ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Maestría en Seguridad Informática Aplicada

"IMPLEMENTACIÓN DE UNA SOLUCIÓN PARA LA ADMINISTRACIÓN CENTRALIZADA DE LOGS GENERADOS EN UN AMBIENTE MULTIPLATAFORMA UTILIZANDO SOFTWARE LIBRE."

EXAMEN DE GRADO (COMPLEXIVO)

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

MAGISTER EN SEGURIDAD INFORMÁTICA APLICADA

FREDDY JAVIER MERCHÁN REYES

GUAYAQUIL – ECUADOR

AÑO: 2015

AGRADECIMIENTO

En este momento importante en mi vida profesional quiero dejar mis sinceros agradecimientos:

A la Escuela Superior Politécnica, formadores de excelentes profesionales por permitirme el privilegio de alcanzar la meta propuesta de ser Magister en Seguridad Informática Aplicada, así como a los docentes que impartieron sus conocimientos y experiencias, por el aporte brindado y sus orientaciones brindadas para mi desenvolvimiento profesional.

Al Ing. Lenin Freire, por sus orientaciones que permitieron culminar de forma exitosa esta etapa.

Con respeto y afecto a mis compañeros de estudio y a todas las personas que me apoyaron durante esta etapa de desarrollo académico y profesional.

DEDICATORIA

A Dios por ser mi guía y fortaleza para los momentos difíciles que se presentaron en esta etapa académica, dándome la fuerza y la fe necesaria para lograr superar los obstáculos que se me presentaron, permitiéndome superar aquellas barreras para poder lograr culminar con éxito la maestría.

A mis padres: Francisco Merchán y Fanny Reyes por ser ejemplo en mi vida, cuyas virtudes morales y espirituales guían siempre todos los actos que realizo, motivo para tenerle gratitud porque sin sus guías y consejos mi formación no hubiera llegado a feliz culminación.

A mis hermanos: John, Fanny y Jorge por el apoyo y ánimos recibido para culminar mis estudios y cumplir el objetivo propuesto.

TRIBUNAL DE SUSTENTACIÓN

_____	_____
MGS. KARINA ASTUDILLO	**ING. JUAN CARLOS GARCÍA**
PROFESOR DELEGADO	PROFESOR DELEGADO
POR LA UNIDAD ACADÉMICA	POR LA UNIDAD ACADÉMICA

RESUMEN

El Proyecto presentado consistió en el diseño de una solución integral para la administración centralizada de eventos de seguridad (logs) en ambientes heterogéneos utilizando software libre, por esta motivo se realizó un estudio sobre las herramientas y estrategia para su administración.

En la primera parte se presenta los conceptos básicos y el análisis de las tecnologías utilizadas, para luego definir los requerimientos técnicos, tecnológicos y procedimentales necesarios para la centralización de los elementos objetos de la investigación.

En la segunda parte como resultado de este trabajo se presenta una propuesta de solución implementada para la administración de eventos de seguridad mediante el uso de software libre, describiendo la definición de procedimientos y el uso de programas que permitieron asegurar y centralizar los logs de toda una infraestructura tecnológica que permitan cumplir los requisitos de seguridad que se deben tener para los mismos.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ABREVIATURA Y SIMBOLOGÍA

DNS	Domain Name System. Sistema de Nombres de Dominio
ESXI	VMware vSphere Hypervisor. Hypervisor para infraestructura de virtualización de VMware.
FTP	File Transfer Protocol – Protocolo de Transferencia de Archivos
IDS	Intrusion Detection System – Sistema de Detección de Intrusos
IP	Internet Protocol
IPS	Intrusion Prevention System – Sistema de Prevención de Intrusos
ISO	International Organization for Standardization – Organización Internacional de Normalización
MV	Machine Virtual. Máquina virtual
NAS	Network Attached Storage – Almacenamiento conectado en Red.
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol – Protocolo de Red
RSYNC	Remote Sync
SAN	Storage Area Network – Red de área de almacenamiento
SSH	Secure Shell.
VAPP	Colección de máquinas virtuales
VPN	Virtual Private Network – Red Privada Virtual
WLC	Wireless Lan Controller - Controladores para redes LAN inalámbricas

INTRODUCCIÓN

Este proyecto tiene como principal propósito de estudio y aplicación la administración centralizada de los eventos de seguridad conocidos en términos anglosajón como log que se producen en cada uno de los sistemas, dispositivos o aplicaciones de una infraestructura tecnológica. En concreto, este trabajo se realiza con el objetivo de establecer una propuesta de solución concebida principalmente a la gestión de dichos eventos mediante la utilización de software libre.

La característica principal de estos eventos generados es la diversidad de formatos en que se pueden presentar, y que cada uno de ellos son almacenados de forma local en el lugar donde se generan dificultando su seguimiento y administración en el caso de que sucede algún incidente informático.

Otra de la problemática que se presenta en la administración de los log es que por lo general son almacenados en archivos de texto que pueden ser fácilmente modificables reduciendo la confiabilidad e integridad que se les pueden tener, por lo que es necesario definir un mecanismo y herramientas que permitan asegurar el contenido de esos archivos y que sirvan como medio de prueba en algún incidente informático que ocurra.

Además, un aspecto que también influye en que no se implemente soluciones de administración centralizada de log es el uso de las herramientas necesarias para realizar esta tarea, debido a que por lo general se considera que solo con programas privativos se puede lograr este objetivo, sin tener en consideración que existen soluciones que utilizan software libre.

Así, el sentido final de este proyecto es demostrar mediante el uso de software libre la implementación de una arquitectura centralidad de log que permita asegurar y analizar los eventos ocurridos en cualquier infraestructura tecnológica independiente de su nivel de heterogeneidad, para lograr esto se utilizará además protocolos estándar a todos estos orígenes de eventos.

Para lograr los objetivos propuesto en este trabajo, en el capítulo I se presenta los lineamientos generales que dieron origen a esta investigación, en el capítulo II se detalla la metodología de desarrollo y las aplicaciones utilizadas, en el capítulo III se presentará el análisis de los resultados de la solución implementada, para finalmente exponer las conclusiones y recomendaciones obtenidas en la elaboración de este proyecto.

CAPÍTULO 1

1. GENERALIDADES

1.1. DESCRIPCIÓN DEL PROBLEMA

En la actualidad, las instituciones tienen en su infraestructura tecnológica un sinnúmero de dispositivos (servidores, routers, switches, access point, sistemas de almacenamientos, etc.) y aplicaciones (sistemas operativos, bases de datos, etc.) interrelacionadas entres sí, que son generadores de eventos de seguridad denominados generalmente como logs relacionados con alguna actividad o estado del dispositivo o aplicación, con la particularidad de que cada uno de esos sucesos manejan su propia estructura de registro, lo que ocasiona que la interpretación y análisis sea una tarea complicada.

Por lo general los eventos de seguridad (LOGS) de los dispositivos y aplicaciones son grabados en los almacenamientos locales donde se generan, existiendo el inconveniente que muchas veces los mismos poseen una capacidad reducida, ocasionando que se graben por un corto periodo de tiempo, para luego borrar o sobrescribir el espacio de asignado para su registro, lo que imposibilita el cumplimiento de directrices establecidas en la norma ISO 27001 referente a que se debe mantener registros de los eventos ocurridos en la infraestructura tecnológica con el objetivo de permitir realizar auditorías informáticas, así como el cumplimiento de normativas referentes a seguridad de la información y de materia legislativa.

Debido a la diversidad de tecnologías y aplicaciones existentes es necesario que se implemente una solución que permita centralizar de una manera normalizada y homogeneizada todos los eventos que se generan en el ambiente tecnológico de cualquier institución, además de que permita garantizar la seguridad en el almacenamiento de cada uno de los eventos que se generen en la infraestructura tecnológica. Lograr esto en un ambiente muy heterogéneo es una ardua tarea, por lo que se necesita tener un conjunto de software que permite recolectar, almacenar, registrar, y generar reportes de los eventos ocurridos en el ambiente administrado.

Situándonos en este escenario, las instituciones públicas e incluso las privadas están teniendo problemas para administrar de una forma centralizada los logs que se generan en cada una de las partes de sus infraestructuras, además de no poseer una herramienta que les permita de manera gráfica el análisis de los mismos. Para solucionar este inconveniente existen herramientas comerciales que son muy costosas y que no están al alcance de muchas de las organizaciones, por lo que existe la necesidad de implementar herramientas Open Source para permitir a los organismos con poco recurso económico gestionar de una manera eficiente y eficaz sus logs teniendo en consideración las normas referentes al cumplimiento tanto de las leyes nacionales e internaciones en el manejo de estos eventos.

1.2. SOLUCIÓN PROPUESTA

Como una forma de hacer frente a las necesidades de las instituciones públicas y privadas de tener una solución para la gestión centralizada de logs que permita almacenar, administrar y generar reportes de los eventos de seguridad que se presenta en su infraestructura tecnológica heterogénea, se propone la implementación de un esquema centralizado de logs utilizando software libre.

En esta propuesta se analiza la homologación de los eventos que se presenten en cada uno de los dispositivos y aplicaciones que integren la infraestructura tecnológica de una institución, así como las metodologías que se utilizan actualmente para poder hacer frente a estas amenazas, con el objetivo de que la solución propuesta permita el descubrimiento, monitoreo, protección y administración de la información de una forma eficaz a través del establecimiento de políticas, que contribuya con el objetivo de disminuir la incidencia de pérdida de datos.

Otro punto a tener en consideración es que por lo general la implementación de un esquema de Centralizado de logs puede ser muy costoso y que solo se justifica en instituciones que estén obligadas por el cumplimiento de alguna ley. Por esta razón, en este trabajo se hace uso de herramientas libres por tres razones fundamentales: la primera como una medida de fomentar la investigación en este tipo de solución, la segunda permitir estar en concordancia con la política de Estado referente al uso del software libre y la última con el objetivo de reducir costos considerable en la implementación sin que eso signifique reducción de la eficiencia y eficacia de la solución implementada.

La solución que se implementó debe permitir mediante una interfaz de gestión verificar la información de los logs almacenados. Con lo expresado anteriormente los beneficios de esta solución son:

- ✓ Administración centralizada de logs de diferentes fuentes heterogéneas de la infraestructura tecnológica de las instituciones.
- ✓ Análisis y categorización de eventos.
- ✓ Centralización de eventos según su origen.
- ✓ Descubrimiento de actividades en los equipos
- ✓ Almacenamiento de log en Sistemas de Gestión de Base de datos, para un análisis de eventos más personalizados.

1.3. OBJETIVO GENERAL

- ✓ Implementar una solución para la administración centralizada de LOGS generados en una infraestructura tecnológica multiplataforma para almacenar, analizar y generar informes utilizando software libre.

CAPÍTULO 2

2. METODOLOGÍA DE DESARROLLO DE LA SOLUCIÓN

2.1. Definición de Log

Log es un término anglosajón que de acuerdo al NIST [1] se define como un registro de los eventos ocurridos dentro de los sistemas o redes de una organización. El log nos permite tener un registro de la actividad realizada durante un rango de tiempo en particular, permitiendo ser utilizado como evidencia para auditoria informática o verificación de algún riesgo informático por el motivo de que estos registros permiten almacenar información sobre quién, qué, dónde, cuándo y por qué ocurrió.

2.2. Definición de Software Libre

Según Free Software Fundation [2] el software libre: “es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software”.

En otras palabras el software libre es aquel que puede ejecutarse con cualquier propósito o finalidad, se lo puede modificar sin restricción, redistribuir las copias que se consideren necesarias.

2.3. Característica de una infraestructura tecnológica típica

Por lo general las infraestructuras tecnologías de las organizaciones poseen varios de los siguientes dispositivos y aplicaciones: switch, router, access point, WLC, firewall servidores físicos y virtuales, controlador de dominio, sistemas de almacenamiento, IPS, IDS, granja de servicios (web, correo, gestión documental, ftp, ssh, balanceador de cargas) y aplicativos propios, lo que de manera resumida y generalizada se puede apreciar en la Figura 2.1, donde cada uno de esos elementos son fuentes generadores de eventos de seguridad susceptibles de ser almacenados para su revisión y evaluación.

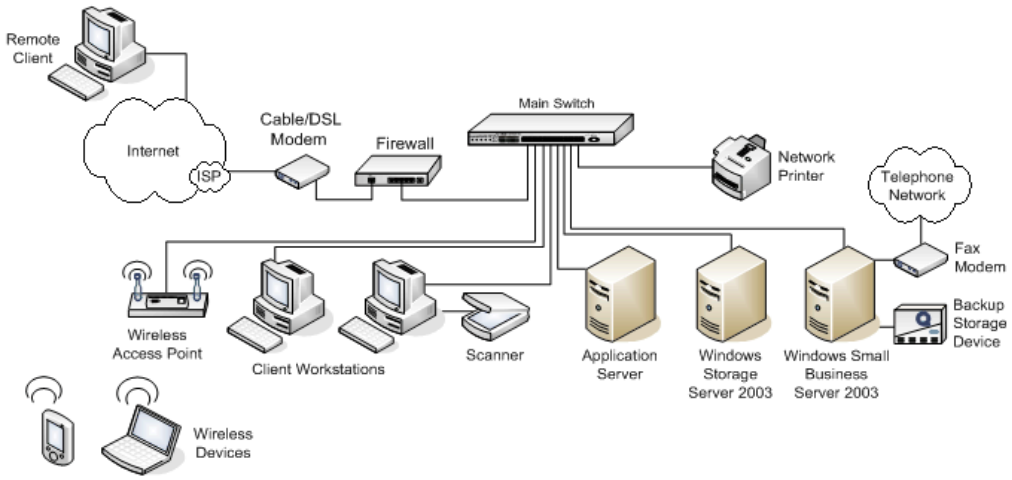


Figura 2.1 Infraestructura típica de una red
Fuente:
https://www.microsoft.com/latam/technet/mediana/images/DESLAN01_big.gif

2.4. Definición de requerimientos

Para lograr el propósito de asegurar y centralizar los LOGS empresarial es necesario definir ciertos requisitos tantos funcionales como no funcionales, los mismos que se describen a continuación [3]:

2.4.1. Requerimientos funcionales

Este tipo de requisitos se basan entre la interacción existente entre un sistema o aplicación con otro usuario o sistema, para lo que se han definido como principales aspectos a cumplir por este proyecto los siguientes puntos:

- ✓ Eventos centralizado

- ✓ Formato homologado con mínimo los siguientes datos: identificación de la fuente, marca de tiempo de ocurrido y detalle del evento ocurrido.
- ✓ Comprensión de los datos almacenados
- ✓ Todos los dispositivos y aplicaciones que son monitoreados deben estar sincronizados con alguna fuente externa, la información que debe tener mínima son la hora y fecha del evento
- ✓ Almacenamiento escalable de los log guardados.
- ✓ Acceso rápido
- ✓ Soportar cualquier tipo de fuente generadora de log.

2.4.2. Requerimientos no funcionales

Este apartado se refiere a los requisitos que no afectan directamente al comportamiento del sistema o aplicación generadora del evento almacenado; pero que son necesarios para lograr tener confiabilidad de todos sucesos que se registran, entre los aspectos considerados se menciona los siguientes:

- ✓ Protocolo de transporte utilizado en la transferencia de los archivos debe ser confiable.
- ✓ Asegurar la integridad y confiabilidad de los log guardados.
- ✓ Asegurar la disponibilidad de los elementos almacenados, para su utilización en cualquier momento que se lo requiera con fines de auditoría o análisis de alguna vulnerabilidad.

2.5. Sincronización de marca de tiempo

Con la finalidad de tener un control y fiabilidad de los log centralizados, es necesario que todos los orígenes de eventos posean una hora común, por lo que se procede a la configuración en todos los equipos de la hora, mediante la ubicación de la dirección IP o nombre DNS del servidor NTP ubicado en la red. (Equipo asignado con la dirección IP: 10.128.21.168)

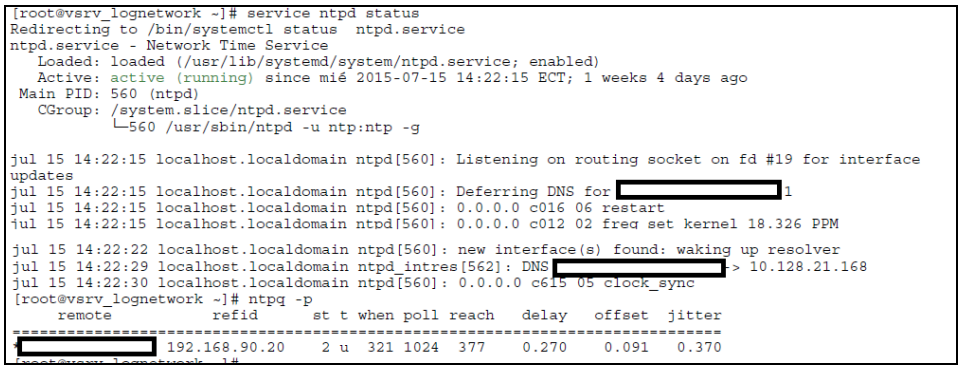


Figura 2.2 Verificación de servicio ntp. Activo en unos de los dispositivos que conforma la red.
Fuente: Autor

2.6. Protocolos de transporte de log

Para transportar logs puede ser utilizado diferentes protocolos, los mismos que tienen sus ventajas, desventajas y características particulares, a continuación se describe de breve manera los principales protocolos existentes [4]:

- ✓ **BSD syslog**.- Protocolo diseñado únicamente para el registro de log y descrito en el RFC 4164.
- ✓ **UDP basado en texto plano**, que utiliza de manera eficaz los recursos, pero con la desventaja que es poco fiable y no seguro
- ✓ **IETF syslog (2009)** – Protocolo que suporta el traslado de la estructura de datos incluida como parte del mensaje, el transporte puede ser basados en paquetes UDP and TCP, permite cifrado y autenticación, además de colocar marcas de tiempo en los mensajes.
- ✓ **CEE (Common Event Expression)**. Estándar de log creado en el 2012, y que hace de formatos JSON para recibir los datos
- ✓ Otros protocolos que no tienen un RFC definido pero que pueden ser usados para el registro de logs tales como BSD syslog over TCP, GELF, SNMP trap messages, etc.

2.6.1. Estructura del estándar syslog

En este proyecto se utiliza syslog por ser un estándar ampliamente utilizado que permite la captura, procesamiento y la transferencia de los eventos de seguridad de un sistema.

En el estándar syslog [5] el mensaje que contiene los log está estructurado por tres campos, que entre todos no pueden tener una longitud mayor de 1024 bytes:

- ✓ El primero de los campos consiste en la **Cabecera** que contiene la información de prioridad del mensaje. Este valor es el resultado de calcular los valores de la facilidad por ocho y sumarle la severidad que cada mensaje tiene (PRI= Facility *8 + Severity), es así que mientras más pequeño sea el valor obtenido mayor será la prioridad. Otra información que contiene este campo son la versión del protocolo, el timestamp del mensaje en formato MMDDHHmmss, hostname (longitud de 255 caracteres como máximo), app-name (nombre del dispositivo o aplicación que origina el mensaje), identificar MSGID para identificar el tipo de mensaje.

Como se puede apreciar el syslog se encarga de recolectar y guardar los eventos en función de dos elementos principales la **Facility** (tipo de origen que genera el mensaje) y **Severidad** (importancia del mensaje)

Tabla 1. Códigos de Facilidad.

Código	Facility (facilidad)
0	Mensajes de kernel
1	Mensajes de nivel de usuario
2	Sistema de correo
3	Demonios del sistema
4	Mensaje de seguridad/autorización[1]
5	Mensaje generado internamente por syslogd
6	Subsistema de impresora en línea
7	Subsistema de noticias de red
8	Subsistema UUCP
9	Demonio de reloj[2]
10	Mensaje de seguridad/autorización[1]
11	Demonio FTP
12	Subsistema NTP
13	Auditoria de eventos[1]
14	Alerta de eventos[1]
15	Demonio de reloj[2]
16	Uso local 0
17	Uso local 1
18	Uso local 2
19	Uso local 3
20	Uso local 4
21	Uso local 5
22	Uso local 6
23	Uso local 7

Tabla 2. Códigos de severidad

Valor	Denominación		Descripción
0	Emergencia	EMERG	Sistema inutilizable
1	Alerta	ALERT	Requiere intervención inmediata
2	Crítico	CRIT	Condición crítica
3	Error	ERR	Condición de error
4	Peligro	WARN	Condición de peligro
5	Aviso	NOTICE	Funcionamiento normal pero con condiciones reseñables
6	Información	INFO	Mensajes informativos
7	Depuración	DEBUG	Mensajes de depuración de bajo nivel

- ✓ Los Datos Estructurados (STRUCTURED-DATA) es el segundo de los campos, donde existe un mecanismo que permite estructurar información para que sea fácil de interpretar, además facilita almacenar meta-información sobre el evento que sea de interés.

- ✓ El tercer elemento es el **Mensaje (MSG),** que consiste en una secuencia de caracteres que puedes ser utilizado por cualquier generador de evento para indicar información de dicho evento, los datos que se ingresan en este campo debe ser de tipo Unicode o según el estando RFC3629.

2.7. Implementación Rsyslog

Rsyslog [6] es un software versátil y robusto que permite el rápido procesamiento de eventos del sistema incluso para grandes entornos empresariales. Se encuentra desarrollado de manera modular lo que permite tener un alto nivel de desempeño con características de seguridad apropiados, permitiendo obtener datos de log desde fuentes variadas, transformación de los datos y la presentación de informes hacia varios tipos de destinos, además de permitir almacenar los registros obtenidos en diferentes gestores de bases de datos.

Las principales características de este software y por el que fue escogido en la implementación de este proyecto son:

- ✓ Soporte para protocolos TCP, UDP, TLS, SSL, RELP
- ✓ Filtrado de cualquier parte de un mensaje en formato syslog.
- ✓ Multi-threading
- ✓ Soporte para almacenar en gestores de base de datos como MySQL, PostgreSQL, Oracle y otros.
- ✓ Permite varios tipos de configuraciones para los reportes de salidas.

2.8. Estandarización de LOGS

Las diferentes fuentes de eventos de seguridad utilizan diferentes formatos para guardar los registros de log generados, por lo que es común encontrar como destino de esos datos archivos separados por coma o tabuladores (CVS), XML, Syslog, json, SNMP, archivos binarios e incluso archivos con formatos propietarios. Estos archivos en algunos casos pueden ser visualizados e interpretados por el personal tecnológico, pero en otros es necesario algún aplicativo específico para su análisis, por lo que es necesario que se estandarice la salida en un solo formato con el número de campos necesarios y suficientes para una posterior interpretación de la misma. En el desarrollo de esta solución se utiliza el esquema definido por syslog y que va hacer estandarizado para usar con todas las fuentes que generen logs.

2.9. Arquitectura centralizada de LOGS

Para una correcta centralización de logs es necesario definir la arquitectura de la solución que se necesita para permitir generar, transmitir, almacenar y analizar los datos guardados, es así que una infraestructura típica incluye los siguientes puntos:

- ✓ **Orígenes o generadores de LOGS.** En este punto encontramos tanto hardware y software que son capaces de generar eventos de seguridad.

- ✓ **Almacenamiento de LOGS.**- Este componente se refiere a los servidores que receptan y que actúan como repositorio o recolectores centrales de los eventos generados en los diferentes orígenes, así tenemos que el almacenamiento puede ser realizado en tiempo real o puede ser ejecutado como procesos batch dependiendo de la estrategia empleada.

  Este ítem también se relaciona con la infraestructura de almacenamiento (NAS o SAN) que una organización posee para guardar dicha información, además si las posibilidades lo permiten se pueden incorporar la utilización de base de datos para el almacenamiento de log.

- ✓ **Herramientas de monitoreo y visualización.**- Una vez almacenado los datos es necesario tener una herramienta que permita de manera gráfica el monitoreo y evaluación de los eventos que están ocurriendo en la organización.

2.10. Identificación de orígenes de LOGS.

Con la finalidad de diferenciar y tener un mejor análisis de la infraestructura a controlar, se procede con la separación de tres tipos principales de orígenes de fuentes de eventos de seguridad (logs), los mismos que son:

- ✓ **Log de Aplicaciones de red y dispositivos de Seguridad.-** Entre los orígenes a controlar tenemos:
  - ✓ Router
  - ✓ Switches
  - ✓ Access Point
  - ✓ Firewall
  - ✓ IDS
  - ✓ IPS
  - ✓ WLC
  - ✓ VPN
  - ✓ Antivirus

- ✓ **Log de Sistemas Operativos.-** Los eventos que se van a recoger en provienen de:
  - ✓ Servidores con sistemas operativos basados en Linux (Distribuciones Centos y Red Hat Enterprise)
  - ✓ Servidores con sistemas operativos Windows. (Versiones Windows 2012 Server)
  - ✓ Hypervisores de plataformas de virtualización (Esxi de VMware)

- ✓ **Log de Aplicaciones.-** En la primer fase de implementación de la propuesta planteada se recolectan log de las siguientes aplicaciones:
  - ✓ Servicio FTP
  - ✓ Servicio SSH
  - ✓ Servicio de correo empresarial
  - ✓ Servicio de Base de Datos

2.11. Definición de sistema de almacenamiento a utilizar.

Uno de los aspectos a considerar en una solución para administración de logs, es la forma de guardar cada evento que ocurre dentro de los dispositivos y aplicaciones que integran la infraestructura tecnológica de una organización, así se tiene las siguientes opciones: almacenamiento local en el dispositivo generador del evento, almacenamiento centralizado en un sistema específico para dicha finalidad o la combinación de ambos.

En el desarrollo de esta solución se escogió el almacenamiento de los eventos en un repositorio centralizado, por lo que se definió dos repositorio para su utilización, el primer repositorio almacena los logs mensuales clasificados según su origen (red o dispositivos de seguridad, sistemas operativos y aplicaciones) y un repositorio centralizado que almacena de forma permanente todos los log que se genera en la infraestructura y que son copiados desde cada uno de los servidores de almacenamientos intermedios que se han definido, esto se realiza con la finalidad de que si se requiere hacer un análisis de actividad sospechosa o auditoria de seguridad del mes actual se lo realice en los servidores intermedios; pero si requiere información más antigua se analiza la información del log centralizado principal.

Para el almacenamiento centralizado se dispone de equipos virtuales con una capacidad de 1TB definidos directamente desde la infraestructura, mientras que para el almacenamiento principal se dispuso un acceso mediante el protocolo NFS de un espacio de 3TB para guardar los registros de eventos de seguridad, valor que puede ser expandido en caso de ser necesario mediante el administrador del sistema de almacenamiento. Ambos repositorios son provistos mediante accesos SAN y NAS respectivamente con la debida redundancia en el acceso y con mecanismos que permiten tener tolerancia a fallas.

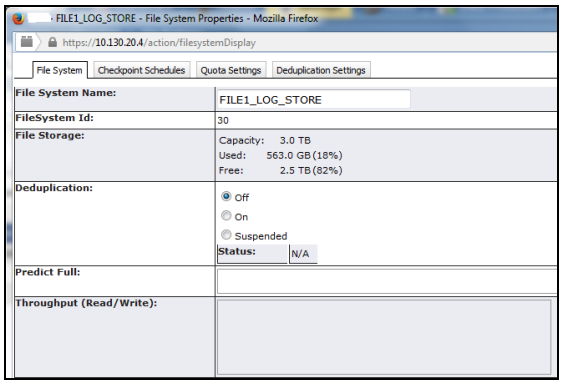


Figura 2.3. Definición del espacio de almacenamiento definido para la solución
Fuente: Autor

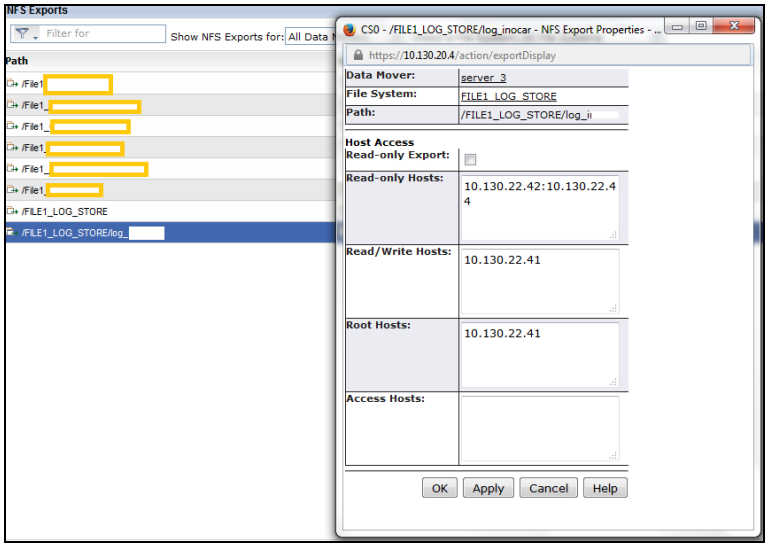


Figura 2.4. Definición del permiso de acceso para el sistema de almacenamiento
Fuente: Autor

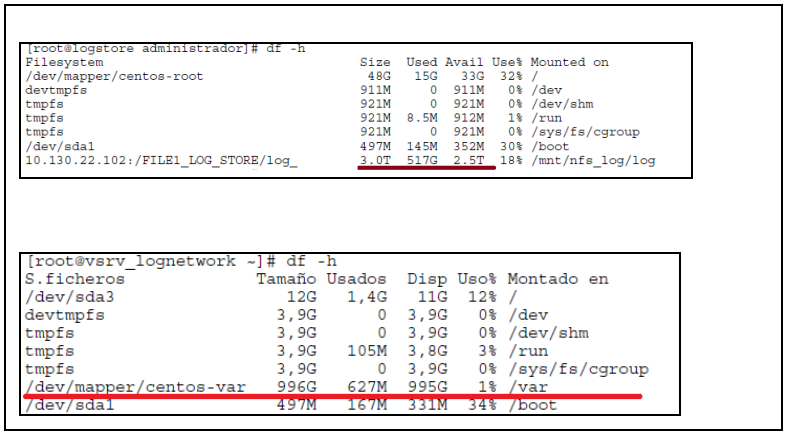


Figura 2.5. Definición del permiso de acceso para el sistema de almacenamiento
Fuente: Autor

2.12. Organización de repositorios centralizado de archivos logs

Para una mejor administración de los archivos recolectados, se estableció una estructura centralizada y clasificada de los log basado en carpetas jerárquicas clasificadas por Año, Mes y Día, tal como se aprecia en la siguiente imagen.

Figura 2.6. Estructura de directorio visualizada desde un navegador.

2.13. Implementación y configuración de recolectores de log centralizados

Toda la estructura necesaria para la administración centralizada de logs se lo realizó en una infraestructura de virtualización donde se agrupa las máquinas virtuales (MV) según su funcionalidad en un concepto llamada VAPPS.

Las máquinas virtuales creadas fueron las siguientes:

- ✓ 3 MV para la administración de log según el origen de los datos (Redes, Sistemas operativos, y Aplicación)
- ✓ 1 MV para base de datos
- ✓ 1 MV para la consola
- ✓ 1 MV de repositorio central principal de los log.

Cada una de las máquinas descritas anteriormente tiene las siguientes características mostradas en las siguientes tablas:

Tabla 3. Requerimientos para servidores de Log según Origen:

Requisito	**Característica implementada**
Procesador	2 Socket – 1 Core por Socket. Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz.
Disco Duro	1 TB
Unidad compartida	3 TB A través de NFS. Modo solo lectura
Memoria	8 GB
Tarjeta de Red	1000 Mbps
Sistema Operativo	Centos 7 64 bits

Tabla 4. Requerimientos para servidor para base de datos

Requisito	**Característica implementada**
Procesador	2 Socket – 1 Core por Socket. Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz.
Disco Duro	1 TB
Memoria	8 GB
Tarjeta de Red	1000 Mbps
Sistema Operativo	Centos 7 64 bits

Tabla 5. Requerimiento máquina virtual para Consola de administración de Log

Requisito	**Característica implementada**
Procesador	1 Socket – 1 Core por Socket. Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz.
Disco Duro	100 GB
Unidad compartida	3 TB A través de NFS. Modo solo lectura
Memoria	4 GB
Tarjeta de Red	1000 Mbps
Sistema Operativo	Centos 7 de 64 bits

Tabla 6. Requerimientos para la máquina virtual de gestión y visualización de log

Requisito	Característica implementada
Procesador	2 Socket – 1 Core por Socket. Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz.
Disco Duro	50 GB
Unidad compartida	3 TB A través de NFS. Modo lectura y escritura.
Memoria	4 GB
Tarjeta de Red	1000 Mbps
Sistema Operativo	Centos 7 de 64 bits

2.13.1. Configuración de Rsyslog como recolectores de log

En los tres equipos configurados para recolectar los eventos según su procedencia (red o dispositivos de seguridad, sistemas operativos o aplicaciones) se necesita instalar y configurar rsyslog como servidor de log, actividad que se realizan configurando los siguientes aspectos:

2.13.1.1. Instalación y verificación de ejecución de Rsyslog.

Lo primero que debemos realizar para poder almacenar los logs es la instalación del paquete rsyslog con el objetivo de que recepte los eventos y en casos de servidores Linux permitan reenviar los eventos a los repositorios centralizados.

Para instalar el paquete principal y ciertos módulos adicionales para permitir almacenar los datos en base de datos Mysql se procede mediante la siguiente instrucción:

#yum -y install rsyslog rsyslog-gnutls rsyslog-gssapi rsyslog-mysql

Una vez instalada el mismo es necesario iniciar el servicio y activar para que el mismo se ejecute al iniciar el sistema operativo, para lo cual se ejecutan las siguientes líneas

Para iniciar el servicio:

#systemctl start rsyslog ***-> Para Centos 7***
#service rsyslog start ***-> Para Centos 6***

Para activar el servicio al iniciar el servicio:

#systemctl enable rsyslog ***-> Para Centos 7***
#chkconfig rsyslog on ***-> Para Centos 6***

2.13.1.2. Creación de un directorio de cache destinados a BD

Con la finalidad de que exista un lugar donde puedan cachear los eventos que se van a almacenar en una base de datos se crea un directorio para este fin ejecutando la siguiente instrucción:

#mkdir –p /var/spool/rsyslog

2.13.1.3. Configuración estándar para recepción de eventos

Para la recepción y personalización de eventos se escoge como referencia el archivo de configuración prevista en el documento desarrollado por el Darío Ortega [6], para lo cual se explicará de forma separada cada parte que se debe configurar en el archivo /etc/rsyslog.conf

a) Configuración de usuario que graba los log

$FileOwner administrador
$FileGroup administrador

b) Activación de puertos para recepción de log. Para permitir la recepción de los eventos de log enviados por otras fuentes es necesario que en los servidores que receptan los mismos se configure en el archivo /etc/rsyslog.con las siguientes líneas:

 # Para recepción UDP syslog
$ModLoad imudp
$UDPServerRun 514

 # Para recepción TCP syslog
$ModLoad imtcp
$InputTCPServerRun 514

c) Activación de soporte para módulo Mysql

$ModLoad ommysql
$ActionOmmysqlServerPort 3306

d) Template para generación de log de orígenes estándares

$template DynFileCron,"/var/log/%HOSTNAME%-exp-cron-%$YEAR%%$MONTH%%$DAY%.log"
$template DynFileMessages,"/var/log/%HOSTNAME%-exp-messages-%$YEAR%%$MONTH%%$DAY%.log"
$template DynFileAuthPriv,"/var/log/%HOSTNAME%-exp-authpriv-%$YEAR%%$MONTH%%$DAY%.log"
$template DynFileMail,"/var/log/%HOSTNAME%-exp-mail-%$YEAR%%$MONTH%%$DAY%.log"
$template DynFileDaemon,"/var/log/%HOSTNAME%-exp-daemon-%$YEAR%%$MONTH%%$DAY%.log"
$template DynFileKern,"/var/log/%HOSTNAME%-exp-kernel-%$YEAR%%$MONTH%%$DAY%.log"

Donde Hostname es el nombre del generador del evento log.

e) Template para generación de log de orígenes específicos

$template DynFileProFTPD,"/var/log/proftpd/%HOSTNAME%-exp-proftpd-%$YEAR%%$MONTH%%$DAY%.log"
$template DynFileSquidError,"/var/log/squid/%HOSTNAME%-exp-squid-error-%$YEAR%%$MONTH%%$DAY%.log"
$template DynFileCisco,"/var/log/%HOSTNAME%-exp-cisco-%$YEAR%%$MONTH%%$DAY%.log"

f) Template para almacenar los log en la base de datos Mysql.

$template MonitorWareMySQLInsert,"insert into SystemEvents (ReceivedAt, DeviceReportedTime, Facility, Priority, FromHost, Message, InfoUnitID, SysLogTag, processid) values ('%timegenerated:::date-mysql%','%timereported:::date-mysql%', %syslogfacility%, %syslogpriority%, '%HOSTNAME:::UPPERCASE%', '%msg%', %iut%, '%programname%', '%PROCID%')",SQL

g) Almacenamiento de log de errores en base de datos

$ActionQueueFileName dbq_error
$ActionResumeRetryCount -1
**.error*
:ommysql:10.130.22.45,RSYSLOGDB_ERRORES,root,password;MonitorWareMySQLInsert

Las opciones que se específica es la dirección IP del servidor de base de datos, nombre de la base de datos, usuario, contraseña y template de ingreso de datos

.

###################################
#Almacenamiento de log de errores en base de datos
###################################

 # Use asynchronous processing
$ActionQueueType LinkedList
$ActionQueueFileName dbq_error
$ActionResumeRetryCount -1

```
*.error
:ommysql:10.130.22.45,RSYSLOGDB_ERROR,root,passw
ord;MonitorWareMySQLInsert

###################################
# The authpriv file has restricted access.
## ########################3
authpriv.* ?DynFileAuthPriv
# Use asynchronous processing
$ActionQueueType LinkedList
$ActionQueueFileName dbq_authpriv
$ActionResumeRetryCount -1
authpriv.*
:ommysql:10.130.22.45,RSYSLOGDB_AUTHPRIV,root,pa
ssword;MonitorWareMySQLInsert



##########################################
#Log all the mail messages in one place.
##########################################
mail.* -?DynFileMail
# Use asynchronous processing
$ActionQueueType LinkedList
$ActionQueueFileName dbq_mail
$ActionResumeRetryCount -1
mail.*
:ommysql:10.130.22.45,RSYSLOGDB_MAIL,root,passwor
d;MonitorWareMySQLInsert
# Log cron stuff
cron.* ?DynFileCron


###################################
## Receive networking messages
#####################################
Local6.* -?DynFileCiscoVPN
## Use asynchronous processing
$ActionQueueType LinkedList
$ActionQueueFileName dbq_local6
$ActionResumeRetryCount -1
Local6.*
:ommysql:10.130.22.45,RSYSLOGDB_NETWORK,root,pa
ssword;MonitorWareMySQLInsert
```

h) Configuración estándar para otros eventos.

Dependiendo del servidor que se quiera recibir los eventos generados se configura los facilitys necesarios para la recepción de eventos, a continuación se especifica un ejemplo para recepción de eventos Windows y para equipos Cisco.

Local4. -?DynFileWinServer*
$ActionQueueType LinkedList
$ActionQueueFileName dbq_local4
$ActionResumeRetryCount -1
*Local4.**
:ommysql:<SERVIDOR_MYSQL>,RSYSLOGDB_WINDOWS_EVENTS,<DB_USER>,<DB_PASS>;MonitorWareMySQLInsert
 # Receive networking messages
Local6. -? DynFileCisco*
$ActionQueueType LinkedList
$ActionQueueFileName dbq_local6
$ActionResumeRetryCount -1
*Local6.**
:ommysql:<SERVIDOR_MYSQL>,RSYSLOGDB_NETWORK,<DB_USER>,<DB_PASS>;MonitorWareMySQLInse

2.14. Implementación y configuración de BD para almacenamiento de logs

Con el objetivo de garantizar los datos de logs generados y tener otra fuente de análisis y correlación de eventos se procede con la instalación de una base de datos Mysql que almacenará la misma

información que se encuentra contenida en los diferentes archivos planos generados.

Para una mejor administración y segmentación de los eventos que ocurren en la infraestructura tecnológica se ha escogido diferenciar el destino de dicha información en diferentes esquemas de bases según lo antes mencionado, por lo que se procedió a crear las siguientes estructuras:

- ✓ **RSYSLOGDB_AUTHPRIV**.- Para almacenar eventos referentes a la seguridad respecto a accesos a servidores, escalamiento de privilegios, entre otros.
- ✓ **RSYSLOGDB_ERROR.-** Para registro de errores que son detectados por algún servicio o sistema en particular.
- ✓ **RSYSLOGDB_MAIL.-** Almacenamiento de registros correspondientes a servidores de correo.
- ✓ **RSYSLOGDB_NETWORK.-** Registro de eventos correspondientes a todo lo referente a los dispositivos de redes y seguridad existentes en el la infraestructura de red.
- ✓ **RSYSLOGDB_WINDOWS_EVENTS.-** Esquema relacionado a los

eventos que puedan presentarse en los servidores que tenga sistemas operativos basados los sistemas Windows de Microsoft.

2.14.1. Instalación de Base de Datos

Para la instalación de la base de datos Mysql en Centos 7, se realizó los siguientes pasos:

```
yum install http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm
yum repolist enabled | grep "mysql.*-community.*"
yum install mysql-community-server
service mysqld status
yum install mysql-server
systemctl start mysqld
systemctl enable mysqld
systemctl status mysqld
```

Con el objetivo de asegurar el acceso al servidor de Base de datos se procede a configurar la contraseña del usuario root, con la siguiente instrucción:

```
# mysqladmin -u root password Nuevo_password
```

Una vez verificada instalada Mysql podemos a ingresar y verificar el contenido de la misma como se muestra en la siguiente imagen.

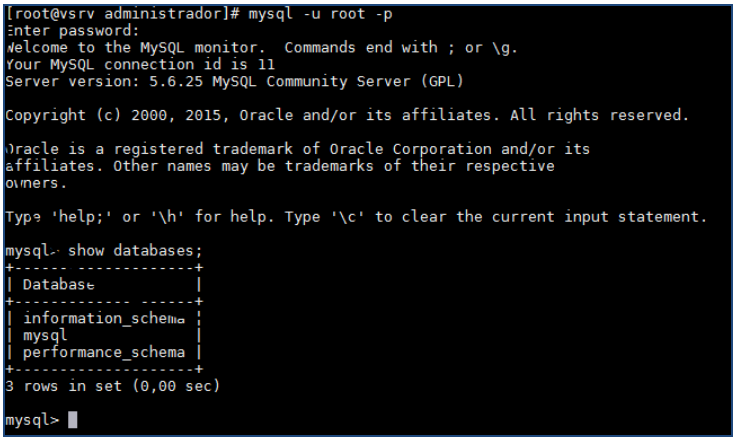
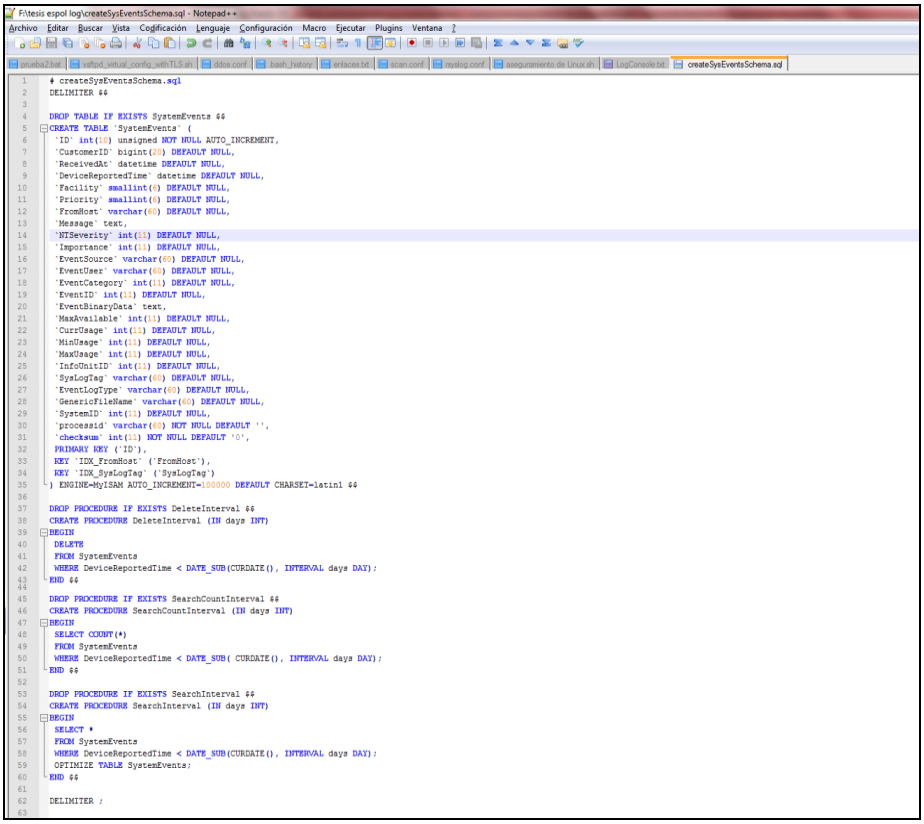


Figura 2.7. Configuración de base de datos Mysql

2.14.2. Esquema de base de datos

Para almacenar los eventos en la base de datos Mysql se va a crear la estructura del esquema sugerido en Rsyslog para cada registro de logs que se requiera almacenar, la misma consiste en la creación de las tablas con sus respectivos campos, y por el motivo de que es una actividad repetitiva se va a crear un script sql para la creación de cada uno de las tablas de las bases de datos previamente definidas tal como se muestra en la siguiente figura.



```
# createSysEventsSchema.sql
DELIMITER $$

DROP TABLE IF EXISTS SystemEvents $$
CREATE TABLE `SystemEvents` (
 `ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
 `CustomerID` bigint(20) DEFAULT NULL,
 `ReceivedAt` datetime DEFAULT NULL,
 `DeviceReportedTime` datetime DEFAULT NULL,
 `Facility` smallint(6) DEFAULT NULL,
 `Priority` smallint(6) DEFAULT NULL,
 `FromHost` varchar(60) DEFAULT NULL,
 `Message` text,
 `NTSeverity` int(11) DEFAULT NULL,
 `Importance` int(11) DEFAULT NULL,
 `EventSource` varchar(60) DEFAULT NULL,
 `EventUser` varchar(60) DEFAULT NULL,
 `EventCategory` int(11) DEFAULT NULL,
 `EventID` int(11) DEFAULT NULL,
 `EventBinaryData` text,
 `MaxAvailable` int(11) DEFAULT NULL,
 `CurrUsage` int(11) DEFAULT NULL,
 `MinUsage` int(11) DEFAULT NULL,
 `MaxUsage` int(11) DEFAULT NULL,
 `InfoUnitID` int(11) DEFAULT NULL,
 `SysLogTag` varchar(60) DEFAULT NULL,
 `EventLogType` varchar(60) DEFAULT NULL,
 `GenericFileName` varchar(60) DEFAULT NULL,
 `SystemID` int(11) DEFAULT NULL,
 `processid` varchar(60) NOT NULL DEFAULT '',
 `checksum` int(11) NOT NULL DEFAULT '0',
 PRIMARY KEY (`ID`),
 KEY `IDX_FromHost` (`FromHost`),
 KEY `IDX_SysLogTag` (`SysLogTag`)
) ENGINE=MyISAM AUTO_INCREMENT=100000 DEFAULT CHARSET=latin1 $$

DROP PROCEDURE IF EXISTS DeleteInterval $$
CREATE PROCEDURE DeleteInterval (IN days INT)
BEGIN
 DELETE
 FROM SystemEvents
 WHERE DeviceReportedTime < DATE_SUB(CURDATE(), INTERVAL days DAY);
END $$

DROP PROCEDURE IF EXISTS SearchCountInterval $$
CREATE PROCEDURE SearchCountInterval (IN days INT)
BEGIN
 SELECT COUNT(*)
 FROM SystemEvents
 WHERE DeviceReportedTime < DATE_SUB( CURDATE(), INTERVAL days DAY);
END $$

DROP PROCEDURE IF EXISTS SearchInterval $$
CREATE PROCEDURE SearchInterval (IN days INT)
BEGIN
 SELECT *
 FROM SystemEvents
 WHERE DeviceReportedTime < DATE_SUB(CURDATE(), INTERVAL days DAY);
 OPTIMIZE TABLE SystemEvents;
END $$

DELIMITER ;
```

Figura 2.8. Script de creación de los esquemas de base de datos

2.14.3. Creación de la Bases de Datos

Antes de crear los esquemas es necesario que previamente estén definidas las respectivas bases de datos de acuerdo a los diferentes tipos de orígenes antes señalados, por lo que se procede con la creación de las mismas, tal como se muestra en la Figura 2.9:

```
mysql> create database RSYSLOGDB_AUTHPRIV;
Query OK, 1 row affected (0,00 sec)

mysql> create database RSYSLOGDB_ERROR
    -> ;
Query OK, 1 row affected (0,00 sec)

mysql> create databases RSYSLOGDB_FTP
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'databases RSYSLOGDB_FTP' at line 1
mysql> create database RSYSLOGDB_FTP;
Query OK, 1 row affected (0,00 sec)

mysql> create database RSYSLOGDB_HTTP;
Query OK, 1 row affected (0,00 sec)

mysql> create database RSYSLOGDB_LOG_CENTRALIZER;
Query OK, 1 row affected (0,00 sec)

mysql> create database RSYSLOGDB_MAIL;
Query OK, 1 row affected (0,00 sec)

mysql> create database RSYSLOGDB_NETWORK;
Query OK, 1 row affected (0,00 sec)

mysql> create database
    ->
    ->
    -> RSYSLOGDB_WINDOWS_EVENTS;
Query OK, 1 row affected (0,00 sec)

mysql> exit
```

Figura 2.9. Creación de esquema de base de datos

Para la verificación de que se crearon correctamente la base de datos se procede como lo indicado en la Figura 2.10.

```
mysql> show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'database' at line 1
mysql> show databases;
+---------------------------+
| Database                  |
+---------------------------+
| information_schema        |
| RSYSLOGDB_AUTHPRIV        |
| RSYSLOGDB_ERROR           |
| RSYSLOGDB_FTP             |
| RSYSLOGDB_HTTP            |
| RSYSLOGDB_LOG_CENTRALIZER |
| RSYSLOGDB_MAIL            |
| RSYSLOGDB_NETWORK         |
| RSYSLOGDB_WINDOWS_EVENTS  |
| mysql                     |
| performance_schema        |
+---------------------------+
11 rows in set (0,00 sec)

mysql>
```

Figura 2.10. Verificación base de datos creados

2.14.4. Creación de esquemas

Una vez creada las bases de datos descritas anteriormente es necesario crear las tablas de cada una de ellas por lo que se va a utilizar el script sql de creación de esquema.

Para realizar esta actividad es necesario indicar nombre de usuario con privilegios para la creación de esquemas, servidor, puerto y nombre de base de datos.

Las instrucciones a ejecutar son las siguientes:

```
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_AUTHPRIV <  createSysEventsSchema.sql
Enter password:
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_ERROR <  createSysEventsSchema.sql
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_ERROR <  createSysEventsSchema.sql
Enter password:
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_FTP <  createSysEventsSchema.sql
Enter password:
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_HTTP <  createSysEventsSchema.sql
Enter password:
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_LOG_CENTRALIZER <
createSysEventsSchema.sql
Enter password:
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_MAIL <  createSysEventsSchema.sql
Enter password:
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_NETWORK <  createSysEventsSchema.sql
Enter password:
[root@vsrv tmp]# mysql -u root -p -h localhost  RSYSLOGDB_WINDOWS_EVENTS <  createSysEventsSchema.sql
```

Figura 2.11. Creación de esquemas en las bases de datos

2.14.5. Verificación de Bases de datos y tablas creadas.

Luego de realizarse el proceso de creación de base de datos con sus tablas desde consola se puede verificar ya sea por medio de comando o por algún administrador gráfico de Base de datos la correspondiente creación de la estructura descrita anteriormente. En este caso se utilizó el software Mysql Workbench para administrar de manera gráfica las bases de datos creadas.

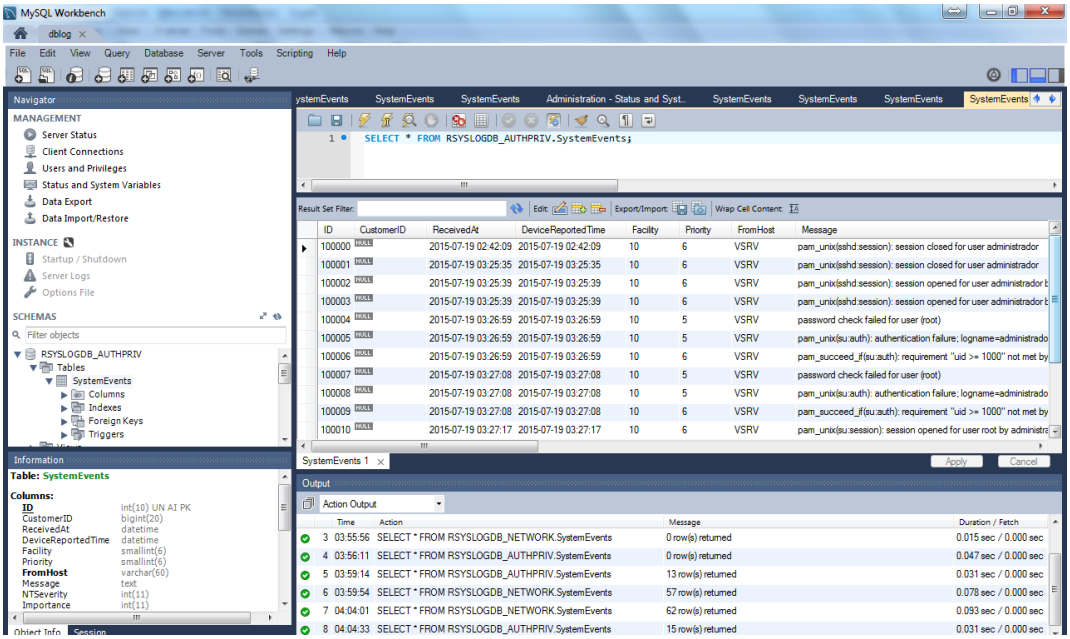


Figura 2.12. Administración de las bases de datos mediante una Interfaz gráfica

2.15. Implementación y configuración de interfaz gráfica para generación de reportes

Para una visualización gráfica de los eventos que se están registrando y centralizando simultáneamente en la base de datos como en el sistema de almacenamiento se procedió a instalar el software LogAnalyzer y el paquete PhpFileTree para la respectiva visualización, ambos programas requiere el uso de un servidor web por esta razón se escogió instalar Apache como servidor web, además de tener instalado Php.

2.15.1. Instalación de Apache

En este proyecto se utilizó Apache 2.4.6-31, que es la versión estable en este momento para el uso en Centos 7, a continuación se puede apreciar la instalación de dicho paquete en la máquina virtual destinada para el uso como consola de administración.

```
[root@vsrv-logconsole administrador]# yum install httpd httpd-devel httpd-manual mod_ssl
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.cedia.org.ec
 * epel: mirror.cedia.org.ec
 * extras: mirror.cedia.org.ec
 * updates: mirror.cedia.org.ec
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete httpd.x86_64 0:2.4.6-31.el7.centos debe ser instalado
--> Procesando dependencias: httpd-tools = 2.4.6-31.el7.centos para el paquete: httpd-2.4.6-
31.el7.cento                s.x86_64
--> Procesando dependencias: /etc/mime.types para el paquete: httpd-2.4.6-31.el7.centos.x86_64
--> Procesando dependencias: libaprutil-1.so.0()(64bit) para el paquete: httpd-2.4.6-
31.el7.centos.x86_6                4
--> Procesando dependencias: libapr-1.so.0()(64bit) para el paquete: httpd-2.4.6-31.el7.centos.x86_64
--> Ejecutando prueba de transacción
---> Paquete apr.x86_64 0:1.4.8-3.el7 debe ser instalado
---> Paquete apr-util.x86_64 0:1.5.2-6.el7 debe ser instalado
---> Paquete httpd-tools.x86_64 0:2.4.6-31.el7.centos debe ser instalado
---> Paquete mailcap.noarch 0:2.1.41-2.el7 debe ser instalado
--> Resolución de dependencias finalizada
Instalado:
  httpd.x86_64 0:2.4.6-31.el7.centos

Dependencia(s) instalada(s):
  apr.x86_64 0:1.4.8-3.el7      apr-util.x86_64 0:1.5.2-6.el7 httpd-tools.x86_64 0:2.4.6-
31.el7.centos
  mailcap.noarch 0:2.1.41-2.el7

¡Listo!
[root@vsrv-logconsole administrador]#
```

Figura 2.13. Instalación de Apache

2.15.2. Instalación de Php

Luego de instalado el paquete de Apache se procede con la instalación de Php y los diferentes módulos necesarios para una correcta ejecución de LogAnalyzer y PhpFileTree

```
[root@vsrv-logconsole administrador]# yum install php-ldap php-cli php php-devel php-common php-mysql
php-pear php-pdo php-xml php-gd
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.cedia.org.ec
 * epel: mirror.cedia.org.ec
 * extras: mirror.cedia.org.ec
 * updates: mirror.cedia.org.ec
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete php.x86_64 0:5.4.16-36.el7_1 debe ser instalado
---> Paquete php-cli.x86_64 0:5.4.16-36.el7_1 debe ser instalado
---> Paquete php-common.x86_64 0:5.4.16-36.el7_1 debe ser instalado
--> Procesando dependencias: libzip.so.2()(64bit) para el paquete: php-common-5.4.16-36.el7_1.x86_64
---> Paquete php-devel.x86_64 0:5.4.16-36.el7_1 debe ser instalado
--> Procesando dependencias: pcre-devel(x86-64) para el paquete: php-devel-5.4.16-36.el7_1.x86_64
--> Procesando dependencias: automake para el paquete: php-devel-5.4.16-36.el7_1.x86_64
--> Procesando dependencias: autoconf para el paquete: php-devel-5.4.16-36.el7_1.x86_64
---> Paquete php-gd.x86_64 0:5.4.16-36.el7_1 debe ser instalado
--> Procesando dependencias: libt1.so.5()(64bit) para el paquete: php-gd-5.4.16-36.el7_1.x86_64
--> Procesando dependencias: libXpm.so.4()(64bit) para el paquete: php-gd-5.4.16-36.el7_1.x86_64
---> Paquete php-ldap.x86_64 0:5.4.16-36.el7_1 debe ser instalado
--> Instalado:
     php.x86_64 0:5.4.16-36.el7_1            php-cli.x86_64 0:5.4.16-36.el7_1      php-common.x86_64
   0:5.4.16-36.el7_1
     php-devel.x86_64 0:5.4.16-36.el7_1      php-gd.x86_64 0:5.4.16-36.el7_1       php-ldap.x86_64
   0:5.4.16-36.el7_1
     php-mysql.x86_64 0:5.4.16-36.el7_1      php-pdo.x86_64 0:5.4.16-36.el7_1      php-pear.noarch
   1:1.9.4-21.el7
     php-xml.x86_64 0:5.4.16-36.el7_1

   Dependencia(s) instalada(s):
     autoconf.noarch 0:2.69-11.el7          automake.noarch 0:1.13.4-3.el7          libXpm.x86_64
   0:3.5.10-5.1.el7
     libxslt.x86_64 0:1.1.28-5.el7          libzip.x86_64 0:0.10.1-8.el7            m4.x86_64 0:1.4.16-
   9.el7
     pcre-devel.x86_64 0:8.32-14.el7        perl-Test-Harness.noarch 0:3.28-2.el7   perl-Thread-
   Queue.noarch 0:3.02-2.el7
     php-process.x86_64 0:5.4.16-36.el7_1   t1lib.x86_64 0:5.1.2-14.el7

   ¡Listo!
   [root@vsrv-logconsole administrador]#
```

Figura 2.14. Instalación de Php

2.15.3. Instalación de LogAnalyzer

Para utilizar LogAnalyzer lo primero que se debe realizar es descargar del sitio oficial el software, luego descomprimirlo y ubicarlo en la carpeta raíz del servidor Apache que para el caso de esta instalación es /var/www/html/logreport.

```
[root@vsrv-logconsole tmp]# wget http://download.adiscon.com/loganalyzer/loganalyzer-3.6.6.tar.gz
--2015-07-23 15:18:45--  http://download.adiscon.com/loganalyzer/loganalyzer-3.6.6.tar.gz
Resolviendo download.adiscon.com (download.adiscon.com)... 176.9.39.152
Conectando con download.adiscon.com (download.adiscon.com)[176.9.39.152]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1047243 (1023K) [application/x-gzip]
Grabando a: "loganalyzer-3.6.6.tar.gz"

100%[=======================================================================>] 1.047.243
89,1KB/s   en 10s

2015-07-23 15:18:56 (101 KB/s) - "loganalyzer-3.6.6.tar.gz" guardado [1047243/1047243]

[root@vsrv-logconsole tmp]# tar -xzf loganalyzer-3.6.6.tar.gz
[root@vsrv-logconsole tmp]# mkdir /var/l
lib/   local/ lock/  log/
[root@vsrv-logconsole tmp]# mkdir /var/www/html/logreport
[root@vsrv-logconsole tmp]# cp -R /tmp/loganalyzer-3.6.6/src/* /var/www/html/logreport/
[root@vsrv-logconsole tmp]# cp -R /tmp/loganalyzer-3.6.6/contrib/* /var/www/html/logreport/
[root@vsrv-logconsole tmp]# cd /var/www/html/logreport/
[root@vsrv-logconsole logreport]# ls
```

Figura 2.15. Instalación de LogAnalyzer

Luego de que se encuentra desempaquetado y ubicado en la carpeta raíz es necesario configurar los archivos configure.sh y secure.sh que se encuentra en la carpeta de instalación con permisos de ejecución para posteriormente proceder a ejecutar el archivo configure.sh. Es necesario indicar que en este momento puede existir algún inconveniente en la instalación con los permisos del archivo config.php si es que se encuentra activado Selinux en el sistema Centos, por lo que es necesario que se deshabilite temporalmente dicha característica o que se configure con los permisos necesarios para su ejecución.

```
[root@vsrv-logconsole logreport]# chmod +x configure.sh secure.sh
[root@vsrv-logconsole logreport]# ls
admin               configure.sh  chartgenerator.php  favicon.ico  install.php  reportgenerator.php
statistics.php
asktheoracle.php    convert.php   details.php         images       js           reports.php
templates
BitstreamVeraFonts  cron          doc                 include      lang         search.php
themes
classes             css           export.php          index.php    login.php    secure.sh
userchange.php
[root@vsrv-logconsole logreport]# ./configure.sh
[root@vsrv-logconsole logreport]# ls
admin               configure.sh        details.php  include      login.php            statistics.php
asktheoracle.php    convert.php         doc          index.php    reportgenerator.php  templates
BitstreamVeraFonts  cron                export.php   install.php  reports.php          themes
classes             css                 favicon.ico  js           search.php           userchange.php
config.php          chartgenerator.php  images       lang         secure.sh
[root@vsrv-logconsole logreport]#
```

Figura 2.16. Configuración de archivo configure.sh de Loganalyzer

Posteriormente se comienza con la instalación del software accediendo mediante un navegador a la ubicación donde se encuentra el archivo install.php del paquete LogAnalyzer instalado en el servidor Centos.

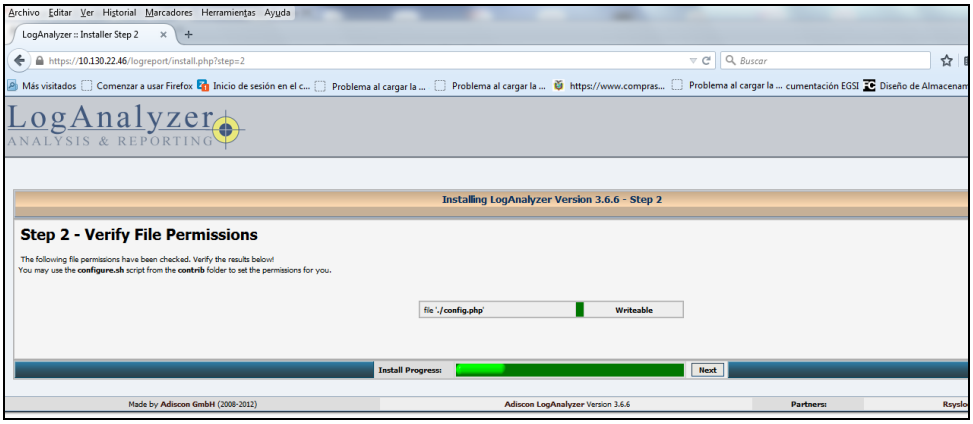


Figura 2.17. Ejecución de asistente de instalación de LogAnalyzer

El primer paso que el asistente verifica es los permisos del archivo config.php que deben ser modificado como se detalló en el párrafo anterior, una vez culminado este paso se continúa con el siguiente actividad que se refiere a la creación de los usuarios que pueden acceder a la interfaz gráfica de administración, para esto se debe crear previamente una base de datos en Mysql que para este caso se denomina LogAnalyzer, y colocar un usuario con permisos para crear esquemas en el gestor de base de datos. En ese momento la aplicación se encarga de crear la correspondiente estructura de las tablas, es necesario indicar que en este apartado también se permite la creación de usuarios integrándolo con usuarios de un Directorio Activo, opción que no se va a utilizar en este caso.

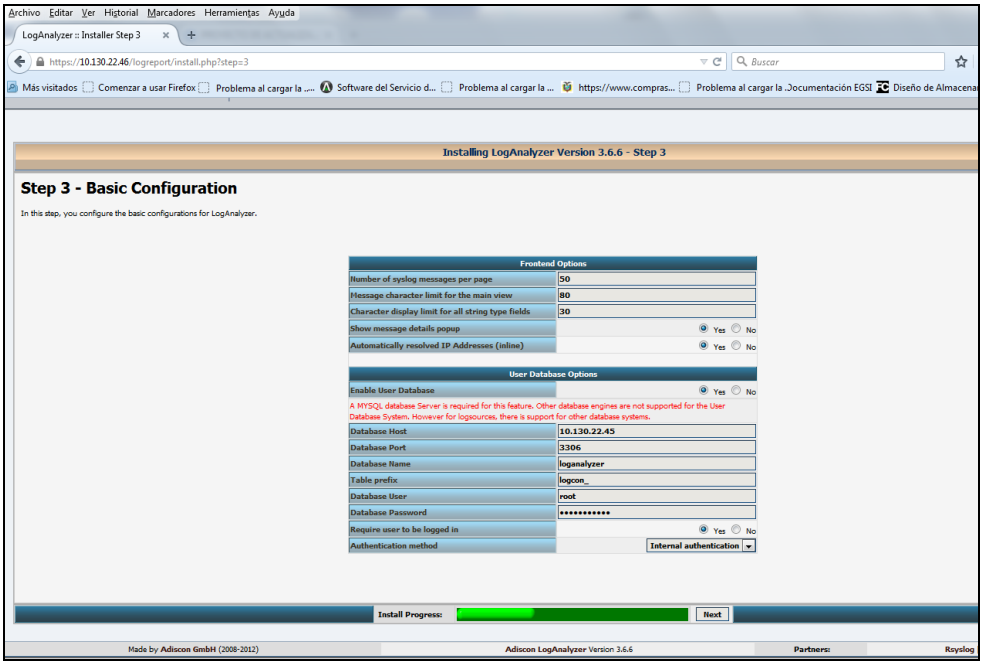


Figura 2.18. Configuración de Base de datos para usuarios de LogAnalyzer

Si todo sale correcto se debe mostrar que las tablas fueron creadas correctamente como se demuestra en la figura 2.19.

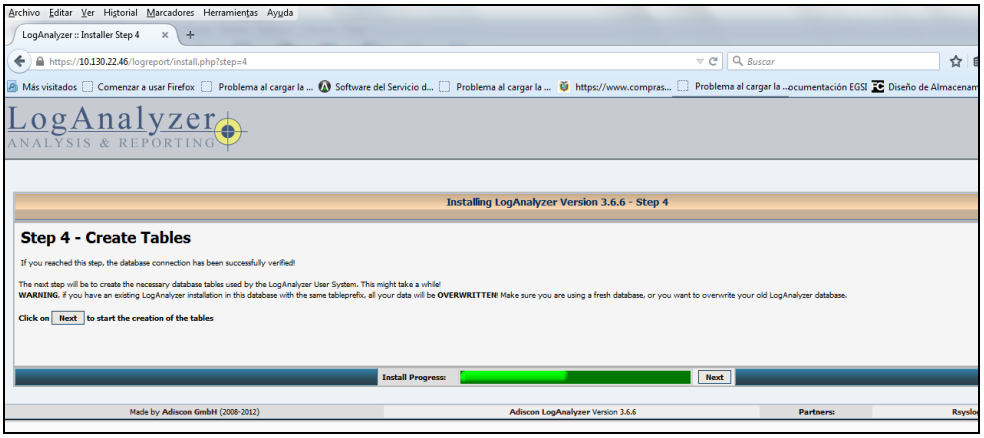


Figura 2.19. Creación de tablas en la base de datos para usuarios de LogAnalyzer

Luego de crearse la base de datos con sus respectivas tablas el asistente permite generar los usuarios que serán almacenados en Mysql y que van a tener permiso de acceso a la interfaz gráfica de LogAnalyzer que para este caso se denominará usuario: adm_log.

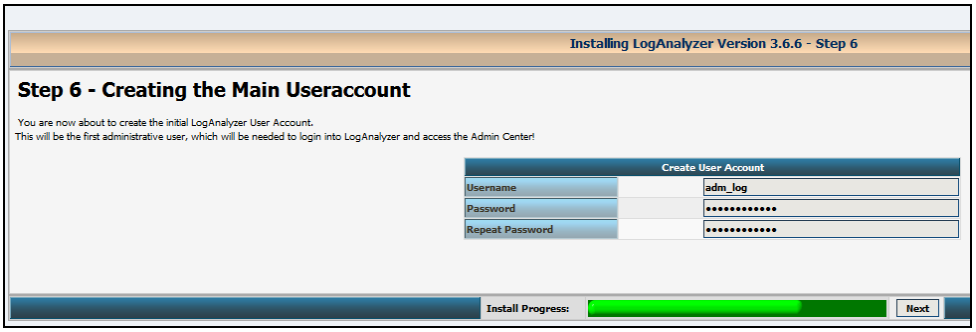


Figura 2. 20. Creación de usuarios para administración de LogAnalyzer

La siguiente actividad que se realiza es la conexión de LogAnalyzer con las bases de datos y tablas que van a almacenar los logs que se generan en la infraestructura, por lo que es necesario incluir la información del nombre del esquema, nombre de la tabla, dirección IP de Mysql y el usuario con permiso para acceder a las mismas como se demuestra en la Figura 2.21

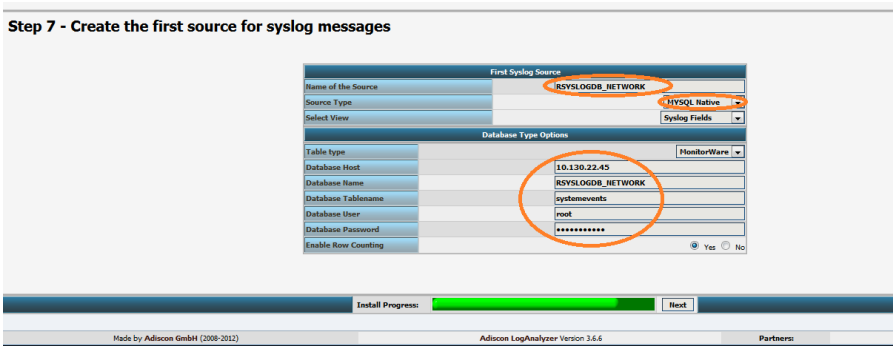


Figura 2. 21. Configuración de acceso de la base de datos que almacenan los logs

Si no existe ningún inconveniente de conexión con la base de datos se mostrará un mensaje indicando que todo fue realizado de manera satisfactoria como se puede apreciar en la figura 2.22

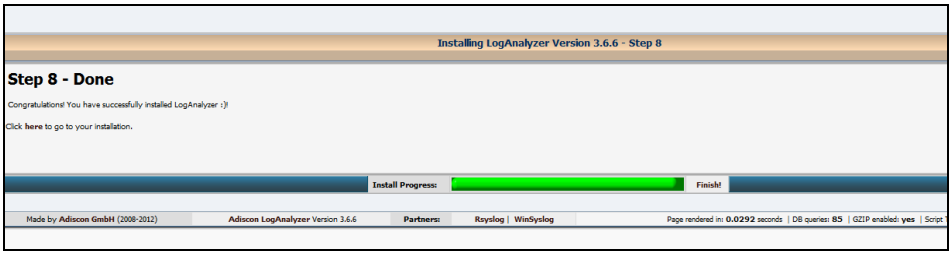


Figura 2. 22. Confirmación de instalación correcta de LogAnalyzer

Una vez culminado satisfactoriamente el proceso de instalación del software se puede dirigir a la página principal del sitio para ingresar con las credenciales creadas en los puntos anteriores, para visualizar la generación de logs que se están almacenando en las bases de datos creadas para este fin.

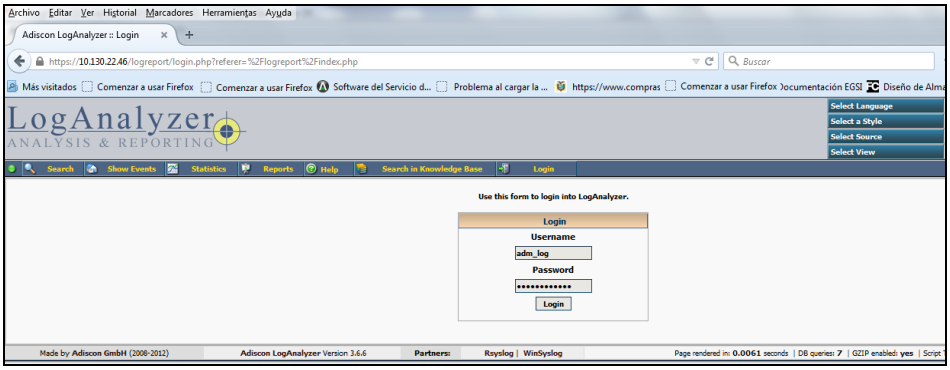


Figura 2. 23. Interfaz de acceso para la administración de LogAnalyzer

En los pasos anteriores solo se conectó a una esquema de base de datos creadas para los diferentes tipos de log generados por lo que para poder conectarse con las demás estructuras es necesario realiza la respectiva conexión con cada una de las BD generadas, esta actividad se la realiza a través del módulo de administración de LogAnalyzer, donde se va ingresar los correspondientes datos de conexión a las base de datos creadas con anterioridad, tal como se puede apreciar en la Figura 2.24

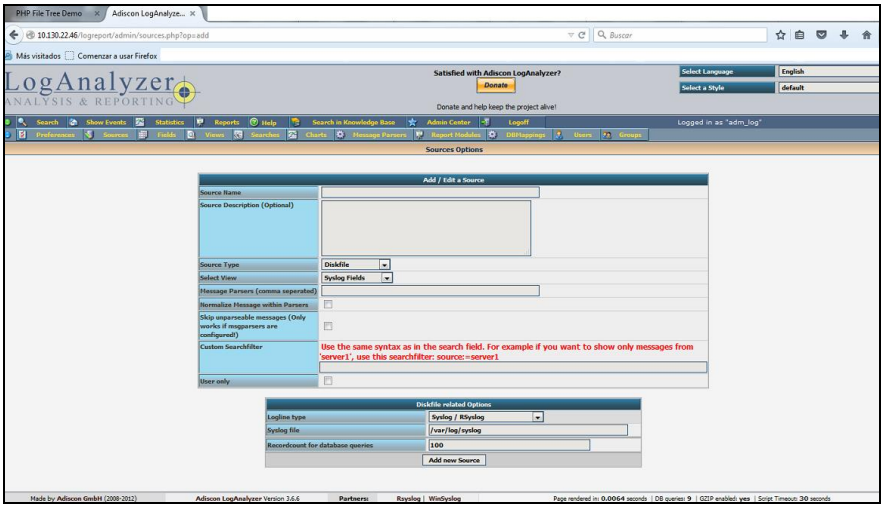


Figura 2. 24. Configuración para acceso de las otras bases de datos de Log

2.15.4. Instalación de PhpFileTree

Con la finalidad de poder apreciar cómo se están almacenando cada uno de los archivos de logs en el repositorio central se procede con la instalación del paquete PhpFileTree, que nos permite mostrar de manera gráfica como está estructurado el directorio del sistema archivo que se desea visualizar en el navegador.

Lo primero que se realiza es descargar el paquete de la página oficial para luego descomprimir y colocar en el directorio por defecto del servidor Apache instalado con anterioridad.

Figura 2. 25. Descarga del paquete phpFileTree

Luego de que se han copiado los archivos de ese paquete en la ruta especificada se puede crear un archivo php donde se coloque la referencia al directorio que se quiere mostrar en el navegador, que en nuestro caso es el punto de montaje del servidor NFS (/mtn/va_log/log que contiene los archivos log guardados)

Archivo de configuración Filelog.php que muestra el contenido de los archives logs guardados en el repositorio central.

```
<?php
// Main function file
// Filelog.php
```

```
include("php_file_tree.php");
?>


<html xmlns="http://www.w3.org/1999/xhtml">

	<head>
		<title>PHP File Tree Demo</title>
		<meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
		<link href="styles/default/default.css" rel="stylesheet" type="text/css" media="screen" />

		<!-- Makes the file tree(s) expand/collapsae dynamically -->
		<script src="jquery-1.3.2" type="text/javascript"></script>
		<script src="php_file_tree_jquery.js" type="text/javascript"></script>
	</head>

	<body>

	<h2>Browing...</h2>

		<?php
		// This displays a JavaScript alert stating which file the user clicked on
		echo php_file_tree("/mnt/nfs_log/log", "javascript:alert('Este es un archivo solo de visualizacion');");
		?>

	</body>

</html>
```

2.16. Configuración de clientes generadores de log

A continuación se procede con la configuración de los diversos clientes generadores de logs para redirigir los registros a los diferentes servidores recolectores configurados para realizar dicha actividad.

2.16.1. Configuración de clientes basados en sistemas operativos Linux

La configuración de los clientes generadores de logs basados en sistemas operativos Linux utilizará la misma plantilla base descrita en el punto 2.13.1.3 de este documento, personalizando aquellas partes necesarias cuando se ejecute algún servicio específico.

Aparte de esto, la única variante que se tiene que activar en el archivo /etc/rsyslog.conf las siguientes líneas dependiendo si es para enviar los mensajes por tcp o udp.

Para udp:

**.* @ipservidorcentral.com*

Para tcp:

**.* @@ ipservidorcentral.com*

2.16.2. Configuración de clientes de dispositivos de redes.

2.16.2.1. Configuración de WLC de Cisco

En el siguiente gráfico se configura un WLC colocando la dirección IP del servidor LOG que almacena estos registros, en este caso la configuración del mismo no presenta ninguna

complicación en su configuración.

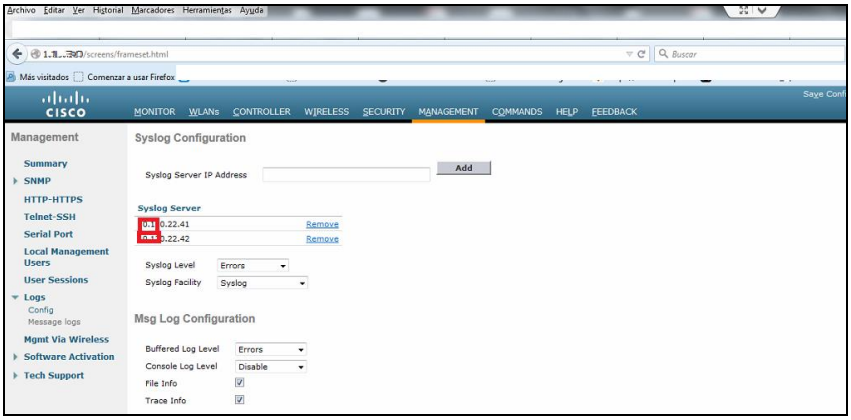


Figura 2.26. Configuración de WLC de Cisco

2.16.2.2. Configuración de Firewall ASA de Cisco

Para configurar el firewall ASA de Cisco primero se habilita el logging en los parámetros básicos, para eso se debe ir al siguiente menú del ADSM a la opción **configuración del registro** ubicada en el submenú **configuración,: características: propiedades: registro**, y luego para habilitar los servidores Syslog externos se activa la casilla de verificación de la opción **registro de permiso**.

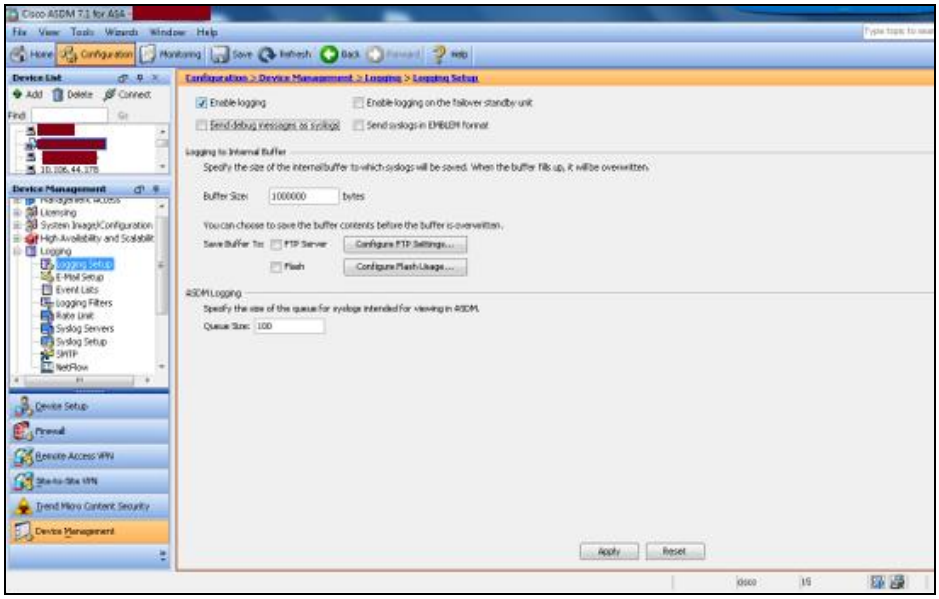


Figura 2.27. Configuración de Parámetros básicos del ASA para activar los logs.

Luego se debe indicar un servidor externo como el destino para los Syslog, para eso elegimos los **servidores de Syslog** en el registro y se escoge add para agregar un servidor de Syslog, a continuación se ingresa los detalles del servidor de Syslog, se escoge el protocolo (TCP/UDP) y se puede modificar el número de puerto en caso de ser necesario, a continuación se de click en **OK** para aceptar la configuración.

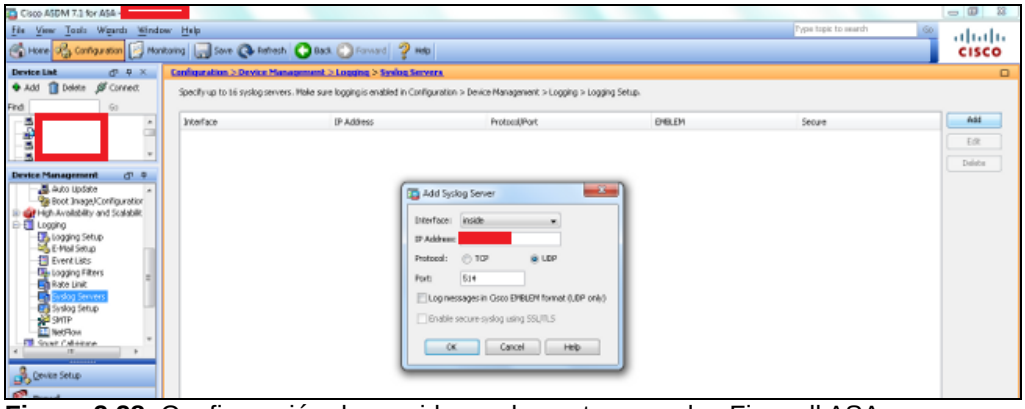


Figura 2.28. Configuración de servidor syslog externo en los Firewall ASA

2.16.2.3. Configuración de switch Cisco

Para configurar un switch Cisco se digita los siguientes comandos:

configure terminal
logging **IpServidorLog**
service timestamps log uptime
logging facility facility-type

2.16.3. Configuración de clientes basados en sistemas operativos ESXI

Para configura un esxi de Vmware tenemos que indicar en la opción Advance System buscar la opción referente a syslog y ubicar la dirección IP del equipo que va a receptar los logs. El acceso a la configuración del esxi se lo realiza a través del Vsphere Client de Vmware.

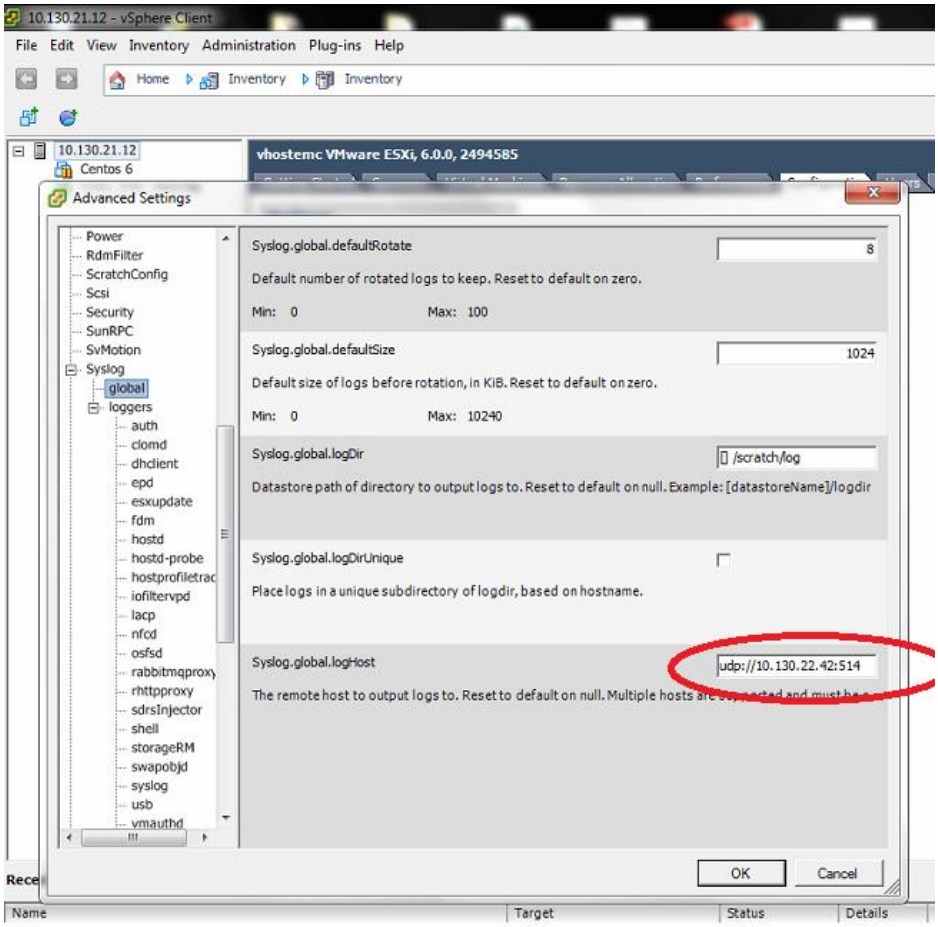


Figura 2. 29. Configuración de servidor syslog externo en los host ESXI

Y el siguiente paso es activar el servicio Syslog en la opción de Security Profiles e iniciar el servicio en caso de que se encuentre detenido.

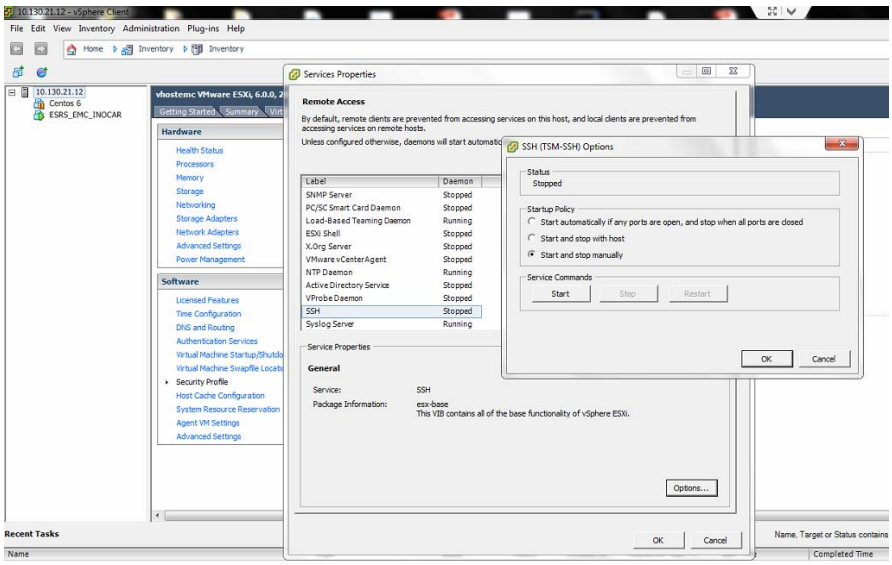


Figura 2. 30. Activación del servicio rsyslog en el ESXI

Uno de los inconvenientes que se encontró cuando se procedió con la activación del syslog en el ESXI es que el archivo log se estaba enviado en un tipo de formato diferente a la norma utilizada, por lo que se debe modificar el archivo config.xml que se encuentra en /etc/vmware/hostd/ para que se envíe en el formato establecido por el estándar de syslog, esta actividad se la realiza mediante un acceso ssh al servidor esxi que se va a configurar y luego se ejecuta el siguiente comando:

#vi /etc/vmware/hostd/config.xml

El archivo de configuración debe quedar así en la sección referente a log:

```
<log>
  <useOldLogPrefix>true</useOldLogPrefix>
```

```
<directory>/var/log/vmware/</directory>
<level>info</level>
<maxFileNum>8</maxFileNum>
<maxFileSize>524288</maxFileSize>
<name>hostd</name>
<outputToConsole>false</outputToConsole>
<outputToFiles>false</outputToFiles>
<outputToSyslog>true</outputToSyslog>
<syslog>
  <facility>local4</facility>
  <ident>Hostd</ident>
  <logHeaderFile>/var/run/vmware/hostdLogHeader.txt</logHeaderFile>
</syslog>
<useOldLogPrefix>true</useOldLogPrefix>
</log>
```

Se debe establecer las siguientes etiquetas:

```
<useOldLogPrefix>true</useOldLogPrefix>
<facility>local4</facility>
 <ident>Hostd</ident>
```

2.16.4. Configuración de clientes basados en SO. Windows

Para enviar los log de Windows se utilizara el paquete rsyslog agent que permite enviar los eventos de Windows en formato log de una manera sencilla.

Lo primero que se realiza es descargar el agente desde la página oficial de Rsyslog.

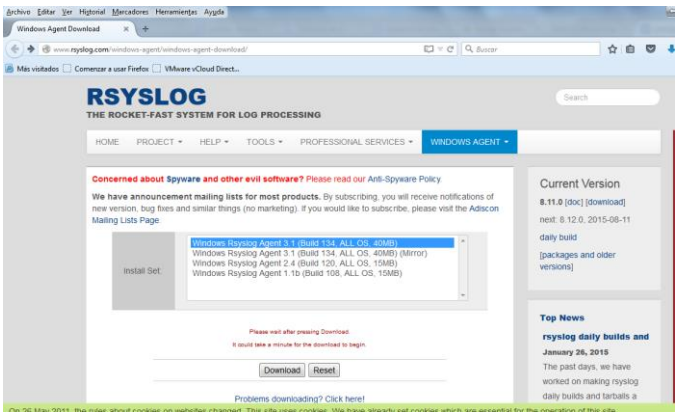


Figura 2. 31. Descarga del agente rsyslog.

Luego se ejecuta el instalador y se deja las versiones por defecto

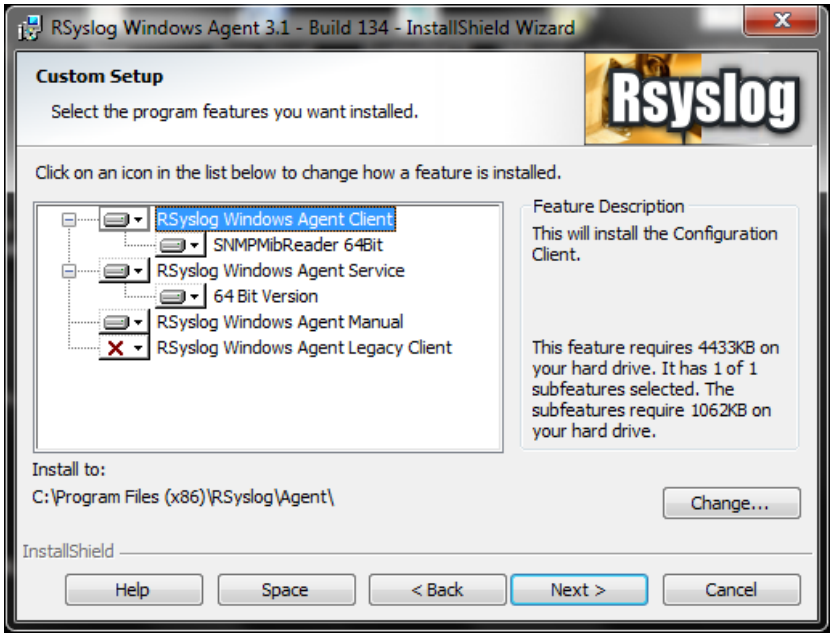


Figura 2.32. Instalación del agente rsyslog en Windows

Una vez instalado el software se lo inicia y se procede con la configuración básica que consiste en indicar el servidor Syslog externo que va a recibir los logs, en la opción Forwarding Syslog.

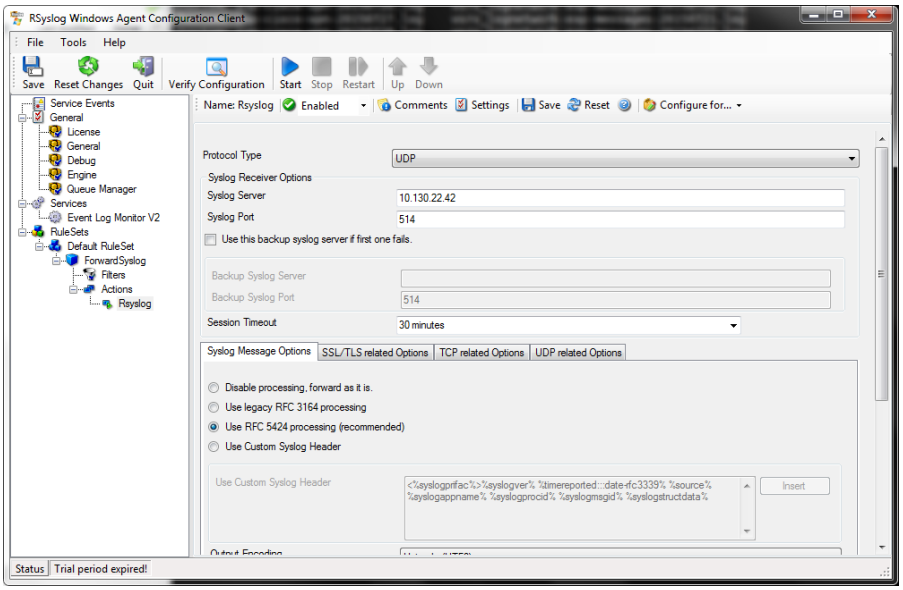


Figura 2.33. Configuración para envío de log a servidor syslog externo en Windows

Para indicar los servicios que van a ser monitoreados en Windows se dirige a la opción service, Event Channels y se escoge los servicios con su respectivo facility y severity, según las necesidades de monitoreo, una vez escogido el mismo se procede a grabar las configuraciones realizadas y se procede a iniciar el agente mediante la opción Start que se encuentra en la parte superior de la aplicación.

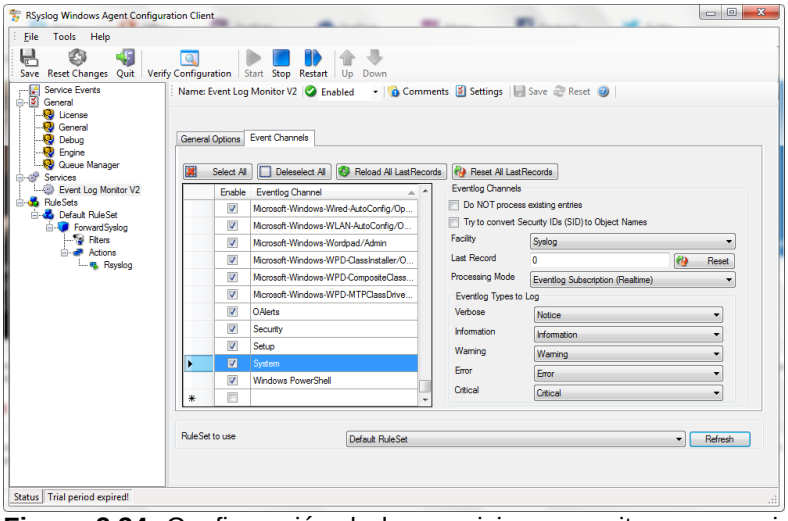


Figura 2.34. Configuración de los servicios a monitorear y enviar log de Windows**.**

2.17. Transferencia de archivos a repositorio central mediante rsync

Con la finalidad de centralizar los archivos en un repositorio central se utiliza la aplicación rsync que por lo general viene instalado en las distribuciones de Linux y que sirve para transferir archivos de manera eficiente permitiendo también la compresión y el cifrado de los ficheros transmitidos.

Otra de la característica de esta aplicación es que permite la sincronización de ficheros y carpetas entre dos o más equipos que tengan conectividad de red, además para asegurar la transmisión de los datos por un canal seguro se puede configurar que la transferencia se la realice utilizando el protocolo ssh como medio de transporte.

2.17.1. Instalación de Rsync

Lo primero que se debe realizar es la instalación del paquete en los almacenamientos intermedios para esto hacemos uso del utilitario yum para proceder con la instalación como se demuestra en la imagen

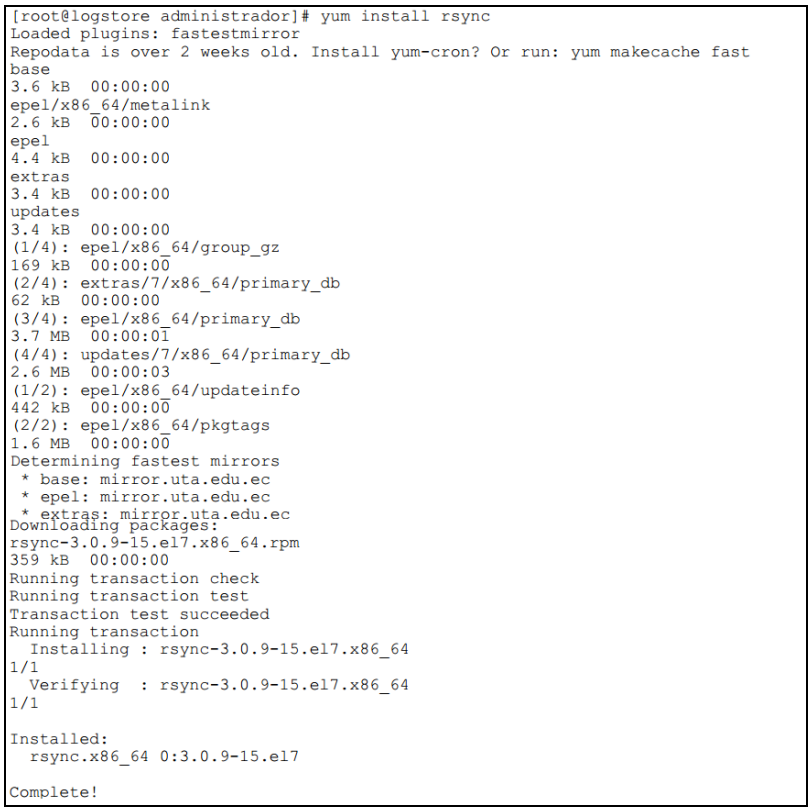


Figura 2. 35. Instalación de Rsync

2.17.2. Script de transferencia de archivos mediante Rsync

Los aspectos principales que se consideraron en el script rsync.sh que se ejecuta como una rutina cron diaria son:

- Compresión de archivo con tar

- Para garantizar la integridad de los archivos generados y luego transferidos la generación de hash mediante el comando ***sha512sum nombre_archivo >>resultado.hash***,, el mismo que generara un hash de cada uno de los archivos que se comprimieron con tar y el resultado va hacer guardado en un archivo diario de extensión .hash

- Y la transferencia de los archivos se la realiza mediante el comando

  rsync -e ssh -avzc /directorio_local administrador@-IPservidorCentral:/ruta_de_almacenamiento.

  En el comando anterior se especifica que la transferencia se la realiza mediante ssh con las credenciales del usuario administrador, en este punto se solicita la contraseña de dicho usuario para proceder con la transferencia respectiva. Para solucionar este inconveniente y que se ejecute como tarea programada, en el apartado 2.15.3 se explicará el uso de claves públicas y privadas de SSH para omitir el ingreso de contraseña, y en lugar de esto utilizar las llaves públicas para la transferencia.

```
#!/bin/bash
    LOG_BASE_DIR=/var/log
    LOG_BASE_TEMP=/var/log/tmp
    RSYNC_MODULE=/mnt/nfs_log/log
    RSYNC_SERVER=10.130.22.41
    if ! [ -d ${LOG_BASE_DIR} ];
    then
        echo "Error! No existe el directorio base ${LOG_BASE_DIR}" > /root/log_rsync.ERORR
        exit -1
     fi
     if [ -x /usr/bin/which ];
     then
        DATE_CMD=`/usr/bin/which --skip-alias date`
        HOSTNAME_CMD=`/usr/bin/which --skip-alias hostname`
        FIND_CMD=`/usr/bin/which --skip-alias find`
        PERL_CMD=`/usr/bin/which --skip-alias perl`
        GZIP_CMD=`/usr/bin/which --skip-alias gzip`
        TAR_CMD=`/usr/bin/which --skip-alias tar`I
        MKDIR_CMD=`/usr/bin/which --skip-alias mkdir`
        MV_CMD=`/usr/bin/which --skip-alias mv`
        LOGGER_CMD=`/usr/bin/which --skip-alias logger`
        SLEEP_CMD=`/usr/bin/which --skip-alias sleep`
        RSYNC_CMD=`/usr/bin/which --skip-alias rsync`
        MD5SUM_CMD=`/usr/bin/which --skip-alias md5sum`
        SHA512SUM_CMD=`/usr/bin/which --skip-alias sha512sum`
        GREP_CMD=`/usr/bin/which --skip-alias grep`
     fi
     if ! [ -x ${DATE_CMD} ];
     then
        echo "date cmd not found. Using default /bin/date."
        if ! [ -x /bin/date ];
        then
            echo "Error! date required" > ${LOG_BASE_DIR}/log_rsync.ERORR
            exit -1 else DATE_CMD="/bin/date"
        fi
     fi

LOG_RSYNC="${LOG_BASE_DIR}/log_rsync-`${DATE_CMD} +"%A"`.log"
LOCAL_UTC=`${DATE_CMD} +"%s"`

echo $LOG_RSYNC

${DATE_CMD} > ${LOG_RSYNC}

if ! [ -x ${HOSTNAME_CMD} ];
then
    echo "hostname cmd not found. Using default /bin/hostname."
    if ! [ -x /bin/hostname ];
    then
        echo "Error! hostname required" >> ${LOG_RSYNC}
        exit -1
    else
        HOSTNAME_CMD="/bin/hostname"
    fi
fi

HOSTNAME=`${HOSTNAME_CMD} -s`

echo "el nombre del equipo es "$HOSTNAME
if ! [ -x ${RSYNC_CMD} ];
then
        echo "rsync cmd not found. Using default /usr/bin/rsync."
        if ! [ -x /usr/bin/rsync ];
        then
            echo "Error! rsync required" >> ${LOG_RSYNC}
            exit -1
        else
            RSYNC_CMD="/usr/bin/rsync"
        fi
    fi
    if ! [ -x ${PERL_CMD} ];
    then
        echo "perl cmd not found. Using default /usr/bin/perl."
        if ! [ -x /usr/bin/perl ];
        then
            echo "Error! perl recommended, using date --date='1 days ago'" >> ${LOG_RSYNC}
            DST_DIR=`date --date='1 days ago' +'%Y/%m/%d'`
                LOCAL_DAY_FILTER=`date --date='1 days ago' +'%Y%m%d'`
                DAY=`date --date='1 days ago' +'d'`
        else
            PERL_CMD="/usr/bin/perl"
        fi
    fi

    if ! [ -x ${FIND_CMD} ];
    then
        echo "find cmd not found. Using default /usr/bin/find."
        if ! [ -x /usr/bin/find ];
        then
            echo "Error! find required" >> ${LOG_RSYNC}
            exit -1 else FIND_CMD="/usr/bin/find"
        fi
        fi

FILE_LIST=`${FIND_CMD} ${LOG_BASE_DIR} -iname "*${LOCAL_DAY_FILTER}.log"`


for FILE in ${FILE_LIST};
do
    echo "Compressing ${FILE}" >> ${LOG_RSYNC}
    echo $FILE
    #${GZIP_CMD} ${FILE} >> ${LOG_RSYNC} 2>> ${LOG_RSYNC}
    ${TAR_CMD} -czvf ${FILE}.tar.gz  ${FILE} >> ${LOG_RSYNC} 2>> ${LOG_RSYNC}
    ${SHA512SUM_CMD} ${FILE}.tar.gz  >> ${HOSTNAME}-sha512sum-daily-${LOCAL_DAY_FILTER}.hash
done


${MKDIR_CMD} ${LOG_BASE_TEMP}/${DAY} >> ${LOG_RSYNC} 2>> ${LOG_RSYNC}
${MV_CMD} *.tar.gz  ${LOG_BASE_TEMP}/${DAY} >> ${LOG_RSYNC} 2>> ${LOG_RSYNC}
${MV_CMD} *.hash  ${LOG_BASE_TEMP}/${DAY} >> ${LOG_RSYNC} 2>> ${LOG_RSYNC}

rsync -e ssh -avzc ${LOG_BASE_TEMP}/${DAY}/ administrador@${RSYNC_SERVER}:${RSYNC_MODULE}/${DST_DIR}
```

Figura 2.36. Archivo de configuración para transferencia rsync.sh

2.17.3. Activación de SSH para transporte de archivos

Para activar SSH como protocolo que permita el transporte de los archivos seguro y que permita la transferencia de los archivos mediante rsync sin tener que ingresar a cada momento las claves de usuario se realiza el siguiente proceso:

- ✓ En la computadora cliente se genera el par de llaves públicas y privadas del usuario que quiere acceder de forma remota mediante el comando ***ssh-keygen***, y luego dejamos por defecto todas las opciones pulsando las teclas enter. Este proceso genera dos archivos en la carpeta de usuario que para nuestro caso es ***/home/administrador/.ssh/id_rsa*** y la pública en ***/home/administrador/.ssh/id_rsa.pub,*** la primera corresponde a la clave privada mientras que la segunda es la clave pública.

- ✓ Una vez generadas las claves es necesarios que en el archivo /***home/administrador/.ssh/authorized_keys*** del computador destino se le copie el contenido de la clave pública generada en el paso anterior.

Luego para verificar el acceso sin la solicitud de clave se digita el comando ***ssh administrador@10.130.22.46*** si es que se realiza la autenticación mediante SSH, caso contrario si se quiere utilizar rsync sobre ssh se digita:

rsync -e ssh -avzc /directorio_local administrador@-IPservidorCentral:/ruta_de_almacenamiento.

2.18. Medidas de seguridad implementadas

Con la finalidad de garantizar la integridad, confiabilidad y disponibilidad de los archivos log almacenados como de la base de datos se configuraron las siguientes medidas de seguridad.

- ✓ Hardening de los sistemas operativos Centos 7 utilizados para la infraestructura de centralización de log diseñada para el efecto.
- ✓ Instalación y configuración de IPTABLES en los sistemas operativos antes mencionados con el objetivo de controlar el acceso a cada uno de las máquinas virtuales, de acuerdo a los servicios que se ejecutan en la maquina estará permitido los accesos ssh, tcp 514, udp 514, mysql, ntp, nfs.

- ✓ Para garantizar que ningún archivo de logs que se encuentra almacenado no ha sido modificado una vez que se ha copiado al repositorio central se genera el hash en sha256 de cada uno de estos ficheros.

- ✓ Acceso limitado de usuarios a los sistemas operativos que integran la solución propuesta.

CAPÍTULO 3

3. ANÁLISIS DE RESULTADOS

3.1 Diagrama de red de la estructura de almacenamiento de log centralizado

Luego de realizarse todas las configuraciones descritas en el capítulo 2 de este documento se puede decir de manera resumida que todos los dispositivos y aplicaciones que generen logs enviaran estos eventos a un servidor centralizado, de manera general la configuración de lo realizado se puede apreciar en la figura 3.1

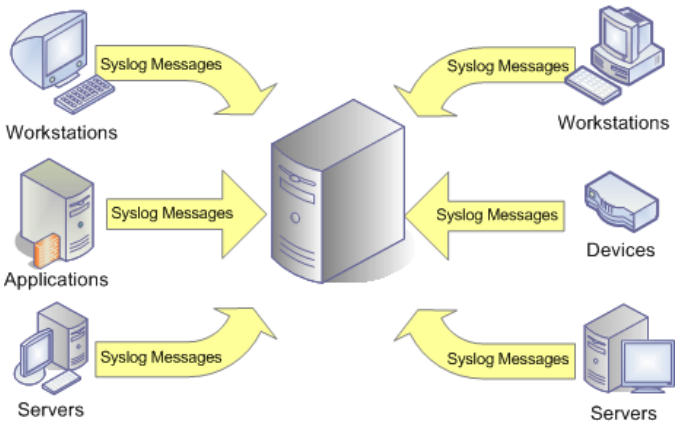


Figura 3.1. Diagrama general de la infraestructura propuesta
Fuente: http://www.malditonerd.com/howto-recibir-logs-remotos-usando-syslog/

La infraestructura que se generó para la implementación de la solución planteada se le desarrolló en un ambiente virtual donde la máquinas virtuales se crearon en un VAPP denominada VAPP_LOG que contiene las maquinas necesarias para la ejecución de este proyecto.

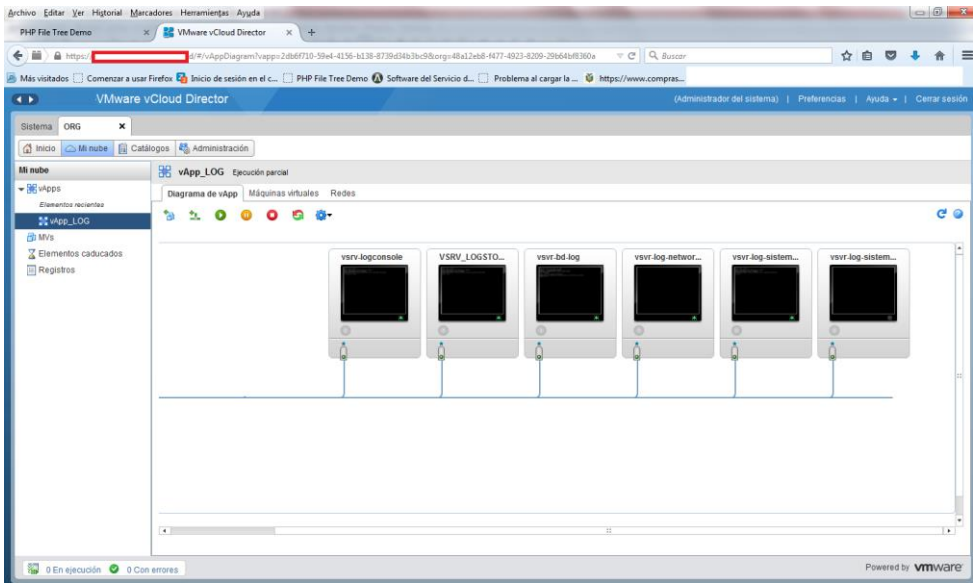


Figura 3. 2. Diagrama de la infraestructura implementada.

3.2. Plan de prueba ejecutado y validado

Para comprobar el correcto funcionamiento de la solución propuesta se procedió a configurar y utilizar los equipos que se detalla en la tabla 7 con la descripción de la función que realizaban en el momento de ejecución de este proyecto.

Tabla 7. Detalles de equipo que intervinieron en el plan de prueba

CARACTERÍSTICAS TÉCNICAS	SISTEMA OPERATIVO	FUNCIÓN	ELEMENTO EN LA ARQUITECTURA DE CENTRALIZACIÓN DE LOG
Máquina virtual	Centos 7	Servidor	Servidor de base de datos para log
Máquina virtual	Centos 7	Servidor	Servidor de centralización de log
Máquina virtual	Centos 7	Servidor	Servidor de recepción de log de sistemas operativos
Máquina virtual	Centos 7	Servidor	Servidor de recepción de log de dispositivos de red y seguridad
Máquina virtual	Centos 7	Servidor	Servidor de recepción de log de aplicaciones
Máquina virtual	Centos 7	Servidor	Servidor para consola de administración de LogAnalyzer y PhpFileTree
Firewall Asa	CISCO IOS	Firewall	Cliente log
Wireless Lan Controlller	Cisco IOS	WLC	Cliente log
Servidor IBM	ESXI	Hypervisor	Cliente log
DELL	Windows	Máquina de desarrollo	Cliente log
Máquina virtual	Centos 7	Servidor	Servidor de FTP

Con la finalidad de validar que toda la solución funcione como se esperaba se procedió a realizar un plan de prueba, el mismo que se detalla a continuación, indicando la actividad así como los resultados esperados y obtenidos.

Tabla 8. Detalle de prueba realizada de la solución

Prueba	Equipo	Resultado (SI/NO/NA)
Sincronización de la hora mediante NTP	Todos	SI
Localización de ficheros a centralizar	Servidor de almacenamiento central	SI
Compresión de los archivos	Servidores intermedios de almacenamiento (redes, sistemas operativos, aplicaciones)	SI
Generación de archivo sha512sum con los hash de los archivos comprimidos	Servidores intermedios de almacenamiento (redes, sistemas operativos, aplicaciones)	SI
Centralización de los archivos al repositorio central	Servidores intermedios de almacenamiento (redes, sistemas operativos, aplicaciones)	SI
Verificación de generación de log de sistemas operativos Linux	Servidores intermedios de almacenamiento (sistemas operativos)	SI
Verificación de generación de log de sistemas operativos Windows	Servidores intermedios de almacenamiento (sistemas operativos)	SI
Verificación de generación de log de dispositivos de redes (firewall y WLC)	Servidores intermedios de almacenamiento (redes)	SI
Verificación de generación de log de aplicación (ftp y ssh)	Servidores intermedios de almacenamiento (aplicaciones)	SI
Verificación de almacenamiento centralizado según la estructura definida	Servidor de almacenamiento central, servidor de consola	SI
Verificación de ingreso de los registros en la base de datos de los log generados	Servidor de base de datos	SI
Verificación del funcionamiento de la consola de administración web basada en LogAnalyzer	Servidor de consola gráfica	SI
Verificación del funcionamiento de la consola de visualización web de ficheros mediante paquete PhpFileTree	Servidor de consola gráfica	SI
Sincronización remota de archivos mediante RSYNC	Servidores intermedios y central de almacenamiento	SI

3.3. Verificación de espacio de almacenamiento de archivos de log clasificados por tipo.

El espacio utilizado en un mes de ejecución del proyecto es de 582G de almacenamiento, tal como se puede apreciar en la figura 3.3.

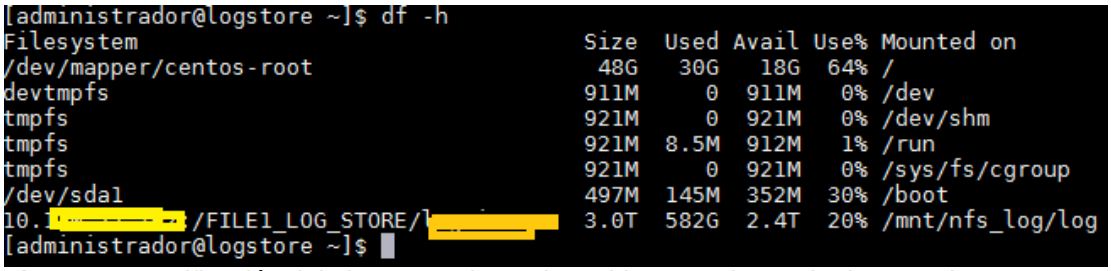


Figura 3.3. Verificación del almacenamiento de archivos en el repositorio central

Es necesario indicar que el consumo promedio de almacenamiento diario de los orígenes de logs es el siguiente:

Tabla 9. Promedio de almacenamiento diario de diferentes fuentes de logs

Origen	Consumo diario
Firewall	25GB
WLC	700 KB
Switch Cisco	890 KB
Servidor Linux	850KB
Equipo Windows	32 MB
ESXI	2,5MB
Aplicación SSH	183KB

3.4. Verificación del almacenamiento de los registros de log en motor de base de datos.

El espacio de almacenamiento utilizado por la base de datos en una semana de ejecución del proyecto es 12MB, además los archivos log (id_logfile0 – id_logfile1) de la base de datos han consumido hasta el momento 96MB tal como se puede apreciar en la figura 3.3.

```
[root@vsrv lib]# cd m
misc/    mlocate/ mysql/
[root@vsrv lib]# cd mysql/
[root@vsrv mysql]# ls -lah
total 109M
drwxr-xr-x. 13 mysql mysql 4,0K jul 23 15:27 .
drwxr-xr-x. 33 root  root  4,0K jul 30 03:44 ..
-rw-rw----.  1 mysql mysql   56 jul 17 12:22 auto.cnf
drwx------.  2 mysql mysql 4,0K jul 23 15:31 db_loganalyzer
-rw-rw----.  1 mysql mysql  12M jul 19 03:53 ibdata1
-rw-rw----.  1 mysql mysql  48M jul 19 03:53 ib_logfile0
-rw-rw----.  1 mysql mysql  48M jul 17 12:22 ib_logfile1
drwx------.  2 mysql mysql 4,0K jul 17 12:22 mysql
srwxrwxrwx.  1 mysql mysql    0 jul 19 03:53 mysql.sock
drwx------.  2 mysql mysql 4,0K jul 17 12:22 performance_schema
drwx------.  2 mysql mysql   88 jul 18 23:54 RSYSLOGDB_AUTHPRIV
drwx------.  2 mysql mysql   88 jul 18 23:55 RSYSLOGDB_ERROR
drwx------.  2 mysql mysql   88 jul 18 23:55 RSYSLOGDB_FTP
drwx------.  2 mysql mysql   88 jul 18 23:55 RSYSLOGDB_HTTP
drwx------.  2 mysql mysql   88 jul 18 23:56 RSYSLOGDB_LOG_CENTRALIZER
drwx------.  2 mysql mysql   88 jul 18 23:56 RSYSLOGDB_MAIL
drwx------.  2 mysql mysql   88 jul 18 23:56 RSYSLOGDB_NETWORK
drwx------.  2 mysql mysql   88 jul 18 23:57 RSYSLOGDB_WINDOWS_EVENTS
[root@vsrv mysql]#
```

Figura 3.4. Verificación del almacenamiento en la base de datos.

3.5. Funcionamiento de interfaz gráfica para verificación de log.

Para verificar el funcionamiento correcto de la aplicación, se ingresa mediante el navegador web al sitio configurado por defecto con las respectivas credenciales y se verifica que los registros han sido ingresados sin inconveniente y que se está realizando la actualización constante con la información generada.

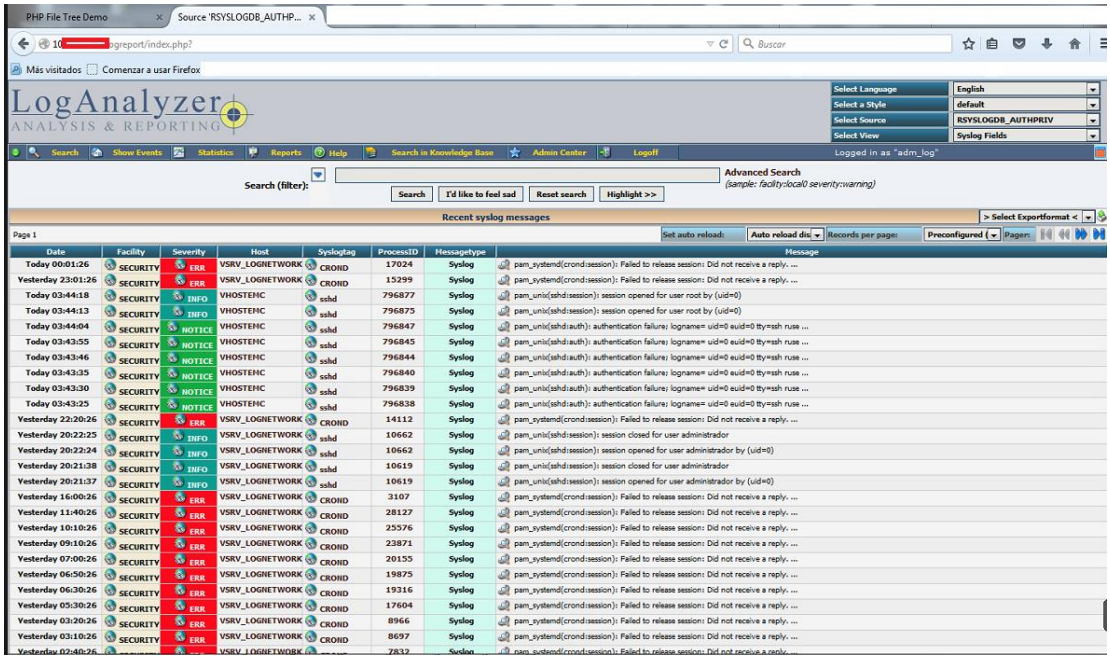


Figura 3.5. Verificación por medio de LogAnalyzer de los registros almacenados en la base de datos

Además para verificar que los archivos se encuentra enviados en el almacenamiento centralizado se visualiza los ficheros mediante el acceso a la dirección URL definida para la utilización de PhpFileTree

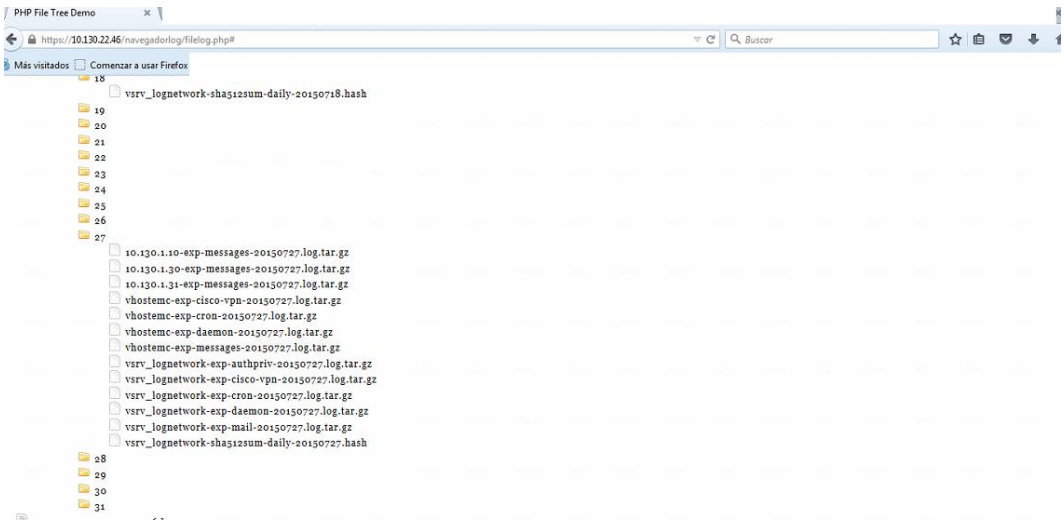


Figura 3.6. Verificación por medio de phpFileTree de los registros almacenados en el sistema de almacenamiento.

CONCLUSIONES

1. Una correcta administración de los logs generados en una infraestructura tecnológica permite una gestión adecuada de los registros de la seguridad informática, permitiendo el cumplimiento de los requerimientos institucionales así como obtener métricas de usos e incluso orígenes de algún error o ataque.

2. El uso de software libre permite que se administre de forma centralizada los log a un bajo costo económico sin perder la eficiencia y eficacia de la solución propuesta, así como la calidad de los registros almacenados en comparación con herramientas comerciales costosas, por lo que lo descrito en este proyecto servirá como un aporte para las instituciones que tienen recursos económicas para inversión en el área tecnológica.

3. La utilización de protocolos estándares y la homologación de los registros que se generan, permiten tener una consolidación y correlación de los eventos ocurridos en un único punto central.

4. El uso de un almacenamiento centralizado, registro en base de datos, hash, compresión y cifrado de datos, transferencia de datos por canales seguros aumentan la confiabilidad e integridad de los logs permitiendo ser utilizados como medio de prueba en casos de incidentes informáticos.

5. Si en una infraestructura tecnológica no existe un mecanismo de sincronización de tiempo (NTP) el almacenamiento de los eventos de seguridad perdería su importancia, por el motivo de que no se podría establecer una línea de tiempo confiable de lo que está ocurriendo con cada uno de las diferentes fuentes de logs de una organización.

6. El éxito de la implementación centralizada de logs no se logra únicamente con la correcta instalación y configuración de los software descritos, por lo que es necesario la implementación de políticas de seguridad institucional bien definidas e implementadas.

7. La centralización de log permite gestionar de una manera eficiente y eficaz todos los eventos de seguridad que ocurren de una infraestructura tecnológica con la finalidad de evaluar y analizar algún incidente informático.

RECOMENDACIONES

1. En la propuesta implementada se generó una esquema de inicial para una solución centralizada para la gestión y administración de logs, el siguiente aspecto a considerar es la integración de lo propuesto con herramientas forense que permitan generar y recrear toda la línea de acción que se hubiera presentado en un caso particular.

2. Adaptar la solución propuesta a las necesidades particulares de cada de una las instituciones con la finalidad de obtener el mejor provecho de las herramientas descritas en el trabajo permitiendo mejorar los controles de seguridad, y, que estos permitan mejorar los procesos de negocios.

3. Es necesario promover el uso de software libre como una forma alternativa frente a los software propietario permitiendo de esta manera tener soluciones eficientes con un valor económico alcanzable para las empresas que tiene poco recurso financiero para implementar soluciones costosas.

4. Al utilizar estándares abiertos para la recolección de log la solución planteada puede ser utilizada para registrar eventos de seguridad de aplicaciones realizadas por desarrolladores propios de una empresa, necesitando solo almacenar los registros según el estándar empleado.

5. Con la finalidad de mejorar e incrementar la seguridad de todos los registros que se generen es muy recomendable seguir evaluando las propiedades y características de los software utilizados y que no han fueron implementadas en este proyecto.

BIBLIOGRAFÍA

[1] Ken, Karen. Souppaya, Murugiah. Guide for Computer Security Log Management. http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf, fecha de consulta julio del 2015

[2] Free Software Foundation's. What is the Free Software. http://www.gnu.org/philosophy/free-sw.en.html#mission-statement. Fecha de consulta julio del 2015

[3] Niño, Diana; Sierra, Alejandro. Guía metodológica para la gestión centralizada de registro de eventos de seguridad en pymes. http://pegasus.javeriana.edu.co/~regisegu/Docs/documento.pdf. Fecha de consulta julio del 2015

[4] Varandi, Risto. Log Management with open-Source tools. http://www.eisay.ee/vvfiles/0/RistoVaarandi.pdf. Fecha de consulta julio del 2015

[5] Gómez, Juan. Servidor de Logs Centralizado, https://riunet.upv.es/bitstream/handle/10251/43428/Memoria.pdf?sequence=1 . Fecha de consulta julio del 2015

[6] Barrios, Joel. Configuración de Rsyslog. http://www.alcancelibre.org/staticpages/index.php/configuracion-rsyslog. Fecha de consulta julio del 2015

[7] Ortega, Darío. Registro, Centralización y Análisis de Eventos en un entorno Corporativo Multiplataforma. http://openaccess.uoc.edu/webapps/o2/bitstream/10609/22750/6/iortegavTFC0313memoria.pdf Fecha de consulta: julio del 2015.