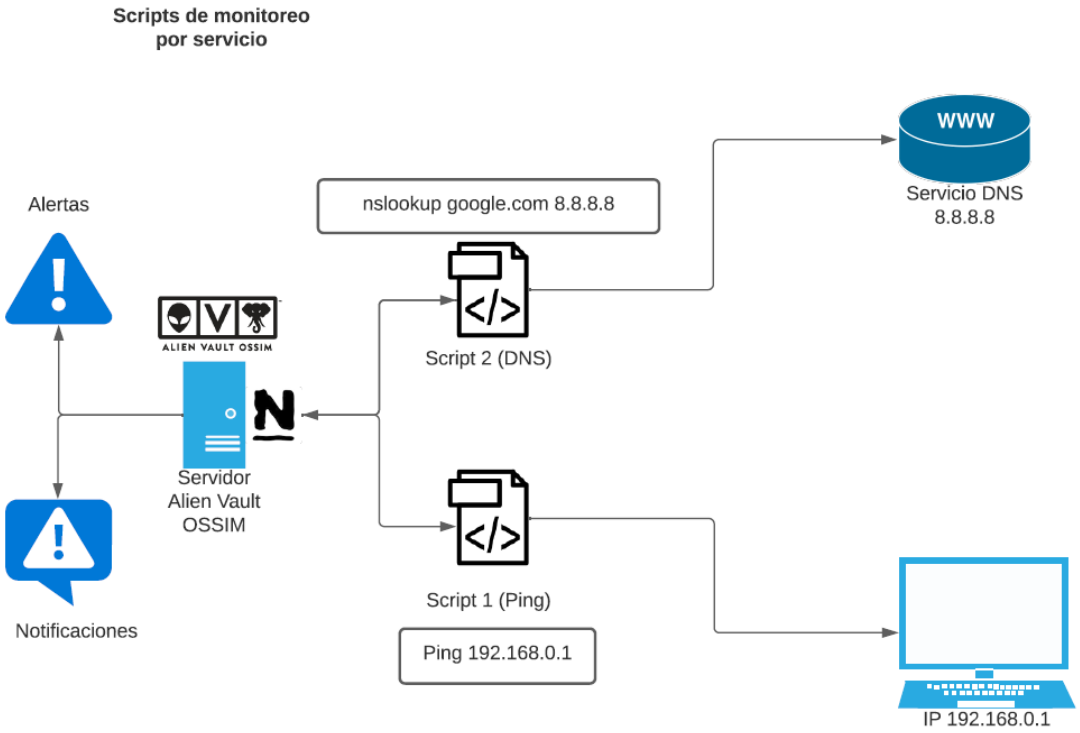


Figura 90: Scripts de monitoreo por servicio
Elaborado por: Jorge Añazco

Básicamente para la obtención de datos para este tipo de script es la ejecución de comandos para ver los estados de los protocolos, por ejemplo, se ejecuta el comando nslookup definiendo un dominio como también la dirección del servidor DNS según la respuesta o salida del comando se puede validar si el servidor se encuentra activo.

4.7.3 Scripts de monitoreo en servidor (obtención de estados)

Este tipo de script ejecuta comandos nativos del sistema operativo o de un servicio de manera local, para obtener información según las necesidades del monitoreo, de acuerdo a los siguientes parámetros:

- Tamaños de logs, carpetas o archivos.
- Modificación y estados de servicios y de archivos.
- Obtener información de consumo y estado de los procesos que se ejecuta en el sistema operativo.

- Verificación de números sesiones activas en el servidor.

- Verificación de usuarios autenticados.

Nagios tiene dos clientes para entornos Linux y Windows, lo que realizan estos clientes es ejecutar los scripts de manera local y la salida o el resultado del script es enviado al Nagios para su interpretación, el cliente para entornos Windows se llama NSClient++ y para entornos Linux el cliente se llama NRPE, básicamente con este cliente es un servicio que ejecuta scripts remotos de manera local.

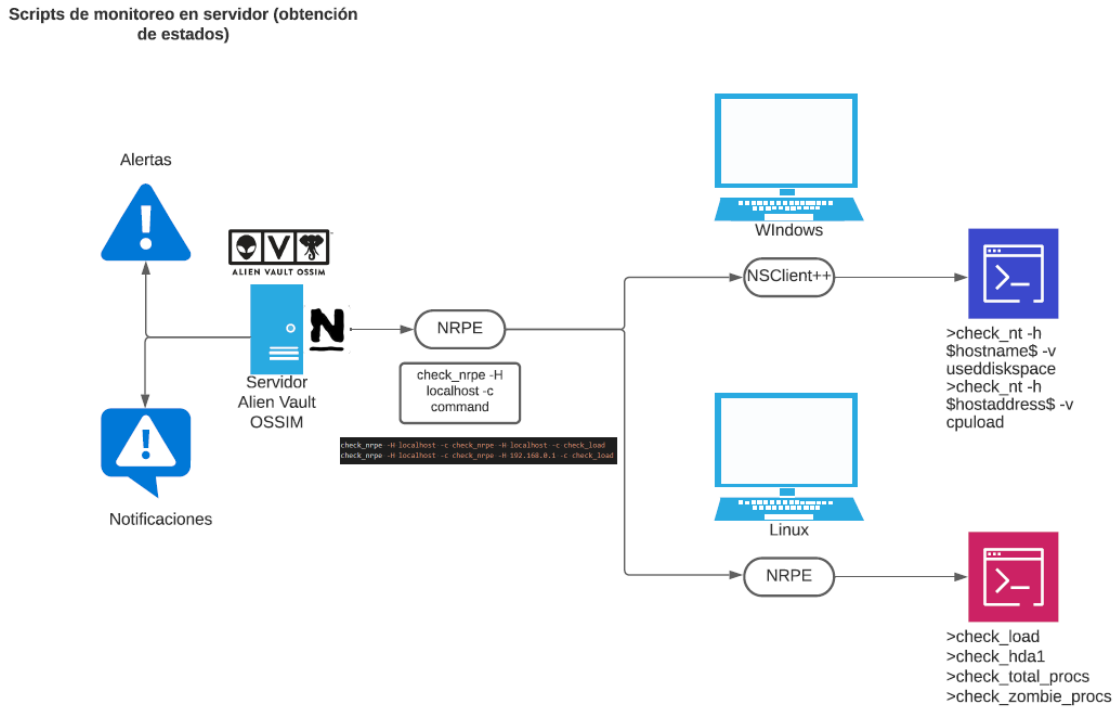


Figura 91: Scripts de monitoreo en servidor (obtención de estados)
Elaborado por: Jorge Añazco

4.7.4 Scripts de monitoreo por análisis de datos

Este tipo de script como principal objetivo es la obtención de información de un banco de datos, ya sea de Bases de datos (SQL, Mysql, Postgresql, MongoDB etc.), archivos tipo banco de datos (CSV, XLSX, XLS, XML) o información de los registros de los sistemas o servicios, luego de tener la información lo procesa y manda un resultado.

Hay que tomar en cuenta que este tipo de script son los que generalmente consume más recursos del sistema, ya que, según el volumen de los datos obtenidos de procesamiento puede consumir recursos afectando al funcionamiento de la estabilidad del sistema.

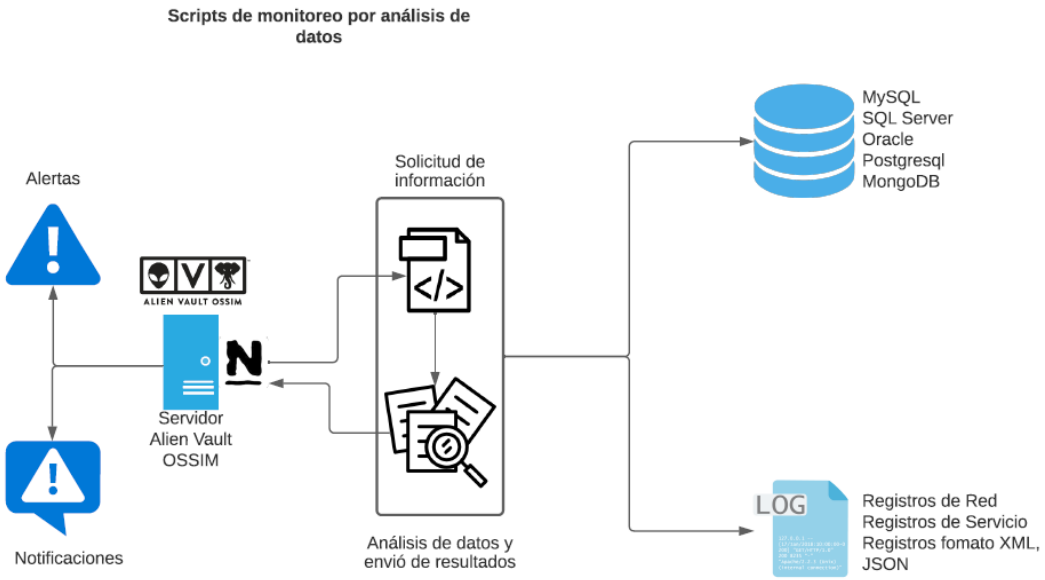


Figura 92: Scripts de monitoreo por análisis de datos
Elaborado por: Jorge Añazco

4.7.5 Scripts de monitoreo por interacción

Este tipo de scripts como una de sus principales funcionalidades es interactuar y enviar sentencias o comandos en los diferentes CLI (Command Line Interface) que existe por marca o dispositivo, para esta funcionalidad se utilizó el paquete/módulo Expect, el cual se encuentra en los ambientes Linux (Debian, Red Hat) y se puede implementar en lenguaje SHELL, este paquete ayuda a realizar desde la conexión y mandar las sentencias o comandos para obtener la información necesaria para el monitoreo.

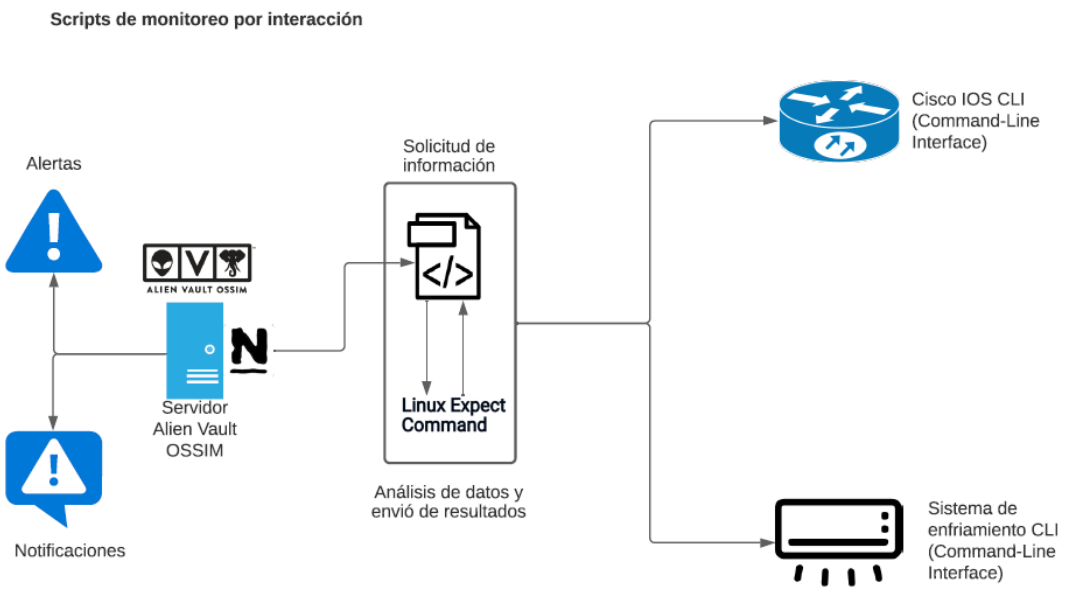


Figura 93: Scripts de monitoreo por interacción
Elaborado por: Jorge Añazco

La herramienta que se utiliza para realizar las sesiones es el paquete Expect que es necesario instalar en el servidor OSSIM AlienVault con el siguiente comando:

- apt-get install expect

Por último un ejemplo de cómo se realiza una sesión con esta herramienta, básicamente en este comando hay que definir un proceso de inicio de sesión con el usuario y contraseña como la IP del dispositivo y el proceso de integración para la obtención de los datos, a continuación, se detalla los comando que tiene la herramienta.

- spawn: Iniciar un script o un programa.
- expect: Esperar la salida de un programa.
- end: Enviar una respuesta a tu programa.
- Interact: Permitirte interactuar con tu programa.

En el siguiente grafico de detalla un ejemplo de inicio de sesión y obtención de datos de un sistema de enfriamiento en un *Data Center* para la obtención de la temperatura promedio:

```
HOST="172.16.254.160"
USER="DCPrueba"
PASS="Adminprueba"
PASS_ROOT="temporalPass"
LOOP=2
    VAR=$(expect -c "
        match_max 100000
        spawn ssh -o StrictHostKeyChecking=no $USER@$HOST
        expect \"Authenticated with partial success.\"
        expect \"*?assword:*\"
        send -- \"$PASS\r\"
        expect  \">\"
        send -- \"1\r\"
        expect \"* Device Manager *\"
        expect \">\"
        send -- \"1\r\"
        expect \"* Environment  *\"
        expect \">\"
        send -- \"\033\r\"
        expect \"* Device Manager *\"
        expect \">\"
        send -- \"\033\r\"
        expect \"* Control Console *\"
        expect \">\"
        send -- \"4\r\"
        match_max 100000
        expect eof
    ")
```

Figura 94: Ejemplo del comando expect.
Elaborado por: Jorge Añazco

4.7.6 Scripts de monitoreo por consumo de microservicios

Este tipo de scripts como uno de sus principales objetivos es consumir un API o microservicios y una de las ventajas es que la manera en que se consulta los datos y la respuesta de los mismos es de manera uniforme a través de una petición HTTP/GET y requiere menos recursos del servidor ya que son autónomos a tal grado que el procesamiento de los datos se puede delegar al API o microservicios y solo consultar el dato específico para el análisis del monitoreo.

Hay que tomar en cuenta que la respuesta es en formato JSON y hay que procesar este tipo de datos, para este proyecto se utilizó programación en lenguaje Python para enviar y procesar los datos.

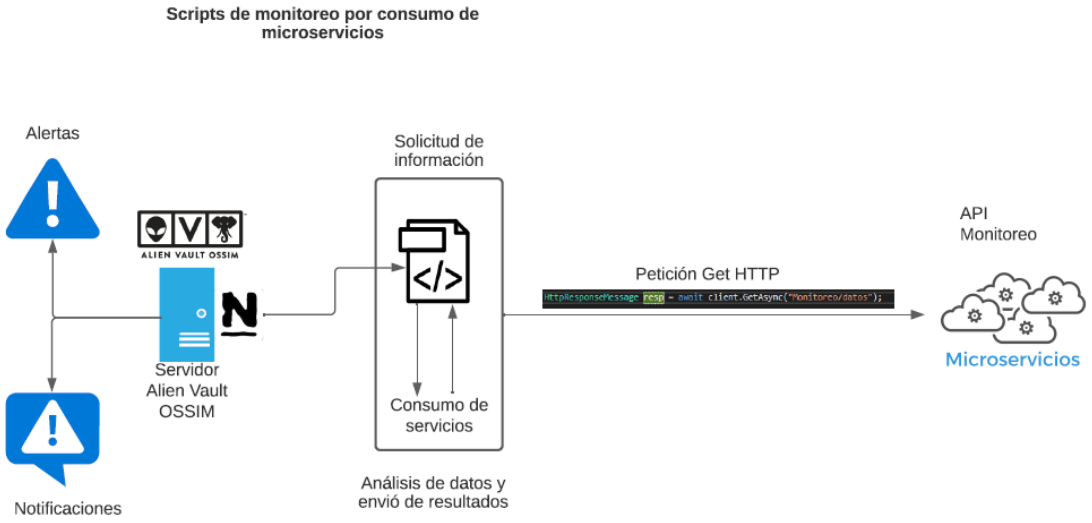


Figura 95: Scripts de monitoreo por consumo de microservicios
Elaborado por: Jorge Añazco

4.8 Creación de scripts para el módulo de Nagios

Para el desarrollo de scripts para el módulo de Nagios se debe tomar en cuenta el tipo de lenguaje en el que se van a desarrollar, en este caso se utiliza SHELL con otras herramientas y lenguajes que ofrece el sistema operativo del OSSIM AlienVaulr como, por ejemplo, el lenguaje Ruby para tratamiento de datos y el módulo Expect que sirve para automatizar sesiones interactivas.

4.8.1 Características de un script para funcionar con Nagios

Los scripts de Nagios cuentan con la siguiente estructura:

Creación del script: El script se puede crear con el comando touch con extensión .sh además de tener permisos de ejecución.

```
alienvault:/nagios/scripts/Script_Prueba# touch nagios_scripts_HolaMundo_v2.sh
alienvault:/nagios/scripts/Script_Prueba# chmod +x nagios_scripts_HolaMundo_v2.sh
```

Figura 96: Características de un script: Creación
Elaborado por: Jorge Añazco

Ruta del script: Ubicación del script dentro del servidor, para este ejemplo:

```
1  /nagios/scripts/Script_Prueba/nagios_scripts_HolaMundo_v1.sh
```

Figura 97: Características de un script: Ubicación.
Elaborado por: Jorge Añazco

Cuerpo del script: es el contenido del documento dónde se encuentra los algoritmos y se define lo que hace el script:

```
#!/bin/bash
#Path: /nagios/scripts/Script_Prueba/nagios_scripts_HolaMundo_v1.sh
#Version 1.0 TesisPrueba
#INICIO SCRIPT

#Variables
ARCHIVO_PRUEBA="nagios/scripts/Script_Prueba/nagios_scripts_HolaMundo_v1.sh"

#Validar el archivo prueba existe
if [ -f $ARCHIVO_IP ]
    then
        echo "OK - El archivo existe"
        exit 0
    else
        echo "CRITICAL - El archivo no existe"
        exit 2
fi

echo "UNKNOWN - Estado desconocido"
exit 3
```

Figura 98: Características de un script: Cuerpo/código.
Elaborado por: Jorge Añazco

Dónde:

Los mensajes que serán mostrados en el Nagios serán las líneas que imprime el comando echo

Host	Status	Last Check	Duration	Status Information
Host-192-168-0-161	UP	2021-03-28 19:31:18	269d 18h 34m 39s	PING OK - Packet loss = 0%, RTA = 0.98 ms
localhost	UP	2021-03-28 19:33:38	270d 21h 56m 24s	PING OK - Packet loss = 0%, RTA = 0.13 ms

Figura 99: Características de un script: Mensajes.
Elaborado por: Jorge Añazco

Y el código *exit*, define el estado de la notificación para generar alarmas según el código

Tabla 6: Características de un script: Estados Nagios.

Código	Estado	Color
exit 0	OK	
exit 1	WARNING	
exit 2	CRITICAL	
exit 3	UNKNOWN	

Elaborado por: Jorge Añazco

Depuración/Debug

Como ayuda al momento de crear y visualizar paso a paso la ejecución del script (depurar) se puede brindar seguimiento con los siguientes comandos:

$bash –x nombredelscript.sh

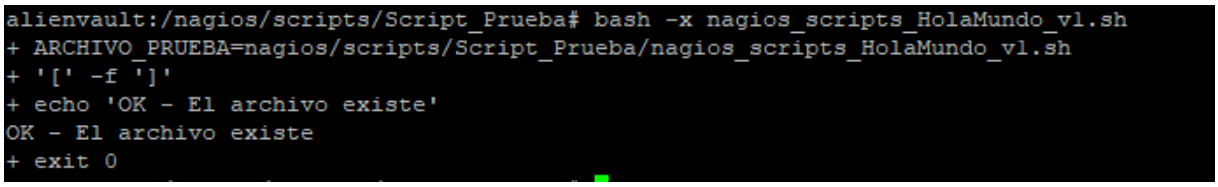


Figura 100: Depuración/Debug: Comando bash.
Elaborado por: Jorge Añazco

Esto permite dar seguimiento y control de línea por línea de lo que ejecuta el script y facilita la detección de errores de programación.

Por último, existe una manera de mostrar el valor de la ejecución que son los valores de los exit:

$echo $?

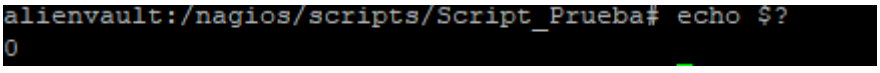


Figura 101: Depuración/Debug: Comando echo $?.
Elaborado por: Jorge Añazco

Este comando captura el valor del último script ejecutado.

4.8.2 Agregar un nuevo comando-script a Nagios para monitoreo

Una vez creado el script, el siguiente paso es la configuración del Nagios para acceder al mismo:

Agregar un nuevo archivo de configuración donde se define y configura el nuevo comando, en este caso se crea una nueva carpeta en la ruta /etc/nagios3 llamada comandosnuevos la cual contendrá el archivo comandos_nuevos.cfg.

$cd /etc/nagios3 && mkdir comandosnuevos && touch comandos_nuevos.cfg

```
alienvault:/etc/nagios3# pwd
/etc/nagios3
alienvault:/etc/nagios3# cd comandosnuevos/
alienvault:/etc/nagios3/comandosnuevos# ll
total 4
-rw-rw-rw- 1 nagios nagios 196 Mar 28 22:27 comandos_nuevos.cfg
```

Figura 102: Agregar un nuevo comando Nagios: Creación nuevo archivo de configuración.
Elaborado por: Jorge Añazco

Se ingresa al archivo comandos_nuevos.cfg e ingresar la configuración del nuevo comando para el Nagios:

```
alienvault:/etc/nagios3/comandosnuevos# cat comandos_nuevos.cfg
# 'comando de prueba' command definition
define command{
        command_name    check_script_prueba
        command_line    /nagios/scripts/Script_Prueba/nagios_scripts_HolaMundo_v1.sh
        }
```

Figura 103: Agregar un nuevo comando Nagios: Estructura de creación de un nuevo comando.
Elaborado por: Jorge Añazco

Donde:

Tabla 7: Agregar un nuevo comando Nagios: Descripción de comandos.

Comando	Descripción
command_name	Nombre del comando
command_line	Ruta del script

Elaborado por: Jorge Añazco

Luego se ingresa en la configuración de Nagios (/etc/nagios3/nagios.cfg), e indicar la ruta del nuevo archivo de configuración:

```
#Definitions for monitoring with custom scripts
cfg_dir=/etc/nagios3/comandosnuevos
```

Figura 104: Agregar un nuevo comando Nagios: Agregar nuevo archivo/directorio de configuración.
Elaborado por: Jorge Añazco

En este caso se utilizó cfg_dir, para indicar que ingrese en la configuración del Nagios todos los archivos del directorio /etc/nagios3/comandosnuevos, pero también se puede indicar un solo archivo para la configuración con el siguiente comando: cfg_file=/etc/nagios3/comandosnuevos/comandos_nuevos.cfg.

Luego de configurar el nuevo comando en el Nagios, hay que ingresarlo a la plantilla del host para el monitoreo, en este caso se utiliza el localhost como ejemplo. Se ingresa al archivo /etc/nagios3/conf.d /localhost_nagios2.cfg y se crea un nuevo servicio de monitoreo:

```
#Nuevo servicio de prueba

define service{
        use                             generic-service         ; Name of service template to use
        host_name                       localhost
        service_description             Prueba-Script
        check_command                   check_script_prueba
        }
```

Figura 105: Agregar un nuevo comando Nagios: Agregar comandos al host.
Elaborado por: Jorge Añazco

Donde:

Tabla 8: Agregar un nuevo comando Nagios: Comandos para agregar servicios al host.

Comando	Descripción
use	Se indica la plantilla que se va utilizar por defecto
host_name	El nombre de host a cuál se le va ingresar el servicio
service_description	Descripción del servicio
check_command	Nombre del comando que se va utilizar

Elaborado por: Jorge Añazco

Para finalizar, compilamos los archivos de configuración de Nagios y visualizar si existe o no algún error con el siguiente comando:

$nagios3 -v /etc/nagios3/nagios.cfg

Y con esto se valida si existe algún error

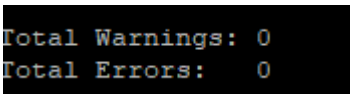


Figura 106: Agregar un nuevo comando Nagios: Errores de compilación Nagios.
Elaborado por: Jorge Añazco

Por último, reiniciamos el servicio de Nagios:

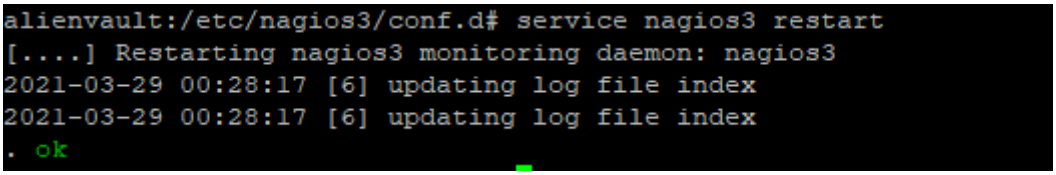
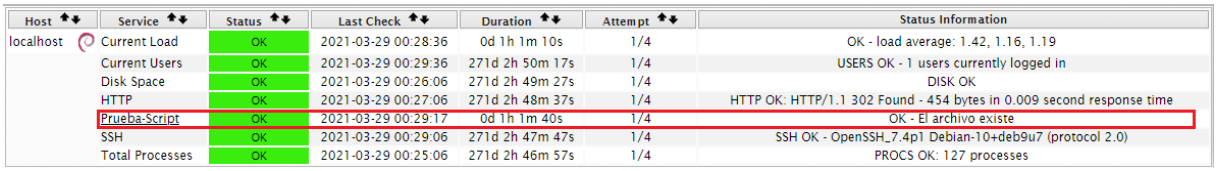


Figura 107: Agregar un nuevo comando Nagios: Reinicio del servicio de Nagios.
Elaborado por: Jorge Añazco

De esta manera se refleja el nuevo servicio en el panel de disponibilidad del AlienVault:



Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	2021-03-29 00:28:36	0d 1h 1m 10s	1/4	OK - load average: 1.42, 1.16, 1.19
	Current Users	OK	2021-03-29 00:29:36	271d 2h 50m 17s	1/4	USERS OK - 1 users currently logged in
	Disk Space	OK	2021-03-29 00:26:06	271d 2h 49m 27s	1/4	DISK OK
	HTTP	OK	2021-03-29 00:27:06	271d 2h 48m 37s	1/4	HTTP OK: HTTP/1.1 302 Found - 454 bytes in 0.009 second response time
	Prueba-Script	OK	2021-03-29 00:29:17	0d 1h 1m 40s	1/4	OK - El archivo existe
	SSH	OK	2021-03-29 00:29:06	271d 2h 47m 47s	1/4	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u7 (protocol 2.0)
	Total Processes	OK	2021-03-29 00:25:06	271d 2h 46m 57s	1/4	PROCS OK: 127 processes

Figura 108: Agregar un nuevo comando Nagios: Nuevo servicio.
Elaborado por: Jorge Añazco

4.9 Notificaciones y alertas a través de Telegram

Para las notificaciones y alertas a través de Telegram se necesita una herramienta llamada telegram-cli que es un software para sistemas Linux ya sea basados en Red Hat (Centos, Fedora, etc.) o Debian (Ubuntu, Kubuntu, Raspbian, etc.) que permite utilizar un terminal para el envío de mensajes a través de la plataforma Telegram.

La ventaja principal de usar telegram-cli en el terminal es que se puede unir con otras aplicaciones con scripts y automatizar envió de notificaciones directamente a los usuario y grupos como a cualquier tipo de dispositivos ya sean de escritorio o móviles.

4.9.1 Instalación telegram-cli en CentOS 8

Para la instalación se necesita el paquete telegram-cli que se puede conseguir en este enlace: https://github.com/vysheng/tg o se puede ejecutar el siguiente comando en el terminal git clone --recursive https://github.com/vysheng/tg.git

```
[root@localhost telegram_git]# git clone --recursive https://github.com/vysheng/tg.git
Clonando en 'tg'...
remote: Enumerating objects: 4511, done.
remote: Total 4511 (delta 0), reused 0 (delta 0), pack-reused 4511
Recibiendo objetos: 100% (4511/4511), 2.99 MiB | 1.91 MiB/s, listo.
Resolviendo deltas: 100% (3041/3041), listo.
Submódulo 'tgl' (https://github.com/vysheng/tgl.git) registrado para ruta 'tgl'
Clonando en '/home/janazco/telegram_git/tg/tgl'...
remote: Enumerating objects: 1555, done.
remote: Total 1555 (delta 0), reused 0 (delta 0), pack-reused 1555
Recibiendo objetos: 100% (1555/1555), 1.03 MiB | 2.00 MiB/s, listo.
Resolviendo deltas: 100% (1137/1137), listo.
Ruta de submódulo 'tgl': check out realizado a 'ffb04caca71de0cddf28cd33a4575922900a59ed'
Submódulo 'tl-parser' (https://github.com/vysheng/tl-parser) registrado para ruta 'tgl/tl-parser'
Clonando en '/home/janazco/telegram_git/tg/tgl/tl-parser'...
remote: Enumerating objects: 87, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 87 (delta 1), reused 1 (delta 1), pack-reused 83
Ruta de submódulo 'tgl/tl-parser': check out realizado a '36bf1902ff3476c75d0b1f42b34a91e944123b3c'
```

Figura 109: Descargar telegram-cli.
Elaborado por: Jorge Añazco

Antes de continuar con la instalación se debe instalar los siguientes paquetes

- dnf install python3
- dnf install gcc gcc-c++
- dnf install libevent-devel
- dnf install openssl-devel
- dnf install readline-devel
- dnf install jansson-devel
- dnf install libgcrypt-devel
- dnf install jansson-devel

- dnf install lua-devel
- dnf install readline-devel
- dnf install libconfig-devel
- dnf install libevent-devel

Si la instalación es en Centos 8 el paquete lua-devel no se encuentra en el repositorio por defecto y se realizó la instalación manual descargando el paquete del siguiente enlace: https://centos.pkgs.org/8/centos-powertools-x86_64/lua-devel-5.3.4-11.el8.x86_64.rpm.htm

Y para instalar con el comando:

```
$ rpm –ivh lua-devel-5.3.4-11.el8.x86_64.rpm
```

Por último y para servidores con el sistema operativo Centos 8 en el proyecto de telegram-cli hay que modificar el archivo /tg/tgl/mtproto-utils.c t comentar las líneas 101 y 115 para que al momento de instalar no se presente un problema de configuración por el sistema operativo.

Figura 110: Prerrequisito para la instalación de telegram-cli en Centos 8
Elaborado por: Jorge Añazco

Y para instalar se ejecuta los siguientes comandos en la ubicacion /tg

```
$ ./configure --disable-openssl CFLAGS="-w"
$ make
```

Luego de la instalación y para ejecutar directamente el telegram-cli como comando del sistema se realiza una copia del ejecutable en una carpeta para luego crear la referencia dentro /usr/bin.

```
$ mkdir /opt/telegram-cli
  && cp bin/telegram-cli tg-server.pub /opt/telegram-cli/
```

&& ln -s /opt/telegram-cli/telegram-cli /usr/bin/

Para configurar el número celular que va a utilizar telegram-cli se introduce el comando:

$ telegram-cli -k tg.pub

Luego se ingresa número celular para enviar un código de autenticación y este código lo recibirá en la aplicación de Telegram del dispositivo, se ingresa y presiona enter y ya está configurado telegram-cli.

```
$ telegram-cli -k tg.pub
Telegram-cli version 1.4.1, Copyright (C) 2013-2015 Vitaly Valtman
Telegram-cli comes with ABSOLUTELY NO WARRANTY; for details type `show_license'
This is free software, and you are welcome to redistribute it
under certain conditions; type `show_license' for details.
Telegram-cli uses libtgl version 2.1.0
Telegram-cli includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit. (http://www.openssl.org/)
I: config dir=[/home/pi/.telegram-cli]
phone number: +59396666666
code ('CALL' for phone code): 58964
```

Figura 111: Instalación telegram-cli
Elaborado por: Jorge Añazco

Para ingresar a la consola del telegram-cli es con el comando telegram-cli -W -k server.pub

```
[root@localhost telegram_git]# telegram-cli -W -k server.pub
change_user_group: can't find the user telegramd to switch to
Telegram-cli version 1.4.1, Copyright (C) 2013-2015 Vitaly Valtman
Telegram-cli comes with ABSOLUTELY NO WARRANTY; for details type `show_license'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show_license' for details.
Telegram-cli uses libtgl version 2.1.0
I: config dir=[/root/.telegram-cli]
12 [16:49]  Nagios Alert <<< * Nagios * Notification Type: PROBLEM   Service: Current Load  Host: localhost  Address: 127.0.0.1
  State: CRITICAL     Date/Time: Fri Jun 11 20:58:17 -05 2021  Additional Info:   CRITICAL - load average: 5.89, 3.82, 4.11
```

4.9.2 Script de comunicación para envió de alerta y notificaciones por telegram-cli

Para este proyecto se realizó un script para la comunicación entre la herramienta OSSIM AlienVault y telegram-cli que se encuentra en servidores diferentes para este se utilizó la herramienta Expect para su funcionamiento.

Primero se define el comando para enviar mensajes sin interactuar con el cliente del telegram-cli que es el siguiente:

```
(sleep 3;echo "msg $destination \"$message\""; echo "safe_quit") | telegram-cli -k tg-server.pub -W
```

Figura 112: Comando local para el envío un mensaje en telegram-cli
Elaborado por: Jorge Añazco

Donde:

$destination: Es el contacto a enviar el mensaje.

$message: Es el contenido del mensaje

Ya definido el comando se puede automatizar un script de conexión y ejecución con la ayuda de la herramienta Expect.

```
HOST="192.168.0.148"
USER="root"
PASS="Adminlocal2k11"
VAR=$(expect -c "
        spawn ssh -o StrictHostKeyChecking=no $USER@$HOST
        match_max 100000
        expect \"*?assword:*\"
        send -- \"$PASS\r\"
        expect \"*$\"
        send -- \"cd /home/janazco/script_telegram\r\"
        expect \"*$\"
        send -- {./SendMessageTelegram.sh -c $destination -m \"$message\"}
        send -- \"\r\"
        expect eof
")
```

Figura 113: Comando remoto para el envío un mensaje en telegram-cli
Elaborado por: Jorge Añazco

4.9.3 Script Telegram

- **Objetivo:** Script para enviar noticiones o alertas por el aplicativo Telegram.
- **Entrada:**

  Entrada 1: Ejecución local para él envió de mensaje por Telegram

  SendMessageTelegram.sh -c $destination -m $message

  Entrada 2: Ejecución remota para él envió de mensaje por Telegram

  SendMessageTelegram.sh –r -c $destination -m $message

- **Salida del script:** El envío de mensaje.

- **Código**: Ver Anexo 1.

- **Flujo**:

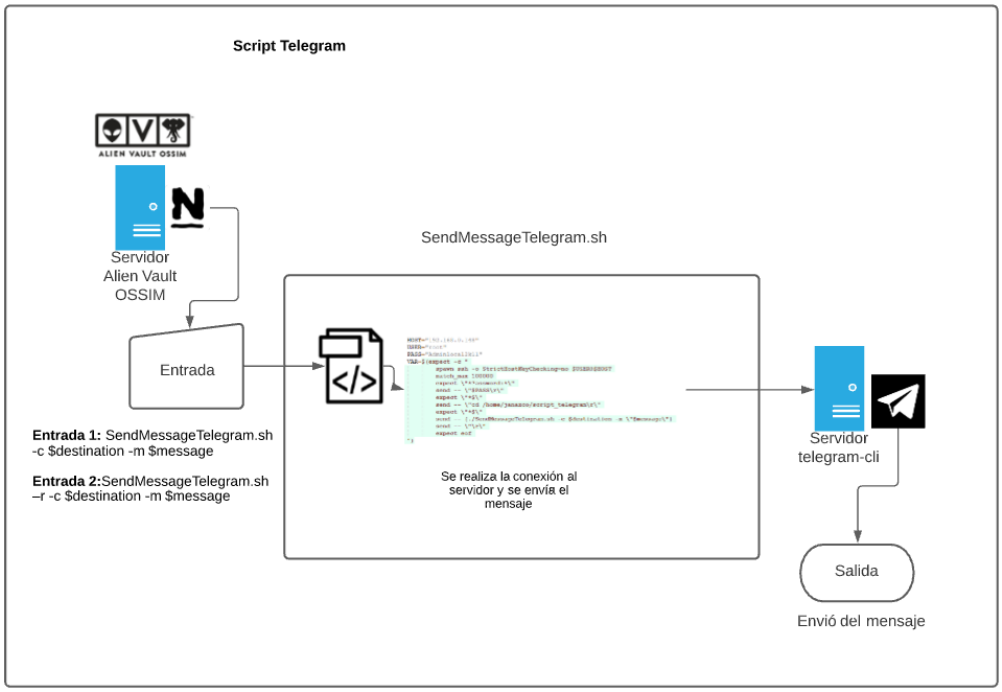


Figura 114: Flujo script SendMessageTelegram.sh
Elaborado por: Jorge Añazco

4.9.4 Configuración Nagios para el envío de notificaciones o alertas por Telegram

Primero en el archivo commands.cfg se define la llamada del script SendMessageTelegram y se parametriza los argumentos de la entrada del script con las variables locales del Nagios, esta configuración es para las notificaciones de los servicios como de los dispositivos.

```
# 'notify-host-by-telegram' command definition
define command{
        command_name    notify-host-by-telegram
        command_line    /home/telegram/ScriptComunicacionTelegram.sh -r -c $_CONTACTTELEGRAM$ -m
        "***** Nagios ***** Notification Type: $NOTIFICATIONTYPE$ Host: $HOSTNAME$ State: $HOSTSTATE$
        Address: $HOSTADDRESS$ Info: $HOSTOUTPUT$ Date/Time: $LONGDATETIME$"
        }

# 'notify-service-by-telegram' command definition
define command{
        command_name    notify-service-by-telegram
        command_line    /home/telegram/ScriptComunicacionTelegram.sh -r -c $_CONTACTTELEGRAM$ -m
        "***** Nagios ***** Notification Type: $NOTIFICATIONTYPE$   Service: $SERVICEDESC$  Host: $HOSTALIAS$
        Address: $HOSTADDRESS$  State: $SERVICESTATE$     Date/Time: $LONGDATETIME$  Additional Info:   $SERVICEOUTPUT$"
        }
```

Figura 115: Configuración del comando para el envío de mensajes por Telegram
Elaborado por: Jorge Añazco

Se crea una nueva plantilla de notificación por contactos donde definimos la llamada de los nuevos comandos para notificar por Telegram

```
define contact{
        name                            operaciones-contact-telegram        ; The name of this contact template
        service_notification_period     24x7                    ; service notifications can be sent anytime
        host_notification_period        24x7                    ; host notifications can be sent anytime
        service_notification_options    w,u,c,r,f,s             ; send notifications for all service states, flapping events, and scheduled downtime events
        host_notification_options       d,u,r,f,s               ; send notifications for all host states, flapping events, and scheduled downtime events
        service_notification_commands   notify-service-by-telegram ; send service notifications via email
        host_notification_commands      notify-host-by-telegram     ; send host notifications via email
        register                        0                       ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL CONTACT, JUST A TEMPLATE!
        }
```

Figura 116: Plantilla para el envío de mensajes por Telegram por contacto.
Elaborado por: Jorge Añazco

Se configura el usuario y se inicializa una nueva variable _telegram para definir el nombre de usuario del contacto de Telegram.

```
define contact{
        contact_name                    janazco-telegram                    ; Short name of user
        use                             operaciones-contact-telegram          ; Inherit default values from generic-contact template (defined above)
        alias                           Jorge Anazco cell "Desarrollo"            ; Full name of user
        _telegram                       Jorge_Desarrollo_Telegram
        }
```

Figura 117: Configuración del contacto para el envío de mensajes por Telegram
Elaborado por: Jorge Añazco

Por último, se realiza el envío de la notificación de prueba

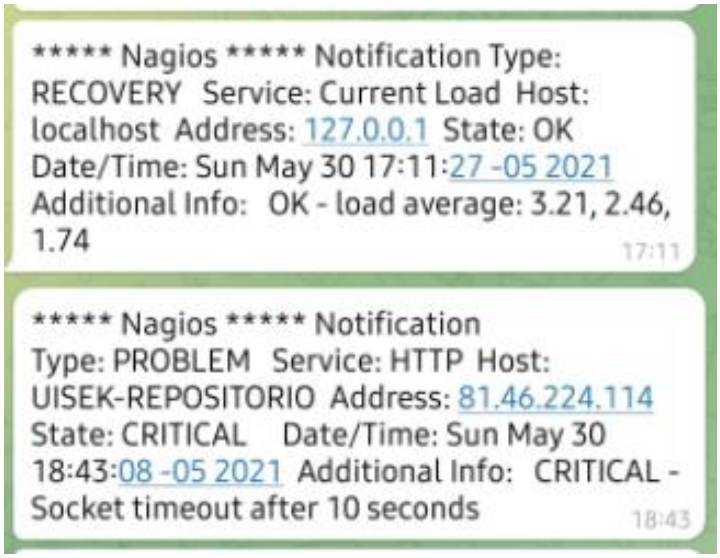


Figura 118: Ejemplo de envío de notificación por Telegram.
Elaborado por: Jorge Añazco

4.10 Scripts desarrollados

Actualmente se ha desarrollado 5 scripts para el monitoreo de Nagios, los cuales se detallan a continuación:

4.10.1 Script para la obtención de la temperatura del sistema de enfriamiento de un *Data Center*

- **Objetivo:** Obtener los valores de temperatura a través de consultas directas en el sistema de enfriamiento, según la temperatura se analiza y se alerta.
- **Tipo de script:** Scripts de monitoreo por interacción.
- **Alerta:** Se emite cuando la temperatura es superior a 28º y la humedad es menor al 25%, para evitar posibles daños a los equipos que se encuentran en el data center, si por alguna razón existe un problema con el sistema de refrigeración.
- **Salida del script:** Temperatura = "$TEMP" Humedad = "$HUMEDAD”
- **Código**: Ver Anexo 2.
- **Flujo**:

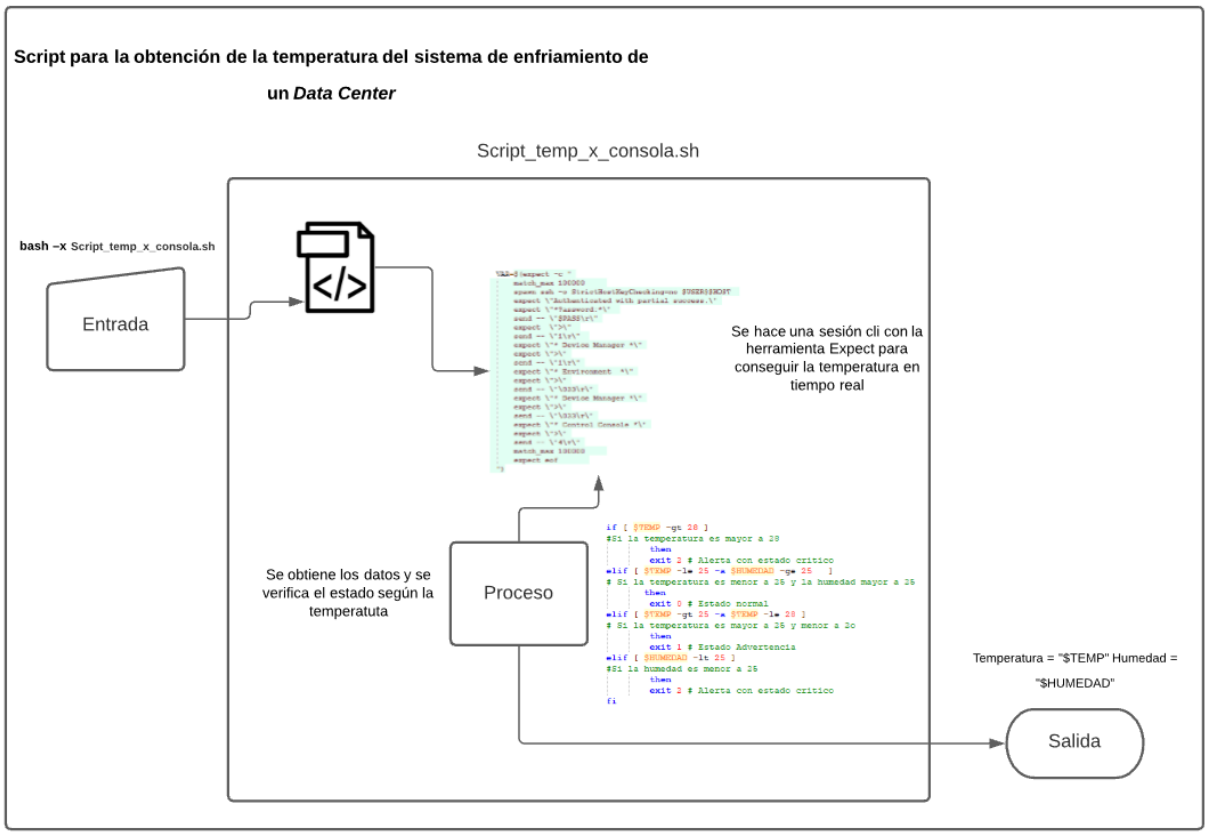


Figura 119: Flujo script para la obtención de la temperatura del sistema de enfriamiento de un Data Center
Elaborado por: Jorge Añazco

4.10.2 Script para la obtención de un archivo .csv donde se detalla número de *snapshots* de un hipervisor, para el análisis y él envió de los estados.

- **Objetivo:** Obtener un archivo .csv a través del servicio ftp donde se detalla número de *snapshots* que se tiene en las máquinas virtuales de un hipervisor, para luego analizarlo y alertar según los estados.

- **Tipo de script:** Scripts de monitoreo por análisis de datos (archivo .csv).

- **Alerta:** Cuando existe algún *snapshots* en el servidor, ya que para mantenerlo activo consume recursos y puede generar problemas con la máquina virtual.

- **Salida del script:** EXISTEN SNAPSHOTS EN "$NOMBRE_SERVIDOR” -- Numero de Snapshots = "$NUMERO_SNAPSHOTS”

- **Código**: Ver Anexo 3.

- **Flujo**:

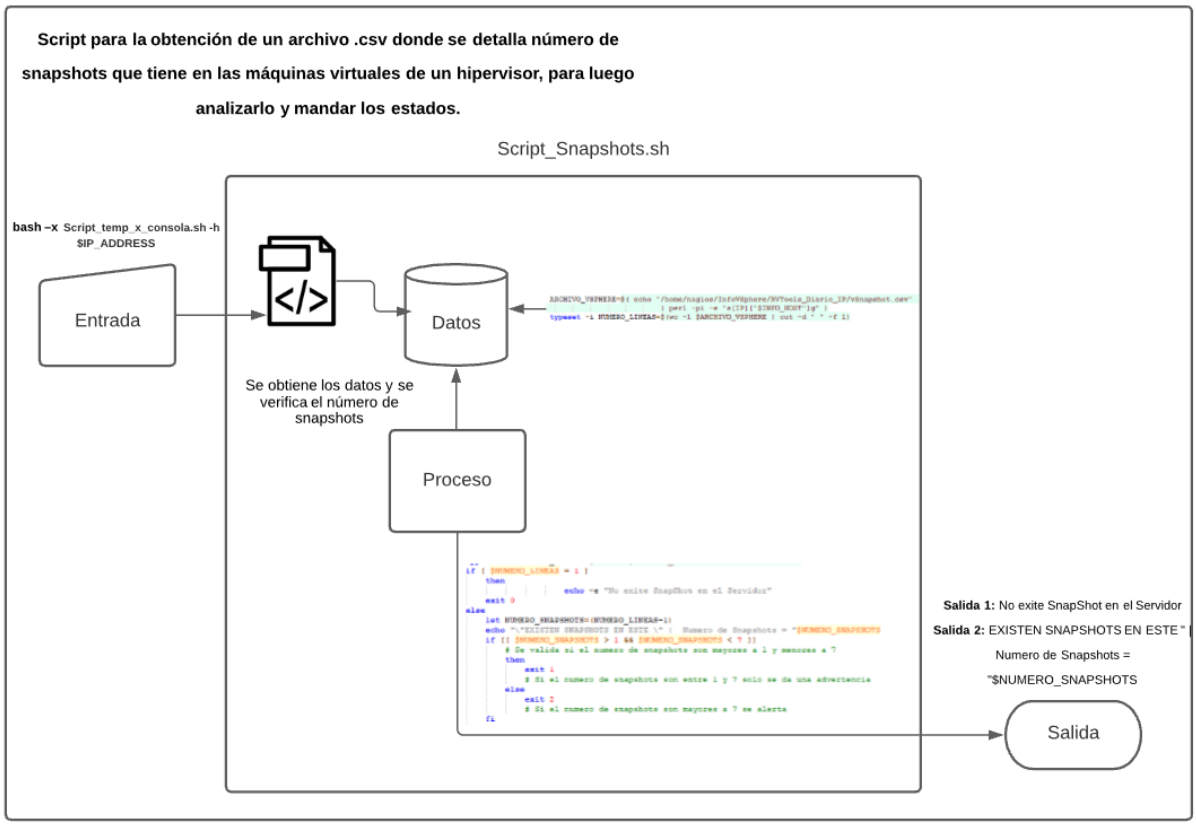


Figura 120: Flujo script para la obtención de un archivo .csv donde se detalla número de snapshots que tiene en las máquinas virtuales de un hipervisor, para luego analizarlo y mandar los estados.
Elaborado por: Jorge Añazco

4.10.3 Script que analiza el log de acceso del servicio de apache

- **Objetivo:** Analizar el log de acceso del servicio de apache.

- **Tipo de script:** Scripts de monitoreo por análisis de datos (archivo .log).

- **Alerta:** Se calcula la media del número de peticiones http que ingresan por día y por hora, con este valor se puede determinar si hay un ataque en proceso a la página web del log monitoreado.

- **Salida del script:**

  Salida 1

  Número de Peticiones son: "$PETICIONES" MAX-NORMAL = "$MAX" MAX-CRITICAL = "$MAX_CRI"

  Salida 2

  Tamaño del LOG es: "$SIZE_MATH$TYPE" MAX-NORMAL = "$i"M MAX-CRITICAL = "$j"M Revisar los LOGS anteriores existe un log que pesa más de 1 GB

- **Código**: Ver Anexo 4.

- **Flujo**:

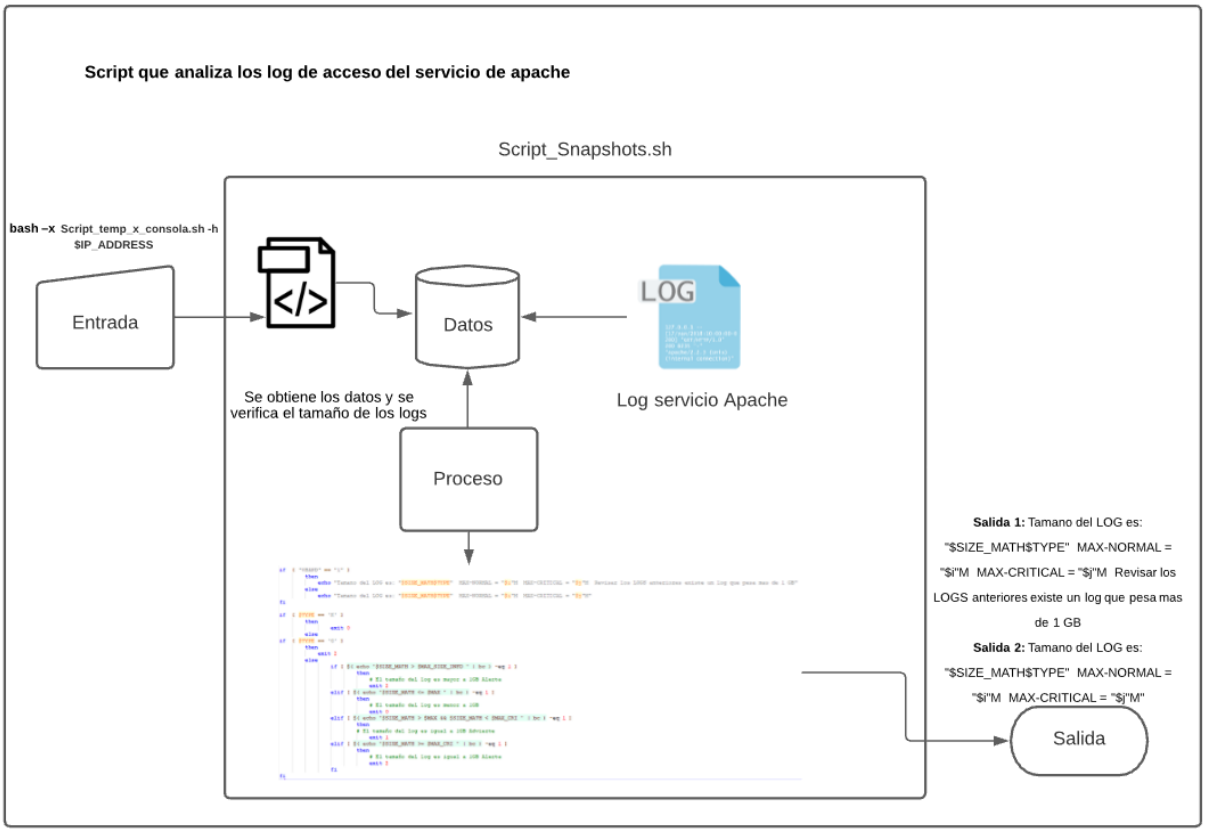


Figura 121: Flujo script que analiza e l log de acceso del servicio de apache
Elaborado por: Jorge Añazco

4.10.4 Script que analiza los tamaños de los directorios y logs de la base de datos Postgresql

- **Objetivo:** Obtener el tamaño de los log, espacio en disco y el tamaño del directorio de la base de datos Postgresql.

- **Tipo de script:** Scripts de monitoreo en servidor (obtención de estados).

- **Alerta:** Según el tamaño de los logs se calcula la media de tamaños de los directorios para alertar, esto ayuda a evitar caídas del servicio por falta de almacenamiento y además se podría detectar ataques a la base de datos.

- **Salida del script:**

  Salida 1

  Tamaño del LOG es: "$SIZE_MATH$TYPE" MAX-NORMAL = "$i"M MAX-CRITICAL = "$j"M

  Salida 2:

  "Tamaño del DIR LOG es: " $SIZE_MATH$TYPE"

  Salida 3:

El porcentaje es: "$SIZE_MATH$TYPE" TAMAÑO = "$TAMANO" OCUPADO = "$UTILIZADO" DISPONIBLE = "$DISPONIBLE"

- **Código**: Ver Anexo 4.

- **Flujo**:

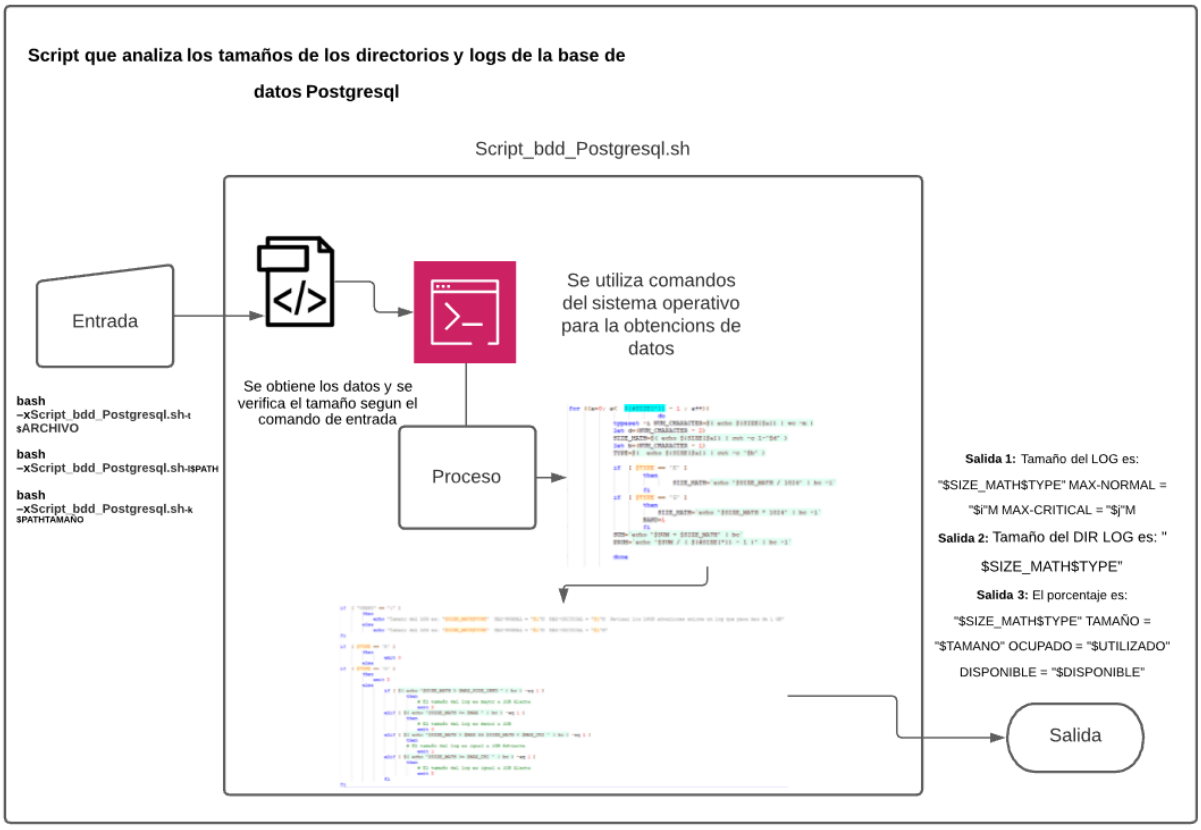


Figura 122: Script que analiza los tamaños de los directorios y logs de la base de datos Postgresql
Elaborado por: Jorge Añazco

4.10.5 Script que analiza el número de tickets de la mesa de servicio

- **Objetivo:** Obtener el número de tickets nuevos, asignados abiertos, asignados cerrados y tickets en seguimiento de la mesa de servicios.

- **Tipo de script:** Scripts de monitoreo por análisis de datos (consultas a la base de datos).

- **Alerta:** Se establece una franja de recepción de tickets abiertos con un mínimo de 20 y máximo de 25 sobre un tema particular de los establecidos en la base de datos del catálogo de servicios de la mesa de ayuda relacionados con el tema de seguridad, para alertar de un posible problema o mal funcionamiento de los sistemas de la universidad.

- **Salida del script:** Número de tickets nuevos son: "$TICKETNUEVO" El número de tickets abiertos son "$TICKETABIERTO" El número de tickets cerrados son "$TICKETCERRADO" El número de tickets en seguimiento son "$TICKETSEGUIMIENTO"

- **Código**: Ver Anexo 6.

- **Flujo**:

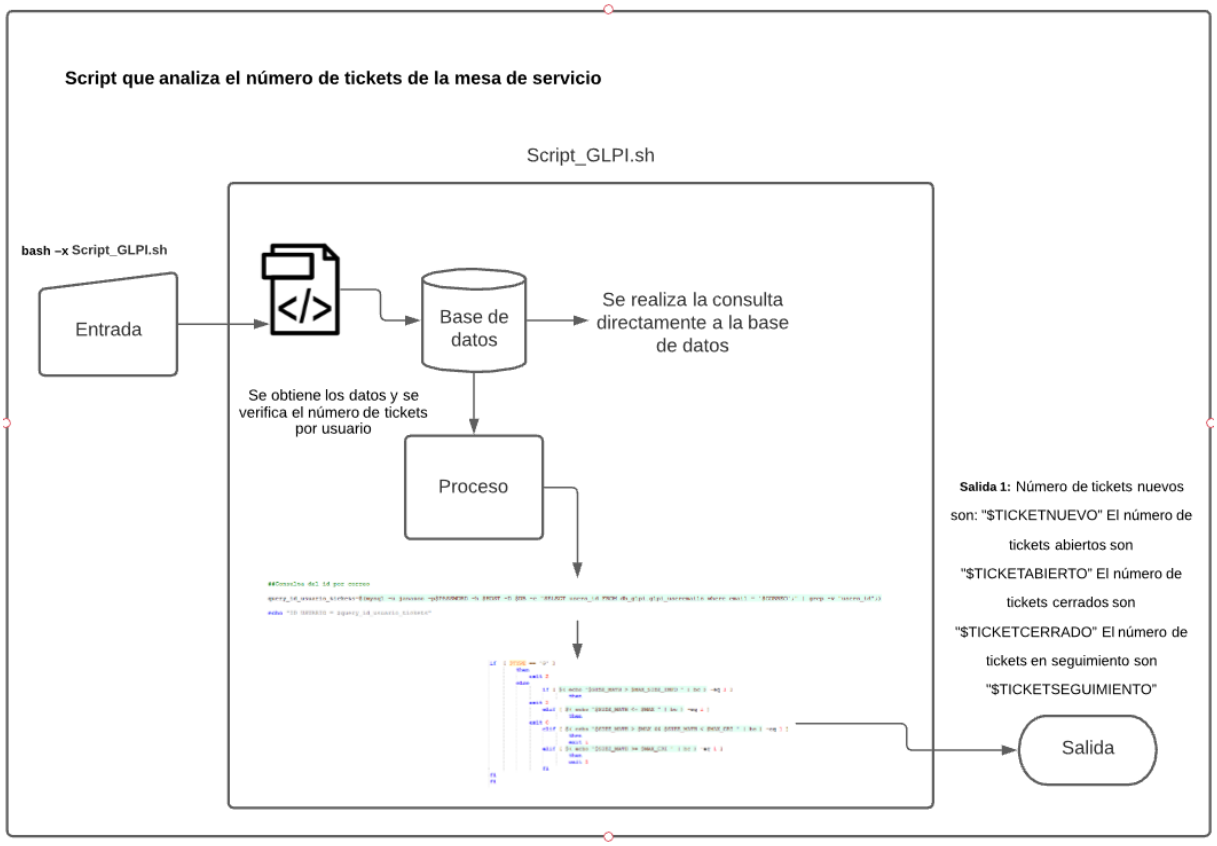


Figura 123: Script que analiza el número de tickets de la mesa de servicio
Elaborado por: Jorge Añazco

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- De acuerdo al análisis, se puede concluir que, el estado actual de la infraestructura tecnológica de la Universidad Internacional SEK no se encuentra de manera óptima ni recomendable; refleja que la institución no cuenta con una estructura, cultura, estándar, esquema, metodología o un sistema para implementar procesos de seguridad de la información, que contribuya con el monitoreo y control de los diferentes sistemas que maneja la Universidad. En el caso de la adquisición o desarrollos de nuevos sistemas, se requiere estandarizar inventarios y actualizarlos, creación de más ambientes para el desarrollo de software, metodologías, definir procesos de seguimiento y control ya sea de software y hardware, control de calidad sobre el código, por último, se puede mejorar el reglamento interno sobre seguridad de la información, e invertir en personal con experiencia en infraestructura informática para mejorar la seguridad de la información de la Universidad.

- La herramienta OSSIM AlienVault tiene una interfaz amigable y fácil de configurar al permitir acceso vía web a su consola, tiene como valor agregado a su fortaleza técnica que contiene también varios reportes orientados a la administración y toma de decisiones. Es importante reconocer que la curva de

aprendizaje es rápida y didáctica ya que cuenta con muchos tutoriales y ayudas en Internet.

- La implementación de la herramienta y sus costos depende del tamaño de la organización y el alcance que debe tener la herramienta, ya que, entre más complejo, se pueden elevar los costos y necesitar más personal como equipos, y gracias a sus módulos autónomos de Gestión de Archivos, Gestión de Disponibilidad, Gestión de Notificaciones, Gestión de Vulnerabilidades y Gestión de Riesgos, se puede monitorear en diferentes niveles la disponibilidad de los activos como de los servicios, además, de alertar al momento de encontrar alguna anomalía, así como también, se puede generar varios tipos de reportes según las necesidades de la Universidad.

- Se ha logrado establecer cinco tipos de scripts y definir métodos como procesos para la obtención de información, uno de los malestares que es parte del monitoreo es como obtener la información para procesarla y dar un resultado válido para la herramienta OSSIM AlienVault, con estas definiciones se pueden iniciar de manera clara los scripts, implementar una estructura más limpia y se optimiza el tiempo de desarrollo.

- El desarrollo de scripts para las necesidades de monitoreo requiere de una planificación previa para cumplir con el objetivo. Para este proyecto se realizaron cinco desarrollos donde se analizó los mejores métodos de obtención de datos y algoritmos de procesamiento de datos y se estableció periodos de ejecución para no afectar el rendimiento del servidor, se utilizó diferentes herramientas como tecnologías para cumplir el objetivo del desarrollo y se estandarizó procesos para que se pueda reutilizar códigos que ayuden a bajar

tiempo de desarrollos y optimizar funcionalidades. Una de las ventajas encontradas del desarrollo personalizado es que, se puede expandir y controlar el núcleo de la seguridad informática ya que con esto se puede implementar nuevos recursos que ayuden a minimizar los posibles riesgos.

5.2 Recomendaciones

- Dado los resultados del análisis de los 20 controles de la CIS v7 es recomendable ir ajustando, modificando y crear políticas de seguridad e ir estableciendo fases para el cumplimento de los controles de CIS, en las primeras fases como principal objetivo sería definir procesos de inventario de los diferentes activos de la organización, ya que, con esto se podrá dimensionar y definir alcances para el progresivo cumplimiento de los controles además se tendría una perspectiva de la situación real de la organización.

- Con la instalación de la herramienta OSSIM AlienVault se debe considerar la creación de lineamientos para la implementación de metodologías, estándares de calidad y procesos para comenzar a utilizar todos los módulos que ofrece la herramienta, además con esta herramienta se puede centralizar algunos requisitos que piden los controles de la CIS y así estandarizar el flujo de la seguridad de la información de la entidad.

- Se recomienda para futuros desarrollos utilizar herramientas adicionales para cumplir con las necesidades o los objetivos del monitoreo, ya que el lenguaje Shell ofrece la opción de incluir otros lenguajes de programación dentro del código, cómo por ejemplo, estos pueden ser PHP, Ruby, Perl y Python, por último, se puede adaptar nuevas tecnologías para complementar la obtención de datos y su análisis.

BIBLIOGRAFÍA

AT&T. (13 de septiembre de 2021). *AlienVault OSSIM*. Obtenido de AT&T Cybersecurity: https://cybersecurity.att.com/products/ossim

AuditScripts. (2018). *Critical Security Controls*. Obtenido de The CIS Critical Security Controls: https://www.auditscripts.com/free-resources/critical-security-controls/

Balarezo, A., & Poveda, D. (2015). PROPUESTA DE MEJORAMIENTO DE LA HERRAMIENTA OSSIM SIEM. *(Trabajo de titulación).* UNIVERSIDAD POLITÉCNICA SALESIANA, Quito.

Bravo, Á., Villafuerte, Á., & Patiño, J. (2015). Implantación De Una Herramienta Ossim Para El Monitoreo Y Gestión De La Seguridad De La Red Y Plataformas Windows Y Linux Aplicado A Empresas Medianas. *Artículos de Tesis de Grado - FIEC*.

CIS Controls. (2018). *CIS Controls Spanish Translation V7*. CIS Controls. Obtenido de https://learn.cisecurity.org/CIS-Controls-v7.1

EcuRed. (15 de junio de 2016). *https://www.ecured.cu/*. Obtenido de http://www.nagios.org/ Nagios: https://www.ecured.cu/Nagios

El Comercio. (12 de enero de 2021). *Ecuador, una de las naciones más atacadas por los 'hackers'.* Obtenido de El Comercio: https://www.elcomercio.com/tendencias/ecuador-naciones-atacadas-hackers-tecnologia.html

El Comercio. (23 de julio de 2021). *Virus RansomEXX es el responsable del ciberataque a CNT*. Obtenido de El Comercio:

https://www.elcomercio.com/actualidad/negocios/virus-ransomeware-cnt-ministerio-telecomunicaciones.html

Espinoza, D. (2015). ESTUDIO DE LA HERRAMIENTA DE SEGURIDAD OPEN. *(TESIS DE GRADO).* UNIVERSIDAD DE GUAYAQUIL, GUAYAQUIL.

Gagne, G., Bear, P., & Silberschatz, A. (2006). *Fundamento de Sistemas Operativos.* Madrid: McGraw-Hill.

Gómez, Á. (2011). *Enciclopedia de la Seguridad Informática 2ª EDICIÓN ACTUALIZADA.* Alfaomega Ra-Ma.

González, R. (2011). *Python para todos.* Recuperado de: https://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf.

helpsystem. (2019 de Diciembre de 2020). *https://www.hostdime.com.pe/*. Obtenido de https://www.helpsystems.com/es/blog/que-es-un-siem

Kaspersky. (2021). *Kaspersky cybermap*. Obtenido de CIBERAMENAZA MAPA EN TIEMPO REAL: https://cybermap.kaspersky.com/es

Labrador, R. M. (2003). *PROGRAMACIÓN AVANZADA EN SHELL.*

Marchionni, A. (2011). *Administrador de Servidores.* Buenos Aires: USERS.

Membrey, P., Verhoeven, T., & Angenendt, R. (2009). *The Definitive Guide to CentOS.* New York: Apress.

Ministerio de Telecomunicaciones y de la Sociedad de la Información. (15 de abril de 2019). *Ministerio de Telecomunicaciones y de la Sociedad de la Información*. Obtenido de Más de 40 millones de ataques al Ecuador neutralizados desde el retiro del asilo a Julian

Assange: https://www.telecomunicaciones.gob.ec/mas-de-40-millones-de-ataques-al-ecuador-neutralizados-desde-el-retiro-del-asilo-a-julian-assange/

Noguera, B. (19 de marzo de 2011). *¿Qué son los Scripts?* Obtenido de Culturación: https://culturacion.com/que-son-los-scripts/

Scott, G., & LaQuey, T. (Febrero de 1993). *IETF Tools (RFC1392).* Obtenido de https://tools.ietf.org/html/rfc1392

Universida Internacional SEK. (2021). *Direccionamiento Estratégico*. Obtenido de Uisek Ecuador: https://www.uisek.edu.ec/es/uisek/direccionamiento-estrategico

Universidad Internacional Sek. (2021). *Historia*. Obtenido de Universidad Internacional Sek : https://www.uisek.edu.ec/es/uisek/nosotros/historia

Universidad Internacional Sek. (2021). *Uisek Ecuador.* Obtenido de https://www.uisek.edu.ec

ANEXOS

Anexo 1

```
1
2   #!/bin/bash
3   #12/05/2021
4   #AUTOR: JORGE AÑAZCO
5   #VERSION: 0.7
6   #In the the there will be only chaos :^}
7
8   FUNCTIONS_SEND_TELEGRAM_LOCAL() {
9           (sleep 3;echo "msg $destination \"$message\""; echo "safe_quit") |
            telegram-cli -k tg-server.pub -W
10  }
11
12
13  FUNCTIONS_SEND_TELEGRAM_REMOTO() {
14
15          echo "Destino: $destination | -r | Mensaje: $message" >>
            /home/telegram/logtelegram.txt
16
17          HOST="192.168.0.148"
18          USER="root"
19          PASS="Adminlocal2k11"
20          VAR=$(expect -c "
21                  spawn ssh -o StrictHostKeyChecking=no $USER@$HOST
22                  match_max 100000
23                  expect \"*?assword:*\"
24                  send -- \"$PASS\r\"
25                  expect \"*$\"
26                  send -- \"cd /home/janazco/script_telegram\r\"
27                  expect \"*$\"
28                  send -- {./SendMessageTelegram.sh -c $destination -m \"$message\"}
29                  send -- \"\r\"
30                  expect eof
31          ")
32  echo "==============="
33  echo "$VAR"
34  }
35
36
37
38  #Entrada del comando validicion y funcionamiento
39  ##############################################################################################
    ######################
40  if [ "$1" == "" ]
41          then
42          echo " \"ALERTA\" ERROR EN EL COMANDO NO SE PUEDE PROCESAR"
43          exit 3
44  fi
45  ##############################################################################################
    ######################
46  while [ "$1" != "" ]; do
47      case $1 in
48  #-------------------------------------------------------------------------------------------
    --------------------#
49          -c | --contact )
50                                                                  shift
51                                                                  destination=$1
52                                                                  shift
53                                                                  if [ "$1" == "" ]
54                                                                          then
55                                                                                  echo
    " \"ALERTA\" ERROR EN EL COMANDO NO SE PUEDE PROCESAR"
56                                                                                  exit 3
57                                                                  elif [ "$1" == "-m" ]
58                                                                          then
59                                                                                  shift
60
    message=$1
61
    FUNCTIONS_SEND_TELEGRAM_LOCAL
62                                                                  else
63                                                                                  echo
    " \"ALERTA\" ERROR EN EL COMANDO NO SE PUEDE PROCESAR"
64                                                                                  exit 3
```

```
65                                                                      fi
66                                                                      ;;
67      #-------------------------------------------------------------------------------
        ----------------------#
68                      -r | --remote )                                 shift
69                                                                      if [ "$1" == "-c" ]
70                                                                              then
71                                                                                      shift
72
        destination=$1
73                                                                      else
74                                                                              echo "
        \"ALERTA\" ERROR EN EL COMANDO NO SE PUEDE PROCESAR"
75                                                                              exit 3
76                                                                      fi
77
78                                                  shift
79                                                                      if [ "$1" == "" ]
80                                                                              then
81                                                                                      echo
        " \"ALERTA\" ERROR EN EL COMANDO NO SE PUEDE PROCESAR"
82                                                                                      exit 3
83                                                                      elif [ "$1" == "-m" ]
84                                                                              then
85                                                                                      shift
86
        message=$1
87
        FUNCTIONS_SEND_TELEGRAM_REMOTO
88                                                                      else
89                                                                                      echo
        " \"ALERTA\" ERROR EN EL COMANDO NO SE PUEDE PROCESAR"
90                                                                                      exit 3
91                                                                      fi
92                                                                      ;;
93      #-------------------------------------------------------------------------------
        ----------------------#
94              -h | --help )                                           echo "          -r
                {Remoto} | -c {contanto} | -m {mensaje} | -h HELP"
95                                                                      echo "
                                                                        --remote {Remoto} |
                                                                        --contact {contacto}
                                                                        | -m {mensaje} |
                                                                        --help {HELP}"
96                                                                      echo "
                                                                        [Ejemplo para enviar
                                                                        mensaje de manera
                                                                        local]
                                                                        ./ScriptComunicacionTe
                                                                        legram.sh -c
                                                                        TELEGRAM_CONTACT -m
                                                                        MENSAJE"
                                                                                        $
97                                                                      echo "
                                                                        [Ejemplo para enviar
                                                                        mensaje desde otro
                                                                        servidor]
                                                                        ./ScriptComunicacionTe
                                                                        legram.sh -r -c
                                                                        TELEGRAM_CONTACT -m
                                                                        MENSAJE"
98                                                                      echo "
                                                                        Requisitos local
                                                                        {Telegram-cli}
                                                                        Requisitos remotos
                                                                        {Expect package}"
99                                                                      exit 3
100                                                                     ;;
101     #-------------------------------------------------------------------------------
        ----------------------#
102             * )                                                     echo "COMANDO NO
```

```
            ENCONTRADO"
103                                                     echo "Para mas ayuda
                                                        --help|-h"
104                                                     exit 3
105                                                     ;;
106         esac
107     shift
108     done
109
110
```

Anexo 2

```bash
#!/bin/bash
#12/04/2021
#AUTOR: JORGE AÑAZCO
#VERSION: 1
#In the the there will be only chaos :^}

FECHA=$(date +%Y-%m-%d--%H:%M)

HOST="172.16.254.160"
USER="DCPrueba"
PASS="Adminprueba"
PASS_ROOT="temporalPass"

    VAR=$(expect -c "
        match_max 100000
        spawn ssh -o StrictHostKeyChecking=no $USER@$HOST
        expect \"Authenticated with partial success.\"
        expect \"*?assword:*\"
        send -- \"$PASS\r\"
        expect  \">\"
        send -- \"1\r\"
        expect \"* Device Manager *\"
        expect \">\"
        send -- \"1\r\"
        expect \"* Environment  *\"
        expect \">\"
        send -- \"\033\r\"
        expect \"* Device Manager *\"
        expect \">\"
        send -- \"\033\r\"
        expect \"* Control Console *\"
        expect \">\"
        send -- \"4\r\"
        match_max 100000
        expect eof
    ")

TEMP=$(echo "$VAR" | grep "LP1: SensorH/T" | perl -pi -e "s[!][ ]g" |cut -d " " -f 13,14 | cut -d "," -f 1)
HUMEDAD=$(echo "$VAR" | grep "LP1: SensorH/T" | perl -pi -e "s[!][ ]g" |cut -d " " -f 13,14 | cut -d "," -f 2)

echo "Temperatura = "$TEMP" Humedad = "$HUMEDAD
TEMP=$( echo $TEMP |  perl -pi -e "s[C][]g" )
HUMEDAD=$( echo $HUMEDAD |  perl -pi -e "s[%][]g" )

if [ $TEMP -gt 28 ]
                        then
                        exit 2
                elif [ $TEMP -le 25 -a $HUMEDAD -ge 25   ]
                       then
                        exit 0
                elif [ $TEMP -gt 25 -a $TEMP -le 28 ]
                        then
                        exit 1
                elif [ $HUMEDAD -lt 25 ]
                        then
                        exit 2
                fi
exit 4

```

Anexo 3

```
#!/bin/bash
#12/02/2021
#AUTOR: JORGE AÑAZCO
#VERSION: 1.1
#In the the there will be only chaos :^}

#Declaración de Variables
ARCHIVO_SETUP="/home/nagios/Setup/config-server.info"
ARCHIVO_ESCRITURA="/home/nagios/Setup/serverwrite.info"
ARCHIVO_LOG="/home/nagios/logs/resumendiario.log"
FECHA=$(date +%Y-%m-%d)

#Numero de host que se encuntran cofigurados

typeset -i NUMERO_HOSTS=$(wc -l $ARCHIVO_SETUP | cut -d " " -f 1)

#CABECERA DEL MAIL
echo "************************************" >> $ARCHIVO_LOG
echo "INICIO SCRIPT COPIA A TRAVES DE FTP" >> $ARCHIVO_LOG
echo "Fecha = "$FECHA >> $ARCHIVO_LOG
echo "------------------------------------" >> $ARCHIVO_LOG

#####################################################################
#Comienzo del Script copiado de los archivos $ARCHIVO_ESCRITURA permite el ingreso a las carpetas para obtener los archivos

for ((a=4; a <= $NUMERO_HOSTS; a++))
    do
    INFO_HOST=$(awk NR==$a $ARCHIVO_ESCRITURA)
    #VALIDACION DE ARCHIVO
    cd /home/nagios/InfoVSphere/RVTools_Diario_"$INFO_HOST"

    rm vSnapshot.csv

                if [ $? = 0 ]
                        then
                        echo "ARCHIVO ANTIGUO ELIMINADO " >> $ARCHIVO_LOG
                        else
                        echo "NO SE ENCONTRO EL ARCHIVO " >> $ARCHIVO_LOG
                fi

#COPIA DEL ARCHIVO A TRAVEZ DE UN SERVICIO FTP

                ftp -inv 172.16.0.33 <<FTP
                        user admin Adminlocal2k11
                        cd RVTools_Diario_"$INFO_HOST"
                        lcd /home/nagios/InfoVSphere/RVTools_Diario_"$INFO_HOST"
                        get vSnapshot.csv
                        bye
FTP

#VALIDACION COPIA DEL ARCHIVO POR SERVICIO FTP

                if [ -f /home/nagios/InfoVSphere/RVTools_Diario_"$INFO_HOST"/vSnapshot.csv ]
                        then
                        echo "SE REALIZO LA COPIA CORRECTAMENTE" >> $ARCHIVO_LOG
                        else
                        echo "NO SE PUDO REALIZAR LA COPIA" >> $ARCHIVO_LOG
                fi

            echo "************************************" >> $ARCHIVO_LOG

done


exit
```

Anexo 4

```bash
#!/bin/bash
#12/03/2021
#AUTOR: JORGE AÑAZCO
#VERSION: 1
#In the the there will be only chaos :^}

FECHA=$(date +%Y-%m-%d)
MES=$(date  +%m)
YEAR=$(date  +%Y)
SUM=0
PROM=0
MAX=0

FUNCTIONS_REPORT_SIZE_LOGS () {

for (( a = 10; a > 0 ; a-- ))
    do
        FECHA_SIZE=$(date  --date="-$a day" +"%Y-%m-%d")
        SIZE+=($( cat $ARCHIVO_SIZEDIR | grep "$FECHA_SIZE" | awk '{print $2}' ))
    done

SIZE+=($( cat $ARCHIVO_INFO | grep "$LOG_INFO" | awk '{print $6}' ))


for ((a=0; a<  ${#SIZE[*]} - 1 ; a++))
    do
        typeset -i NUM_CHARACTER=$( echo ${SIZE[$a]} | wc -m )
        let d=(NUM_CHARACTER - 2)
        SIZE_MATH=$( echo ${SIZE[$a]} | cut -c 1-"$d" )
        let b=(NUM_CHARACTER - 1)
        TYPE=$(  echo ${SIZE[$a]} | cut -c "$b" )

        if  [ $TYPE == 'K' ]
            then
                SIZE_MATH=` echo "$SIZE_MATH / 1000" | bc -l`
            fi
            if  [ $TYPE == 'G' ]
                    then
                        SIZE_MATH=`echo "$SIZE_MATH * 1000" | bc -l`
                        BAND=1
                    fi
            SUM=`echo "$SUM + $SIZE_MATH" | bc`

            done

PROM=`echo " $SUM / ( ${#SIZE[*]} - 1 )" | bc -l`
MAX=`echo " $PROM * 2 " | bc -l`
MAX_CRI=`echo " $PROM * 3 " | bc -l`

k=`echo "${#SIZE[*]} - 1 " | bc -l`

NUM_CHARACTER=$( echo ${SIZE[$k]} | wc -m )
let d=(NUM_CHARACTER - 2)
SIZE_MATH=$( echo ${SIZE[$k]} | cut -c 1-"$d" )
let b=(NUM_CHARACTER - 1)
TYPE=$(  echo ${SIZE[$k]} | cut -c "$b" )

i=$( echo $MAX | cut -c 1-5  )
j=$( echo $MAX_CRI | cut -c 1-5  )


if  [ "$BAND" == "1" ]
        then
                echo "Tamano del LOG es: "$SIZE_MATH$TYPE"  MAX-NORMAL = "$i"M
                MAX-CRITICAL = "$j"M  Revisar los LOGS anteriores existe un log que
                pesa mas de 1 GB"
        else
        echo "Tamano del LOG es: "$SIZE_MATH$TYPE"  MAX-NORMAL = "$i"M  MAX-CRITICAL =
        "$j"M"
```

```
67  fi
68
69
70  if  [ $TYPE == 'K' ]
71          then
72                  exit 0
73          else
74  if  [ $TYPE == 'G' ]
75          then
76              exit 2
77          else
78                  if [ $( echo "$SIZE_MATH > $MAX_SIZE_INFO " | bc ) -eq 1 ]
79                          then
80              exit 2
81                  elif [ $( echo "$SIZE_MATH <= $MAX " | bc ) -eq 1 ]
82                          then
83              exit 0
84                  elif [ $( echo "$SIZE_MATH > $MAX && $SIZE_MATH < $MAX_CRI " | bc ) -eq
    1 ]
85                          then
86                          exit 1
87                  elif [ $( echo "$SIZE_MATH >= $MAX_CRI " | bc ) -eq 1 ]
88                          then
89                          exit 2
90                  fi
91  fi
92  fi
93
94  }
95
96  FUNCTIONS_REPORT_APACHE_ACCESS () {
97
98  for (( a = 0; a <= 10 ; a++ ))
99                          do
100             FECHA_INFO=$(date  --date="-$a day" +"%Y-%m-%d")
101             SIZE+=($( cat $ARCHIVO_SIZEDIR | grep "$FECHA_INFO" | awk '{print $2}' ))
102 done
103
104 SIZE+=($( cat $ARCHIVO_INFO | grep "Numero de Consultas Totales son:" | awk '{print
    $7}' ))
105
106
107 for ((a=0; a<  ${#SIZE[*]} - 1 ; a++))
108                         do
109                         let SUM=(SUM + SIZE[$a])
110                         done
111 x=${#SIZE[*]}
112 let z=( x - 1 )
113 let PROM=( SUM / z )
114 let MAX=( PROM * 2 )
115 let MAX_CRI=( PROM * 3 )
116
117 k=`echo "${#SIZE[*]} - 1 " | bc -l`
118
119 echo "Numero de Peticiones son: "${SIZE[$k]}"  MAX-NORMAL = "$MAX"  MAX-CRITICAL =
    "$MAX_CRI
120
121         if [ ${SIZE[$k]} -gt $MAX_SIZE_INFO  ]
122                         then
123                         exit 2
124                 elif [  ${SIZE[$k]} -le $MAX  ]
125                         then
126                         exit 0
127                 elif [  ${SIZE[$k]} -gt $MAX -a ${SIZE[$k]} -lt $MAX_CRI  ]
128                         then
129                         exit 1
130                 elif [ ${SIZE[$k]} -ge $MAX_CRI   ]
131                         then
132                         exit 2
```

```
                fi


}

while [ "$1" != "" ]; do
   case $1 in
       -t | --LOGSSIZE )       shift
                               ARCHIVO_INFO=$1
                               shift
                               ARCHIVO_SIZEDIR=$1
                               shift
                               LOG_INFO=$1
                               shift
                                                                MAX_SIZE_INFO=$1
                               FUNCTIONS_REPORT_SIZE_LOGS
                               ;;
       -k | --BDDSIZE  )       shift
                               ARCHIVO_INFO=$1
                                                                shift
                                                                ARCHIVO_SIZEDIR=$1
                                                                shift
                                                                LOG_INFO=$1
                                                                shift
                                                                MAX_SIZE_INFO=$1
                    FUNCTIONS_REPORT_APACHE_ACCESS
                               ;;
       -h | --help )           echo "-c {REPORTE COMPLETO} -l {ULTIMA HORA} -h HELP"
                               echo "--complete|-c       --lasthour|-l         --help|-h"
                               exit
                               ;;
        * )                    echo "COMANDO NO ENCONTRADO"
                               echo "Para mas ayuda --help|-h"
                               exit
                               ;;
esac
      shift

done

exit
```

Anexo 5

```bash
#!/bin/bash
#12/05/2021
#AUTOR: JORGE AÑAZCO
#VERSION: 1
#In the the there will be only chaos :^}

FECHA=$(date +%Y-%m-%d)
MES=$(date  +%m)
YEAR=$(date  +%Y)
SUM=0
PROM=0
MAX=0

FUNCTIONS_REPORT_SIZE_LOGS () {
ARCHIVO_SIZE_LOG="/SIZEDIRPERDAYLOG.txt"
LOG_FINAL=$(echo "$ARCHIVO_INFO" | cut -d "/" -f 1,2,3,4,5,6)
LOG="$LOG_FINAL$ARCHIVO_SIZE_LOG"

for (( a = 10; a > 0 ; a-- ))
                        do
                        FECHA=$(date  --date="-$a day" +"%Y-%m-%d")
                        SIZE+=($( cat $LOG | grep "$LOG_INFO""$FECHA" | awk '{print $1}' ))
done
FECHA=$(date  --date="-$a day" +"%Y-%m-%d")
SIZE+=($( cat $ARCHIVO_INFO | grep "$LOG_INFO$FECHA" | awk '{print $1}' ))

for ((a=0; a<  ${#SIZE[*]} - 1 ; a++))
                        do
            typeset -i NUM_CHARACTER=$( echo ${SIZE[$a]} | wc -m )
            let d=(NUM_CHARACTER - 2)
            SIZE_MATH=$( echo ${SIZE[$a]} | cut -c 1-"$d" )
            let b=(NUM_CHARACTER - 1)
            TYPE=$(  echo ${SIZE[$a]} | cut -c "$b" )

            if  [ $TYPE == 'K' ]
                then
                        SIZE_MATH=`echo "$SIZE_MATH / 1024" | bc -l`
                fi
            if  [ $TYPE == 'G' ]
                then
                    SIZE_MATH=`echo "$SIZE_MATH * 1024" | bc -l`
                BAND=1
                fi
            SUM=`echo "$SUM + $SIZE_MATH" | bc`
            PROM=`echo "$SUM / ( ${#SIZE[*]} - 1 )" | bc -l`

            done

MAX=`echo "$PROM * 2 " | bc -l`
MAX_CRI=`echo "$PROM * 3 " | bc -l`

k=`echo "${#SIZE[*]} - 1 " | bc -l`
NUM_CHARACTER=$( echo ${SIZE[$k]} | wc -m )
let d=(NUM_CHARACTER - 2)
SIZE_MATH=$( echo ${SIZE[$k]} | cut -c 1-"$d" )
let b=(NUM_CHARACTER - 1)
TYPE=$(  echo ${SIZE[$k]} | cut -c "$b" )

i=$( echo $MAX | cut -c 1-5 )
j=$( echo $MAX_CRI | cut -c 1-5 )


if  [ "$BAND" == "1" ]
        then
                echo "Tamano del LOG es: "$SIZE_MATH$TYPE"  MAX-NORMAL = "$i"M
                MAX-CRITICAL = "$j"M  Revisar los LOGS anteriores existe un log que
                pesa mas de 1 GB"
        else
```

```
67          echo "Tamano del LOG es: "$SIZE_MATH$TYPE"  MAX-NORMAL = "$i"M  MAX-CRITICAL =
            "$j"M"
68  fi
69
70
71  if  [ $TYPE == 'K' ]
72          then
73                  exit 0
74          else
75  if  [ $TYPE == 'G' ]
76          then
77              exit 2
78          else
79                  if [ $( echo "$SIZE_MATH > $MAX_SIZE_INFO " | bc ) -eq 1 ]
80                      then
81              exit 2
82                  elif [ $( echo "$SIZE_MATH <= $MAX " | bc ) -eq 1 ]
83                      then
84              exit 0
85                  elif [ $( echo "$SIZE_MATH > $MAX && $SIZE_MATH < $MAX_CRI " | bc ) -eq
                    1 ]
86                      then
87                      exit 1
88                  elif [ $( echo "$SIZE_MATH >= $MAX_CRI " | bc ) -eq 1 ]
89                      then
90                      exit 2
91                  fi
92  fi
93  fi
94
95  }
96
97
98  FUNCTIONS_REPORT_SIZE_DIR (){
99
100 SIZE=$( cat $ARCHIVO_INFO | grep "$DIR_LOG" | grep -v "postgresql" | awk '{print $1}' )
101 typeset -i NUM_CHARACTER=$( echo $SIZE | wc -m )
102 let d=(NUM_CHARACTER - 2)
103 SIZE_MATH=$( echo $SIZE | cut -c 1-"$d" )
104 let b=(NUM_CHARACTER - 1)
105 TYPE=$(  echo $SIZE | cut -c "$b" )
106 echo "Tamano del LOG es: " $SIZE_MATH$TYPE
107
108 SIZE_DIR=$(cat $ARCHIVO_SIZEDIR | grep "$FECHA" | awk '{print $2}')
109
110 typeset -i NUM_CHARACTER=$( echo $SIZE_DIR | wc -m )
111 let d=(NUM_CHARACTER - 2)
112 SIZE_MATH_DIR=$( echo $SIZE_DIR | cut -c 1-"$d" )
113 let b=(NUM_CHARACTER - 1)
114 TYPE_DIR=$(  echo $SIZE_DIR | cut -c "$b" )
115
116
117 if  [ $TYPE == 'K' ]
118     then
119     SIZE_MATH=`echo "$SIZE_MATH / 1024" | bc -l`
120     fi
121 if  [ $TYPE == 'G' ]
122     then
123     SIZE_MATH=`echo "$SIZE_MATH * 1024" | bc -l`
124     fi
125
126 if  [ $TYPE_DIR == 'K' ]
127         then
128         SIZE_MATH_DIR=`echo "$SIZE_MATH_DIR / 1024" | bc -l`
129         fi
130 if  [ $TYPE_DIR == 'G' ]
131         then
132         SIZE_MATH_DIR=`echo "$SIZE_MATH_DIR * 1024" | bc -l`
133         fi
```

```
134
135   DIR_PORCENTAJE=`echo "$SIZE_MATH_DIR * 0.1 " | bc -l`
136
137   SUM1=`echo "$SIZE_MATH_DIR + $DIR_PORCENTAJE " | bc -l`
138   SUM2=`echo "$SIZE_MATH_DIR + ( $DIR_PORCENTAJE * 2 ) " | bc -l`
139
140    if [ $( echo "$SIZE_MATH <= $SUM1 " | bc ) -eq 1 ]
141                            then
142                            exit 0
143                    elif [ $( echo "$SIZE_MATH > $SUM1 && $SIZE_MATH < $SUM2 " | bc ) -eq 1 ]
144                            then
145                            exit 1
146                    elif [ $( echo " $SIZE_MATH > $SUM2 " | bc ) -eq 1 ]
147                            then
148                            exit 2
149                    fi
150
151
152   }
153
154   FUNCTIONS_REPORT_SIZE_BDD (){
155   SIZE=$( cat $ARCHIVO_INFO | grep "$DIRECTORIO" | awk '{print $5}' )
156   TAMANO=$( cat $ARCHIVO_INFO | grep "$DIRECTORIO" | awk '{print $2}' )
157   UTILIZADO=$( cat $ARCHIVO_INFO | grep "$DIRECTORIO" | awk '{print $3}' )
158   DISPONIBLE=$( cat $ARCHIVO_INFO | grep "$DIRECTORIO" | awk '{print $4}' )
159
160   typeset -i NUM_CHARACTER=$( echo $SIZE | wc -m )
161   let d=(NUM_CHARACTER - 2)
162   SIZE_MATH=$( echo $SIZE | cut -c 1-"$d" )
163   let b=(NUM_CHARACTER - 1)
164   TYPE=$(  echo $SIZE | cut -c "$b" )
165   echo "El porcentaje es: " $SIZE_MATH$TYPE "TAMANO = "$TAMANO"  OCUPADO = "$UTILIZADO"
      DISPONIBLE = "$DISPONIBLE
166
167    if [ $SIZE_MATH -le 80 ]
168                            then
169                            exit 0
170                    elif [ $SIZE_MATH -gt 80 -a $SIZE_MATH -lt 90  ]
171                            then
172                            exit 1
173                    elif [ $SIZE_MATH -ge 90  ]
174                            then
175                            exit 2
176                    fi
177
178
179   }
180
181   while [ "$1" != "" ]; do
182      case $1 in
183           -t | --LOGSSIZE )       shift
184                                   ARCHIVO_INFO=$1
185                                   shift
186                                   ARCHIVO_SIZEDIR=$1
187                                   shift
188                                   LOG_INFO=$1"-"
189                                   shift
190                                                                   MAX_SIZE_INFO=$1
191                                   FUNCTIONS_REPORT_SIZE_LOGS
192                                   ;;
193           -l | --DIRSIZE  )       shift
194                                   ARCHIVO_INFO=$1
195                                       shift
196                                                                   ARCHIVO_SIZEDIR=$1
197                                   shift
198                                   DIR_LOG=$1
199                                   FUNCTIONS_REPORT_SIZE_DIR
200                                   ;;
201           -k | --BDDSIZE  )       shift
```

```
                                ARCHIVO_INFO=$1
                                shift
                                DIRECTORIO=$1
                                FUNCTIONS_REPORT_SIZE_BDD
                                ;;
        -h | --help )           echo "-c {REPORTE COMPLETO} -l {ULTIMA HORA} -h HELP"
                                echo "--complete|-c       --lasthour|-l         --help|-h"
                                exit
                                ;;
         * )                    echo "COMANDO NO ENCONTRADO"
                                echo "Para mas ayuda --help|-h"
                                exit
                                ;;
  esac
        shift

  done

  exit
```

Anexo 6

```bash
#!/bin/bash
#10/06/2021
#AUTOR: JORGE AÑAZCO
#VERSION: 1
#In the the there will be only chaos :^}

FUNCTIONS_OPTION_IP () {
   INFO_HOST=$(echo $IP_ADDRESS | perl -pi -e "s[\.][""]g")
   ARCHIVO_VSPHERE=$( echo "/home/nagios/InfoVSphere/RVTools_Diario_IP/vSnapshot.csv" | perl -pi -e "s[IP]["$INFO_HOST"]g" )
   if [ -f $ARCHIVO_VSPHERE ]
        then
            typeset -i NUMERO_LINEAS=$(wc -l $ARCHIVO_VSPHERE | cut -d " " -f 1)
            if [ $NUMERO_LINEAS = 1 ]
                then
                                echo -e "No exite SnapShot en el Servidor"
                exit 0
            else
                let NUMERO_SNAPSHOTS=(NUMERO_LINEAS-1)
                echo "\"EXISTEN SNAPSHOTS EN ESTE SERVIDOR\" |  Numero de Snapshots = "$NUMERO_SNAPSHOTS
                if [[ $NUMERO_SNAPSHOTS > 1 && $NUMERO_SNAPSHOTS < 7 ]]
                    # Se valida si el numero de snapshots son mayores a 1 y menores a 7
                    then
                        exit 1
                        # Si el numero de snapshots son entre 1 y 7 solo se da una advertencia
                    else
                        exit 2
                        # Si el numero de snapshots son mayores a 7 se alerta
                fi

        fi
   else
                echo "\"ALERTA NO SE PUDO ENCONTRAR EL ARCHIVO .CSV FAVOR REVISAR\""
        exit 3
   fi
}


##############################################################################################################
if [ "$1" == "" ]
    then
    echo " \"ALERTA\" ERROR EN EL COMANDO NO SE PUEDE PROCESAR"
    exit 3
fi
##############################################################################################################
while [ "$1" != "" ]; do
    case $1 in
#-------------------------------------------------------------------------------------------------#
        -p | --ipaddress )      shift
                IP_ADDRESS=$1
                FUNCTIONS_OPTION_IP
                ;;
#-------------------------------------------------------------------------------------------------#
        -h | --help )           echo "-p {IP ADDRESS} | -h HELP"
                                echo "--ipaddress|-p --help|-h"
                                exit 3
                                ;;
#-------------------------------------------------------------------------------------------------#
         * )                    echo "COMANDO NO ENCONTRADO"
                echo "Para mas ayuda --help|-h"
                exit 3
                ;;
```

```
62        esac
63    shift
64    done
65
66
67    exit
68
```