UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
INGENIERO DE SISTEMAS

TEMA:
DESARROLLO DE UNA APLICACIÓN PARA WINDOWS DE TÉCNICAS DE HARDENING SOBRE SISTEMAS OPERATIVOS LINUX PARA LA OBTENCIÓN DE REPORTES

AUTOR:
JOSÉ OSWALDO BENÍTEZ BUENAÑO

DIRECTOR:
CALDERÓN HINOJOSA XAVIER ALEXANDER

Quito, febrero del 2015

DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE GRADO

Yo, JOSÉ OSWALDO BENÍTEZ BUENAÑO, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además declaro que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

_____

José Oswaldo Benítez Buenaño

CC 1717303125

DEDICATORIA

Esta meta muy anhelada se la dedico a mis padres y mi abuela Georgina que con mucho amor siempre me estuvo apoyando, a mis padres que con su ejemplo me impulsaron a seguir siempre adelante. A mis familiares, amigos y profesores por sus consejos que nunca faltaron.

José Oswaldo Benítez Buenaño

AGRADECIMIENTO

A todos los que conforman la Universidad Politécnica Salesiana, por inculcar el conocimiento y afecto por la profesión.

También un profundo agradecimiento a mi director de tesis y a los profesores que aportaron con sus consejos para el desarrollo del presente trabajo de grado.

José Oswaldo Benítez Buenaño

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El presente trabajo de grado está enfocado a la auditoria informática de servidores GNU/Linux y su aporte es facilitar el trabajo a un administrador de sistemas operativos, para optimizar las tareas de identificar vulnerabilidades mediante la ejecución de comandos necesarios que reflejen la información necesaria, para posteriormente analizarla, y presentarla conjuntamente con recomendaciones de como mitigar los riesgos del servidor, en el contenido teórico del presente proyecto se detalla las sugerencias de cada tipo de análisis

Aplicando las técnicas de HARDENING en lo que respecta permisos, configuraciones y revisión de los LOGS, se ha identificado el conjunto de información necesaria para poder identificar las vulnerabilidades del servidor.

Para este desarrollo se ha utilizado la tecnología de VB.net, librerías nativas de Microsoft y otras adicionales para el manejo del protocolo SSH, también se hace uso de Microsoft Word para la obtención de informes generales y la carga de las recomendaciones en cada tipo de análisis.

La aplicación permite realizar la auditoría a 4 servidores simultáneamente estableciendo conexiones síncronas y es compatible con las distribuciones Centos, OpenSUSE, Ubuntu, Fedora y las derivadas de las mencionadas, se ha personalizado en análisis para cada distribución debido a las diferencias de los directorios y nombres de los archivos de configuración, por consiguiente se llega finalmente a un informe general de extensión .docx donde se tiene las recomendaciones y resultados de los análisis que se han seleccionado.

Se escogió la metodología SCRUM para el desarrollo del proyecto en consecuencia se detallan las actividades y con sus respectivos tiempos y pruebas documentados.

ABSTRACT

This paper grade is focused on computer audit of GNU/Linux servers and your contribution is to facilitate the work to a manager operating systems to optimize the tasks of identifying vulnerabilities through of executing commands necessary to reflect the necessary information for later analyze, and present it along with recommendations on how to mitigate risks server in the theoretical content of this project suggestions for each type of analysis is detailed.

Applying techniques HARDENING regarding permissions, settings and review of LOGS, has identified the set of information needed to identify vulnerabilities in the server.

For this development has been used VB.net technology, native libraries from Microsoft and additional for managing SSH protocol, using Microsoft Word for obtaining general reports and the burden of the recommendations in each type is also made analysis.

The application allows the audit to 4 servers simultaneously establishing synchronous connections and is compatible with the distributions, Centos, OpenSUSE, Ubuntu, Fedora and those derived from the above, is customized analysis for each distribution due to differences in the directories and names configuration files, therefore it finally comes to a general report in *.docx* extension where you have the recommendations and results of analyzes that have been selected.

The SCRUM methodology was chosen for the project accordingly detailed activities and their respective times and documented evidence.

INTRODUCCIÓN

De la seguridad informática se deriva la seguridad de los servidores GNU/Linux, que agrupan conceptos, técnicas y procedimientos llamados HARDENING, que nos ayudan a mitigar los riesgos de seguridad, realizar todos estos procedimientos es complejo y es donde nace la necesidad para el administrador de sistemas operativos de una herramienta para la identificación de vulnerabilidades, donde se tenga las recomendaciones de cada análisis en un informe general de cada sistema operativo.

Para auditar un servidor GNU/Linux es necesario conocer acerca de los tipos de análisis involucrados en el HARDENING a realizarse con sus correspondientes recomendaciones que se verán en el primer capítulo.

Para el desarrollo del software se ha escogido la metodología SCRUM la que nos ayuda a detallar las actividades secuenciales para la construcción del software. En el diseño de la arquitectura del software se has creado diagramas *UML (Lenguaje Unificado de Modelado)* necesarios para entendimiento de la arquitectura se puede visualizar en la fase descritos en el segundo capítulo.

La codificación se ha realizado con estándares y criterios basados en los diagramas *UML (Lenguaje Unificado de Modelado)* que describen la arquitectura, se detallan las librerías utilizadas el código fuente generado, diagramas de implementación y pruebas de la aplicación en el tercer capítulo.

En la fase de soporte se detalla conjuntamente con los requerimientos del software y las configuraciones para el ambiente de desarrollo que se ha implementado en el presente trabajo de grado, las configuraciones de las máquinas virtuales así como también los documentos de historiales, detalles y aprobaciones que exige la metodología SCRUM en el capítulo 4.

La aplicación es una herramienta para el administrador de sistemas operativos para la identificación inmediata de vulnerabilidades, que se complementa con las recomendaciones correspondientes en cada análisis convirtiéndose en una guía para corregir y reforzar la seguridad en el servidor, haciendo más difícil la labor del atacante y evitar consecuencias por un inminente incidente de seguridad.

CAPÍTULO 1

INTRODUCCIÓN Y CONCEPTOS

1.1 Diagnóstico de la situación

En la mayoría de servidores se configura los servicios y aplicaciones sin considerar los huecos de seguridad que pueden dejar, al no modificar las configuraciones por defecto de los sistemas operativos. Es común ver que no protegen adecuadamente sus activos de información, razón por la que, los administradores de sistemas operativos deben garantizar la protección de las aplicaciones, configuraciones, permisos e información sobre ellos, evaluando los servidores uno por uno cada periodo de tiempo y en muchos casos sin saber qué información relevante se debe obtener.

Son diversas las estrategias defensivas que salvaguarda a los servidores contra los ataques informáticos, la evaluación de arquitectura de seguridad de una empresa y la auditoría de la configuración de sus sistemas, son los más importantes y generales que se consideran con el fin de desarrollar e implementar procedimientos de consolidación para asegurar sus recursos críticos.

En la actualidad los huecos de seguridad están en los empleados de la misma empresa que por falta de políticas de seguridad internas, son vulnerables a ataques, otra amenaza de seguridad son los hackers informáticos que constantemente actualizan sus herramientas que son cada vez más sofisticados, obligando a las empresas implementar planes de contingencia de prevención y que sus sistemas estén siempre al día para defenderse de posibles ataques con una adecuada configuración de un sistema operativo y un cuidado minucioso de cada uno de los aspectos más básicos como son los permisos y configuraciones base, se puede proteger del robo de la propiedad intelectual, la malversación de información de clientes y sobre todo el escamoteo de contraseñas.

“GNU/Linux es uno de los sistemas operativos más extendidos, principalmente en el ámbito empresarial actuando como servidores para diferentes tipos de toles. Su principal atractivo suele ser su alto grado de configuración y flexibilidad que ofrece a los administradores.” (Álvarez Martín & Gonzales Pérez, 2013, pág. 11)

1.2 Objetivos

1.2.1 Objetivo general

Analizar, diseñar y desarrollar una aplicación, para Windows, de técnicas de HARDENING sobre sistemas operativos GNU/Linux para la obtención de reportes.

1.2.2 Objetivos específicos

- Diseñar la arquitectura de la aplicación
- Desarrollar un módulo que maneje todas las conexiones síncronas y el protocolo ssh, con la capacidad de almacenar los comandos en una cola fifo y envié al server los comandos, para la ejecución de los mismos.
- Desarrollar un módulo de comunicación que contenga las consolas de conexiones a los servidores, será la capa de comunicación entre los servidores y el cliente, se podrá conectarse hasta 4 servidores.
- Desarrollar un módulo para la configuración de la auditoría donde se selecciona los tipos de análisis a realizar para los 4 servidores.
- Desarrollar un módulo para la obtención de reportes y recomendaciones para cada conexión basados en técnicas de HARDENING, genera un informe general.
- Realizar las pruebas del software.

1.3 Justificación

> En la actualidad la seguridad informática en las empresas es cada vez un aspecto más crítico en la gestión TI. El robo de información confidencial por parte de un usuario sin acceso a dichos datos, la denegación de un servicio, la suplantación de una identidad o la destrucción de la información de la empresa son algunos riesgos a los que día a día el administrador enfrenta. (Álvarez Martín & Gonzales Pérez, 2013, pág. 13).

En las empresas es muy complejo garantizar la seguridad informática, porque existen debilidades en los sistemas operativos, siendo los usuarios también un riesgo debido a la falta de políticas o conocimientos de seguridad informática.

Cada empresa tiene al menos un servidor que se considera un recurso crítico y son muy escasas las herramientas que faciliten la identificación de vulnerabilidades para

los administradores de sistemas operativos, que anualmente tienen que reducir los riesgos identificando puertas abiertas de seguridad.

> La mejora de la seguridad en los sistemas TI es una de las máximas a la que se debe optar en un entorno corporativo. Es por ello que, habitualmente, se deben realizar procesos de fortificación de sistemas. Además, la ejecución de test de intrusión que comprueben hasta donde se puede llegar y que se puede obtener. Los test de intrusión forman parte de las auditorías de seguridad informática. (Álvarez Martín & Gonzales Pérez, 2013, pág. 13)

Estas técnicas de HARDENING se las ejecuta revisando los archivos de configuración y permisos de cada servidor tomando en cuenta que los directorios o los nombres de los archivos de configuración varían dependiendo de la distribución, es por eso la necesidad de aplicación que permita ejecutar rápidamente estos análisis optimizando el tiempo, por consiguiente poder detectar las vulnerabilidades por medio de reportes.

1.4 Marco teórico

El marco teórico que fundamenta esta investigación, se describen tres cosas principales para entender la evolución de este proyecto.

a) Conceptos básicos.
b) Introducción a seguridad informática HARDENING.
c) Metodología de desarrollo SCRUM.

1.4.1 Conceptos básicos

a) Protocolo SSH (Secure Shell)

En español intérprete de órdenes segura" es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, además de la conexión a otros dispositivos, SSH nos permite copiar datos de forma segura, y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado. Este protocolo tiene algoritmos de negociación de parámetros para la seguridad y encriptación de la información transferida, establece llaves de seguridad. Actualmente hay una nueva versión SSH 2 que tiene llaves de seguridad más robustas que la anterior versión, esta aplicación soporta SSH 1 y SSH 2.

b) Thread (Hilos de ejecución)

Un hilo de ejecución o thread, en sistemas operativos, es una característica que permite a una aplicación realizar varias tareas concurrentemente. Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

Todas las aplicaciones se ejecutan en un thread o hilo de ejecución. Pero cada aplicación puede tener más de un thread al mismo tiempo, es decir se pueden estar haciendo varias cosas a un mismo tiempo. En Visual Basic.Net, a diferencia de las versiones anteriores, se pueden crear múltiples threads para que podamos realizar diferentes tareas al mismo tiempo. Cuando se define un nuevo thread, lo que hay que hacer es indicarle al compilador cual será el procedimiento que queremos usar de forma paralela al resto de la aplicación.

c) Sockets

Los sockets son un sistema de comunicación entre procesos de diferentes máquinas de una red, es un punto de comunicación por el cual un proceso puede emitir o recibir información. Es un proceso que permite manejar de una forma sencilla la comunicación entre dos equipos, aunque estos procesos se encuentren en sistemas distintos, sin necesidad de conocer el funcionamiento de los protocolos de comunicación subyacentes, en la figura 1 se puede ver una interpretación de los procesos involucrados para establecer la comunicación entre dos máquinas.

Proceso de comunicación entre dos máquinas con sockets.

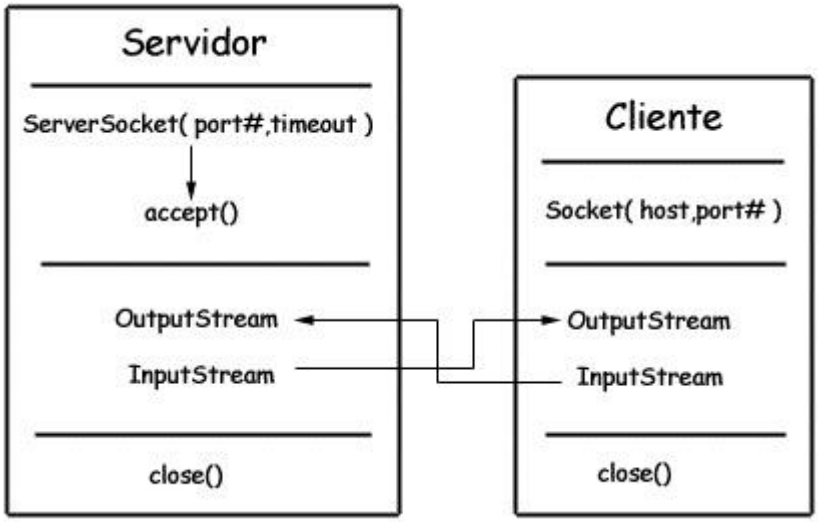


Figura 1. Eventos que interactúan en la comunicación entre sockets.
Fuente: (Gigena, 2012)
Elaborado por: José Benítez

1.4.2 Introducción a seguridad informática HARDENING

La seguridad informática se basa en tres principios fundamentales:

- Mínimo punto de exposición
- Mínimo privilegio posible
- Defensa en profundidad

Mínimo punto de exposición: cuando menor sea el punto de exposición menos probabilidad de que el servidor sea atacado, aquí nace la necesidad de tener un plan de contingencia el cual mitigue el efecto de la amenaza, cuanto menor sea el punto de exposición, menor será la amenaza que hay que mitigar o el impacto de ésta.

Realiza un análisis de los servicios que se ejecutan en un servidor, se clasifica cuales son críticos, medios y bajos, En un servidor donde existe un servicio crítico no se debe compartir los otros servicios, ya que la vulneración de uno de éstos es el hueco de seguridad de acceso a los datos que gestiona otro servicio. Por lo que no es recomendable ejecutar varios servicios en la misma máquina, de aquí viene la importancia de la virtualización de sistemas operativos.

Mínimo privilegio posible: es muy común que los usuarios que ejecutan aplicaciones en una máquina lo hacen con los máximos privilegios posibles esto representa un riesgo para el sistema, las aplicaciones deben ejecutarse con el mínimo privilegio posible, para evitar que se ejecuten aplicaciones o SCRIPTS maliciosos.

Recomienda que los usuarios trabajen sin privilegios y que solo introduzcan éstos en el instante que se requiera ejecutar una instrucción con elevación. En sistemas operativos GNU/Linux se dispone de los SUDOERS con los que se puede implementar estas acciones. Tampoco es recomendable que el usuario root sea conocido por distintos empleados y se debe cambiar el nombre del usuario root.

También se determina los privilegios de los directorios de los archivos de configuración, los directorios de los LOGS de cada usuario en el /home estos deben estar correctamente configurados para cada usuario dando solo los permisos necesarios en caca directorio, se recomienda crear un grupo de usuarios que puedan ver los LOGS del sistema y que no sean visibles a usuarios curiosos.

Los bits especiales ayudan también a proteger los sistemas ante borrados inapropiados, *sticky bit*, o la ejecución de ciertos comandos con la identidad del propietario por eso es necesario revisar el bit SUID y GUID en todo el sistema operativo.

Defensa en profundidad: El modelo de defensa en profundidad proviene del entorno militar, es decir, mantener múltiples líneas de defensa, en vez de disponer de una línea de defensa única muy reforzada, este modelo tiene como objetivo retrasar el posible avance de un intruso o usuario malintencionado lo máximo posible. En la figura 2 se puede visualizar una interpretación del modelo defensa en profundidad.

Figura 2. Capas del modelo defensa en profundidad.
Fuente: (Álvarez Martín & Gonzales Pérez, 2013, pág. 15).
Elaborado por: José Benítez

Políticas, procedimientos y concienciación: son las políticas de la empresa que debe implementar para todos sus empleados con el objetivo de efectuar costumbres y capacitar acerca de la responsabilidad que implica el manejo de sus contraseñas para que no sean víctimas de la ingeniería social.

a) **Seguridad física:** se puede entender desde dos perspectivas: la seguridad física como l procedimiento mediante el uso de cámaras, guardias de seguridad. Y la protección física como mecanismos que son utilizados para asegurar los sistemas o la información del acceso físico a un medio digital por parte de un usuario.
b) **Seguridad del perímetro:** es la barrera dedicada a proteger el entorno o capa interna de la empresa, es el paso previo a la red interna y es una capa que debe estar correctamente configurada, involucra a las configuraciones de firewall con *ACL (Access Control List)* correctamente distribuidos y separar segmentos internos que necesiten más seguridad con el uso de *Virtual Private Network VPN*.

c) **Seguridad de la red interna:** Analizar y segmentar las redes con VPN y la implementación de *IDS (Intrusión Detection System)* que detectan accesos no autorizados a un equipo o incluso a una red.

d) **Seguridad a nivel de servidor:** Se debe tomar en cuenta las actualizaciones del sistema operativo servidor que da soporte y gestiona las aplicaciones y servicios que se ejecutan en dicha máquina.

   Es importante disponer de loggin, tanto local como remoto en el servidor para llevar un registro de actividad, existen diferentes tipos de LOGS estos pueden serlos de kernel, autentificación registro de actividad o tareas y es donde se debe saber gestionar la información.

e) **Seguridad en la aplicación:** generalmente se disponen de varias aplicaciones o servicios que pueden tratar con la parte pública a través de la red, y la exposición debe estar bien controlada y segura, hace énfasis en no tener configuraciones por defecto de las aplicaciones y recomienda disponer de una configuración propia y en la que se sepa que se está realizando en lugar de tener varias aplicaciones configuradas por defecto, las cuales pueden abrir vías de ataque a un usuario malintencionado con las que podría lograr acceder al control remoto del server, una denegación de servicio o simplemente a visualizar la información interna de la máquina.

   Se recomienda tener siempre las aplicaciones actualizadas y manejar adecuadamente los privilegios de los usuarios para controlar las acciones de los mismos. Las cuotas disco también pueden favorecer que los usuarios no se excedan con los recursos, por ello debe haber un control de almacenamiento para cada usuario. (Álvarez Martín & Gonzales Pérez, 2013, págs. 13 - 21)

1.4.2.1 Información importante del servidor

Se considera información importante aquella que nos puede dar una visión general del servidor, la información que se necesita saber si el equipo tiene vulnerabilidades y si han existido eventos inusuales en su normal uso como se puede visualizar en la tabla 1.

Tabla 1.
Tipos de análisis y sus descripciones.

Descripción	Nombre análisis
Obtiene información general del server como; Nombre de la máquina, nombre del sistema operativo, actualización, versión, tipo de procesador, tipo de arquitectura, y cuanto tiempo está sin reiniciar.	Información general del servidor
Informa que usuarios están conectados en el instante que se ejecuta la acción	Listado de conexiones activas
Obtiene información de todos los usuarios que han intentado ingresar al servidor y no han tenido éxito.	Intentos fallidos de conexión
Muestra los usuarios que han estado logeados recientemente en el servidor así como las consolas y terminales virtuales (ttys) que han usado.	Listado conexiones Usuarios
Presenta todos los usos del comando *su* o *sudo* y sus acciones.	Listado uso comando *su o sudo*
Obtiene información de los puertos udp activos, tcp abiertos, rcp activos que el servidor tendrá en el momento de la ejecución.	Servicios activos

Nota: Estos tipos de análisis reflejan información general importante para el atacante.
Elaborado por: José Benítez

1.4.2.1.1 Información general del servidor

Para empezar es necesario la información general del servidor para tener una visión clara de la máquina que se pretende interrogar, en este caso se utilizan los mismos comandos en todas las distribuciones escogidas para este proyecto, a continuación en la tabla 2 se describen los comandos a ejecutarse.

Tabla 2.
Información general del servidor comandos y descripciones.

Información	Comando
Nombre de la máquina	uname -n
Nombre del sistema operativo	uname -s
Versión del núcleo	uname -r
Versión del kernel	uname -v
Tipo de procesador	uname -p
Tipo de arquitectura	uname -m
La hora actual, el tiempo que el sistema ha estado funcionando, cuántos usuarios están actualmente conectados y el promedio de carga del sistema para los últimos 1, 5 y 15 minutos.	uptime

Nota. Los mismos comandos se ejecutan en todas las distribuciones.
Elaborado por: José Benítez

Ejemplos:

El comando *uptime* nos muestra la información en el siguiente orden;

- La hora actual
- Tiempo que el sistema ha estado funcionando
- Número de usuarios actualmente conectados
- El promedio de carga del sistema para los últimos 1, 5 y 15 minutos.

Se puede visualizar en las figura 3 un ejemplo de ejecución del comando *uptime* en Ubuntu.

Ejemplo ejecución comando *uptime* en Ubuntu.

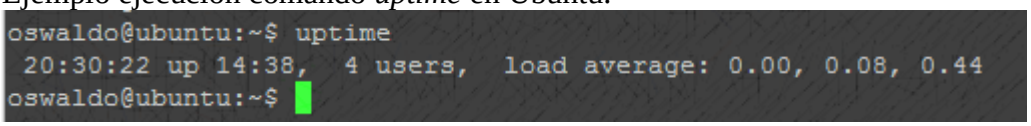


Figura 3. Elaborado por: José Benítez

Y en la figura 4 un ejemplo de ejecución del comando *uptime* en OpenSUSE.

Ejemplo ejecución comandos de información en OpenSUSE.

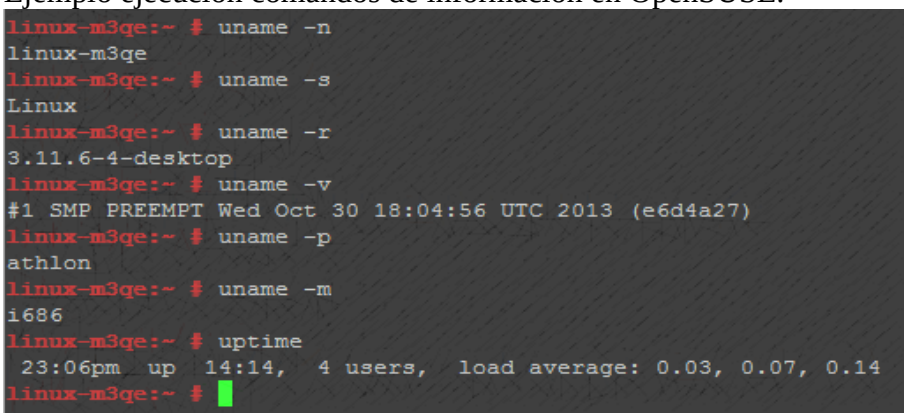


Figura 4. Grupos de comandos necesarios para obtener información de general. Elaborado por: José Benítez

1.4.2.1.2 Intentos Fallidos de conexión

Se utiliza el comando *last* para obtener un listado de intentos fallidos de conexión con la fecha y el usuario que intentó entrar.

Con esta información podríamos detectar a tiempo algún intento de hacking por fuerza bruta como el ataque de diccionario que consiste en ir intentando simultáneamente combinaciones de palabras desde un fichero plano donde contiene las posibles contraseñas, hace varios intentos fallidos de loggin hasta encontrar la contraseña.

Para mitigar esta vulnerabilidad recomendable limitar el número de intentos de loggin que se explicará posteriormente en políticas de cuentas o implementar políticas de contraseñas seguras.

Ejemplos:

En la figura 5 podemos ver un ejemplo de la información que nos facilita el comando *lastb* en OpenSUSE.

Ejemplo ejecución comando *lastb* en OpenSUSE.

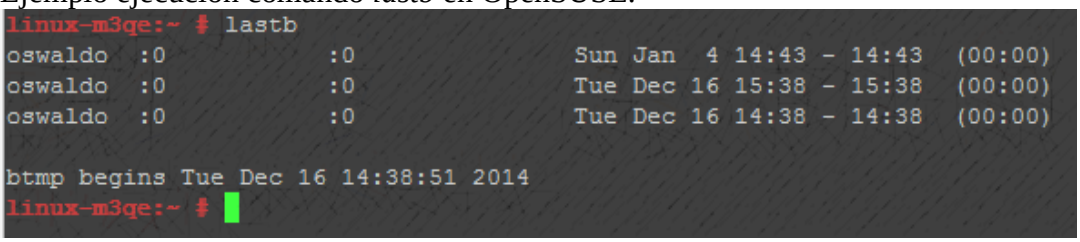


Figura 5. Elaborado por: José Benítez

1.4.2.1.3 Listado de conexiones activas

El informe presenta todos los usuarios conectados en el momento de la ejecución, y los detalles de su conexión.

El comando *who* puede listar los nombres de los usuarios conectados actualmente, su terminal, el tiempo que han estado conectados, y el nombre del host desde el que se han conectado. En la tabla 3 se detalla la descripción y opciones del comando *who.*

Tabla 3.
Opciones del comando who abreviaturas y descripción.

Opción		Descripción
-a	*--all*	Mismo que -b -d --login -p -r -t -T -u
-b	*--boot*	Tiempo del último arranque del sistema
-d	*--dead*	Imprimir procesos muertos
-H	*--heading*	Línea de impresión de encabezados de columna
-l	*--login*	Procesos de inicio de sesión del sistema de impresión
	--lookup	tratar de canonizar nombres de host a través de DNS
-m		Sólo el nombre de host y el usuario asociado con la entrada estándar
-p	*--process*	Imprimir procesos activos generados por init
-q	*--count*	Todos los nombres de usuario y número de usuarios conectados
-r	*--runlevel*	Imprimir nivel de ejecución actual
-s	*--short*	Imprimir sólo el nombre, la línea y el tiempo (por defecto)
-t	*--time*	Imprimir cambio del reloj último sistema
-T	*-w, --mes*	Añadir el estado del mensaje de usuario como +, - o

-u	*--users*	Listado de usuarios conectados
	--help	Muestra esta ayuda y salir
	--version	Salida de información de la versión y salir

Nota. Se puede ver las opciones del comando who con la opción *#man who*
Elaborado por: José Benítez

Ejemplos:

A continuación podemos visualizar en la figura 6 un ejemplo de la ejecución del comando *who –uH* en Fedora.

Ejemplo ejecución comando *who-uH* en Fedora.

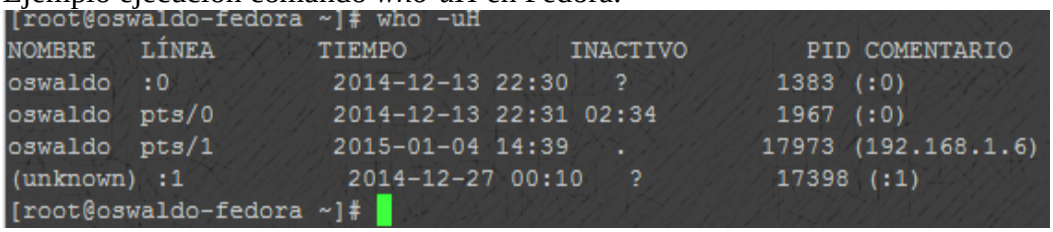


Figura 6. Elaborado por: José Benítez

1.4.2.1.4 Listado de conexiones de usuarios

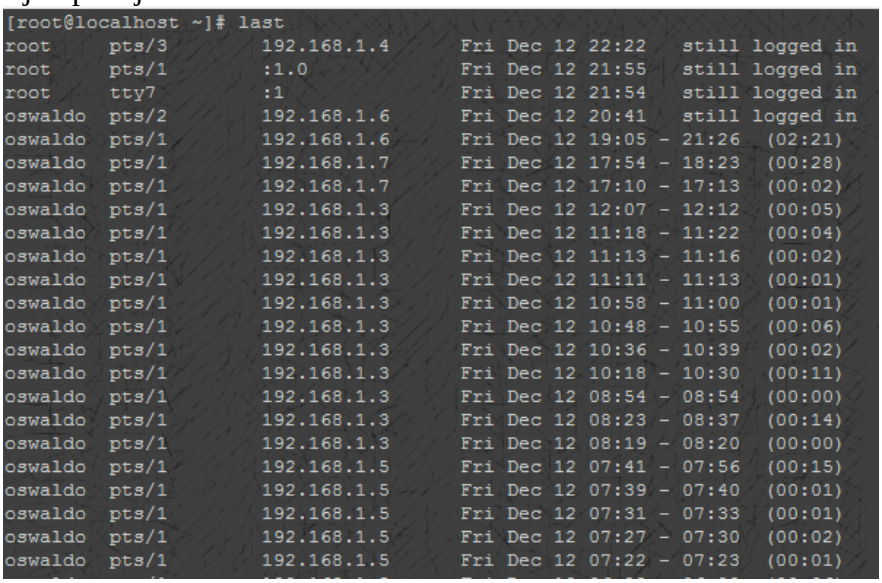
Con el comando *last* se puede visualizar un listado de la última entrada del usuario, y observar la actividad del usuario en el sistema.

Dado que la actividad de todos los usuarios en el sistema se registra en el archivo */var /log/wtmp* el comando *last* buscará ese archivo de registro en particular.

Si es un usuario normal sin privilegios de root también puede utilizar este comando y visualizar las conexiones inclusive del usuario root. Como se puede ver en la figura 7 el orden de la información es el siguiente; nombre, terminal, origen, desde, hasta.

Ejemplos:

Ejemplo ejecución comando last en Centos.



```
[root@localhost ~]# last
root     pts/3        192.168.1.4      Fri Dec 12 22:22   still logged in
root     pts/1        :1.0             Fri Dec 12 21:55   still logged in
root     tty7         :1               Fri Dec 12 21:54   still logged in
oswaldo  pts/2        192.168.1.6      Fri Dec 12 20:41   still logged in
oswaldo  pts/1        192.168.1.6      Fri Dec 12 19:05 - 21:26  (02:21)
oswaldo  pts/1        192.168.1.7      Fri Dec 12 17:54 - 18:23  (00:28)
oswaldo  pts/1        192.168.1.7      Fri Dec 12 17:10 - 17:13  (00:02)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 12:07 - 12:12  (00:05)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 11:18 - 11:22  (00:04)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 11:13 - 11:16  (00:02)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 11:11 - 11:13  (00:01)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 10:58 - 11:00  (00:01)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 10:48 - 10:55  (00:06)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 10:36 - 10:39  (00:02)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 10:18 - 10:30  (00:11)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 08:54 - 08:54  (00:00)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 08:23 - 08:37  (00:14)
oswaldo  pts/1        192.168.1.3      Fri Dec 12 08:19 - 08:20  (00:00)
oswaldo  pts/1        192.168.1.5      Fri Dec 12 07:41 - 07:56  (00:15)
oswaldo  pts/1        192.168.1.5      Fri Dec 12 07:39 - 07:40  (00:01)
oswaldo  pts/1        192.168.1.5      Fri Dec 12 07:31 - 07:33  (00:01)
oswaldo  pts/1        192.168.1.5      Fri Dec 12 07:27 - 07:30  (00:02)
oswaldo  pts/1        192.168.1.5      Fri Dec 12 07:22 - 07:23  (00:01)
```

Figura 7. Elaborado por: José Benítez

1.4.2.1.5 Listado uso del comando “su” o “sudo”

En este tipo de análisis presenta un informe de todo el histórico del uso del comando *su* o *sudo* con los detalles como la actividad, fecha, hora y usuario, dependiendo de la distribución varía el formato de presentación de cada LOG y el nombre del archivo.

Para obtener esta información debemos leer los archivos, en cada distribución es diferente como se detallan en la tabla 4.

Tabla 4.
Comandos para el historial de su o sudo en cada distribución.

Centos 20, Fedora 7	Ubuntu 14	OpenSUSE 13
journalctl /usr/bin/su -n50 journalctl /usr/bin/sudo -n50	cat /var/log/auth.log	cat /var/log/messages

Nota. Identificación de los comandos en cada distribución.
Elaborado por: José Benítez

En este proyecto se utilizó la versión 20 de la distribución Fedora en Centos la versión 7, estas distribuciones tienen una diferencia y es que están ya migradas a SYSTEMD. Posteriormente las nuevas distribuciones que sigan apareciendo también serán migradas.

SYSTEMD tiene su propio sistema de registros LOGS, dado que la mayor parte de las distribuciones *GNU/Linux* utilizaban archivos de texto y se dificultaba la

búsqueda avanzada de LOGS, ahora se utiliza y se utilizará JOURNALCTR que facilita la exploración y clasifica los LOGS, por lo tanto ya no se utilizaría más el servicio *syslog*.

Para añadir un filtro y ver sólo los últimos 50 registros de información generados utilizamos los siguientes comandos:

 # journalctl /usr/bin/su -n50
 # journalctl /usr/bin/sudo -n50

Ejemplos:

En la figura 8 se puede visualizar un ejemplo de *journalctl* en Fedora.

Histórico del archivo journalctl en Fedora.

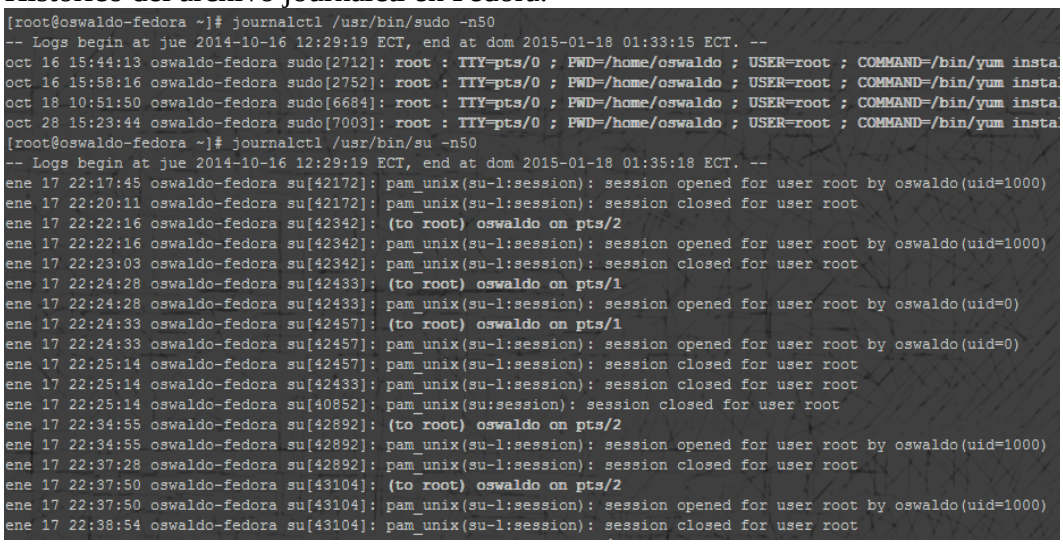


Figura 8. Elaborado por: José Benítez

En la figura 9 tenemos un ejemplo de la visualización de los LOGS en Ubuntu y el formato del fichero */var/log/auth.log*

Histórico del archivo audit.log en Ubuntu.

```
root@ubuntu:/# cat /var/log/auth.log
Jan  5 09:09:01 ubuntu CRON[6823]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan  5 09:09:02 ubuntu CRON[6823]: pam_unix(cron:session): session closed for user root
Jan  5 09:17:01 ubuntu CRON[6855]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan  5 09:17:01 ubuntu CRON[6855]: pam_unix(cron:session): session closed for user root
Jan  5 09:17:13 ubuntu pkexec: pam_unix(polkit-1:session): session opened for user root by (uid=1000)
Jan  5 09:17:13 ubuntu pkexec[6866]: oswaldo: Executing command [USER=root] [TTY=unknown] [CWD=/home/oswaldo] [COMMAND=
/usr/lib/update-notifier/package-system-locked]
Jan  5 09:21:19 ubuntu sshd[5130]: pam_unix(sshd:session): session closed for user oswaldo
Jan  5 09:21:19 ubuntu su[5208]: pam_unix(su:session): session closed for user root
Jan  5 09:36:06 ubuntu compiz: gkr-pam: unlocked login keyring
Jan  5 09:39:01 ubuntu CRON[6939]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan  5 09:39:02 ubuntu CRON[6939]: pam_unix(cron:session): session closed for user root
Jan  5 09:41:20 ubuntu compiz: PAM unable to dlopen(pam_kwallet.so): /lib/security/pam_kwallet.so: cannot open shared o
bject file: No such file or directory
Jan  5 09:41:20 ubuntu compiz: PAM adding faulty module: pam_kwallet.so
Jan  5 09:41:20 ubuntu compiz: pam_succeed_if(lightdm:auth): requirement "user ingroup nopasswdlogin" not met by user "
oswaldo"
Jan  5 09:42:47 ubuntu sshd[5319]: pam_unix(sshd:session): session closed for user oswaldo
Jan  5 09:42:47 ubuntu su[5397]: pam_unix(su:session): session closed for user root
Jan  5 09:48:54 ubuntu su[6998]: Successful su for root by oswaldo
Jan  5 09:48:54 ubuntu su[6998]: + /dev/pts/12 oswaldo:root
Jan  5 09:48:54 ubuntu su[6998]: pam_unix(su:session): session opened for user root by oswaldo(uid=1000)
root@ubuntu:/#
```

Figura 9. Elaborado por: José Benítez

En la figura 10 tenemos un ejemplo de la visualización de LOGS en OpenSUSE, y el formato de información que contiene el archivo */var/log/messages*.

```
linux-m3qe:~ # cat /var/log/messages
2015-01-15T23:20:59.345841-05:00 linux-m3qe sshd[27833]: Accepted keyboard-interactive/pam for
oswaldo from 192.168.1.6 port 49244 ssh2
2015-01-15T23:20:59.353301-05:00 linux-m3qe sshd[27833]: pam_unix(sshd:session): session opened
 for user oswaldo by (uid=0)
2015-01-15T23:20:59.363675-05:00 linux-m3qe systemd-logind[23622]: New session 469 of user oswa
ldo.
2015-01-15T23:20:59.372189-05:00 linux-m3qe systemd[1]: Starting Session 469 of user oswaldo.
2015-01-15T23:20:59.376694-05:00 linux-m3qe systemd[1]: Started Session 469 of user oswaldo.
2015-01-15T23:21:05.292857-05:00 linux-m3qe su: (to oswaldo) oswaldo on none
2015-01-15T23:21:05.297433-05:00 linux-m3qe su: pam_unix(su-l:session): session opened for user
 root by oswaldo(uid=1000)
2015-01-15T23:21:05.304230-05:00 linux-m3qe su: pam_systemd(su-l:session): pam_putenv: delete n
on-existent entry; XDG_RUNTIME_DIR
2015-01-15T23:22:36.481225-05:00 linux-m3qe sshd[27833]: pam_unix(sshd:session): session closed
 for user oswaldo
2015-01-15T23:22:36.504589-05:00 linux-m3qe su: pam_unix(su-l:session): session closed for user
 root
2015-01-15T23:22:36.520828-05:00 linux-m3qe systemd-logind[23622]: Removed session 469.
2015-01-15T23:24:09.466906-05:00 linux-m3qe sshd[27951]: Accepted keyboard-interactive/pam for
oswaldo from 192.168.1.6 port 49247 ssh2
2015-01-15T23:24:09.474452-05:00 linux-m3qe sshd[27951]: pam_unix(sshd:session): session opened
 for user oswaldo by (uid=0)
2015-01-15T23:24:09.483988-05:00 linux-m3qe systemd-logind[23622]: New session 470 of user oswa
ldo.
2015-01-15T23:24:09.495260-05:00 linux-m3qe systemd[1]: Starting Session 470 of user oswaldo.
2015-01-15T23:24:09.506840-05:00 linux-m3qe systemd[1]: Started Session 470 of user oswaldo.
2015-01-15T23:24:31.031023-05:00 linux-m3qe su: (to oswaldo) oswaldo on none
2015-01-15T23:24:31.034169-05:00 linux-m3qe su: pam_unix(su-l:session): session opened for user
```

Figura 10. Histórico del archivo audit.log en OpenSUSE.
Elaborado por: José Benítez

1.4.2.1.6 Servicios activos

Es importante identificar los puertos UDP activos y los puertos TCP abiertos, así como los servicios RCP activos, esta información es muy vital para el atacante porque es el primer paso para entrar al servidor.

Con un reporte de los puertos UDP y TCP activos sabremos cuales son los puertos abiertos, en este caso es importante tener una configuración de los servicios configurando en diferentes puertos que los que se tiene por default.

Y en el caso del servicio RCP se verifica ejecutando *rpcinfo* con la opción *-p* los puertos y protocolos utilizados por los servicios *portmapper, NFS, lockd, mountd, rquotad y statd.*

En la tabla 5 se detallan los comandos utilizados para obtener la información, en este caso son los mismos para todas las distribuciones.

Tabla 5.
Comandos para verificar los servicios activos.

Distribución	Puertos UDP activos	Puertos TCP abiertos	Servicios RCP activos
CENTOS	*netstat -an|grep udp*	*netstat -an|grep tcp|grep LISTEN*	*rpcinfo -p*
FEDORA	*netstat -an|grep udp*	*netstat -an|grep tcp|grep LISTEN*	*rpcinfo -p*
UBUNTU	*netstat -an|grep udp*	*netstat -an|grep tcp|grep LISTEN*	*rpcinfo -p*
OPENSUSE	*netstat -an|grep udp*	*netstat -an|grep tcp|grep LISTEN*	*rpcinfo -p*

Nota. Comandos necesarios para verificar los puertos o servicios activos o abiertos.
Elaborado por: José Benítez

Ejemplos:

Los comandos son los mismos en todas las distribuciones seleccionadas y por ello tenemos un ejemplo en la distribución Ubuntu de los 3 comandos, en la figura 11.

Ejemplos comandos para los servicios activos y puertos.

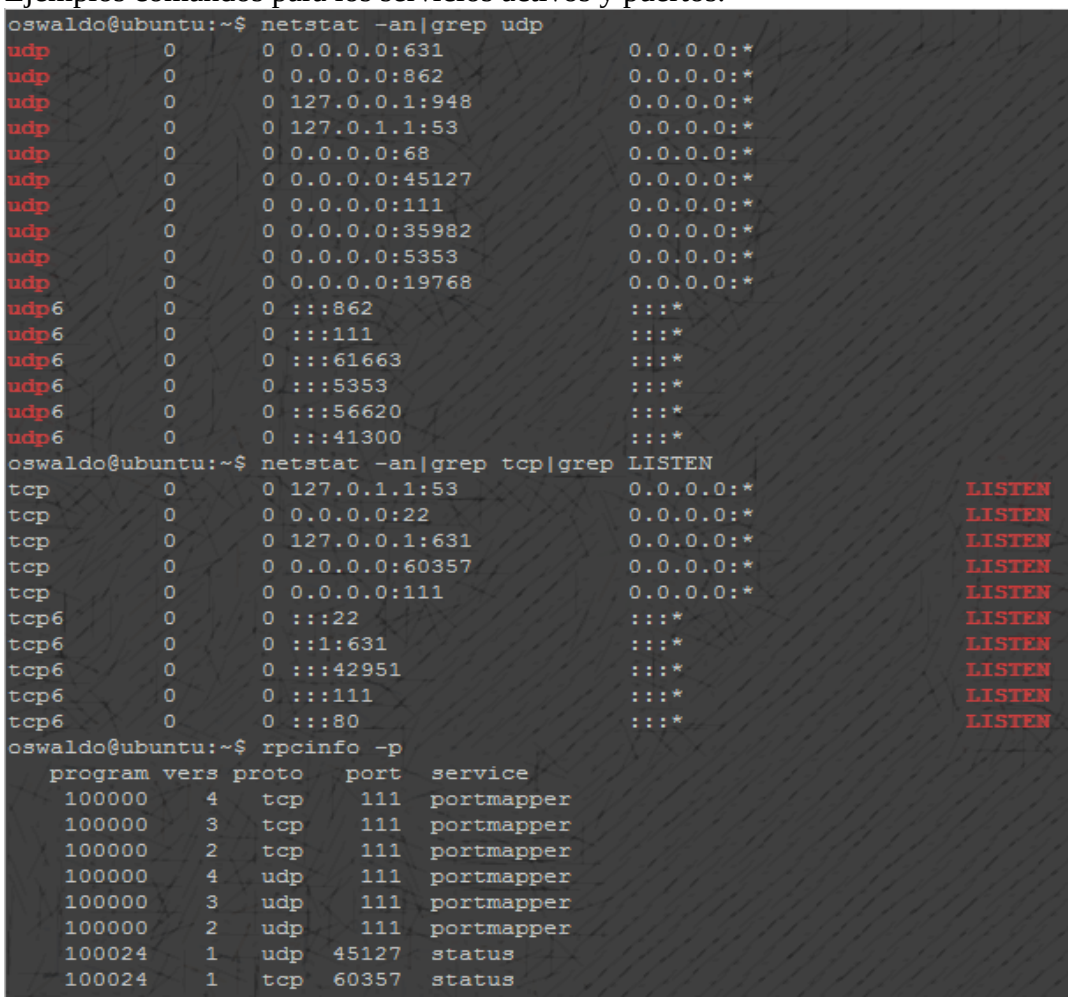


Figura 11. Elaborado por: José Benítez

1.4.2.2 Permisos

En las prácticas de HARDENING se aplica el concepto del mínimo privilegio posible debido a que hay usuarios que no deben tener acceso a archivos especiales y no deberían poder visualizar información relevante así como también se debe restringir los accesos a los directorios de otros usuarios, la mayoría de ataques se hacen después de haber analizado las ventanas del sistemas esto implica que se debe prevenir la curiosidad en los servidores. Delimitar los permisos de las aplicaciones es también recomendable pues los usuarios lo hacen con los máximos privilegios posibles esto representa un riesgo para el sistema.

Se debe buscar los bits especiales, SUID y GUID que ayudan a proteger los sistemas ante borrados inapropiados, o la ejecución de ciertos comandos con la identidad del propietario en todo el sistema operativo.

Todas estas recomendaciones se las clasifica en permisos y las descripciones de cada uno se las detallan en la siguiente tabla 6:

Tabla 6.
Descripción de los tipos de análisis involucrados en la categoría permisos.

Descripción	Nombre análisis
Obtiene un detalle de los usuarios existentes e información de sus contraseñas, plantea recomendaciones en base a criterios técnicos.	Detalle de usuarios y contraseñas
Búsqueda de los directorios y ficheros que tienen el SUID(S) y STICKY BIT(T) activados.	Listado SUID y SGID activos
Listado de los permisos en los archivos especiales potencialmente vulnerables, y de interés para los atacantes.	Permisos archivos especiales
Obtiene información de permisos del archivo *shadow* y plantea recomendaciones de como mitigar este inconveniente.	Lectura shadow
Obtiene información de permisos de los directorios que tiene cada usuario en su respectivo *home.*	Permisos de multiusuarios

Nota. Estos tipos de análisis son los que un atacante necesita saber para enfocar su exploración.
Elaborado por: José Benítez

1.4.2.2.1 Detalle de usuarios y contraseña

Es muy frecuente encontrar servidores donde se permite listar directorios o incluso ver código de aplicaciones y SCRIPTS que se usan, aunque en teoría no se tenga acceso a ellos. Este tipo de problemas son frecuentes debido a una mala gestión de los permisos en los ficheros sensibles.

Por mucho que se proteja un sistema, algunos usuarios son descuidados y no se preocupan lo suficiente a la hora de asignar una contraseña segura o acostumbran a poner sus contraseñas escritas en papeles pegados en los escritorios de trabajo. Cuando eso ocurre, un potencial atacante ya habría superado la primera barrera, y por lo tanto tendría acceso al sistema gracias a la cuenta con seguridad débil proporcionada por el usuario legítimo, la mayoría de ataques son realizados por fuga de información de los mismos empleados a causa de la falta de conocimientos de las políticas de seguridad informática.

Para evitar esta clase de problemas es importante obligar a los usuarios a redefinir su contraseña cada cierto periodo de tiempo, así aseguramos que aunque el usuario pierda la contraseña o cualquier posible atacante la obtenga, al cabo de cierto tiempo esas credenciales se invalidarán, consiguiendo así que el atacante no pueda reutilizarlos.

Para mitigar este problema hacemos uso del comando *chage* su ayuda es bastante explícita y se ejecuta de la siguiente manera:

 # chage <opciones> <usuario>

Con las opciones podemos asignar un día concreto de expiración, inhabilitar una cuenta después de cierto tiempo sin usarse, el número máximo y mínimo en los que habrá que cambiar la clave.

Ejemplo:

 # chage –M 30 –W 5 admin

En este caso estamos diciendo que el usuario *admin* deberá cambiar la contraseña en un máximo de 30 días, y se le avisará durante los 5 días previos de que debe hacerlo. Finalmente podemos comprobar que realmente estas opciones han sido asignadas con la opción "-l" que nos lista las opciones concretas para el usuario especificado:

 # chage -l admin

Último cambio de contraseña: jun 20, 2015
La contraseña caduca: jul 19, 2015
Contraseña inactiva: nunca
La cuenta caduca: nunca
Número de días mínimo entre cambio de contraseña: 0
Número de días máximo entre cambio de contraseñas: 30
Número de días de aviso antes de que expire la contraseña: 5
(Ferran Pichel, 2011, pág. 13)

- */etc/passwd*

Para entender mejor la información se filtra la presentación aplicando AWK y se tiene el siguiente comando.

*#awk -F":" '{print "User= "$1" *UID= "$3" *GID= "$4" *Nombre completo= "$5" *Directorio= "$6}' /etc/passwd*

- **User.-**Nombre de la cuenta para acceder al sistema.
- **UID.-** Identificador único del usuario.
- **GID.-** Identificador único que indica a cual grupo pertenece el usuario.
- **Nombre completo.-** Nombre del usuario completo del usuario
- **Home del usuario.-** Directorio de trabajo del usuario.

En la figura 12 podemos ver un ejemplo de la información procesada con AWK del archivo /*etc/passwd* desde Fedora.

Formato del archivo *passwd* en Fedora.

```
[root@oswaldo-fedora ~]# awk -F":" '{print "User= "$1" *UID= "$3" *GID= "$4" *Nombre completo=
"$5" *Directorio= "$6}' /etc/passwd
User= root *UID= 0 *GID= 0 *Nombre completo= root *Directorio= /root
User= bin *UID= 1 *GID= 1 *Nombre completo= bin *Directorio= /bin
User= daemon *UID= 2 *GID= 2 *Nombre completo= daemon *Directorio= /sbin
User= adm *UID= 3 *GID= 4 *Nombre completo= adm *Directorio= /var/adm
User= lp *UID= 4 *GID= 7 *Nombre completo= lp *Directorio= /var/spool/lpd
User= sync *UID= 5 *GID= 0 *Nombre completo= sync *Directorio= /sbin
User= shutdown *UID= 6 *GID= 0 *Nombre completo= shutdown *Directorio= /sbin
User= halt *UID= 7 *GID= 0 *Nombre completo= halt *Directorio= /sbin
User= mail *UID= 8 *GID= 12 *Nombre completo= mail *Directorio= /var/spool/mail
User= operator *UID= 11 *GID= 0 *Nombre completo= operator *Directorio= /root
User= games *UID= 12 *GID= 100 *Nombre completo= games *Directorio= /usr/games
User= ftp *UID= 14 *GID= 50 *Nombre completo= FTP User *Directorio= /var/ftp
User= nobody *UID= 99 *GID= 99 *Nombre completo= Nobody *Directorio= /
User= avahi-autoipd *UID= 170 *GID= 170 *Nombre completo= Avahi IPv4LL Stack *Directorio= /var/
lib/avahi-autoipd
User= dbus *UID= 81 *GID= 81 *Nombre completo= System message bus *Directorio= /
User= polkitd *UID= 999 *GID= 999 *Nombre completo= User for polkitd *Directorio= /
User= abrt *UID= 173 *GID= 173 *Nombre completo=  *Directorio= /etc/abrt
User= usbmuxd *UID= 113 *GID= 113 *Nombre completo= usbmuxd user *Directorio= /
User= colord *UID= 998 *GID= 998 *Nombre completo= User for colord *Directorio= /var/lib/colord
User= rtkit *UID= 172 *GID= 172 *Nombre completo= RealtimeKit *Directorio= /proc
User= geoclue *UID= 997 *GID= 996 *Nombre completo= User for geoclue *Directorio= /var/lib/geoc
lue
User= chrony *UID= 996 *GID= 995 *Nombre completo=  *Directorio= /var/lib/chrony
User= tss *UID= 59 *GID= 59 *Nombre completo= Account used by the trousers package to sandbox t
he tcsd daemon *Directorio= /dev/null
User= unbound *UID= 995 *GID= 994 *Nombre completo= Unbound DNS resolver *Directorio= /etc/unbo
und
User= openvpn *UID= 994 *GID= 993 *Nombre completo= OpenVPN *Directorio= /etc/openvpn
User= avahi *UID= 70 *GID= 70 *Nombre completo= Avahi mDNS/DNS-SD Stack *Directorio= /var/run/a
vahi-daemon
User= pulse *UID= 993 *GID= 991 *Nombre completo= PulseAudio System Daemon *Directorio= /var/ru
n/pulse
User= gdm *UID= 42 *GID= 42 *Nombre completo=  *Directorio= /var/lib/gdm
User= gnome-initial-setup *UID= 992 *GID= 989 *Nombre completo=  *Directorio= /run/gnome-initia
l-setup/
User= nm-openconnect *UID= 991 *GID= 988 *Nombre completo= NetworkManager user for OpenConnect
*Directorio= /
User= sshd *UID= 74 *GID= 74 *Nombre completo= Privilege-separated SSH *Directorio= /var/empty/
sshd
User= oswaldo *UID= 1000 *GID= 1000 *Nombre completo= oswaldo *Directorio= /home/oswaldo
User= rpc *UID= 32 *GID= 32 *Nombre completo= Rpcbind Daemon *Directorio= /var/lib/rpcbind
User= rpcuser *UID= 29 *GID= 29 *Nombre completo= RPC Service User *Directorio= /var/lib/nfs
User= nfsnobody *UID= 65534 *GID= 65534 *Nombre completo= Anonymous NFS User *Directorio= /var/
lib/nfs
```

Figura 12. Elaborado por: José Benítez

- */etc/shadow*

En este fichero se encuentran almacenados los datos sobre las contraseñas de cada usuario del sistema para entender mejor se usó AWK ejecutando el siguiente comando.

*#awk -F":" '{print "User= "$1" *Último cambio pasword= "$3" *Días notificación cambio= "$4" *Días para expiración contraseña= "$5" *Alarmas pasword= "$6" *Días desactivados= "$7" *Desactivada= "$8}' /etc/shadow*

Obtendremos la respuesta del servidor con la siguiente información:

- **User.-** Es el nombre de usuario.
- **Último cambio pasword.-** Días desde el último cambio de clave.
- **Días notificación cambio:** Días de aviso al usuario antes que expire la clave.

- **Días para expiración contraseña:** Días en que se desactiva la cuenta tras expirar la clave.
- **Alarmas pasword:** Una vez que la contraseña expiró, será deshabilitada después de pasado estos días.
- **Desactivada:** Después de esta fecha la cuenta es deshabilitada y jamás podrá iniciar sesión.

En la figura 13 se puede visualizar un ejemplo del comando *shadow* procesado con AWK desde Ubuntu.

Archivo *shadow* procesado con awk en Ubuntu.

```
root@ubuntu:~# awk -F":" '{print "User= "$1" *Último cambio pasword= "$3" *Días notificación ca
mbio= "$4" *Días para expiración contraseña= "$5" *Alarmas pasword= "$6" *Días desactivados= "$
7" *Desactivada= "$8}' /etc/shadow
User= root *Último cambio pasword= 16393 *Días notificación cambio= 0 *Días para expiración con
traseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= daemon *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración c
ontraseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= bin *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración cont
raseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= sys *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración cont
raseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= sync *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración con
traseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= games *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración co
ntraseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= man *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración cont
raseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= lp *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración contr
aseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= mail *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración con
traseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= news *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración con
traseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= uucp *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración con
traseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= proxy *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración co
ntraseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= www-data *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración
 contraseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= backup *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración c
ontraseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= list *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración con
traseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= irc *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración cont
raseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
User= gnats *Último cambio pasword= 16177 *Días notificación cambio= 0 *Días para expiración co
ntraseña= 99999 *Alarmas pasword= 7 *Días desactivados=  *Desactivada=
```

Figura 13. Respuestas del archivo *shadow* procesado con awk en Ubuntu.
Elaborado por: José Benítez

1.4.2.2.2 Listado SUID y SGUID activos

Los sistemas operativos GNU/Linux tienen configuraciones por defecto que no fortifican o consideran una amenaza estos tipos de permisos, a continuación se ha resumido las recomendaciones de la empresa auditora Isecauditors.

> Linux es un sistema multiusuario, lo que conlleva a mantener la privacidad de estos y un control general para que no todos puedan hacer lo que quieran y comprometer así el sistema.

En la familia de sistemas operativos GNU/Linux se utilizan los permisos para ello, permitiendo o no realizar acciones concretas al resto de usuarios no propietarios del fichero, o incluso al propio propietario.
Cada usuario tiene un identificador UID (*User Identification*) y un grupo identificativo GID (*Group Identification*). Estos usuarios pueden realizar tres acciones distintas en un fichero; Ejecutar, leer y escribir.

Cuando un usuario crea un fichero, el UID y GID propietarios de éste son los del propio usuario, como es lógico. Pero tenemos más permisos que podemos configurar, de hecho si hacemos "ls -l" veremos una cadena, al inicio de cada línea parecida a la siguiente:

-rwx------

Estos permisos nos indican que el usuario propietario del fichero tiene permisos de Lectura (R), Escritura (W) y Ejecución (X), que no es un directorio y que los miembros del grupo y el resto de usuarios del sistema no tienen ningún permiso. Para interpretar correctamente esta cadena se debe entender que cada espacio corresponde a un Bit de permisos, y estos al mismo tiempo se pueden agrupar en cuatro bloques a continuación una explicación en la tabla 6:

Tabla 6.
Bloques de permisos GNU/Linux

-	Indica si es o no un directorio de ser un directorio empieza con la letra d
rwx	Permisos para el propietario del fichero.
---	Permisos para los usuarios del grupo del fichero.
---	Permiso para el resto de usuarios

Nota. Fuente: (Ferran Pichel, 2011, pág. 4)
Elaborado por: José Benítez

Bit 0: En caso de que el fichero sea un directorio nos aparecerá una "d".
Los Bits de permisos tienen distinto comportamiento según se trate de un fichero o de un directorio, en la tabla 7 se describen las diferencias:

Tabla 7.
Diferencias bits de permisos

Bit	Fichero	Directorio
R	Lectura del fichero.	Lectura del contenido del directorio.
W	Escritura del fichero.	Mover y borrar los ficheros en un directorio.
X	Ejecución del fichero.	Poder acceder a un directorio.

Nota. Fuente: (Ferran Pichel, 2011, pág. 4)
Elaborado por: José Benítez

- **SUID (S)**

  El bit SUID es un flag especial que sirve para modificar temporalmente los privilegios del usuario que ejecuta un programa. Si este bit se activa, el usuario que ejecute el fichero pasará a tener los mismos privilegios que el propietario de éste.

  El bit se puede asignar para que al ejecutar la aplicación se cambie el usuario, el grupo o ambas características del usuario que ejecuta.

- **Sticky bit (T)**

  Es un flag adicional que actualmente no tiene función alguna en los ficheros, pero en cambio sí la tiene cuando se trata de directorios. Se representa con la letra "T".

  Si este bit se encuentra activo en un directorio, los ficheros del directorio pueden ser borrados o renombrados únicamente por el propietario del directorio o por el propietario del fichero, por consiguiente puede ser de gran ayuda cuando hablamos de directorios donde se encuentran ficheros sensibles del sistema u otra información necesaria para el buen funcionamiento de éste.

- **Modificación**

  Como el bit 0 nos indica si se trata o no de un directorio, no podemos modificarlo, sino que es asignado en el momento de crear el fichero/directorio.

  Cuando un fichero es creado se le asigna el propietario (ing: owner) y el grupo de ese usuario. Estos valores pueden ser cambiados utilizando el comando *chown* (*Change Owner*). La sintaxis básica se muestra a continuación:

  chown <usuario>:<grupo> fichero

  De esta manera conseguimos que el fichero pertenezca ahora a <usuario> y al grupo <grupo>. Si uno de los dos campos no se especifica, queda sin cambiar. Obviamente este comando debe ser ejecutado por el anterior propietario del fichero. En los bits de permisos, tenemos tres bloques con tres bits cada uno:

  "rwx".

  Si pensamos que cada bloque de tres bits es independiente del resto, podemos diferenciar cada uno según su valor en binario, que dependerá de la posición

en la que se encuentre. Es decir, si tenemos “rxw” (todos los bits a 1) y hacemos la suma binaria de:

100 + 010 + 001 = 111 -> 7 (en base 10)

Dicho de otra manera, cada permiso tiene un único valor, y basta con sumarlos para asignárselos a un fichero en la tabla 8 se puede visualizar los permisos y sus valores en binario y decimal, en la tabla 8 se detallan los permisos y sus valores en decimal y binario.

Tabla 8.
Permisos en binarios y decimales

Permiso	Binario	Decimal
Read (R)	100	4
Write (w)	100	2
Execute (x)	001	1

Nota. Fuente: (Ferran Pichel, 2011, pág. 6)
Elaborado por: José Benítez

Teniendo en cuenta que son tres bloques de bits, ahora solo se debe elegir que permisos dar a cada bloque. Por ejemplo, si se quiere dar “rxw” (7) al propietario, “rw” (6) al grupo y “r” (4) al resto, basta con poner:

 # chmod 764 <fichero>

Y luego al hacer “ls -l <fichero>” veremos que los permisos han sido modificados:

-rwxrw-r--

Hasta ahora se ha jugado solamente con los bits “rwx”, veamos como activar y desactivar el bit SUID y STICKY. Para poder jugar con estos bits hay que añadir un dígito al comando anterior, por ejemplo:

 # chmod 1755 <fichero>

En la tabla 9 están los números y sus interpretaciones:

Tabla 9.
Los permisos y sus valores.

1	Se le asigna el bit STICKY
7	"r" + "w" + "x" para el propietario
5	"r" + "w" para el grupo
5	"r" + "x" para el resto de usuarios

Nota. Fuente: (Ferran Pichel, 2011, pág. 6)
Elaborado por: José Benítez

En la tabla 10 se muestran los bits dependiendo del primer dígito:

Tabla 10.
Bits asignados

1	Sticky
2	SUID del grupo
4	SUID del usuario

Nota. Fuente: (Ferran Pichel, 2011, pág. 6)
Elaborado por: José Benítez

Sumando los valores podemos especificar las opciones necesarias, el funcionamiento es análogo al de los permisos *rwx*." (Ferran Pichel, 2011, págs. 4,5,6)

1.4.2.2.3 Permisos archivos especiales

En este tipo de análisis se verifica los permisos que tienen archivos que son de interés para el atacante, estos deberían tener permisos de lectura y escritura solamente para el propietario y para los otros usuarios solamente deben tener permisos de lectura.

Para este análisis se extrae un listado de ficheros potencialmente vulnerables y muy importantes para el atacante con sus respectivos permisos.
En la tabla 11 se puede visualizar los comandos utilizados en cada distribución, en este caso son los mismos en todos los sistemas operativos seleccionados para este proyecto.

Tabla 11.
Comandos para verificación de permisos.

Distribución	Permisos adecuados	Comandos a ejecutar
CENTOS	-rw-r—r— -rw-r—r— -rw-r—r— -rw-r—r— -rw-r—r—	ls -l /etc/passwd ls -l /etc/init.d ls -l /etc/xinetd.d ls -l /etc/environment ls -l /etc/exports
FEDORA	-rw-r—r— -rw-r—r— -rw-r—r— -rw-r—r— -rw-r—r—	ls -l /etc/passwd ls -l /etc/init.d ls -l /etc/xinetd.d ls -l /etc/environment ls -l /etc/exports
UBUNTU	-rw-r—r— -rw-r—r— -rw-r—r— -rw-r—r— -rw-r—r—	ls -l /etc/passwd ls -l /etc/init.d ls -l /etc/xinetd.d ls -l /etc/environment ls -l /etc/exports

OPENSUSE	-rw-r—r— -rw-r—r— -rw-r—r— -rw-r—r— -rw-r—r—	ls -l /etc/passwd ls -l /etc/init.d ls -l /etc/xinetd.d ls -l /etc/environment ls -l /etc/exports

Nota. Elaborado por: José Benítez

Ejemplos:

En la figura 14 veremos un ejemplo de los permisos por defecto en Centos de los archivos mencionados con el comando *ls -l*.

Permisos de los archvios *passwd* y *environment* desde Centos.

```
[root@localhost ~]# ls -l /etc/passwd
-rw-r--r--. 1 root root 2506 ene 19 18:13 /etc/passwd
[root@localhost ~]# ls -l /etc/init.d
lrwxrwxrwx. 1 root root 11 ene 19 17:21 /etc/init.d -> rc.d/init.d
[root@localhost ~]# ls -l /etc/xinetd.d
total 0
[root@localhost ~]# ls -l /etc/environment
-rw-r--r--. 1 root root 0 jun  9  2014 /etc/environment
[root@localhost ~]# ls -l /etc/exports
-rw-r--r--. 1 root root 0 jun  7  2013 /etc/exports
[root@localhost ~]#
```

Figura 14. Elaborado por: José Benítez

1.4.2.2.4 Lectura shadow

Este archivo es el que almacena la información relacionada de usuarios y contraseñas del sistema encriptados, el fichero */etc/shadow*.

> Cuando un usuario cambia el password o bien se registra en el sistema, lo hace utilizando el programa */bin/login*, el cual tiene activado el atributo, o bit, *SUID*. Dicho de otra manera, mientras *login* está en ejecución, el usuario que lo ejecuta es *root*.
>
> Ahora si miramos los permisos del fichero */etc/shadow* veremos algo así:
>
> *-rw-r----- 1 root shadow 735 2006-09-29 12:51 /etc/shadow*
>
> Puede variar el grupo o incluso los permisos. Ahora tengamos en cuenta que uso se hará de este fichero, es decir, que usuario va a utilizarlo y para qué. Este fichero solamente será leído y escrito por *root*, a no ser que el administrador quiera que alguno más lo haga. Se recomienda eliminar todo acceso a este fichero por parte de cualquiera, exceptuando *root* a menos que sea realmente necesario. (Ferran Pichel, 2011, pág. 8)

Ejemplos:

Tenemos un ejemplo en la figura 15 de los permisos por defecto que vienen en la distribución Ubuntu.

Permisos del archivo *shadow* desde Ubuntu.

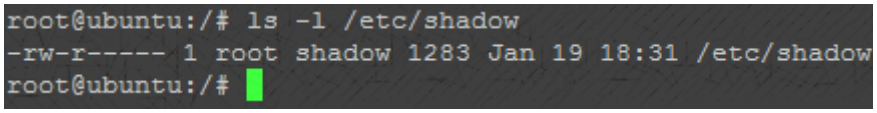


Figura 15. Elaborado por: José Benítez

"Para poder evitar estas situaciones lo mejor es darle permisos de lectura y escritura sólo y exclusivamente al usuario root:

 # chmod 600 /etc/shadow

 # ls -l /etc/shadow

-rw------- 1 root shadow 735 2006-09-29 12:51 /etc/shadow" (Ferran Pichel, 2011, pág. 8)

1.4.2.2.5 Permisos multiusuarios

En la familia de operativos GNU/Linux los usuarios tienen su *home* en */home/<usuario>/* por defecto. Exceptuando el administrador root, el home del cual se encuentra en */root* generalmente.

Hay que mantener la privacidad de todos los usuarios y como proteger el *home* del administrador para que los usuarios no puedan ver que hay en él.

Por defecto, muchas distribuciones de Linux permiten listar el contenido del directorio /root al resto de usuarios del sistema y no es recomendable debido a esto se debe realizar cambios para que esto no ocurra, simplemente basta con asignar permisos de ejecución, lectura y escritura exclusivamente al usuario root:

 # chmod 700 /root

Con esto ya tenemos el directorio protegido contra el listado de ficheros el mismo procedimiento se debe realizar a todos los directorios *home* de los usuarios para evitar que unos vean el contenido de los otros:

*# chmod 700 /home/**

Si queremos hacer grupos de usuarios, y que todos los que formen parte de dicho grupo puedan acceder a sus directorios *home* mutuamente. Simplemente debemos añadir los usuarios a un grupo y hacer lo siguiente:

chown :<grupo> /home/user1 ... chown :<grupo> /home/userN

Y finalmente debemos darles permiso para que puedan listar y ejecutar, por ejemplo, en todos los directorios *home* de los usuarios del grupo:

chmod g+wx /home/user1 ... chmod g+wx /home/userN

Con estos 4 comandos ya tenemos configurados los permisos para los usuarios y para el propio root. (Ferran Pichel, 2011, págs. 8,9)

1.4.2.3 Configuración del servidor

Para esta categoría se han escogido los tipos de análisis que se detallan en la tabla 12 con una breve descripción de lo que representa cada tipo de análisis.

Tabla 12.
Tipos de análisis y descripción de la categoría configuración.

Descripción	Nombre análisis
Obtiene información de los grupos existentes en el servidor y sus usuarios con el GUID.	Detalle de grupos del servidor
Extracción de recursos exportados por NFS. Informe que presenta los directorios exportados a través del servicio NFS.	Listado de recursos exportados por NFS
Análisis de configuraciones básicas y usuarios FTP permitidos y no permitidos.	Listado de usuarios FTP
Listado de usuarios para acceso al CRON	Listado usuarios para CRON
Obtiene información sobre la configuración de las cuentas y las respectivas recomendaciones.	Políticas de cuentas
Obtiene la configuración de gestor de arranque.	Gestor de arranque GRUB 2
Obtiene información de permisos de directorios para los LOGS.	Protección de LOGS
Obtiene información de permisos de lectura de archivos importantes como el shadow.	Inhabilitando el Ctrl+Alt+Del

Nota. Tipos de análisis que realiza el atacante para explorar las configuraciones.
Elaborado por: José Benítez

1.4.2.3.1 Detalle de grupos del servidor

El objetivo de los grupos es dar o restringir permisos sobre algunos archivos a ciertos usuarios es por esto la importancia de tener bien clasificados los grupos, sus integrantes y sus permisos.

Con esta información se puede empezar a enjaular sus propios directorios y archivos, es decir asegurarse de que cada usuario solo podrá utilizar lo que necesita. Este análisis se basa en el concepto del mínimo privilegio posible, como información vital para esto se necesita saber cuántos grupos se tiene y cuáles son los integrantes de cada grupo.

Cada usuario tiene un grupo principal o puede pertenecer a diversos grupos y si conoce la clave de algún grupo puede volverse miembro durante una sesión, esto es una vulnerabilidad que los atacantes consideran importante.

En el archivo */etc/group* se encuentran almacenados los datos sobre los grupos creados en el sistema así como los miembros que pertenecen a cada grupo, y para que la información sea más comprensible se filtra con el uso de AWK, de manera que en todas las distribuciones escogidas para este proyecto se ejecuta el siguiente comando:

*awk -F":" '{print "Nombre del grupo= "$1" *GID= "$3" *Miembros= ("$4")"}' /etc/group*

Se obtiene información procesada que representa lo siguiente:

- **Grupo**: Nombre del grupo
- **GID**: Identificador que indica a cual grupo pertenece el usuario
- **Miembros**: Usuarios pertenecientes al grupo

En la figura 16 se tiene un ejemplo de la respuesta que se obtiene al ejecutar el comando mencionado previamente desde la distribución Ubuntu.

Archivo */etc/group* procesado con *AWK* desde Ubuntu.

```
root@ubuntu:~# awk -F":" '{print "Nombre del grupo= "$1" *GID= "$3" *Miembros= ("$4")"}' /etc/group
Nombre del grupo= root *GID= 0 *Miembros= ()
Nombre del grupo= daemon *GID= 1 *Miembros= ()
Nombre del grupo= bin *GID= 2 *Miembros= ()
Nombre del grupo= sys *GID= 3 *Miembros= ()
Nombre del grupo= adm *GID= 4 *Miembros= (syslog,oswaldo,admin)
Nombre del grupo= tty *GID= 5 *Miembros= ()
Nombre del grupo= disk *GID= 6 *Miembros= ()
Nombre del grupo= lp *GID= 7 *Miembros= ()
Nombre del grupo= mail *GID= 8 *Miembros= ()
Nombre del grupo= news *GID= 9 *Miembros= ()
Nombre del grupo= uucp *GID= 10 *Miembros= ()
Nombre del grupo= man *GID= 12 *Miembros= ()
Nombre del grupo= proxy *GID= 13 *Miembros= ()
Nombre del grupo= kmem *GID= 15 *Miembros= ()
Nombre del grupo= dialout *GID= 20 *Miembros= ()
Nombre del grupo= fax *GID= 21 *Miembros= ()
Nombre del grupo= voice *GID= 22 *Miembros= ()
Nombre del grupo= cdrom *GID= 24 *Miembros= (oswaldo)
Nombre del grupo= floppy *GID= 25 *Miembros= ()
Nombre del grupo= tape *GID= 26 *Miembros= ()
Nombre del grupo= sudo *GID= 27 *Miembros= (oswaldo,admin)
Nombre del grupo= audio *GID= 29 *Miembros= (pulse)
Nombre del grupo= dip *GID= 30 *Miembros= (oswaldo)
Nombre del grupo= www-data *GID= 33 *Miembros= ()
Nombre del grupo= backup *GID= 34 *Miembros= ()
Nombre del grupo= operator *GID= 37 *Miembros= ()
```

Figura 16. Elaborado por: José Benítez

1.4.2.3.2 Listado de recursos exportados por NFS

En este análisis se involucra un servicio RPC (Remote Procedure Call) es un protocolo que facilita la capacidad de ejecutar código o un programa desde una

fuente remota, se utiliza para acceder a servicios de red tales como la comparación de archivos NFS, diversas vulnerabilidades presenta este servicio y la mayoría de los ataques distribuidos de denegación de servicio se hacen por este protocolo.

Se recomienda en el caso de ser posible desactivar o eliminar estos servicios o utilizarlo solamente cuando sea necesario y si se va a compartir algún recurso se recomienda crear un nuevo directorio y asignar los permisos correspondientes de solo lectura al directorio especificado

Este archivo contiene una lista de entradas, cada entrada indica un volumen que se comparte y cómo se comparte.

En la figura 17 que se puede visualizar en la siguiente página tenemos un ejemplo de configuración del archivo /etc/exports desde OpenSUSE:

Recursos exportados NFS en OpenSUSE.

```
linux-m3qe:/etc # cat /etc/exports
# See the exports(5) manpage for a description of the syntax of this file.
# This file contains a list of all directories that are to be exported to
# other computers via NFS (Network File System).
# This file used by rpc.nfsd and rpc.mountd. See their manpages for details
# on how make changes in this file effective.

/var/RecursosCompartidosNFS/     192.168.1.0/24(rw,sync,no_root_squash,no_all_squash)
linux-m3qe:/etc #
```

Figura 17. Elaborado por: José Benítez

En las opciones de permisos NFS que se especifican después de la IP se define las opciones para cada máquina, se describirá el tipo de acceso que la máquina tendrá. Las alternativas que se pueden escoger se detallan en la tabla 13:

Tabla 13.
Opciones de exports del servicio NFS

Opción	Descripción
rw	Permite lectura y escritura en un volumen NFS.
ro	Permite sólo peticiones de lectura en un volumen NFS.
sync	Responde a las solicitudes sólo después de los cambios que se han cometido al almacenamiento estable. (Por defecto)
async	Esta opción permite que el servidor NFS se comunique asíncronamente y responder a las peticiones antes de que los cambios realizados por dicha solicitud se han ejecutado al almacenamiento estable.
secure	Esta opción requiere que las solicitudes que se originan desde la internet sea de un puerto menor a 1024 IPPORT_RESERVED). (Por defecto)
insecure	Esta opción acepta todos los puertos.

wdelay	Tiempo para aceptar una petición de escritura en el disco ligeramente si sospecha que otra petición de escritura relacionada puede estar en curso o puede llegar pronto. (Por defecto)
no_wdelay	Si un servidor NFS recibió principalmente pequeñas solicitudes no relacionadas, este comportamiento podría en realidad reducir el rendimiento, por lo no_wdelay está disponible para apagarlo.
subtree_check	Esta opción permite el control del subárbol. (Por defecto)
no_subtree_check	Esta opción desactiva el control del subárbol, que tiene implicaciones de seguridad, pero puede mejorar la confiabilidad en algunas circunstancias.
root_squash	Mapa peticiones de UID / GID 0 a los anónimos UID / GID.
no_root_squash	No activada la opción root_squash.
all_squash	Mapea todas las UID y GID de usuarios anónimos.
no_all_squash	Desactiva all_squash es decir no permite el mapeo de usuarios anónimos
anonuid=UID anongid=GID	Estas opciones establecen explícitamente el UID y GID de la cuenta anónima. Esta opción es útil cuando se desea permitir el acceso solo a un usuario o a un grupo específico.

Nota. Las opciones para la configuración del servicio NFS. Fuente: (Peña, 2014)
Elaborado por: José Benítez

1.4.2.3.3 Listado de usuarios FTP

El protocolo de transferencia de archivos FTP es antiguo y fue diseñado para transferir archivos en la red, todas las transacciones entre el cliente y servidor, la autenticación de usuarios, no están cifradas, por lo tanto se considera un protocolo inseguro y debe configurarse con cuidado.

Los atacantes podrían enviar un archivo (*shell script .sh*) donde ejecute su ataque a través de FTP, es importante definir qué usuarios tendrán acceso a este servicio y consecuentemente proteger el directorio donde se va a trabajar.

Lo recomendable seria enjaular el espacio de trabajo para este servicio y de ser necesario cambiar la configuración del puerto donde se configura el servicio FTP.

Para este tipo de análisis se necesita los comandos que se detallan en la tabla 14.

Tabla 14. *Configuraciones servicio FTP.*

Fedora y Centos	Ubuntu y OpenSUSE
cat /etc/vsftpd/ftpusers cat /etc/vsftpd/user_list cat /etc/vsftpd/vsftpd.conf	cat /etc/vsftpd.conf cat /etc/ftpusers

Nota. Comandos para la visualización de la información
Elaborado por: José Benítez

1.4.2.3.4 Listado usuarios para CRON

El demonio CRON o conocido también como *crontab* permite realizar acciones regularmente, o calendarizar procesos, todo servidor tiene calendarizadas algunas tareas que apuntan en muchos casos a archivos (shell script .sh), de esta forma se consigue que el sistema se actualice, cada periodo de tiempo pude ser una vez a la semana, mes o año.

> ¿Pero qué pasa si precisamente esta característica nos supone un problema? Tal vez sea preferible que los usuarios no puedan programar tareas en el sistema, ya sea por la falta de necesidad o por los peligros que esto conlleva (bombas DoS programadas a través del *crontab*). Por suerte también hay solución para este problema, si miramos la descripción de *CRON* ejecutando (*man cron)* veremos cómo se nombran dos ficheros; */etc/cron.allow* y */etc/cron.deny*. (Ferran Pichel, 2011, págs. 18,19)

Actualmente en las nuevas distribuciones de Ubuntu estos ficheros ya no se utilizan, porque se considera que solamente el usuario root será el que pueda usar el fichero.

crontab.

El funcionamiento de los ficheros mencionados es el siguiente;

> El fichero */etc/cron.allow*, en caso de existir, es el único al que *CRON* hace caso, es decir, los usuarios que no aparezcan en esta lista no tendrán acceso a *crontab*.
>
> En cambio si en vez de querer evitar a todos los usuarios y deseamos permitir solo a unos pocos, basta con añadir su nombre al fichero */etc/cron.deny* e inhabilitar el fichero */etc/cron.allow* borrándolo o moviéndolo
>
> Por ejemplo, si queremos vetar a todo el mundo y creamos el */etc/cron.deny*, cuando algún usuario intente usar *crontab* recibirá el siguiente mensaje:
>
> *root # touch /etc/cron.allow*
>
> *usuario$ crontab –e*
>
> You (usuario) are not allowed to use this program (crontab)
>
> See crontab(1) for more information. (Ferran Pichel, 2011, págs. 18,19)

Para este análisis se ejecutan los siguientes comandos detallados en la tabla 15:

Tabla 15.
Comandos obtención información permisos a CRON

Distribución	Listado usuarios para cron
CENTOS	cat /etc/at.deny cat /etc/cron.deny
FEDORA	cat /etc/at.deny cat /etc/cron.deny
UBUNTU	cat /etc/cron.d/anacron
OPENSUSE	cat /etc/at.deny cat /etc/cron.deny

Nota. Elaborado por: José Benítez

1.4.2.3.5 Políticas de cuentas

El fichero */etc/login.defs* es el que contiene los parámetros por defecto para el sistema de autenticación estándar de GNU/Linux. Muchos de sus parámetros han caído en el desuso debido a la adopción de *PAM (Módulos de autenticación conectables son un marco común para la autenticación y seguridad)* como mecanismo de autenticación. Sin embargo, para tratar de fortificar la configuración del sistema existen algunos parámetros que pueden resultar interesantes. Los parámetros relevantes para su posterior modificación se pueden visualizar en la figura 18.

Configuración políticas de cuentas recomendaciones.

```
...
PASS_MAX_DAYS   30    # Caducidad del password en días
PASS_WARN_AGE   5     # Aviso de caducidad de password en días

UMASK           077   # Máscara por defecto para creación de ficheros
LOGIN_RETRIES   1     # Número máximo de intentos de login
LOGIN_TIMEOUT   10    # Timeout en la pantalla de login. Expresado en segundos
...
```

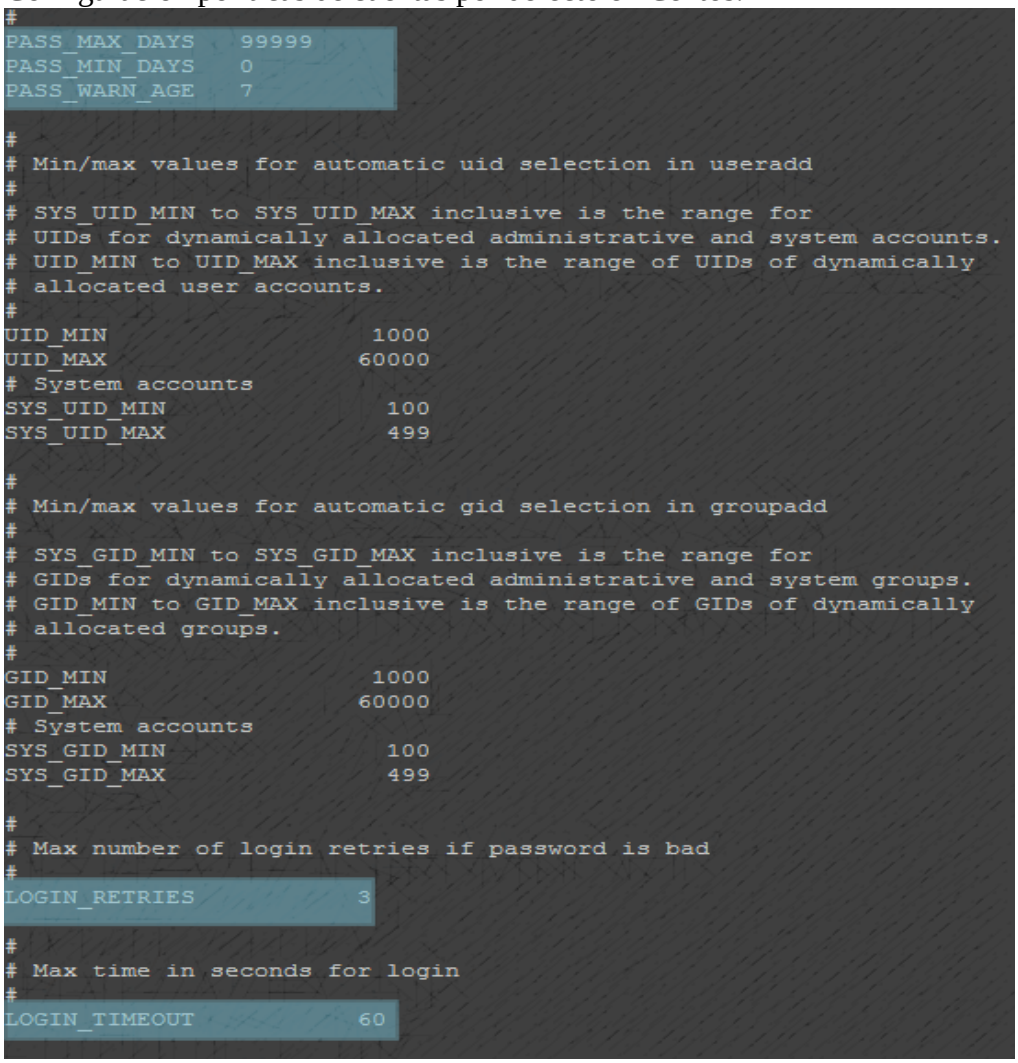
Figura 18. Fuente: (Álvarez Martín & Gonzales Pérez, 2013, pág. 178)

Con los primeros parámetros del fichero */etc/login.defs* se están generando unas entradas en el fichero */etc/shadow.*

Los campos con el valor 99999 y 7 corresponden respectivamente con los valores por defecto mencionados en el fichero */etc/login.defs* (Álvarez Martín & Gonzales Pérez, 2013, pág. 178)

Los sistemas operativos por defecto tienen la configuración menos recomendable según las técnicas de HARDENING este análisis es uno de los más importantes, en la figura 19 se puede ver la configuración por defecto.

Configuración políticas de cuentas por defecto en Centos.



```
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7

#
# Min/max values for automatic uid selection in useradd
#
# SYS_UID_MIN to SYS_UID_MAX inclusive is the range for
# UIDs for dynamically allocated administrative and system accounts.
# UID_MIN to UID_MAX inclusive is the range of UIDs of dynamically
# allocated user accounts.
#
UID_MIN                  1000
UID_MAX                 60000
# System accounts
SYS_UID_MIN               100
SYS_UID_MAX               499

#
# Min/max values for automatic gid selection in groupadd
#
# SYS_GID_MIN to SYS_GID_MAX inclusive is the range for
# GIDs for dynamically allocated administrative and system groups.
# GID_MIN to GID_MAX inclusive is the range of GIDs of dynamically
# allocated groups.
#
GID_MIN                  1000
GID_MAX                 60000
# System accounts
SYS_GID_MIN               100
SYS_GID_MAX               499

#
# Max number of login retries if password is bad
#
LOGIN_RETRIES           3

#
# Max time in seconds for login
#
LOGIN_TIMEOUT           60
```

Figura 19. Elaborado por: José Benítez

1.4.2.3.6 Gestor de arranque GRUB 2

En la segunda versión del gestor de arranque se modifican bastantes aspectos y se apuesta por ofrecer mayor flexibilidad a la hora de configurarlo, todo ello mediante scritpts para automatizar configuraciones y nuevas directivas de configuración entre otros cambios. A pesar de ello el aspecto sigue siendo casi idéntico al de GRUB.

Centrado la atención en la protección mediante password, la principal novedad radica en que es posible la creación de roles y grupos de usuarios con diferentes privilegios en GRUB2. Así pues, es posible disponer del rol por defecto *superusers* y agregar a él diferentes usuarios que obtendrán la

posibilidad de acceder a la terminal del GRUB y modificar entradas de arranque. Para esta configuración se centrará únicamente en el rol *superusers.*

Como ocurría en la primera versión del GRUB, puede establecerse el password en texto plano o cifrado además de bloquear diferentes líneas de arranque. La diferencia es que si se escoge la opción de cifrado, ya no se establecerá en *md5*, sino que en su defecto se utiliza *pbkdf.* La herramienta que se sugiere utilizar para generar el password cifrado es *grub-mkpasswd-pbkdf2*.

Es recomendable hacerlo en el fichero */etc/grub.d/40_custom*. Nuevamente, como sugerencia se ha utilizado un password cifrado y el fichero final tendrá el siguiente aspecto como se ve en la figura 20. (Álvarez Martín & Gonzales Pérez, 2013, pág. 32)

Estableciendo un password en GRUB2.

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.

set superusers="admin"
password_pbkdf2 admin grub.pbkdf2.sha512.10000.BC65DD5C796105A485F223BD3FE4113F
9BA8B7C3FABC101989B39683FC303144482E4109B3F7333548E7D512709F2D2895C0A795B32948D
5A30B7BCD1BA23B.10A75AFBF5A5E3BFE33BA189173FC17B27D6714AA3397B58F5F18251BA40E4F
AD40D4010598D80D25798B228C64A0AB76EB1D5AA0C707EAA4FF4955250949BD
```

Figura 20. Fuente: (Álvarez Martín & Gonzales Pérez, 2013, pág. 32)

Para que la configuración sea efectiva es necesario regenerar el fichero de configuración de GRUB. Para ello se utiliza el comando *update-grub,* que leerá todos los SCRIPTS incluyendo el */etc/grub.d/40_custom*. En el momento en que se reinicie la máquina podrá comprobar que la consola no es accesible a menos que se introduzcan credenciales establecidas. De igual modo estará restringido el acceso a la edición de las líneas de arranque. En el caso del ejemplo, había que introducir el nombre de usuario "admin" y la contraseña establecida. (Álvarez Martín & Gonzales Pérez, 2013, pág. 32)

El comando que utilizamos para obtener la información será el mismo en todas las distribuciones escogidas para este proyecto de grado.

cat /etc/grub.d/40_custom

1.4.2.3.7 Protección de LOGS

Se debe configurar los LOGS del sistema para evitar lecturas por parte de otros usuarios distintos a root o el encargado de los LOGS.

Supongamos el caso que un usuario es el encargado de los LOGS, llamemos le *logger*, y queremos que sea el único con acceso. Bastará entonces con cambiar el propietario de los LOGS a *logger* y darle los permisos adecuados, tal y como se ha hecho con los directorios *home* de los usuarios. Claro que a lo mejor interesa que *logger* solo pueda leer los ficheros, sin llegar a modificarlos, y no queremos que pueda cambiarse los privilegios porque entonces podría saltarse esa prohibición.

Entonces se podría crear un grupo llamado *LOGS*, añadirle el usuario *logger* y cambiar el grupo del directorio de los LOGS, además de asignarle permisos de solo lectura para el grupo.

Paso 1.- Crear el grupo logs:

 # addgroup logs

Paso2.- Añadimos el usuario especificado en este ejemplo *logger* al grupo logs:

 # gpasswd -a logger logs

Paso 3.- Ahora cambiamos el grupo del directorio de los logs, generalmente */var/log*:

 # chown :logs /var/log/

Paso 4.- Asignamos los permisos pertinentes (u=rwx , g=rx, o=-):

 # chmod 750 /var/log

Ahora que ya tenemos el directorio protegido para el resto de usuarios sólo es necesario permitir leer todos los ficheros del directorio al usuario *logger*, como no es el propietario ni forma parte del grupo de muchos de los ficheros, se deberá dar permisos de lectura al resto de usuarios "o". La gracia está en que con los permisos del directorio */var/log* ya no se permite la entrada a éste, entonces dar permisos de lectura al resto, significa dar permisos de lectura a *logger*, porque los demás usuarios no podrán entrar en el directorio.

Paso 5.- Asignamos los permisos de lectura al resto de usuarios

*# chmod o+r /var/log/**

Paso 6.- Finalmente vemos los permisos del directorio y de su contenido:

 # ls -ld /var/log

drwxr-x--- 5 root logs 4096 2006-10-03 10:08 /var/log

 # ls -l /var/log

[..]

-rw-r--r-- 1 root adm 133812 2006-10-03 11:08 messages

[..]

Ya tenemos los LOGS protegidos contra los ojos curiosos que pueda haber en nuestro servidor. Hay que tener en cuenta que bloquear el acceso al recurso */var/log* puede acarrear problemas en alguno de los comandos que cogen información de ahí, como por ejemplo *lastlog*, sólo podrán realizar correctamente dicho comando el usuario *root* y los miembros del grupo *logs*.

Este pequeño ejemplo realizado sobre el directorio de *logs*, es aplicable a cualquier otra operación de este estilo, donde un usuario es el encargado de un recurso en concreto. Además si queremos añadir un segundo usuario para realizar la misma acción bastará con añadirlo al grupo, "LOGS" en este caso, y ya tendrá los mismos privilegios que el primero.

Dependiendo del tipo de servidor, los permisos deberán ser más o menos restrictivos, eso ya depende de cada caso en concreto y del administrador que esté a cargo.

Lo que sí es generalizable es el minucioso cuidado que se debe tener con los ejecutables con el bit SUID activo, se deben extremar las precauciones y mantenerlo fuera del alcance de los usuarios. De lo contrario, si existiera un bug en algún ejecutable de este tipo o se hiciera un mal uso de éste, un usuario podría llegar a realizar una escalada de privilegios y ser entonces *root*, lo que comprometería el sistema. (Ferran Pichel, 2011, págs. 9,10)

En la figura 21 tenemos un ejemplo de una configuración por defecto en la distribución Ubuntu que tiene una configuración no segura.

Permisos directorio */var/log* en Ubuntu.

```
root@ubuntu:~# ls -ld /var/log
drwxrwxr-x 14 root syslog 4096 Jan  5 09:03 /var/log
root@ubuntu:~#
```

Figura 21. Elaborado por: José Benítez

1.4.2.3.8 Inhabilitando el Ctrl+Alt+Del

Dependiendo del entorno en el que se encuentre el servidor, es aconsejable deshabilitar el reinicio por teclado utilizando la secuencia de teclas:

Ctrl.+Alt+Del.

Para hacerlo basta con ir al fichero */etc/inittab* en las distribuciones Fedora y OpenSUSE que se encarga del comportamiento de los RunLevel, o estados del sistema. Ahí podemos encontrar la siguiente línea:

 # What to do when CTRL-ALT-DEL is pressed.

ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

En la tabla 16 tenemos la descripción de cada valor en la línea de configuración mencionada anteriormente.

Tabla 16.
Descripción configuración fichero /etc/inittab

Parámetros	Significado	Valor
Identificador	Id único en /etc/inittab	ca
RunLevels	RunLevels a los que afecta	12345
Acción	En qué acción se ejecutará el proceso	Ctrl alt del
Proceso	Proceso a ejecutar	/sbin/shutdown
Parámetros	Parámetros del proceso	-t1 -a -r now

Nota. Fuente: (Ferran Pichel, 2011, pág. 15)
Elaborado por: José Benítez

Esta sentencia define la acción que se llevará a cabo cuando se produzca la acción *CtrlAltDel* en cualquiera de los runlevel (1-5), excepto el estado de apagado (0) y reboot(6). Lo que ocurrirá es que se llamara a */sbin/shutdown* con los parámetros especificados y por lo tanto se cerrará el sistema (*-r now*) al instante. (Ferran Pichel, 2011, pág. 15)

Recomendaciones:

a) Pasos para las distribuciones OpenSUSE y Fedora

Para distribuciones de GNU/Linux donde se utiliza el tradicional *SystemV* para la gestión de tareas y servicios durante el inicio del sistema, sólo es necesario editar el archivo */etc/inittab:*

Paso 1: Abrimos el archivo */etc/inittab*

[root@oswaldo-fedora ~]# vi /etc/inittab

Paso 2: Localice lo siguiente:

ca::ctrlaltdel:/sbin/shutdown -t3 -r now

Paso 3: Comente la línea anterior con una almohadilla:

 # ca::ctrlaltdel:/sbin/shutdown -t3 -r now

Paso 4: Para que apliquen de inmediato los cambios se debe ejecutar:

[root@oswaldo-fedora ~]# init q

b) Para las distribuciones Centos y Ubuntu

Para distribuciones de GNU/Linux que utilizan *Upstart* para la gestión de tareas y servicios durante el inicio, se edita el archivo */etc/init/control-alt-delete.conf:*

Paso 1: Abrimos el archivo */etc/inittab*

root@ubuntu:~# vi /etc/init/control-alt-delete.conf

Paso 2: Localice lo siguiente:

start on control-alt-delete

exec /sbin/shutdown -r now "Control-Alt-Delete pressed"

Paso 3: Comente la última línea y añada otra que simplemente se envíe un mensaje informativo al pulsar la combinación de teclas *Control-Alt-Delete*.

start on control-alt-delete

#exec /sbin/shutdown -r now "Control-Alt-Delete pressed"

exec echo "Control-Alt-Delete desactivado por el administrador"

Paso 4: Para aplicar de inmediato los cambios, ejecute:

root@ubuntu:~# initctl reload-configuration

Cabe señalar que si el sistema actualiza el paquete *upstart*, el archivo */etc/init/control-alt-delete.conf* será sobre-escrito y se perderán los cambios realizados, por lo que puede ser conveniente realizar todo lo anterior en un archivo denominado */etc/init/control-alt-delete.override*.

1.5. Metodología de desarrollo SCRUM

SCRUM es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto, es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil, incremental basada en iteraciones y revisiones.

Se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo, en este caso se define con el director del proyecto el tiempo para la revisión de los avances.

Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto.

Estas iteraciones son la base del desarrollo ágil, y SCRUM gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente. (Palacio, 2006, pág. 2)

En la figura 22 podemos ver los elementos de la metodología mencionada, donde se puede apreciar que todo inicia a partir de la pila del producto que es un listado de funcionalidades que debe cumplir el proyecto, pero para ello se necesitan actividades o avances para conseguir una funcionalidad, a estos avances se los conoce como sprint que se los revisa cada 15 o 30 días dependiendo el escenario, estas interacciones se repiten hasta tener un incremento final que será una funcionalidad creada.

Elementos de la metodología SCRUM.

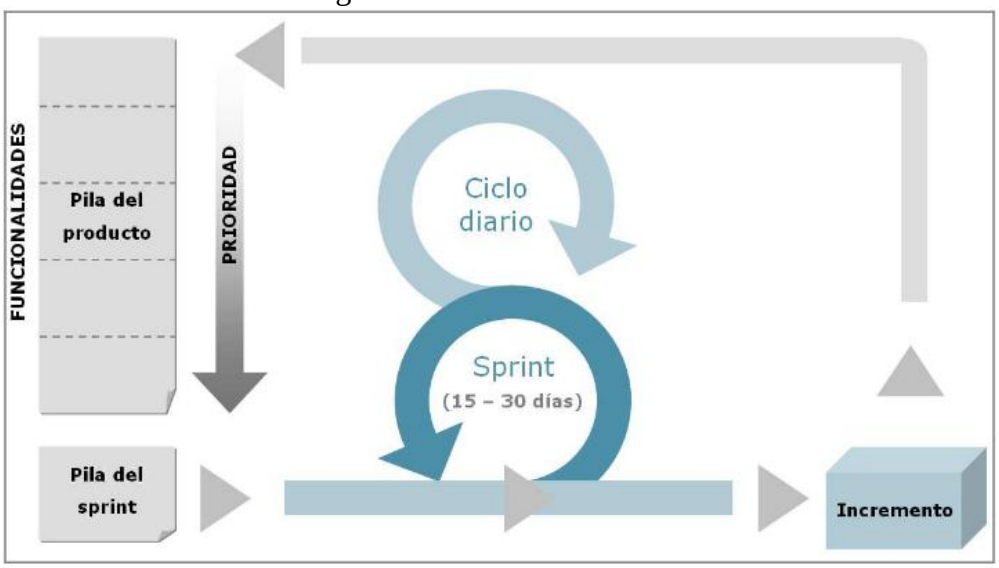


Figura 22. Fuente: (Palacio, 2006, pág. 3)

Una de sus ventajas es que es muy fácil de entender y requiere de poco esfuerzo para comenzar a usarse.

Una parte muy importante de SCRUM son las reuniones que se realizan durante cada una de las iteraciones. Hay distintos tipos:

- SCRUM diario: cada día durante la iteración, tiene lugar una reunión de estado del proyecto. A esta reunión se le domina SCRUM
- Reunión de planificación de iteración (sprint): se lleva a cabo al principio del ciclo de la iteración.
- Reunión de revisión de iteración: al final del ciclo de la iteración.
- Iteración retrospectiva: al final del ciclo de la iteración. (INTECO Instituto Nacional de Tecnologías y la comunicación, 2009, pág. 66)

Podría justificarse el uso de esta metodología puesto que es la adecuada para alcanzar el desarrollo del presente trabajo de grado porque facilita una constante evaluación de los avances del producto, debido a que es una metodología incremental e iterativa.

Con la metodología SCRUM se podrá tener un control en cada una de las etapas del ciclo de vida del software, facilitando las entregas de SPRINTS o interacciones al tutor de trabajo de titulación.

CAPÍTULO 2

FASE INICIAL Y DEFINICIÓN

2.1 Fase inicial

Se la puede llamar también fase de análisis porque es donde vamos a identificar los requerimientos y el producto backlog que corresponde a la metodología SCRUM, se analiza viabilidad técnica y financiera.

2.1.1 Definición del proyecto

Analizar, diseñar y desarrollar una aplicación, para Windows, de técnicas de HARDENING sobre sistemas operativos Linux para la obtención de reportes dirigido para administradores de sistemas operativos.

Con la aplicación se podrá conectar a servidores GNU/Linux de diferentes familias padres de todas las distribuciones existentes, se han escogido las siguientes distribuciones; Fedora, OpenSUSE, Centos, Ubuntu. Esta conexión es a través del protocolo SSH (Secure Shell) compatible con la versión ssh1 y ssh2.

Esta aplicación puede establecer conexiones síncronas por el protocolo ssh, simultáneamente a múltiples equipos máximo 4, se definen los parámetros de conexión como; Distribución, IP, usuario (root), contraseña.

Establecida ya la conexión se deben manejar conexiones independientes con el uso correcto de hilos (Thread). Esto se replica en 4 conexiones independientes entre ellas, porque se aplican hilos para la optimización de los recursos de hardware disponibles, antes de proceder a analizar debemos asegurarnos de tener los privilegios de root haciendo un loggin nuevamente ya en la conexión establecida con los comandos “su –“ e ingresando la contraseña correspondiente.

Definida, controlada y preparada la capa de comunicación se dispondrá de una interfaz donde se procede a realizar la configuración de auditoría donde se debe seleccionar qué tipo de análisis deseamos obtener, se pueden escoger todos o seleccionar los necesarios, después de esto se procede a almacenar en un vector los comandos correspondientes a ejecutar y las respuestas se imprimen en un *textbox*, para posteriormente almacenarlos en un nuevo vector en el módulo de reportes, este

escenario es repetido por cada conexión existente y por lo tanto se tiene un vector por cada conexión. Los comandos a ejecutarse son almacenados, pero esto varía en función de la distribución del servidor conectado.

Para el módulo de reportes se podrá visualizar solo un servidor a la vez y en un nuevo formulario se podrá escoger uno por uno los tipos de análisis que escogimos previamente en la configuración de auditoría y nos mostrará el resultado obtenido conjuntamente con las recomendaciones correspondientes basadas en los principios técnicos de HARDENING, para cada tipo de análisis existe una recomendación o información conjuntamente con el resultado obtenido, para algunos tipos de análisis se tendrá algo similar a una alerta donde detecta directamente una vulnerabilidad, y finalmente se podrá exportar todo el análisis en un documento de Microsoft Word, en un directorio por defecto, los reportes obtenidos aportarán a la toma de decisiones y medidas preventivas de los administradores de sistemas operativos.

2.1.2 Análisis de requerimientos funcionales y no funcionales

Con la metodóloga SCRUM hay que definir la “pila del producto" o conocido también como “pila de tareas” o “backlog”, detallando la prioridad de cada uno.
Para este proyecto se tiene una escala del 1 al 10 siendo 10 más prioritario y 1 menos prioritario, y finalmente sus días estimados en función de la complejidad y alcance, se considera que son 7 horas eficientes de trabajo por cada día, en la tabla 17 se puede ver el número de funcionalidad, el nombre, la descripción, la prioridad y la estimación en días.

Tabla 17.
Pila del producto

Pila de producto SHL (Sistema HARDENING Linux)				
#	**Nombre**	**Descripción**	**Prioridad**	**Tiempo estimado días**
1	Ambiente de desarrollo y pruebas	Diseñar la arquitectura de la aplicación y preparar ambiente de desarrollo y pruebas.	5	5
2	Módulo de conexión	Desarrollar un módulo conexión que maneje el protocolo ssh, de forma síncrona, con la capacidad de almacenar los comandos en una cola fifo y envié al server los comandos después de que termine de ejecutar cada comando, con el uso de banderas.	10	20

3	Módulo de comunicación	Desarrollar un módulo de comunicación que contenga las consolas de conexiones para 4 servidores, y sea el vínculo a los siguientes módulos.	10	20
4	Recomendaciones de HARDENING	Investigar técnicas de HARDENING y redactar recomendaciones, definir los comandos a ejecutar, terminar capítulo 1.	5	30
5	Módulo de configuración de auditoría	Desarrollar un módulo de configuración de auditoría para la configuración de los comandos a ejecutar para los 4 servidores.	8	20
6	Módulo de reportes	Desarrollar un módulo para la obtención de reportes y recomendaciones para cada servidor escaneado, basados en técnicas de HARDENING.	10	30
7	Pruebas	Revisar, corregir, integrar todos los módulos del software.	7	20
8	Documentación	Documentación en normas APA.	8	50
TOTAL DÍAS				195

Nota. Cada día contiene 7 horas efectivas de trabajo. Elaborado por: José Benítez

2.1.2.1 Requerimientos funcionales

Son aquellos requerimientos del sistema que expresan una actividad o tarea para que el software cumpla con el objetivo para el cual fue creado, cada requerimiento funcional se los agrupa y asocia en módulos, para este software se ha identificado 4 módulos:

- **Módulo conexiones ssh**

  Todo el proceso de inicio de la conexión de un socket, encriptación, negociación de las llaves privadas y el manejo del stream de lectura y escritura, se las simplifica con el uso de una librería externa de nombre SharpSSH.

  SharpSSH es una biblioteca para VB.Net y Java, para el uso de Secure Shell (SSH) tiene una suite completa de clases para la versatilidad y escalabilidad del protocolo en sus dos versiones SSH1 y SSH2.

  Con esta librería es posible conectar con servidores a través de SSH y se puede integrar en cualquier aplicación de Framework, la biblioteca se distribuye bajo licencia estilo BSD (Berkeley Software Distribution) es una licencia de software libre permisiva.

  SharpSSH permite leer y escribir datos y transferir archivos a través de canales SSH. Además, proporciona algunas clases adicionales que hacen aún más simple la comunicación por SSH.

Se debe estructurar una capa de lectura y escritura que sincronice las respuestas obtenidas por el STREAM que estará abierto, así mismo se deberá establecer una arquitectura con una cola *FIFO (First In, First Out)* que detecte si hay comandos a ejecutar por parte del cliente y deberá estar siempre escuchando, esto se lo debe controlar mediante el uso de hilos *threads* que deberán manejar múltiples conexiones simultáneas síncronas.

Estas conexiones son independientes porque tienen una arquitectura síncrona y cada una maneja un hilo diferente, para garantizar las conexiones a 4 terminales, es decir existirán 4 clases iguales.

- **Módulo de comunicación**

  Tendrá un formulario previo para el ingreso o de autentificación a la aplicación y otro formulario para inicializar todo el proceso de conexión a través del protocolo ssh utilizando el módulo de conexiones ssh deberá pasar a este como parámetros; dirección ip, contraseña, usuario administrador y que tipo de distribución tiene el servidor a escanear.

  Está sincronizado con el módulo de conexiones ssh y enlaza los hilos de lectura a un evento público que tendrá el módulo de conexiones, de igual manera deberá controlar el manejo de hilos y evitar un volcado de memoria a causa de hilos huérfanos, debe optimizar el uso del procesador con el uso de excepciones.

  Este módulo de comunicación es el central porque es un vínculo con los otros módulos, desde aquí se accede al resto, así mismo tendrá 4 consolas, 4 botones para desconectar y 4 botones para obtener información del protocolo ssh y 4 botones para acceder al módulo de reportes de la conexión ya establecida.

- **Módulo configuración de auditoría**

  Se accede después de haber establecido una conexión adecuada a un servidor mediante el módulo de comunicación, como parámetros se necesita saber qué número de servidor es y la distribución.

  Se necesita una interfaz gráfica para la configuración de auditoría donde se escogerá de un listado qué tipo de información se desea obtener, eso servirá

para almacenar en un vector los comandos a ejecutarse. Después de escoger se envía al módulo de comunicación el vector y el número de servidor con el que se está trabajando para que proceda a la ejecución automática de todos los comandos almacenados.

Los comandos almacenados deben estar validando la distribución a la que pertenece porque en algunos casos varían los directorios y los nombres de los archivos de configuración.

- **Módulo de reportes**

  Se accede desde el módulo de comunicación después de que haya terminado de ejecutarse y de imprimirse los resultados de todos los comandos ejecutados en el módulo de configuración de auditoría, y deberá enviar como parámetros el vector de los tipos de análisis seleccionados y un objeto donde recoja toda la información que se ha imprimido en un cuadro de texto con el número de servidor, por lo tanto se accede a este módulo para visualizar servidor por servidor uno a la vez.

  Tiene una interfaz gráfica donde se visualiza en un listado todos los tipos de análisis escogidos y al seleccionar cada uno imprime los resultados y las recomendaciones conjuntamente en un cuadro de texto.

  Las recomendaciones están almacenadas localmente en archivos de texto de formato .rtf donde se los abrirá uno a uno según corresponda.

  Este módulo permitirá a los administradores de sistemas operativos visualizar en un reporte general el resultado del escaneo con recomendaciones de cada información, permisos y configuraciones que se hayan escogido, esto servirá para la toma de decisiones, basado en la auditoría informática de sistemas operativos HARDENING. Estos reportes serán almacenados en un directorio por defecto en formato .docx que se abrirá después de que se haya cargado toda la información.

2.1.2.2 Requerimientos no funcionales

Son las necesidades no lógicas y en su lugar son los aspectos técnicos que debe incluir un sistema, son necesarios para que el proyecto pueda desarrollarse, implementarse y probarse a través del uso de herramientas para su creación.

- **Red LAN**

  Para el desarrollo y pruebas de software es necesario montar una ambiente el cual consiste en implementar una pequeña red LAN donde interactúan dos máquinas físicamente y virtualmente 5 máquinas, la máquina cliente y 4 servidores virtualizados haciendo uso de las tecnologías de virtualización en este proyecto se usa VMware Workstation.

- **Diseño de imágenes y diagramas.**
  - **Adobe Photoshop:** Es una herramienta utilizada para el diseño de imágenes, se puede editar, nuevas imágenes, esta herramienta será utilizada para las imágenes involucradas en el presente proyecto.
  - **Microsoft Visio:** Aplicación para dibujar los diagramas que ayuda a visualizar, explorar y comunicar y entender la información compleja de un software, bases de datos, diagramas de flujo de programas, *UML (Lenguaje Unificado de Modelado)*, que facilitan la comprensión de la arquitectura de un software.
  - **Power Designer:** Es una aplicación para el diseño de diagramas que abracan la ingeniería del software y se acopla a estándares internacionales, tiene más funcionalidades adicionales que generan código a partir de diagramas diseñados.
- **IDE (Entorno de desarrollo integrado).**
  - **Visual Basic .Net:** es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic.
  - **Net Framework 4.0:** Tecnología que admite la compilación y ejecución del código generado.
  - **AWK:** es un lenguaje de programación diseñado para procesar datos de archivos de texto, nativo de la familia de sistemas operativos Unix, Linux.

2.1.3 Viabilidad técnica

En el presente proyecto de grado hay que considerar que para el desarrollo y pruebas se necesita montar un ambiente en el que simulemos una red con 4 servidores y el equipo cliente donde ejecutamos la aplicación.

Para optimizar el ahorro del presupuesto en el proyecto la mejor alternativa es utilizar dos computadores físicos pero en la primera máquina se recomienda que

tenga un buen procesador con 8GB de ram, instalado windows 7 en adelante como sistema operativo base y con la herramienta de virtualización *VMware® Workstation* virtualizados 4 servidores con las siguientes características como se puede ver en la tabla 18.

Tabla 18.
Descripción de las distribuciones a utilizarse.

Descripción	Fedora	Ubuntu	Centos	OpenSUSE
Versión	20	14	7	13
Año	2014	2014	2014	2014
Software para virtualizar	*VMware® Workstation* 10	*VMware® Workstation* 10	*VMware® Workstation* 10	*VMware® Workstation* 10
Memoria ram virtualizada	2 GB	2 GB	1.5 GB	1,5 GB
Espacio físico Virtualizado	15 GB	15 GB	15 GB	15 GB
Espacio físico real	8,55 GB	6,86 GB	6,54 GB	4,76 GB
Estado del adaptador de red	Bridged Automatic	Bridged Automatic	Bridged Automatic	Bridged Automatic
Usuario administrador	root	oswaldo	root	root
Distribución base	Red Had	Debian	Red Had Enterprise	SUSE Linux
Sistema de ficheros	ext3, ext4	ext3, ext4	ext3, ext4	ext3, ReiserFS, XFS
Arquitectura del SO	x86, x86-64, i386, PowerPC	x86, x86-64, IA64	x86, x86-64, i386, s390x, PowerPC, Alpha	x86, x86-64, IA64, s390, ppc, ppc64
Servicios adicionales	OpenSSH, server FTP, NFS	OpenSSH, server FTP, NFS	OpenSSH, server FTP, NFS	OpenSSH, server FTP, NFS

Nota. Descripción de las distribuciones montadas en máquinas virtuales y sus características.
Elaborado por: José Benítez

Finalmente en la segunda máquina solamente será la que ejecuta la aplicación es decir el cliente, no es necesario que tenga muchos recursos pero al menos 3 GB de ram con un sistema operativo mínimo Windows 7 64 bits.

> En informática, virtualización es la creación a través de software una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento u otros recursos de red. Se refiere a la abstracción de los recursos de una computadora, llamada Hypervisor o VMM (Virtual Machine Monitor) que crea una capa de abstracción entre el hardware de la máquina física y el sistema operativo de la máquina virtual (virtual machine, guest), dividiéndose el recurso en uno o más entornos de ejecución.

Esta capa de software (VMM) maneja, gestiona y arbitra los cuatro recursos principales de una computadora (CPU, Memoria, Dispositivos Periféricos y Conexiones de Red) y así podrá repartir dinámicamente dichos recursos entre todas las máquinas virtuales definidas en el computador central. Esto hace que se puedan tener varios ordenadores virtuales ejecutándose en el mismo ordenador físico. (Wikipedia®, 2014),

2.1.4 Viabilidad financiera

Para el presente proyecto se consideran gastos operativos, los relacionados con pagos de honorarios incluido seguro social para el desarrollo estimado por 6 meses, suponiendo que el salario estándar es de 1000 dólares americanos por las 8 horas laborables menos la hora de almuerzo es decir 7 horas eficientes de trabajo, más los gastos que impliquen la investigación, entendiéndose como gastos en libros, internet en la tabla 19 se tiene un detalle de los valores.

Tabla 19.
Detalles de la viabilidad económica.

Detalle	Valor USD
Salario más seguro social por 6 meses	$6000
Gastos operativos por 6 meses	$480
Subtotal mensual	$6480
Gastos de investigación	$100
Licencia *VMware® Workstation*	$150
Total por 6 meses	$6730

Nota. Estimación del presupuesto de gastos
Elaborado por: José Benítez

2.2 Fase de definición

Los diagramas que implican el diseño para el desarrollo del software son los que se detallan en esta fase, diagramas *UML (Unified Modeling Language)* versión 2.2 por consiguiente se escogieron los siguientes diagramas para el presente proyecto;

- Diagramas casos de uso.
- Diagramas de secuencia.
- Diagramas de actividades.
- Diagramas de clases.

2.2.1 Diagramas de casos de uso

Debido a que el proyecto es dirigido a un administrador de sistemas operativos o a un ingeniero de soporte de servidores, y teniendo en cuenta el concepto de HARDENING del menor punto de exposición, el software solo tiene un único usuario, el mismo que tendrá acceso a todos los módulos como se puede ver en la figura 23 que se encuentra en la siguiente página.

Diagrama casos de uso general de la aplicación.

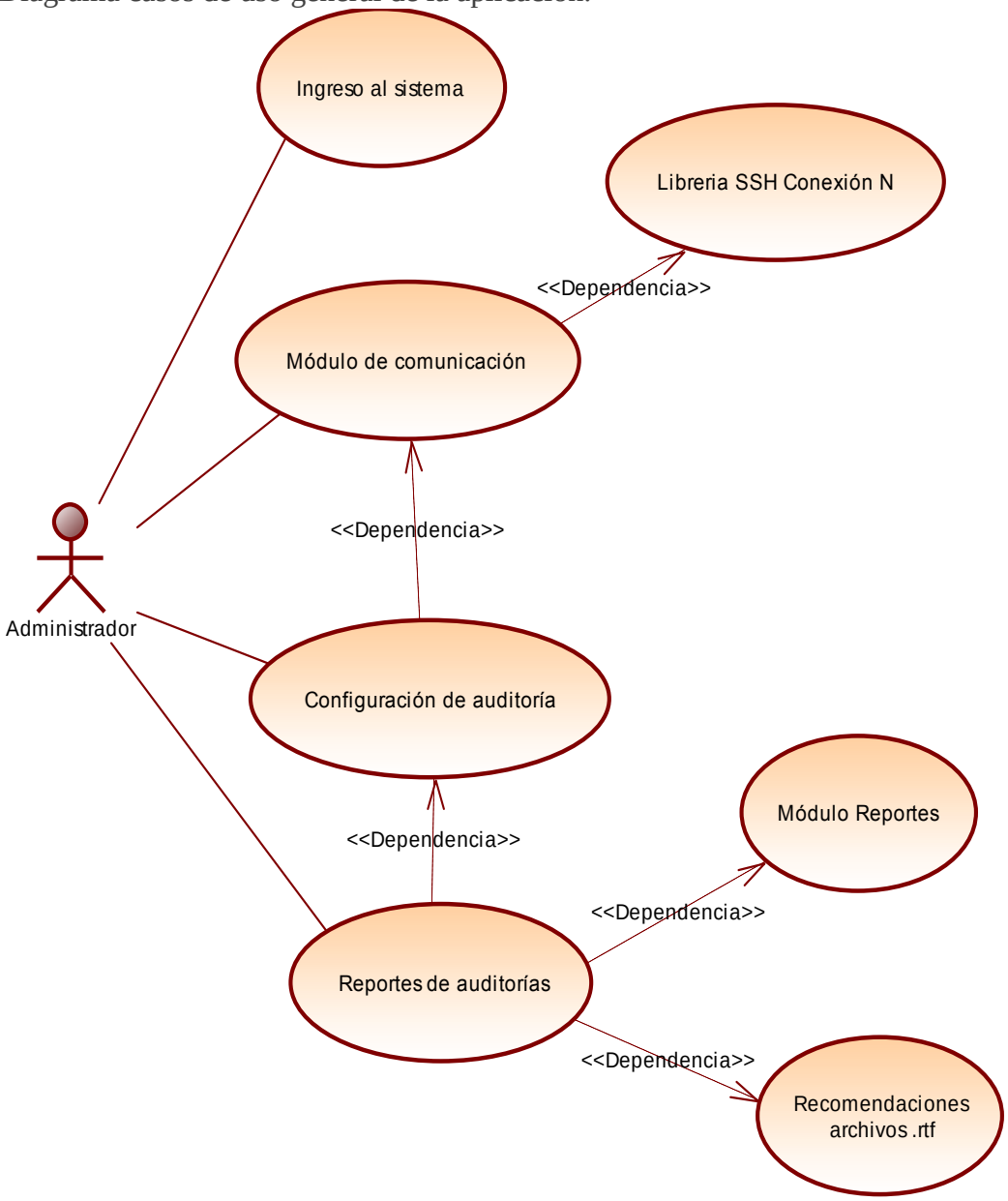


Figura 23. Elaborado por: José Benítez

Caso de uso 1: Módulo de conexión ssh

En el módulo de conexión interactúa con el formulario que está directamente relacionado a las clases con las librerías para la conexión ssh y solo involucra un único usuario como se puede ver en la figura 24.

Diagrama casos de uso módulo conexión ssh.

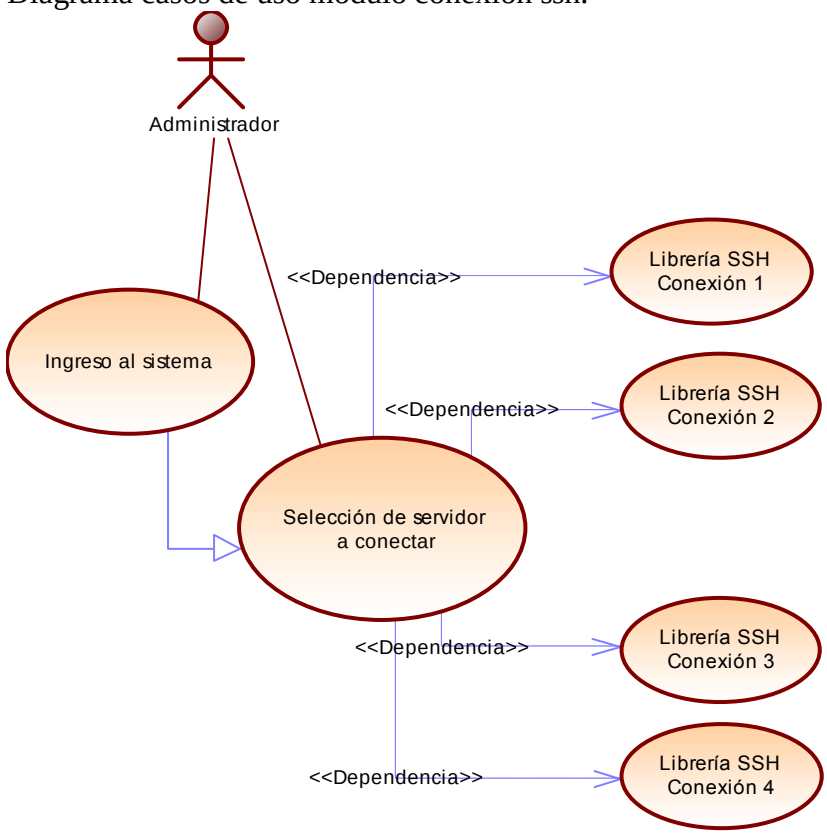


Figura 24. Elaborado por: José Benítez

En la tabla 20 podemos ver las descripciones de este caso de uso.

Tabla 20.
Detalles caso de uso conexiones ssh

Caso de uso	Conexiones SSH
ACTORES	Administrador
PRECONDICIONES	Deberá ingresar el usuario y contraseña de acceso al software, dirección ip, puerto y distribución del servidor a conectarse.
POSCONDICIONES	Se debe esperar a que se imprima la respuesta del servidor.
FLUJO BÁSICO	1. Ingresa al formulario consola de conexiones. 2. Selecciona qué servidor desea conectar. 3. Llena los parámetros de conexión y establece la conexión.
FLUJO ALTERNATIVO	1. Si no se ha enviado los parámetros de conexión correctos muestra un mensaje de error en el cuadro de texto.

Elaborado por: José Benítez

Caso de uso 2: Módulo de comunicación

En el módulo de comunicación es donde se visualiza todas las respuestas al servidor conectado y depende del módulo de conexiones para su sincronía y la clase "Librería SSH Conexión N" se comunica directamente con el servicio *OpenSSH* de cada servidor, en la figura 25 se puede ver lo interpretado con anterioridad.

Diagrama casos de uso módulo de comunicación.

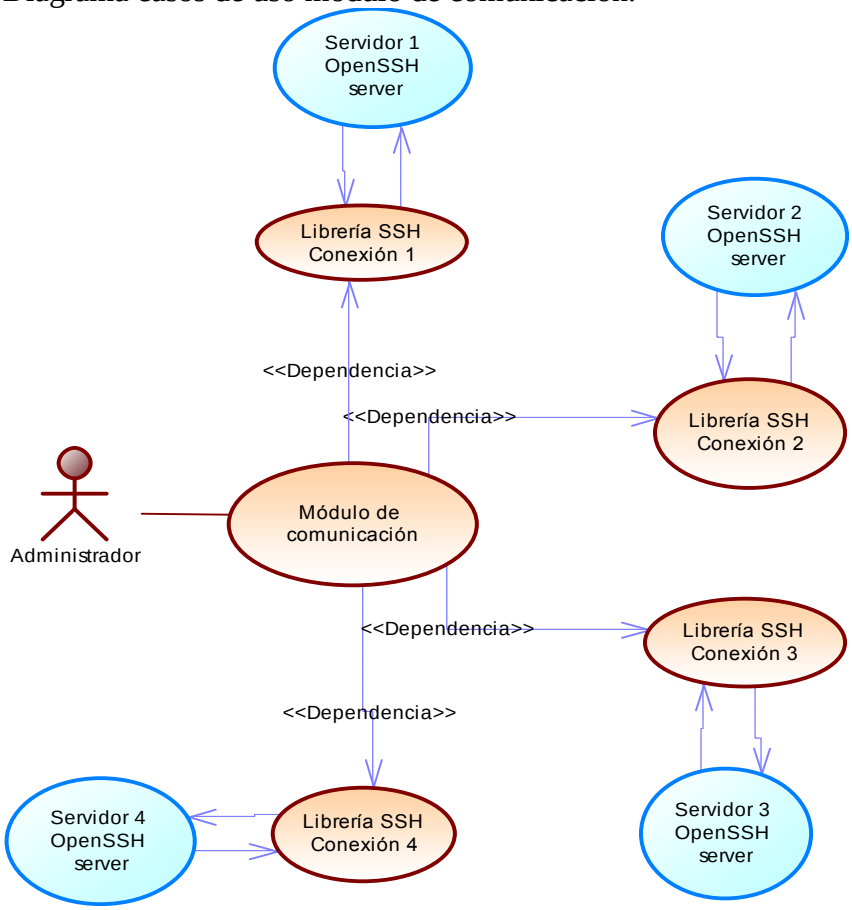


Figura 25. Elaborado por: José Benítez

En la tabla 21 para este caso de uso tenemos la descripción de, sus actores, pre y post condiciones conjuntamente con su flujo básico y alternativo.

Tabla 21.
Caso de uso módulo de comunicación.

Caso de uso	Módulo de comunicación
ACTORES	Administrador
PRECONDICIONES	Al momento de conectarse inicializa y enlaza el hilo de lectura al formulario, se define una conexión síncrona esto se repite para cada conexión que se desea abrir.
POSCONDICIONES	Se debe escoger adecuadamente la distribución a la que pertenece el servidor conectado.

FLUJO BÁSICO	1. Envía comandos por un cuadro de texto. 2. Imprime siempre y cuando haya encontrado el promt.
FLUJO ALTERNATIVO	1. Si no encuentra el promt imprime el resultado después de un tiempo de espera y despliega una alerta advirtiendo que se escoja correctamente el promt.

Nota. Elaborado por: José Benítez

Caso de uso 3: Módulo configuración de auditoría

En este módulo el usuario administrador es el que configura el análisis, ver figura 26.

Diagrama casos de uso módulo de auditoría.

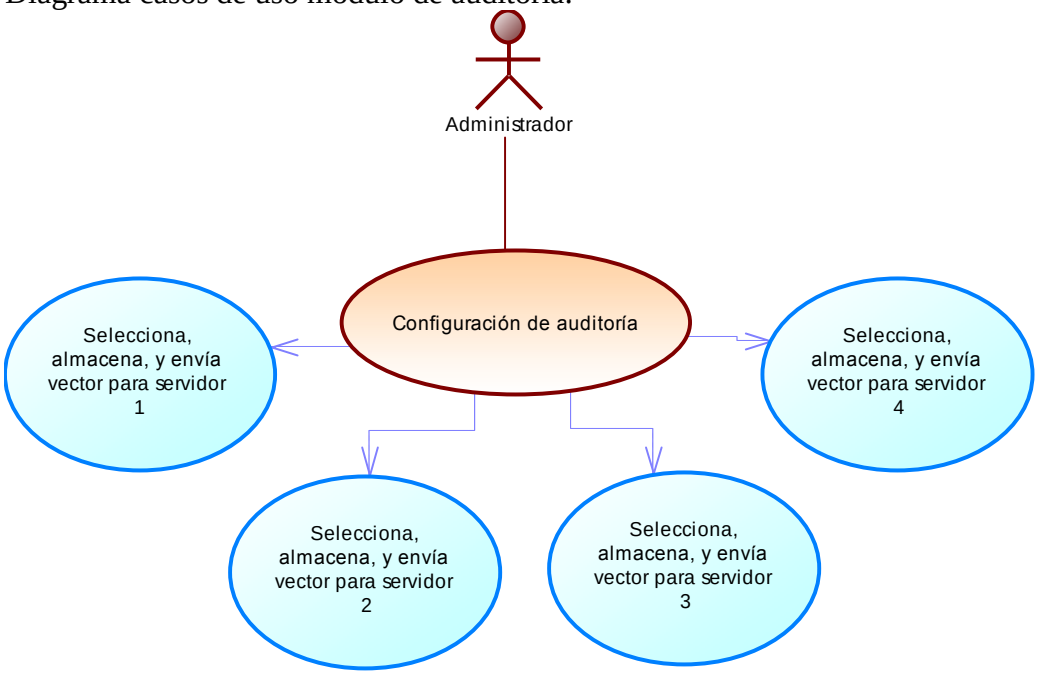


Figura 26. Elaborado por: José Benítez

En la tabla 22 tenemos la descripción de los casos de uso para este módulo.

Tabla 22.
Caso de uso módulo de auditoría.

Caso de uso	Modulo configuración de auditoría
ACTORES	Administrador
PRECONDICIONES	Accede desde el módulo de comunicación enviando el número servidor, la distribución y selecciona automáticamente la pestaña que corresponde al servidor donde se trabaja.
POSCONDICIONES	Después de enviar el vector con todos los comandos a ejecutar se debe regresar al módulo de comunicación para visualizar las respuestas.
FLUJO BÁSICO	1. Abre el formulario y se selecciona la configuración correspondiente al servidor seleccionado 2. Escoge todos los tipos de análisis a ejecutar. 3. Lo envía al módulo de comunicación.
FLUJO ALTERNATIVO	1. Escoge solo algunos tipos de análisis y posteriormente. 2. Los envía a ejecutarse al módulo de comunicación.

Nota. Elaborado por: José Benítez

Caso de uso 4: Módulo de reportes

El usuario administrador usará este módulo y el formulario "Reportes de auditorías" para ver uno por uno los resultados de los análisis realizados, ver figura 27.

Diagrama casos de uso módulo de reportes.

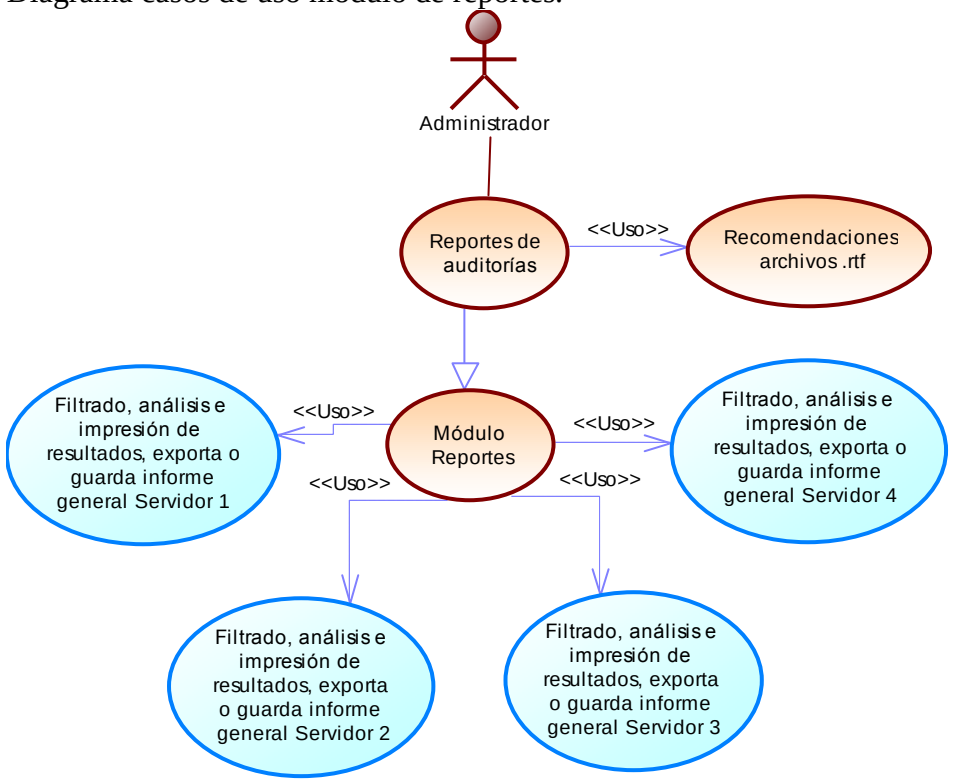


Figura 27. Elaborado por: José Benítez

En la tabla 23 tenemos las descripciones de los casos de usos para este módulo.

Tabla 23.
Caso de uso módulo de reportes.

Caso de uso	Módulo de reportes
ACTORES	Administrador
PRECONDICIONES	Accede desde el módulo de comunicación después de haber terminado de recopilar la información solicitada, Envía el número servidor, el listado de los tipos de análisis escogidos en la configuración de auditoría y finalmente todas las respuestas del servidor.
POSCONDICIONES	Una vez cargado y filtrada la información en un nuevo vector, se procede a visualizar uno por uno cada tipo de análisis seleccionado.
FLUJO BÁSICO	1. Abre el formulario y se selecciona uno por uno los tipos de análisis y en efecto se imprimirán en un cuadro de texto las recomendaciones correspondientes 2. Se imprime la información filtrada que se ha recibido del servidor. 3. Exporta a un informe general y lo guarda en un directorio por defecto.

FLUJO ALTERNATIVO	1. En el caso de acceder a este módulo sin esperar a que termine de ejecutarse todos los comandos no se cargará toda la información debido a que el servidor todavía no ha finalizado toda la ejecución de los comandos.

Nota. Elaborado por: José Benítez

2.2.2 Diagramas de secuencia

Se pude ver la secuencia de acciones realizadas por el usuario y la interacción para la comunicación síncrona con los servidores, desde el establecimiento de la conexión, hasta el cierre de la conexión, en este diagrama se ha escogido dos objetos que se comunican entre sí; el módulo de comunicación y los servidores Linux como se puede ver en la figura 28.

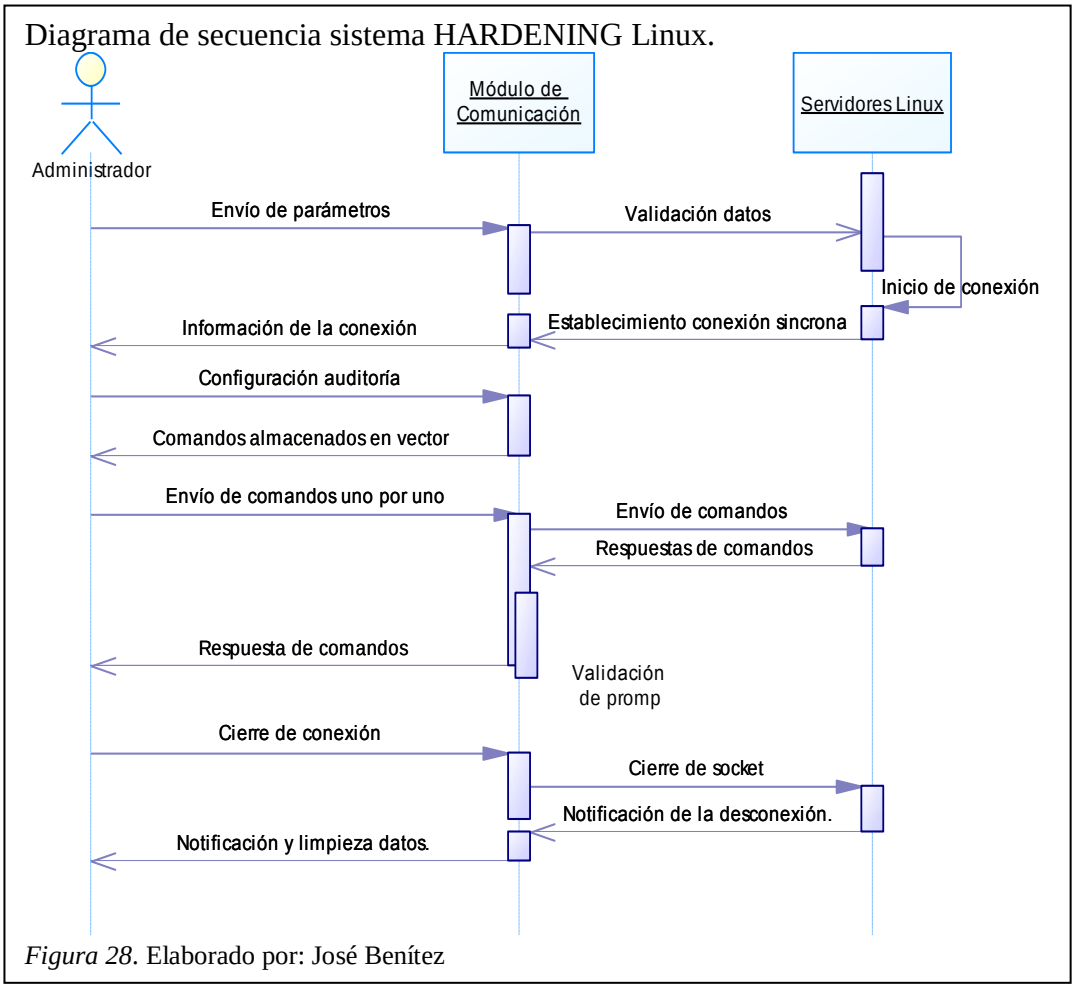


Figura 28. Elaborado por: José Benítez

2.2.3 Diagramas de actividades

En este diagrama se pude observar las actividades relacionadas directamente entre el usuario y la aplicación, detallando las actividades para la obtención del reporte final que nos genera el software desarrollado. En la figura 29 se puede visualizar los diagramas de actividades de la aplicación en general.

Diagrama de actividades usuario y aplicación.

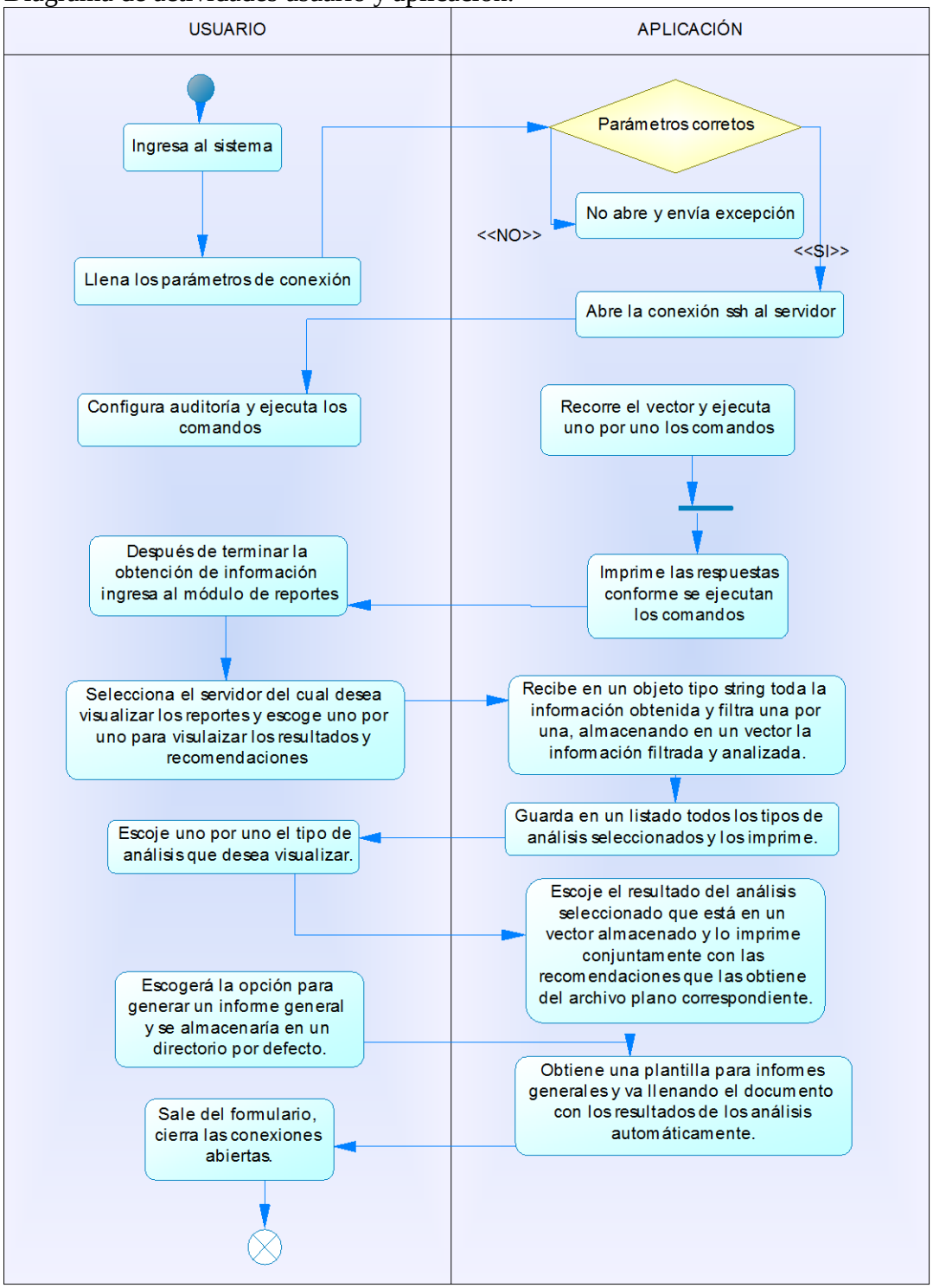


Figura 29. Elaborado por: José Benítez

2.2.4 Diagrama de clases

Se ha programado 4 clases similares considerando que cada una trabaja con un STREAM, se conserva la atomicidad y no se aplica el polimorfismo ver figura 30.

Diagrama de clases en la aplicación.

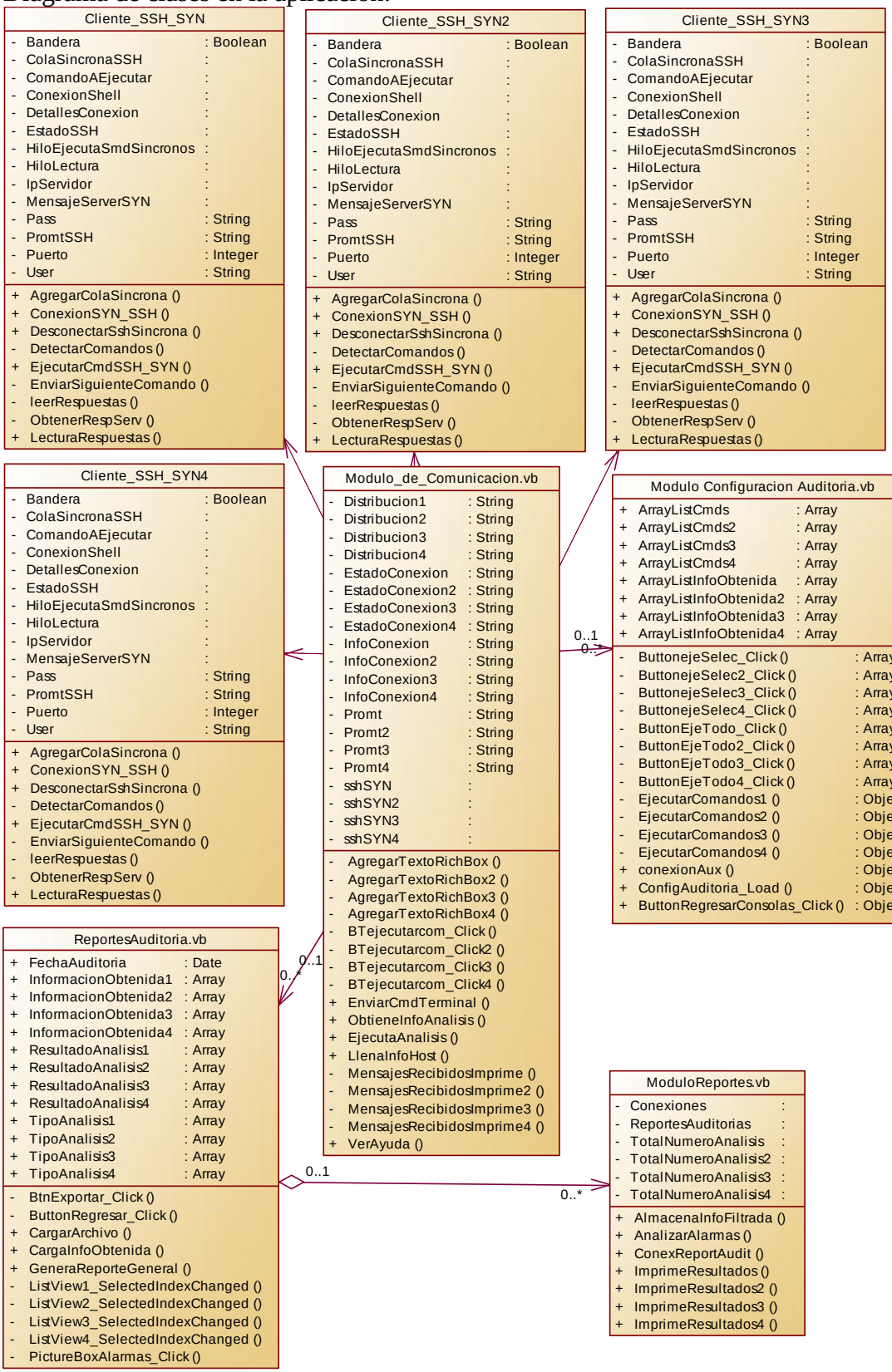


Figura 30. Elaborado por: José Benítez

CAPÍTULO 3

EJECUCIÓN Y ENTREGA

3.1 Fase ejecución

Es la fase que contiene la información relacionada con la codificación del software; descripciones de sus clases y componentes, librerías utilizadas, excepciones controladas, diagramas de implementación y las pruebas del producto.

3.1.1 Desarrollo

Se describen las clases creadas para el desarrollo en la tabla 24 y posteriormente una descripción de cada una con el código de sus métodos o funciones.

Tabla 24.
Descripción de las clases desarrolladas.

Nombre clase	Descripción	Observaciones
Login.vb	Formulario de autenticación para acceder al sistema SHL (Sistema HARDENING Linux). Se tiene 3 intentos antes de que se cierre la aplicación.	Se debe ingresar el usuario y contraseña correctamente.
LibreriaSSH_Conexion.vb LibreriaSSH_Conexion2.vb LibreriaSSH_Conexion3.vb LibreriaSSH_Conexion4.vb	Para establecer la conexión síncrona por ssh, establece métodos de lectura y escritura. Se maneja dos hilos uno de lectura y otro de ejecución de comandos.	Se necesita una clase similar para cada servidor es decir 4 iguales.
ConsolaConexiones.vb	Formulario para establecer las conexiones remotas a los servidores, se escoge como parámetro la IP, usuario, contraseña, puerto y distribución.	Los parámetros de conexión son obligatorios, para cada consola.
ConfigAuditoría.vb	Permite seleccionar que tipos de análisis vamos a ejecutar, rellena un vector y envía a su ejecución.	Los métodos y componentes son iguales para cada conexión es decir en total 4.
ReportesAuditoría.vb	Es la interfaz donde se imprimen todos los resultados obtenidos, filtrados y procesados, carga archivos de texto para imprimirlos conjuntamente con el resultado del análisis escogido.	Los métodos y componentes son iguales para cada conexión es decir en total 4.
ModuloReportes.vb	Es el encargado de recibir toda la información recibida del servidor, filtra, procesa y analiza para después almacenar los resultados en un vector.	Los métodos son iguales e independientes para cada conexión.

Nota. Elaborado por: José Benítez

Clase: Login.vb

Es el formulario de ingreso al sistema solo contiene un método para la validación de autentificación como se puede ver en la tabla 25.

Tabla 25.
Detalles clase Login.vb

Login.vb	
Métodos o funciones	**Descripción**
ValidaIngreso()	Validación del usuario y contraseña para el ingreso al programa. Tiene hasta 3 intentos, después del tercero cierra la aplicación.

Código

```
Private Sub ValidaIngreso()
    usuario = TextBoxUser.Text.Trim
    Dim PaswordIngresado = MaskedTextBox1.Text.Trim
    If usuario = "auditor" And PaswordIngresado = Password Then
        Me.Visible = False
        ConsolaConexiones.Visible = True
        LabelMensaje.Text = "Ingreso exitoso"
        LabelMensaje.Visible = True
    Else
        intentos = intentos + 1
        LabelMensaje.Visible = True
        LabelMensaje.Text = "Usuario o contraseña incorrectos. " & vbCrLf & "Número de intentos: "
& intentos
     If intentos = 3 Then
         MsgBox("Usted ha exedido el número de intentos disponibles", MsgBoxStyle.Information)
         Dispose()
     End If
   End If
End Sub
```

Nota: El texto que se puede ver en el campo código se refiere al lenguaje de programación utilizado y por lo tanto no representa alguna redacción.
Elaborado por: José Benítez

Clase: LibreriaSSH_Conexion.vb, LibreriaSSH_Conexion2.vb, LibreriaSSH_Conexion3.vb, LibreriaSSH_Conexion4.vb.

Se detallan las 4 clases que conforman el módulo de conexiones en la tabla 26 con las mismas funciones debido a que las 4 clases son idénticas en su codificación.

Tabla 26.
Detalles clases del módulo de conexiones.

LibreriaSSH_Conexion.vb, LibreriaSSH_Conexion2.vb, LibreriaSSH_Conexion3.vb, LibreriaSSH_Conexion4.vb		
#	**Métodos o funciones**	**Descripción**
1	ConexionSYN_SSH()	Función que establece la conexión al terminal pide como parámetros la ip, usuario, contraseña, puerto, distribución y promt.
	Código	
	Public Function ConexionSYN_SSH(ByVal Host As String, ByVal UserHost As String, ByVal PaswordHost As String, ByVal PuertoHost As Integer, ByVal Prompt As String) As String Me.Puerto = PuertoHost	

```
        Me.Pass = PaswordHost
        Me.User = UserHost
        Me.IpServidor = Host
        Me.PromptSSH = Prompt
        If (Me.EstadoConexion = False) Then
            Try
                Me.ConexionShell = New SshShell(Me.IpServidor, Me.User)
                If (Me.Pass <> "") Then
                    Me.ConexionShell.Password = Me.Pass
                End If
                Me.ConexionShell.Connect()
                Me._DetallesConexion = "PARÁMETROS DE CONEXIÓN:" & vbCrLf & "   >
                Servidor: 1  " & vbCrLf & "   > IP: " & Me._IPServidor & vbCrLf & "   > Puerto: " &
                Me.Puerto & vbCrLf & "   > Usuario: " & Me.User & "" & vbCrLf &
                "INFORMACIÓN DE PROTOCOLO SSH: " & vbCrLf & "   > Servidor SSH: " &
                Me.ConexionShell.ServerVersion & vbCrLf & "   > Cifrado: " &
                Me.ConexionShell.Cipher & vbCrLf & "   > Código Hash: " &
                Me.ConexionShell.GetHashCode() & vbCrLf & "   > HMAC: " &
                Me.ConexionShell.Mac
                Me._MensajeConexion += "PARÁMETROS DE CONEXIÓN:" & vbCrLf & "Tipo
                de conexión: SinCRONa | Servidor: 1 | IP: " & Me._IPServidor & " | Puerto: " &
                Me.Puerto & " | Usuario: " & Me.User & " exitosa." & vbCrLf & "INFORMACIÓN
                DE CONEXIÓN: " & vbCrLf & " Servidor SSH: " &
                Me.ConexionShell.ServerVersion & vbCrLf & " Cifrado: " &
                Me.ConexionShell.Cipher & vbCrLf & " Código Hash: " &
                Me.ConexionShell.GetHashCode() & vbCrLf & " HMAC: " &
                Me.ConexionShell.Mac & vbCrLf & vbCrLf & "Ingresar el comando a ejecutar en el
                cuadro de texto. Asegurarse que tenemos privilegios de root con $su -" & vbCrLf
                'Cambiamos de estado las banderas de control.
                Me.EstadoConexion = True
                Me.EstadoSSH = "Conectado"
                'Removemos los caracteres especiales ANCI
                Me.ConexionShell.RemoveTerminalEmulationCharacters = True
                'Obtiene el primer mensaje de texto enviado por el server
                Me._MensajeConexion += Me.ConexionShell.Expect() & vbCrLf
            Catch ex As Exception
                Throw New Exception("Clase: ClienteSSH_SYN | Sub: ConexionSYN_SSH |
                Parametros: Ip: " & Me._IPServidor & " User: " & Me._User & " |Descripción:
                usuario o contraseña invalidos. Error: " & ex.Message & ".")
                Me.EstadoConexion = False
                Me.EstadoSSH = "Disponible"
            End Try
        Else
            Me.EstadoSSH = "Indisponible"
            Me._MensajeConexion = "Hay una conexión ya establecida al server " &
            Me._IPServidor & " al puerto: " & Me._Puerto & " estado: " & Me.EstadoSSH &
           "Desconectarse primero. " & vbCrLf
        End If
        If Me.ConexionShell.Connected Then
            'Instanciamos, enlazamos los hilos y los  inicializamos.
            Me.HiloLectura = New Thread(AddressOf Me.leerRespuestas)
            Me.HiloLectura.Start()
            Me.HiloEjecutaCmdSinCRONos = New Thread(AddressOf DetectarComandos)
            Me.HiloEjecutaCmdSinCRONos.Start()
        End If
        Return Me._MensajeConexion
    End Function
```

2	DetectarComandos()	Método que está enviando constantemente comandos en el caso de existir en el vector fifo, envía uno por uno al método EjecutarCmdSSH_SYN().

Código

```
''' <summary>    ''' Método que está enviando a ejecutarse constantemente comandos en el caso de existir en el arraylist fifo    ''' </summary>
Private Sub DetectarComandos()
  While Me.ConexionShell.Connected
    If Me.ColaSinCRONaSSH.Count > 0 Then
       EnviarSiguienteComando()
    Else 'Para mejorar el consumo del cpu se duerme al hilo si no encuentra comandos en la cola fifo.
     Thread.Sleep(300)
    End If
  End While
End Sub
```

3	EjecutarCmdSSH_SYN()	Recibe el comando a ejecutar y almacena la respuesta del server en la variable pública que será leída por el módulo de comunicación.

Código

```
Public Sub EjecutarCmdSSH_SYN(ByVal ObMensaje As String)
  Dim mensCliente As String
  Me.MensajeServerSYN = ""
  mensCliente = ObMensaje
  'Para detectar los comandos "Ctl + C" desde un texto
  If mensCliente = "parar" Or mensCliente = "Parar" Or mensCliente = "PARAR" Then
    Dim controlC As String = "" & Chr(3)
    mensCliente = controlC
    'Para detectar función de la tecla "Barra espaciadora" para uso del comando "more" por ejemplo
  ElseIf mensCliente = "ESPACIO" Or mensCliente = "espacio" Or mensCliente = "Espacio" Then
    Dim TeclaEspacio As String = "" & Chr(32)
    mensCliente = TeclaEspacio
  End If
  Try  'Envía el comando a ejecutar
    Me.ConexionShell.WriteLine(mensCliente)
    Me.Bandera = "Ejecucion Terminada"
  Catch ex As Exception
    Me.Bandera = "Error Ejecucion"
    Throw New Exception("Clase: Cliente_SSH_SYN.vb Sub: LeerMensaje(). | Error al ejecutar el cmd: " & mensCliente & " | Detalles: " + ex.Message)
    Finally
      Finalize()
    End Try
End Sub
```

4	ObtenerRespServ()	Almacena en un objeto público tipo string las respuestas del servidor y esta enlazado a un hilo que constantemente estará escuchando, la lectura finaliza cuando se ha encontrado el prompt.

Código

```
Private Sub ObtenerRespServ()
    Dim ContadorControl As Integer = 0
    Dim FinalizaCMD As Boolean = False
    Dim EncuentraPrompt As Boolean
    Dim StreamRecepcion As String = ""
    While FinalizaCMD = False And Me.ConexionShell.Connected
      Try
        'Se va a obtener el caracter que se identifica que esta activada nuevamente el prompt
        StreamRecepcion = Me.ConexionShell.Expect()
        EncuentraPrompt = StreamRecepcion.Contains(Me.PromptSSH)
        'Si se ejecuta el comando su - el promt cambia
        If EncuentraPrompt = False And Me.ComandoAEjecutar = "su -" Then
```

```
                EncuentraPrompt = StreamRecepcion.Contains(":")
                StreamRecepcion += "" & vbCrLf
            End If
            'Si se ejecuta el comando para verificación de logs el promt cambia
            If EncuentraPrompt = False And Me.ComandoAEjecutar = "q" Then
                EncuentraPrompt = StreamRecepcion.Contains("lines")
                Me.MensajeServerSYN = vbCrLf & Me.MensajeServerSYN & vbCrLf
            End If
            If EncuentraPrompt = True Then
                FinalizaCMD = True
                Me.Bandera = "Lectura Terminada"
                Me.MensajeServerSYN = StreamRecepcion 'Almacena respuesta
            Else
                System.Threading.Thread.Sleep(500)
            End If
        Catch ex As Exception
            Me.Bandera = "Error de Lectura"
            Me.MensajeServerSYN += Me.ConexionShell.Expect()
            Throw New Exception("Clase: Cliente_SSH_SYN.vb Sub: ObtenerRespServ(). Error
            al obtener respuesta detalles: " + ex.Message)
        End Try
        'Para control de bucle infinito en el caso de no ingresar bien el prompt
        ContadorControl = ContadorControl + 1
        If ContadorControl > 100 Then
            Me.Bandera = "Prompt Incorrecto"
            Me.MensajeServerSYN += StreamRecepcion
            Me.MensajeServerSYN += vbCrLf & "!Alerta¡: " & Me.Bandera & ". Por favor
            ingrese correctamente el prompt para un correcto sincronismo." & vbCrLf
            Exit While
        End If
    End While
    'Si ya ha finalizado la lectura y no existen datos en el stream procede a enviar la respuesta
    If FinalizaCMD = True Then
        RaiseEvent LecturaRespuestas(Me.MensajeServerSYN)
    ElseIf FinalizaCMD = False And Me.Bandera = "Prompt Incorrecto" Then
        RaiseEvent LecturaRespuestas(Me.MensajeServerSYN)
    End If
End Sub
```

5	leerRespuestas()	Evento para la lectura constante de los comandos ejecutados, esta enlazado a un hilo y llama al evento de lectura ObtenerRespServ().

Código

```
Private Sub leerRespuestas()
  While Me.ConexionShell.Connected ' Lee mientras exista una conexión establecida
    ObtenerRespServ()
  End While
End Sub
```

6	DesconectarSshSincrona()	Función que desconecta la conexión síncrona establecida si existe alguna y devuelve un string con la notificación de la desconexión, mata los hilos y libera recursos de memoria.

Código

```
Public Function DesconectarSshSinCRONa() As String
    If ConexionShell.ShellConnected = True Then
      Try
        ConexionShell.Close()   'Cerramos el socket principal
        'Matamos al hilo de ejecución de comandos HiloEjecutaCmdSincronos.
        If Not IsNothing(Me.HiloEjecutaCmdSinCRONos) Then
          If Me.HiloEjecutaCmdSinCRONos.IsAlive Then
              Me.HiloEjecutaCmdSinCRONos.Abort()
          End If
```

```
            End If
            Me.EstadoConexion = False
            Me.EstadoSSH = "Disponible"
            Me.MensajeServerSYN = "" & vbCrLf & "Desconexión exitosa al terminal: " &
            Me._IPServidor & " en el puerto: " & Me.Puerto & " estado terminal: " &
            Me.EstadoSSH & "." & vbCrLf
            Catch ex As Exception
               Throw New Exception("Clase: ClienteSSH_SYN | Function:
               DesconectarSshSincrona() | Description: " & ex.Message & vbCrLf)
               Me.EstadoSSH = "Indisponible"
            Finally
               Finalize()
            End Try
         Else
            Me.MensajeServerSYN = "¡Alerta! No existe alguna conexión para proceder a cerrarla."
            & vbCrLf
         End If
         Return Me.MensajeServerSYN
      End Function
```

Nota: El texto que se puede ver en el campo código se refiere al lenguaje de programación utilizado y por lo tanto no representa alguna redacción solo las letras de color verde son comentarios.
Elaborado por: José Benítez

Clase: ConsolaConexiones.vb

En la clase CosolaConexiones.vb se describe los métodos y funciones más importantes del código se los puede ver en la tabla 27.

Tabla 27.
Detalles clase ConsolaConexiones.vb

LibreriaSSH_Conexion.vb		
#	**Métodos o funciones**	**Descripción**
1	AgregarTextoRichBox() AgregarTextoRichBox2() AgregarTextoRichBox3() AgregarTextoRichBox4()	Evento para rellenar el cuadro de texto continuamente con los datos obtenidos por el módulo de conexiones enviado desde la terminal.
	Código	

	`Private Sub AgregarTextoRichBox(ByVal MensajeRecibido As String)`
	`  If RichTextBoxHost1.InvokeRequired Then`
	`  RichTextBoxHost1.Invoke(Sub() AgregarTextoRichBox(MensajeRecibido))`
	`  Else`
	`  RichTextBoxHost1.AppendText(MensajeRecibido)`
	`  RichTextBoxHost1.SelectionStart = RichTextBoxHost1.TextLength`
	`  End If`
	`  If MensajeRecibido.Contains("* Inhabilitando el Ctrl+Alt+Del") Then`
	`  RichTextBoxInfoHost1.BackColor = Drawing.Color.MediumSeaGreen`
	`  End If`
	`  Dim EncontroInicio As Boolean`
	`  EncontroInicio = Me.InfoConexion.Contains("INFORMACIÓN DE PROTOCOLO SSH:")`
	`  If EncontroInicio = True Then`
	`  BtnInfoConexHost1.BackgroundImage = ImageListInfoConex.Images(1)`
	`  End If`
	`End Sub`	

2	MensajesRecibidosImprime() MensajesRecibidosImprime2() MensajesRecibidosImprime3() MensajesRecibidosImprime4()	Evento que será utilizado para enviar a imprimir las respuestas obtenidas por el server lo envía al evento delegado "AgregarTextoRichBox ()" está enlazado con un hilo de lectura.

Código

```
Private Sub MensajesRecibidosImprime(ByVal Mensaje As String)
    Dim texto As String = Mensaje
    AgregarTextoRichBox(Mensaje)
End Sub
```

3	EjecutaAnalisis()	Método para ejecución de los comandos almacenados en un vector previamente llenado en la configuración de la auditoría, y especifica el número de conexión a enviar. Se utiliza para todas las conexiones.

Código

```
Public Sub EjecutaAnalisis(ByVal VectorComandos As ArrayList, ByVal NumConsola As
Integer)
  'Para el servidor número 1
  If NumConsola = 1 Then
  Try
      Dim comando As String
      'Enviamos los comandos a ejecutar al módulo de conexiones
      For indice As Integer = 0 To VectorComandos.Count - 1 Step 1
          comando = VectorComandos(indice)
          sshSYN.AgregarColaSincrona(comando.Trim, Me.Promt)
      Next
  Catch ex As Exception
     RichTextBoxHost1.AppendText(ex.Message & vbCrLf)
   End Try
   End If
  'Para el servidor número 2
  If NumConsola = 2 Then
  Try
     Dim comando As String
     'Enviamos los comandos a ejecutar al módulo de conexiones
     For indice As Integer = 0 To VectorComandos.Count - 1 Step 1
        comando = VectorComandos(indice)
        sshSYN2.AgregarColaSincrona(comando.Trim, Me.Promt2)
     Next
   Catch ex As Exception
           RichTextBoxHost2.AppendText(ex.Message & vbCrLf)
   End Try
   End If
   'Para el servidor número 3 ……
….Se repite el mismo código con los nombres de los objetos correspondientes a la conexión.
```

4	EnviarCmdTerminal()	Método para el envío de los comandos escritos en el cuadro de texto correspondiente a cada servidor.

Código

```
Private Sub EnviarCmdTerminal()
  'Para el servidor 1
  If TabControlConsolas.SelectedIndex = 0 Then
    Try
    If TextBoxComandos.Text.Trim = "clear" Or TextBoxComandos.Text.Trim
       = "CLEAR" Then
      RichTextBoxHost1.Clear()
      RichTextBoxHost1.AppendText(Me.Promt)
      TextBoxComandos.Clear()
      sshSYN.MensajeServerSYN = ""
    Else
      sshSYN.AgregarColaSincrona(TextBoxComandos.Text.Trim, Me.Promt)
      TextBoxComandos.Text = ""
    End If
    Catch ex As Exception
      RichTextBoxHost1.AppendText(ex.Message & vbCrLf)
    End Try
```

```
    End If
    'Para el servidor 2
    If TabControlConsolas.SelectedIndex = 1 Then
      Try
       If TextBoxComandos2.Text.Trim = "clear" Or
          TextBoxComandos2.Text.Trim = "CLEAR" Then
          RichTextBoxHost2.Clear()
          RichTextBoxHost2.AppendText(Me.Promt2)
          TextBoxComandos2.Clear()
       Else
          sshSYN2.AgregarColaSincrona(TextBoxComandos2.Text.Trim,
          Me.Promt2)
          TextBoxComandos2.Text = ""
       End If
      Catch ex As Exception
        RichTextBoxHost2.AppendText(ex.Message & vbCrLf)
      End Try
    End If
    'Para el servidor 3….
….Se repite el mismo código con los nombres de los objetos correspondientes a la conexión.
End Sub
```

5	ObtieneInfoAnalisis()	Método que obtiene e imprime información de los tipos de análisis que se han enviado a ejecutarse.

Código

```
Public Sub ObtieneInfoAnalisis(ByVal VectorInfo As ArrayList, ByVal NumConsola As
Integer)
   If NumConsola = 1 Then 'Para el servidor número 1
     Dim comando As String
     RichTextBoxInfoHost1.Clear()
     For indice As Integer = 0 To VectorInfo.Count - 1 Step 1
        comando = VectorInfo(indice)
        RichTextBoxInfoHost1.AppendText(comando & vbCrLf)
     Next
   End If
   If NumConsola = 2 Then 'Para el servidor número 2
     Dim comando As String
     RichTextBoxInfoHost2.Clear()
     For indice As Integer = 0 To VectorInfo.Count - 1 Step 1
        comando = VectorInfo(indice)
        RichTextBoxInfoHost2.AppendText(comando & vbCrLf)
     Next
   End If
   If NumConsola = 3 Then  'Para el servidor número 3…….
….Se repite el mismo código con los nombres de los objetos correspondientes a la conexión.
End Sub
```

6	LlenaInfoHost()	Evento para cargar las imágenes y obtiene información de los parámetros de conexión en cada botón.

Código

```
Private Sub LlenaInfoHost()
 If RadioButtonConexion1.Checked Then  'Carga los datos del servidor 1
   Dim ParametrosCons1 As String
   ParametrosCons1 = "Servidor 1 Datos: " & vbCrLf & "IP: " _
   & TextBoxIP.Text & vbCrLf & "Puerto: " & TextBoxPuerto.Text _
   & vbCrLf & "Usuario: " & TextBoxUser.Text & vbCrLf _
   & "Distribución: " & CmbDistro.SelectedItem
   BtnHost1.Text = ParametrosCons1
   Select Case CmbDistro.SelectedItem
     Case "Centos"
        BtnHost1.BackgroundImage = ImageListDistros.Images(2)
        Distribucion1 = "Centos"
```

```
        Case "OpenSUSE"
            BtnHost1.BackgroundImage = ImageListDistros.Images(4)
            Distribucion1 = "OpenSUSE"
        Case "Ubuntu"
            BtnHost1.BackgroundImage = ImageListDistros.Images(3)
            Distribucion1 = "Ubuntu"
        Case "Fedora"
            BtnHost1.BackgroundImage = ImageListDistros.Images(1)
            Distribucion1 = "Fedora"
        Case Else
            BtnHost1.BackgroundImage = ImageListDistros.Images(0)
    End Select
End If
Dim ParametrosCons2 As String  'Carga los datos del servidor 2
If RadioButtonConexion2.Checked Then
    ParametrosCons2 = "Servidor 2 Datos: " & vbCrLf & "IP: " _
    & TextBoxIP.Text & vbCrLf & "Puerto: " & TextBoxPuerto.Text _
    & vbCrLf & "Usuario: " & TextBoxUser.Text & vbCrLf _
    & "Distribución: " & CmbDistro.SelectedItem
    BtnHost2.Text = ParametrosCons2
    Select Case CmbDistro.SelectedItem
        Case "Centos"
                BtnHost2.BackgroundImage = ImageListDistros.Images(2)
                Distribucion2 = "Centos"
        Case "OpenSUSE"
                BtnHost2.BackgroundImage = ImageListDistros.Images(4)
                Distribucion2 = "OpenSUSE"
        Case "Ubuntu"
                BtnHost2.BackgroundImage = ImageListDistros.Images(3)
                Distribucion2 = "Ubuntu"
        Case "Fedora"
                BtnHost2.BackgroundImage = ImageListDistros.Images(1)
                Distribucion2 = "Fedora"
        Case Else
                BtnHost2.BackgroundImage = ImageListDistros.Images(0)
        End Select
End If
    'Carga los datos del servidor 3 ….
….Se repite el mismo código con los nombres de los objetos correspondientes a la conexión.
End Sub
```

Nota: El texto que se puede ver en el campo código se refiere al lenguaje de programación utilizado y por lo tanto no representa alguna redacción solo las letras de color verde son comentarios.
Elaborado por: José Benítez

Clase: ConfigAuditoria.vb

En la tabla 28 se tienen los detalles esta clase utilizada para el módulo de configuración de auditoria conjuntamente con su código de los métodos o funciones más importantes.

Tabla 28.
Detalles clase ConfigAuditoria.vb

ConfigAuditoría.vb		
#	**Métodos o funciones**	**Descripción**
1	conexionAux()	Método que enlaza al objeto ya instanciado y que tiene la conexión activa en la clase LibreriaSSH_Conexion, hace referencia a la clase ConsolaConexiones.vb para heredar tus

		objetos.

Código

```
''' <summary>''' Método que enlaza al objeto ya instanciado que tiene la conexión activa en la clase LibreriaSSH_Conexion ''' </summary>
''' <param name="conexion"> Hace referencia a la clase con la que quiere instanciar y heredar sus objetos</param>
Public Sub conexionAux(ByRef conexion As ConsolaConexiones)
    ConsolaConex = conexion
End Sub
```

2	ConfigAuditoría_Load()	Método que escoge la pestaña seleccionada en función del servidor con el que se está trabajando en ConsolaConexiones.vb

Código

```
Public Sub ConfigAuditoría_Load(ByVal NumeroConsola As Integer)
        If NumeroConsola = 0 Then
            Me.TabControlConfig.SelectedIndex = 0
        End If
        If NumeroConsola = 1 Then
            Me.TabControlConfig.SelectedIndex = 1
        End If
        If NumeroConsola = 2 Then
            Me.TabControlConfig.SelectedIndex = 2
        End If
        If NumeroConsola = 3 Then
            Me.TabControlConfig.SelectedIndex = 3
        End If
End Sub
```

3	EjecutarComandos1() EjecutarComandos2() EjecutarComandos3() EjecutarComandos4()	Métodos que almacenan en un vector los comandos para después enviarlos a ejecutar, solo los análisis seleccionados del servidor escogido, el código en estos 4 bloques son similares, únicamente varían los vectores donde almacenan, y el número de conexión.

Código

```
Private Sub EjecutarComandos1()
  Me.ArrayListCmds = New ArrayList
  Me.ArrayListCmds.Clear()
  If CheckListBoxConfig.GetItemChecked(0) Then
  'Información general del servidor
      ArrayListCmds.Add("echo -e '\t********** Información general del servidor **********\n
      Información obtenida:\n'")
      ArrayListCmds.Add("uname -n")
      ArrayListCmds.Add("uname -s")
      ArrayListCmds.Add("uname -r")
      ArrayListCmds.Add("uname -v")
      ArrayListCmds.Add("uname -p")
      ArrayListCmds.Add("uname -m")
      ArrayListCmds.Add("uptime")
  End If
  If CheckListBoxConfig.GetItemChecked(1) Then
     'Listado de conexiones activas
     ArrayListCmds.Add("echo -e '\t********** Listado de conexiones activas **********\n
     Información obtenida:\n'")
     ArrayListCmds.Add("who -uH")
  End If
  If CheckListBoxConfig.GetItemChecked(2) Then
    'Intentos fallidos de conexión
    ArrayListCmds.Add("echo -e '\t********** Intentos fallidos de conexión **********\n
    Información obtenida:\n'")
    ArrayListCmds.Add("lastb")
  End If
```

```
If CheckListBoxConfig.GetItemChecked(3) Then
   'Listado conexiones de usuarios
   ArrayListCmds.Add("echo -e '\t********* Listado conexiones de usuarios *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("last -50")
End If
If CheckListBoxConfig.GetItemChecked(4) Then
  'Listado uso comando su o sudo
If ConsolaConex.Distribucion1 = "OpenSUSE" Then
   ArrayListCmds.Add("echo -e '\t********* Listado uso comando su o sudo *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("tail -n50 /var/log/messages")
ElseIf ConsolaConex.Distribucion1 = "Fedora" Or ConsolaConex.Distribucion1 = "Centos"
 Then
   ArrayListCmds.Add("echo -e '\t********* Listado uso comando su o sudo *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("journalctl /usr/bin/sudo -n50")
   ArrayListCmds.Add("q")
   ArrayListCmds.Add("journalctl /usr/bin/su -n50")
   ArrayListCmds.Add("q")
 Else
   ArrayListCmds.Add("echo -e '\t********* Listado uso comando su o sudo *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("cat /var/log/auth.log")
 End If
End If
If CheckListBoxConfig.GetItemChecked(5) Then
   'Listado de servicios activos
   ArrayListCmds.Add("echo -e '\t********* Listado de servicios activos *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("netstat -an|grep udp")
   ArrayListCmds.Add("netstat -an|grep tcp|grep LISTEN")
   ArrayListCmds.Add("rpcinfo -p")
End If
If CheckListBoxConfig.GetItemChecked(6) Then
   'Detalle de usuarios y contraseñas
   ArrayListCmds.Add("echo -e '\t********* Detalle de usuarios y contraseñas *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("awk -F"":"" '{print ""User= ""$1"" *UID= ""$3"" *GID= ""$4""
   *Nombre completo= ""$5"" *Directorio= ""$6}' /etc/passwd")
   ArrayListCmds.Add("awk -F"":"" '{print ""User= ""$1"" *Último cambio pasword= ""$3""
   *Dias notificación cambio= ""$4"" *Días para expiración contraseña= ""$5"" *Alarmas
    pasword= ""$6"" *Días desactivados= ""$7"" *Desactivada= ""$8}' /etc/shadow")
End If
If CheckListBoxConfig.GetItemChecked(7) Then
   'Listado SUID y SGID activos
   ArrayListCmds.Add("echo -e '\t********* Listado SUID y SGID activos *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("find / -path '/mnt' -prune -o -path '/cdrom' -prune -o -path '/floppy' –
   prune -o -fstype NFS -prune -o -type f -perm -4000 -print")
   ArrayListCmds.Add("find / -path '/mnt' -prune -o -path '/cdrom' -prune -o -path '/floppy' –
   prune -o -fstype NFS -prune -o -type f -perm -2000 -print")
End If
If CheckListBoxConfig.GetItemChecked(8) Then
   'Permisos archivos especiales
   ArrayListCmds.Add("echo -e '\t********* Permisos archivos especiales *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("ls -l /etc/passwd")
   ArrayListCmds.Add("ls -l /etc/init.d")
   ArrayListCmds.Add("ls -l /etc/xinetd.d")
```

```
    ArrayListCmds.Add("ls -l /etc/environment")
    ArrayListCmds.Add("ls -l /etc/exports")
End If
If CheckListBoxConfig.GetItemChecked(9) Then
    'Lectura shadow
    ArrayListCmds.Add("echo -e '\t********* Lectura shadow *********\nInformación
    obtenida:\n'")
    ArrayListCmds.Add("ls -l /etc/shadow")
End If
If CheckListBoxConfig.GetItemChecked(10) Then
    'Permisos de multiusuarios
    ArrayListCmds.Add("echo -e '\t********* Permisos de multiusuarios *********\n
    Información obtenida:\n'")
    ArrayListCmds.Add("ls -l /home")
End If
If CheckListBoxConfig.GetItemChecked(11) Then
   'Detalle de grupos del servidor
   ArrayListCmds.Add("echo -e '\t********* Detalle de grupos del servidor *********\n
   Información obtenida:\n'")
   ArrayListCmds.Add("awk -F"":"" '{print ""Nombre del grupo= ""$1"" *GID= ""$3""
   *Miembros= (""$4"")""}' /etc/group")
End If
If CheckListBoxConfig.GetItemChecked(12) Then
    'Listado de recursos exportados por NFS
    ArrayListCmds.Add("echo -e '\t********* Listado de recursos exportados por NFS
    *********\n Información obtenida:\n'")
    ArrayListCmds.Add("cat /etc/exports")
End If
If CheckListBoxConfig.GetItemChecked(13) Then
    'Listado usuarios FTP
   If ConsolaConex.Distribucion1 = "Fedora" Or ConsolaConex.Distribucion1 = "Centos"
Then
    ArrayListCmds.Add("echo -e '\t********* Listado usuarios FTP *********\n Información
    obtenida:\n'")
    ArrayListCmds.Add("cat /etc/vsftpd/FTPusers")
    ArrayListCmds.Add("cat /etc/vsftpd/user_list")
    ArrayListCmds.Add("cat /etc/vsftpd/vsftpd.conf")
   Else
     ArrayListCmds.Add("echo -e '\t********* Listado usuarios FTP *********\nInformación
     obtenida:\n'")
     ArrayListCmds.Add("cat /etc/vsftpd.conf")
     ArrayListCmds.Add("cat /etc/FTPusers")
   End If
End If
If CheckListBoxConfig.GetItemChecked(14) Then
   'Listado usuarios para cron
   If ConsolaConex.Distribucion1 = "Ubuntu" Then
      ArrayListCmds.Add("echo -e '\t********* Listado usuarios para cron *********\n
      Información obtenida:\n'")
      ArrayListCmds.Add("cat /etc/cron.d/anacron")
   Else
      ArrayListCmds.Add("echo -e '\t********* Listado usuarios para cron *********\n
      Información obtenida:\n'")
      ArrayListCmds.Add("cat /etc/at.deny")
      ArrayListCmds.Add("cat /etc/cron.deny")
   End If
End If
If CheckListBoxConfig.GetItemChecked(15) Then
    'Políticas de cuentas
    ArrayListCmds.Add("echo -e '\t********* Políticas de cuentas *********\nInformación
```

```
        obtenida:\n"')
        ArrayListCmds.Add("cat /etc/login.defs")
        ArrayListCmds.Add("cat /etc/pam.d/passwd")
    End If
    If CheckListBoxConfig.GetItemChecked(16) Then
        'Gestor de arranque GRUB 2
        ArrayListCmds.Add("echo -e '\t********* Gestor de arranque GRUB 2 *********\n
        Información obtenida:\n"')
        ArrayListCmds.Add("cat /etc/grub.d/40_custom")
    End If
    If CheckListBoxConfig.GetItemChecked(17) Then
        'Protección de logs
        ArrayListCmds.Add("echo -e '\t********* Protección de logs *********\nInformación
        obtenida:\n"')
        ArrayListCmds.Add("ls -ld /var/log")
    End If
    If CheckListBoxConfig.GetItemChecked(18) Then
        'Inhabilitando el Ctrl+Alt+Del"
        If ConsolaConex.Distribucion1 = "Fedora" Or ConsolaConex.Distribucion1 = "OpenSUSE"
        Then
            ArrayListCmds.Add("echo -e '\t********* Inhabilitando el Ctrl+Alt+Del *********\n
            Información obtenida:\n"')
            ArrayListCmds.Add("cat /etc/inittab")
        ElseIf ConsolaConex.Distribucion1 = "Centos" Then
            ArrayListCmds.Add("echo -e '\t********* Inhabilitando el Ctrl+Alt+Del *********\n
            Información obtenida:\n"')
            ArrayListCmds.Add("cat /usr/lib/systemd/system/ctrl-alt-del.target")
        Else
            ArrayListCmds.Add("echo -e '\t********* Inhabilitando el Ctrl+Alt+Del *********\n
            Información obtenida:\n"')
            ArrayListCmds.Add("cat /etc/init/control-alt-delete.conf")
        End If
    End If
        'Enviamos los comandos cargados para su ejecución al módulo de comunicación.
        ConsolaConex.EjecutaAnalisis(Me.ArrayListCmds, 1)
End Sub
```

4	ButtonEjeTodo_Click_1 () ButtonEjeTodo2_Click() ButtonEjeTodo3_Click() ButtonEjeTodo4_Click()	Evento que selecciona los tipos de análisis y los almacena en el vector ArrayListInfoObtenida, el código en los 4 eventos son similares con la diferencia que almacenan en distintos vectores la información.

Código

```
''' <summary>''' Evento para ejecutar todos los tipos de análisis para el servidor 1 ''' </summary>
Private Sub ButtonEjeTodo_Click_1(ByVal sender As Object, ByVal e As EventArgs) Handles
ButtonEjeTodo.Click
    'Activamos todas las opciones del CheckListBoxConfig
    Dim i As Integer
    For i = 0 To CheckListBoxConfig.Items.Count - 1
        CheckListBoxConfig.SetItemChecked(i, True)
    Next
    MsgBox("Ya se está ejecutando los comandos de los tipos de análisis seleccionados, después
                de unos 30 segundos aproximadamente podrá revisar los reportes obtenidos. " _
               & "Si desea puede regresar al módulo de comunicación para ver las respuestas del
               servidor.", MsgBoxStyle.OkOnly)
    'Enviamos el Arraylist para su ejecución en la consola de conexiones
    ButtonSelec_Click_1(sender, e)
    ConsolaConex.TabControlConsolas.SelectTab(0)
End Sub
```

Nota: El texto que se puede ver en el campo código se refiere al lenguaje de programación utilizado y por lo tanto no representa alguna redacción solo las letras de color verde son comentarios.
Elaborado por: José Benítez

Clase: ReportesAuditoria.vb

Clase perteneciente al módulo de reportes, trabaja conjuntamente con la clase ModuloReportes.vb a continuación en la tabla 29 se detallan sus métodos y funciones más importantes:

Tabla 29.
Detalles clase ReportesAuditoria.vb

ConfigAuditoría.vb		
#	**Métodos o funciones**	**Descripción**
1	CargaInfoObtenida()	Método que carga toda la información obtenida en el listado de donde se selecciona uno por uno para ver los resultados.
	Código	

```
Public Sub CargaInfoObtenida(ByVal TiposAnalisis As ArrayList, ByVal InfoObtenida As
String, ByVal NumConsola As Integer)
    ModReportes = New ModuloReportes
    OpenFileRTF = New OpenFileDialog()
    Me.FechaAuditoría = "fecha: " & DateString & " hora: " & TimeString
    If NumConsola = 1 Then
      Me._TipoAnalisis1 = New ArrayList
      Me._ResultadoAnalisis1 = New ArrayList
      Me._TipoAnalisis1.Clear()
      Dim TipoInfo As String ' Para rellenar el listview
      Me._InformacionObtenida1 = InfoObtenida
      Me._TipoAnalisis1 = TiposAnalisis
      'Almacenamos localmente el arraylist enviado por la clase ConsolaConexiones
      If Me._TipoAnalisis1.Count > 0 Then
        For indice As Integer = 0 To Me._TipoAnalisis1.Count - 1 Step 1
          TipoInfo = Me._TipoAnalisis1(indice)
          ListView1.Items.Add(TipoInfo)
        Next
      Else
        MsgBox("Primero debe analizar el servidor 2 antes de revisar los reportes",
                MsgBoxStyle.Information)
      End If
      ModReportes.AlmacenaInfoFiltrada(1)
      'Enviamos 1 especificando que corresponde al servidor 1
    End If
    If NumConsola = 2 Then
      Me._TipoAnalisis2 = New ArrayList
      Me._ResultadoAnalisis2 = New ArrayList
      Me._TipoAnalisis2.Clear()
      Dim TipoInfo As String ' Para rellenar el listview
      Me._InformacionObtenida2 = InfoObtenida
      Me._TipoAnalisis2 = TiposAnalisis
      'Almacenamos localmente el arraylist enviado por la clase ConsolaConexiones
      If Me._TipoAnalisis2.Count > 0 Then
        For indice As Integer = 0 To Me._TipoAnalisis2.Count - 1 Step 1
          TipoInfo = Me._TipoAnalisis2(indice)
          ListView2.Items.Add(TipoInfo)
        Next
      Else
        MsgBox("Primero debe analizar el servidor 2 antes de revisar los reportes",
                MsgBoxStyle.Information)
      End If
      ModReportes.AlmacenaInfoFiltrada(2)
      'Enviamos 2 especificando que corresponde al servidor 2
```

```
        End If
….Se repite el mismo código con los nombres de los objetos correspondientes a la conexión.
    End Sub
```

2	CargaArchivo()	Método público que carga las recomendaciones de archivos planos .rtf y los imprime en un cuadro de texto.

Código

```
''' <summary> ''' Método público que carga las recomendaciones de archivos planos .rtf y los imprime en un cuadro de texto.   ''' </summary>
''' <param name="NombreArchivo">Nombre del archivo plano a cargar</param>
''' <param name="InformacionObtenida">Resultado obtenido del servidor</param>
''' <remarks></remarks>
Public Sub CargaArchivo(ByVal NombreArchivo As String, ByVal InformacionObtenida As String)     'Carga la información del RichTextBoxRecomendaciones e inserta respuesta
    OpenFileRTF.DefaultExt = "*.rtf"
    OpenFileRTF.FileName = NombreArchivo
    RichTextBoxRecomendaciones.LoadFile("..\..\DocumentosHARDENING\" +
    OpenFileRTF.FileName)
    RichTextBoxRecomendaciones.AppendText("REPORTE: " & vbCrLf &
    InformacionObtenida)     'Para cargar las alarmas
    RichTextBoxAlertas.Clear()
    If InformacionObtenida.Contains("VULNERABILIDAD ENCONTRADA") Then
        PictureBoxAlarmas.BackgroundImage = ImageAlarmas.Images(0)
    Else
        PictureBoxAlarmas.BackgroundImage = ImageAlarmas.Images(1)
    End If
    RichTextBoxAlertas.AppendText("Los siguientes archivos o directorios presentan
    vulnerabilidades: " & vbCrLf & InformacionObtenida)
End Sub
```

3	GeneraReporteGeneral()	Método que genera el reporte a partir de un archivo de formato y rellena en los marcadores correspondientes la información obtenida, filtrada y analizada.

Código

```
Public Sub GeneraReporteGeneral(ByVal ResultadoAnalisis As ArrayList)
    Dim path As String
    Try
        FolderBrowserDialogDestino.ShowDialog()
        path = FolderBrowserDialogDestino.SelectedPath()
        MsgBox("Después de unos pocos segundos se abrira el documento: " & path &
        "\Informe_General_" + DateString + ".docx", MsgBoxStyle.MsgBoxHelp)
    FileCopy("..\..\DocumentosHARDENING\HistoricoReportes\FormatoInformeGeneral.docx",
        path & "\Informe_General_" + DateString + ".docx")
        Documento = MSWord.Documents.Open(path & "\Informe_General_" + DateString +
        ".docx")
        'Llenamos la información general en los marcadores correspondientes
        Documento.Bookmarks.Item("Fecha").Range.Text = DateString
        Documento.Bookmarks.Item("Datos_generales").Range.Text =
        RichTextBoxDetallesConexion.Text
        'Rellena uno por uno en los marcadores correspondientes.
        For Each Resultado As String In ResultadoAnalisis
            If Resultado.Contains("Información general del servidor") Then
                Documento.Bookmarks.Item("InformacionMáquina").Range.Text = Resultado
            End If
            If Resultado.Contains("Listado de conexiones activas") Then
                Documento.Bookmarks.Item("ListadoConexionesActivas").Range.Text = Resultado
            End If
            If Resultado.Contains("Intentos fallidos de conexión") Then
                Documento.Bookmarks.Item("IntentosFallidosConexion").Range.Text = Resultado
            End If
            If Resultado.Contains("Listado conexiones de usuarios") Then
```

```
            Documento.Bookmarks.Item("ListadoConexionesUsuarios").Range.Text = Resultado
        End If
        If Resultado.Contains("Listado uso comando su o sudo") Then
            Documento.Bookmarks.Item("ListadoUsoComandoSu").Range.Text = Resultado
        End If
        If Resultado.Contains("Listado de servicios activos") Then
            Documento.Bookmarks.Item("ServiciosActivos").Range.Text = Resultado
        End If
        If Resultado.Contains("Detalle de usuarios y contraseñas") Then
            Documento.Bookmarks.Item("DetalleUsuariosContraseña").Range.Text = Resultado
        End If
        If Resultado.Contains("Listado SUID y SGID activos") Then
            Documento.Bookmarks.Item("ListadoSUIDySGUID").Range.Text = Resultado
        End If
        If Resultado.Contains("Permisos archivos especiales") Then
            Documento.Bookmarks.Item("PermisosArchivosEspeciales").Range.Text = Resultado
        End If
        If Resultado.Contains("Lectura shadow") Then
            Documento.Bookmarks.Item("LecturaShadow").Range.Text = Resultado
        End If
        If Resultado.Contains("Permisos de multiusuarios") Then
            Documento.Bookmarks.Item("PermisosMultiusuarios").Range.Text = Resultado
        End If
        If Resultado.Contains("Detalle de grupos del servidor") Then
            Documento.Bookmarks.Item("DetalleGruposServidor").Range.Text = Resultado
        End If
        If Resultado.Contains("Listado de recursos exportados por NFS") Then
            Documento.Bookmarks.Item("ListadoRecursosNFS").Range.Text = Resultado
        End If
        If Resultado.Contains("Listado usuarios FTP") Then
            Documento.Bookmarks.Item("ListadoUsuariosFTP").Range.Text = Resultado
        End If
        If Resultado.Contains("Listado usuarios para CRON") Then
            Documento.Bookmarks.Item("ListadoUsuariosCRON").Range.Text = Resultado
        End If
        If Resultado.Contains("Políticas de cuentas") Then
            Documento.Bookmarks.Item("PoliticasCuentas").Range.Text = Resultado
        End If
        If Resultado.Contains("Gestor de arranque GRUB 2") Then
            Documento.Bookmarks.Item("ArranqueGRUB").Range.Text = Resultado
        End If
        If Resultado.Contains("Protección de LOGS") Then
            Documento.Bookmarks.Item("ProteccionLOGS").Range.Text = Resultado
        End If
        If Resultado.Contains("Inhabilitando el Ctrl+Alt+Del") Then
            Documento.Bookmarks.Item("InabilitandoCTRL_ALT_DEL").Range.Text =
            Resultado
        End If
    Next  'Guardamos el archivo generado a partir de una plantilla
        Documento.Save()
        MSWord.Visible = True 'Abrimos el documento
    Catch ex As Exception
        MsgBox("ERROR AL GUARDAR: " + ex.Message)
        MSWord.Quit()
    End Try
End Sub
```

Nota: El texto que se puede ver en el campo código se refiere al lenguaje de programación utilizado y por lo tanto no representa alguna redacción solo las letras de color verde son comentarios.
Elaborado por: José Benítez

Clase: ModuloReportes.vb

Trabaja conjuntamente con la clase ReportesAuditoría.vb y realiza los procesos de filtrado y análisis de la información obtenida como se puede ver en la tabla 30.

Tabla 30.
Detalles clase ModuloReportes.vb

ModuloReportes.vb		
#	**Métodos o funciones**	**Descripción**
1	ConexReportAudit()	Método que enlaza al objeto ya instanciado que tiene la conexión activa del módulo de comunicación, hace referencia a la clase ReportesAuditoría.vb para heredar sus objetos.
	Código	
	Public Sub ConexReportAudit(ByRef conexion As ReportesAuditoría) ReportesAuditorías = conexion End Sub	
2	ImprimeResultados() ImprimeResultados2() ImprimeResultados3() ImprimeResultados4()	Método que filtra la información en función del tipo de análisis seleccionado por el usuario y envía a su impresión conjuntamente con las recomendaciones y el análisis.
	Código	

```
Public Sub ImprimeResultados()
  TotalNumeroAnalisis = ReportesAuditorías.ListView1.Items.Count
  Dim TipoEscogido As String
  Dim Respuesta As String
  For indice As Integer = 0 To TotalNumeroAnalisis - 1 Step 1
   Try
    If ReportesAuditorías.ListView1.Items(indice).Focused = True And
    ReportesAuditorías.ListView1.Items(indice).Selected = True Then
     TipoEscogido = ReportesAuditorías.ListView1.Items(indice).Text
     ReportesAuditorías.LabelTitulo.Text = TipoEscogido.Replace("*", "Tipo análisis:")
     Respuesta = vbCrLf & ReportesAuditorías.ResultadoAnalisis1.Item(indice)
     'Carga la info del RichTextBoxRecomendaciones e inserta respuesta
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Información general del servidor" Then
     Call ReportesAuditorías.CargaArchivo("InformacionMáquina.rtf", Respuesta)
     End If
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Listado de conexiones activas" Then
     Call ReportesAuditorías.CargaArchivo("ListadoConexionesActivas.rtf", Respuesta)
    End If
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Intentos fallidos de conexión" Then
      Call ReportesAuditorías.CargaArchivo("IntentosFallidosConexion.rtf", Respuesta)
    End If
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Listado conexiones de usuarios" Then
      Call ReportesAuditorías.CargaArchivo("ListadoConexionesUsuarios.rtf", Respuesta)
    End If
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Listado uso comando su o sudo" Then
      Call ReportesAuditorías.CargaArchivo("ListadoUsoComandoSu.rtf", Respuesta)
    End If
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Listado de servicios activos" Then
      Call ReportesAuditorías.CargaArchivo("ServiciosActivos.rtf", Respuesta)
    End If
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Detalle de usuarios y contraseñas" Then
      Call ReportesAuditorías.CargaArchivo("DetalleUsuariosContraseña.rtf", Respuesta)
    End If
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Listado SUID y SGID activos" Then
      Call ReportesAuditorías.CargaArchivo("ListadoSUIDySGUID.rtf", Respuesta)
    End If
    If ReportesAuditorías.TipoAnalisis1(indice) = "* Permisos archivos especiales" Then
```

```
        Call ReportesAuditorías.CargaArchivo("PermisosArchivosEspeciales.rtf", Respuesta)
      End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Lectura shadow" Then
        Call ReportesAuditorías.CargaArchivo("LecturaShadow.rtf", Respuesta)
      End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Permisos de multiusuarios" Then
        Call ReportesAuditorías.CargaArchivo("PermisosMultiusuarios.rtf", Respuesta)
      End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Detalle de grupos del servidor" Then
        Call ReportesAuditorías.CargaArchivo("DetalleGruposServidor.rtf", Respuesta)
      End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Listado de recursos exportados por NFS"
      Then
        Call ReportesAuditorías.CargaArchivo("ListadoRecursosNFS.rtf", Respuesta)
      End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Listado usuarios FTP" Then
        Call ReportesAuditorías.CargaArchivo("ListadoUsuariosFTP.rtf", Respuesta)
      End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Listado usuarios para CRON" Then
        Call ReportesAuditorías.CargaArchivo("ListadoUsuariosCRON.rtf", Respuesta)
      End If
        If ReportesAuditorías.TipoAnalisis1(indice) = "* Políticas de cuentas" Then
      Call ReportesAuditorías.CargaArchivo("PoliticasCuentas.rtf", Respuesta)
        End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Gestor de arranque GRUB 2" Then
        Call ReportesAuditorías.CargaArchivo("ArranqueGRUB.rtf", Respuesta)
      End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Protección de LOGS" Then
        Call ReportesAuditorías.CargaArchivo("ProteccionLOGS.rtf", Respuesta)
      End If
      If ReportesAuditorías.TipoAnalisis1(indice) = "* Inhabilitando el Ctrl+Alt+Del" Then
        Call ReportesAuditorías.CargaArchivo("InabilitandoCTRL+ALT+DEL.rtf", Respuesta)
      End If
    End If
    Catch ex As Exception
      ReportesAuditoría.RichTextBoxRecomendaciones.AppendText("Clase: ReportesAuditoría |
      Error al leer recomendaciones: " & ex.Message & ".")
    End Try
  Next
End Sub
```

AlmacenaInfoFiltrada()	Método que almacena en un arraylist la información filtrada de cada análisis. Se usa para todos los servidores y recibe como parámetro el número del servidor con el que se está trabajando.

Código

3

```
Public Sub AlmacenaInfoFiltrada(ByVal NumeroConsola As Integer)
  If NumeroConsola = 1 Then
    TotalNumeroAnalisis = ReportesAuditorías.ListView1.Items.Count
    For indice As Integer = 0 To TotalNumeroAnalisis - 1 Step 1
      Dim ResultadoFiltrado As String
      Dim InfoAlarmas As String = ""
      Dim encontrar As Boolean
      Dim posicionInicial, posicionFinal, totalRecortar, Inicio, Final As Integer
      encontrar = ReportesAuditorías.InformacionObtenida1.Contains
      (ReportesAuditorías.TipoAnalisis1(indice))
      If encontrar Then
      Try
        encontrar = ReportesAuditorías.InformacionObtenida1.Contains
        (ReportesAuditorías.TipoAnalisis1(indice))
        'Si es la última posición cambia el patrón de búsqueda
        If indice = ReportesAuditorías.ListView1.Items.Count - 1 Then
```

```
            posicionInicial = ReportesAuditorías.InformacionObtenida1.LastIndexOf
            (ReportesAuditorías.TipoAnalisis1(indice))
            posicionFinal = ReportesAuditorías.InformacionObtenida1.LastIndexOf("#")
            totalRecortar = posicionFinal - posicionInicial
            ResultadoFiltrado =      ReportesAuditorias.InformacionObtenida1.Substring
            (posicionInicial, (totalRecortar - 1))
            InfoAlarmas = AnalizarAlarmas(ResultadoFiltrado, ReportesAuditorias.TipoAnalisis1
            (indice))
            ReportesAuditorias.ResultadoAnalisis1.Add(InfoAlarmas)
          Else
            posicionInicial = ReportesAuditorías.InformacionObtenida1.LastIndexOf
            (ReportesAuditorías.TipoAnalisis1(indice))
            posicionFinal = ReportesAuditorías.InformacionObtenida1.LastIndexOf
            (ReportesAuditorías.TipoAnalisis1(indice + 1))
            totalRecortar = posicionFinal - posicionInicial
          If posicionFinal > 0 Then
            'Filtramos más el texto recortando los últimos caracteres del filtro
            ResultadoFiltrado = ReportesAuditorías.InformacionObtenida1.Substring
            (posicionInicial, (totalRecortar))
            Inicio = ResultadoFiltrado.LastIndexOf("echo")
            Final = ResultadoFiltrado.Length
            If Inicio > 0 Then
              ResultadoFiltrado = ResultadoFiltrado.Remove(Inicio, (Final - Inicio))
            Else
              ResultadoFiltrado = ResultadoFiltrado.Remove(Final)
            End If
              ResultadoFiltrado = "Extracción datos de la " & ReportesAuditorías.FechaAuditoría
              & vbCrLf & "*********" & ResultadoFiltrado
              InfoAlarmas = AnalizarAlarmas(ResultadoFiltrado,
              ReportesAuditorías.TipoAnalisis1(indice))
              ReportesAuditorías.ResultadoAnalisis1.Add(InfoAlarmas)
          Else
            posicionFinal = ReportesAuditorías.InformacionObtenida1.LastIndexOf
            (ReportesAuditorías.TipoAnalisis1(indice + 1), posicionInicial)
          End If
        End If
        Catch ex As Exception
          ReportesAuditorías.ResultadoAnalisis1.Add("No se ha cargado la info del análisis: " +
          ReportesAuditorías.TipoAnalisis1(indice) + ". Vuelva a ejecutar el análisis. Error: " +
          ex.Message)
          MsgBox("No se ha cargado la infotmación del análisis: " +
          ReportesAuditorías.TipoAnalisis1(indice)  + ". Vuelva a ejecutar el análisis. Error: " +
          ex.Message)
        End Try
      End If
      Next
    End If  ' final de condición para servidor 1
    If NumeroConsola = 2 Then
    …….. ' Se repite el mismo código para cada servidor 2,3, 4.
End Sub
```

4	AnalizarAlarmas()	Función que analiza la información obtenida en búsqueda de vulnerabilidades según las técnicas de HARDENING.

Código

```
Public Function AnalizarAlarmas(ByVal TextoRecibido As String, ByVal TipoAnalisis As
String) As String
  Dim FiltoAlarma As String = ""
  Dim LineaTexto As String() ' objeto para recorrer línea por línea el texto
  Dim LineaAnalizada As String 'objeto auxiliar para comparar
  Dim PosicionSalto As Integer
  'Recorremos línea por línea para ir analizando la información
```

```
LineaTexto = TextoRecibido.Split(New [Char]() {CChar(vbLf)})
If TipoAnalisis = "* Permisos archivos especiales" Then
  For Each LineaAnalizada In LineaTexto
            'Identificamos si es una línea de permisos
  If LineaAnalizada.StartsWith("-") Or LineaAnalizada.StartsWith("d") Or
     LineaAnalizada.StartsWith("l") Then
     PosicionSalto = LineaAnalizada.Length
     LineaAnalizada = LineaAnalizada.Substring(0, PosicionSalto - 1)
  If LineaAnalizada.Contains("-rwxr--r--") Or LineaAnalizada.Contains("drwxr--r--") Then
     FiltoAlarma += LineaAnalizada + " CORRECTO" + vbLf
  End If
     FiltoAlarma += LineaAnalizada + " VULNERABILIDAD ENCONTRADA" + vbLf
  Else
     FiltoAlarma += LineaAnalizada
  End If
Next
End If
If TipoAnalisis = "* Lectura shadow" Then
   For Each LineaAnalizada In LineaTexto
     'Identificamos si es una línea de permisos
     If LineaAnalizada.StartsWith("-") Or LineaAnalizada.StartsWith("d") Or
        LineaAnalizada.StartsWith("l") Then
        PosicionSalto = LineaAnalizada.Length
        LineaAnalizada = LineaAnalizada.Substring(0, PosicionSalto - 1)
        If LineaAnalizada.Contains("-rw-------") Then
           FiltoAlarma += LineaAnalizada + " CORRECTO" + vbLf
        End If
           FiltoAlarma += LineaAnalizada+" VULNERABILIDAD ENCONTRADA"+
           vbLf
     Else
       FiltoAlarma += LineaAnalizada
     End If
   Next
   End If
   If TipoAnalisis = "* Permisos de multiusuarios" Then
     For Each LineaAnalizada In LineaTexto
     'Identificamos si es una línea de permisos
     If LineaAnalizada.StartsWith("-") Or LineaAnalizada.StartsWith("d") Or
       LineaAnalizada.StartsWith("l") Then
       PosicionSalto = LineaAnalizada.Length
       LineaAnalizada = LineaAnalizada.Substring(0, PosicionSalto - 1)
       If LineaAnalizada.Contains("drwxr-----") Then
         FiltoAlarma += LineaAnalizada + " CORRECTO" + vbLf
       End If
         FiltoAlarma += LineaAnalizada + " VULNERABILIDAD ENCONTRADA" + vbLf
     Else
       FiltoAlarma += LineaAnalizada
     End If
    Next
   End If
     If TipoAnalisis = "* Protección de logs" Then
       For Each LineaAnalizada In LineaTexto
        'Identificamos si es una línea de permisos
        If LineaAnalizada.StartsWith("-") Or LineaAnalizada.StartsWith("d") Then
          PosicionSalto = LineaAnalizada.Length
          LineaAnalizada = LineaAnalizada.Substring(0, PosicionSalto - 1)
          If LineaAnalizada.Contains("drwxr-x---") Then
             FiltoAlarma += LineaAnalizada + " CORRECTO" + vbLf
          Else
          FiltoAlarma += LineaAnalizada + " VULNERABILIDAD ENCONTRADA" + vbLf
```

```
            End If
           Else
              FiltoAlarma += LineaAnalizada
           End If
        Next
    End If
    If TipoAnalisis = "* Inhabilitando el Ctrl+Alt+Del" Then
       For Each LineaAnalizada In LineaTexto
        If LineaAnalizada.StartsWith("e") Or LineaAnalizada.StartsWith("i") Or
           LineaAnalizada.StartsWith("A") Then
           If LineaAnalizada.Contains("shutdown") Or LineaAnalizada.Contains(
             "id:5:initdefault:") Or LineaAnalizada.Contains("AllowIsolate") Then
             FiltoAlarma += LineaAnalizada + " VULNERABILIDAD ENCONTRADA" +
             vbLf
    End If
       Else
            FiltoAlarma += LineaAnalizada
       End If
       Next
    End If
   'Si no coincide devuelve el mismo string recibido
   If FiltoAlarma = "" Then
      FiltoAlarma = TextoRecibido
   End If
   Return FiltoAlarma
End Function
```

Nota: El texto que se puede ver en el campo código se refiere al lenguaje de programación utilizado y por lo tanto no representa alguna redacción solo las letras de color verde son comentarios.
Elaborado por: José Benítez

3.1.1.1 Librerías utilizadas

a) Librería SharpSSH

Se ha utilizado una librería llamada *Tammir.SharpSSH* para que maneje toda la encriptación y las llaves públicas y privadas del protocolo ssh. Para este desarrollo se utilizaron las instancias *Tamir. SshShell* y *Tamir.Streams.*

La librería utilizada tiene las siguientes características:

- Soporta la versión SSH2
- SSH File Transfer Protocol (SFTP)
- Copia segura SCP (Secure Copy)
- Intercambio de claves: diffie-hellman-group-exchange-sha1, diffie-hellman-group1-sha1
- Cifrado: 3des-cbc**,** aes128-cbc
- Mensaje de código autenticación: hmac-md5
- Tipo de clave de hots: ssh-rsa**,** ssh-dss
- Autentificación de usuario: password**,** public key (RSA, DSA)
- Puerto de reenvío

- Stream de reenvío
- Ejecución remota (exec)
- Generación de pares de claves DSA y RSA**.**
- Cifrado de contraseña a una clave privada

b) Control de office

Para exportar los reportes en un documento final de formato *.docx* correspondiente a Microsoft Word se ha importado tres librerías adicionales:

- Microsoft.Office.Interop.Word: Para el control de Microsoft Word
- System.IO: Para el control de sistema de archivos
- Microsoft.Office.Interop: Para importar componentes de Microsoft Office

c) Librería de hilos

Se utiliza para este proyecto la librería *System.Threading* que proporciona clases e interfaces que permiten la programación multiproceso. Se utilizan dos hilos para cada conexión.

d) Otras librerías utilizadas

Se utilizan estas librerías adicionales muy comunes en visual basic .net para el procesamiento de texto y controles más generales.

- System.Text
- System.Collections

3.1.1.2 Control de excepciones

Es importante tener en cuenta que es indispensable el control de excepciones porque se mejora el rendimiento de la aplicación en la tabla 31 se puede ver lo mencionado.

Tabla 31.
Detalles de las excepciones implementadas

Nombre clase	Nombre función, método o evento	Mensaje recibido
Cliente_SSH_SYN Cliente_SSH_SYN2 Cliente_SSH_SYN3 Cliente_SSH_SYN4	ConexionSYN_SSH()	Clase: ClienteSSH_SYN | Sub: ConexionSYN_SSH | Parametros: Ip: 192.168.1.9 User: root | Descripción: usuario o contraseña inválidos. Error: Auth fail.

Cliente_SSH_SYN Cliente_SSH_SYN2 Cliente_SSH_SYN3 Cliente_SSH_SYN4	EjecutarCmdSSH_SYN()	Clase: Cliente_SSH_SYN.vb Sub: LeerMensaje(). | Error al ejecutar el comando: " & mensCliente & " | Detalles: " + ex.Message
Cliente_SSH_SYN Cliente_SSH_SYN2 Cliente_SSH_SYN3 Cliente_SSH_SYN4	ObtenerRespServ()	Clase: Cliente_SSH_SYN.vb Sub: ObtenerRespServ(). Error al obtener respuesta detalles: " + ex.Message
Cliente_SSH_SYN Cliente_SSH_SYN2 Cliente_SSH_SYN3 Cliente_SSH_SYN4	DesconectarSshSincrona()	Clase: ClienteSSH_SYN | Function: DesconectarSshSinCRONa() | Description: " & ex.Message
Cliente_SSH_SYN Cliente_SSH_SYN2 Cliente_SSH_SYN3 Cliente_SSH_SYN4	ObtenerRespServ()	!Alerta¡ Promt no encontrado. Por favor ingrese correctamente el prompt para un correcto sincronismo
ModuloReportes.vb	ImprimeResultados() ImprimeResultados2() ImprimeResultados3() ImprimeResultados4()	Clase: ReportesAuditoría | Error al leer recomendaciones: " & ex.Message & "."
ModuloReportes.vb	AlmacenaInfoFiltrada()	No se ha cargado la información del análisis: " + "Tipo análisis" + ". Vuelva a ejecutar el análisis. Error: " + ex.Message
ReportesAuditoría.vb	GeneraReporteGeneral()	MsgBox("Error al guardar el informe general. Descripción del error: " + ex.Message, MsgBoxStyle.Critical)

Nota: El texto que se puede ver en el campo *Mensaje recibido* no representa redacción literaria, expresa el lenguaje de programación que hace posible la visualización del mensaje.
Elaborado por: José Benítez

3.1.2 Integración del producto

En el presente proyecto de grado se ha desarrollado una aplicación que puede ser vista desde la perspectiva de 3 capas y como se comunican entre ellas secuencialmente, considerando que en el software creado no necesita una base de datos en consecuencia no se aplica la capa de datos y en su lugar identificamos una capa de conexiones y otra capa de comunicación, así pues se pude entender de una mejor manera la arquitectura que se ha implementado, se pude ver una interpretación gráfica de la comunicación entre cada una de ellas en la figura 31 que se encuentra en la siguiente página.

Interpretación gráfica de las capas identificadas.

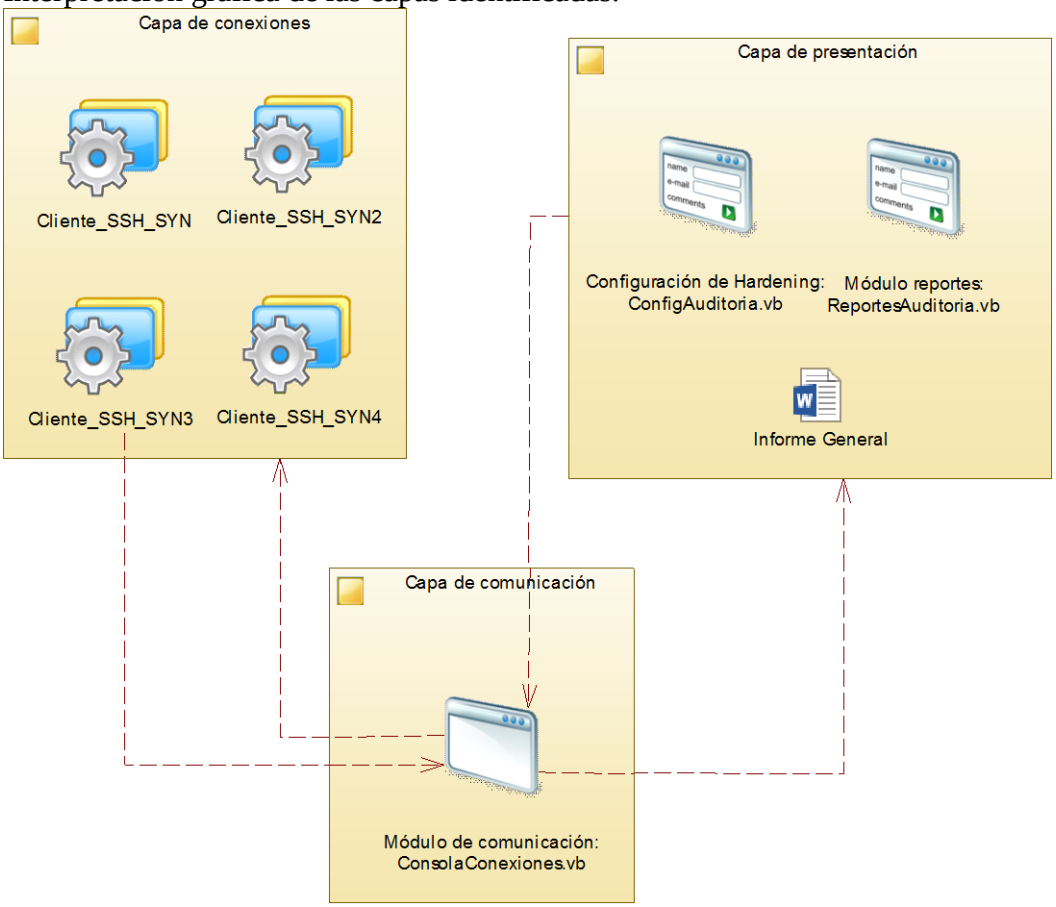


Figura 31. Elaborado por: José Benítez

Capa de conexiones

Es la capa que se conecta directamente con el servicio *OpenSSH* instalado y activo de los servidores Linux, establece una comunicación síncrona es decir que el cliente estará escuchando constantemente pero imprimirá los bits enviados por el servidor solamente después de que ha recibido el *promt* (últimos caracteres), este identifica que una consola de Linux está lista para recibir comandos.

Esta capa también es la encargada de controlar todo el protocolo ssh y enviar a la capa de comunicación las respuestas recibidas por los servidores y así mismo envía los comandos a ejecutar especificados por el usuario.

Capa de comunicación

En esta capa se recibe toda la información en cadenas de texto, se imprimen todos los bits ya interpretados y decodificados en un cuadro de texto, como se ha delimitado el alcance a 4 terminales consecuentemente se tiene 4 cuadros de texto, todos ellos agrupados en diferentes pestañas.

Está sincronizada con un evento público en la capa de conexiones que estará constantemente escuchando las respuestas del servidor conectado.

Para el envío de mensajes al servidor se escribe en un cuadro de texto el comando que deseamos y presionando el botón ejecutar del formulario correspondiente al servidor de interés, envía los mensajes a la capa de conexiones para su ejecución.

También es el enlace a la siguiente capa de presentación que se conforma por dos módulos; el módulo de configuración y el de reportes.

Capa de presentación

Es la capa relacionada con el módulo de reportes, y con el módulo de configuración de auditoría, es el usuario final el que se comunica y selecciona los tipos de análisis que desea obtener y posteriormente revisa toda la información filtrada y procesada, la misma información le servirá para la toma de decisiones, en función de las recomendaciones incluidas en cada análisis.

Se presenta al final un informe general, en un documento de Microsoft Word que incluye todas las recomendaciones y resultados obtenidos de cada servidor. Para la navegación en el módulo de reportes lo realiza un servidor a la vez y se genera un documento por cada equipo.

3.1.3 Diagramas de implementación

Se refieren a la descripción para la implementación del software, donde hace referencia al código y a la estructura física del mismo.

Existen dos tipos de diagramas de implementación;

- Diagramas de componentes
- Diagramas de despliegue

a) **Diagrama de componentes:** se usa para modelar la vista lógica del software y ayuda a documentar el comportamiento funcional de los componentes o elementos del programa desarrollado, en este caso el componente módulo de comunicación es el medio de información entre la aplicación, los servidores y el usuario final, como se puede ver en la figura 32.

Diagrama de componentes en la aplicación.

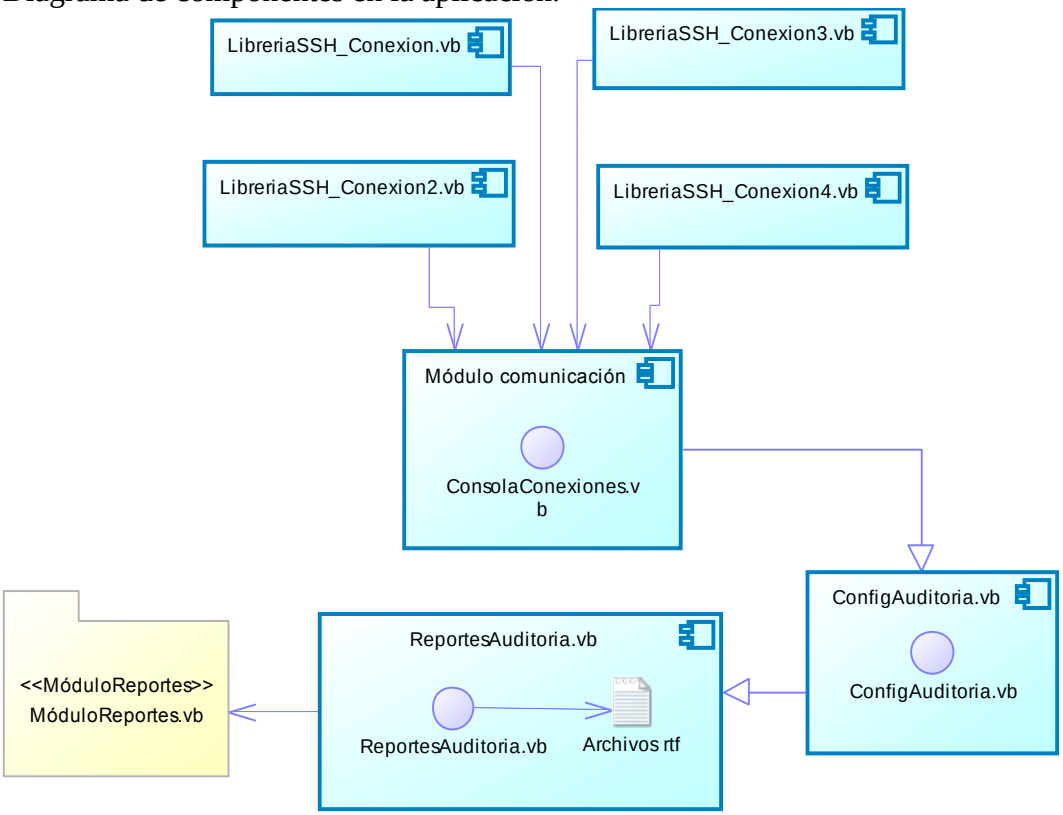


Figura 32. Elaborado por: José Benítez

b) Diagramas de despliegue: con este diagrama podemos visualizar la disposición física y la arquitectura en tiempo de ejecución del software, cada nodo es representado por un equipo físico y se interpreta por los cubos.

Para este proyecto, en el caso del nodo <<Servidores Linux>> se puede interpretar que tiene virtualizados 4 máquinas, con los sistemas operativos linux, este diagrama aplica solo para la presentación del proyecto pero en realidad deberían ser servidores físicos independientes. Pero en la actualidad este escenario se repite puesto que en la mayoría de empresas ya hacen uso del concepto de virtualización y separan los servicios más importantes en diferentes servidores como se puede ver en la figura 33 se tiene dos nodos para el proyecto.

Diagrama de despliegue en la aplicación.

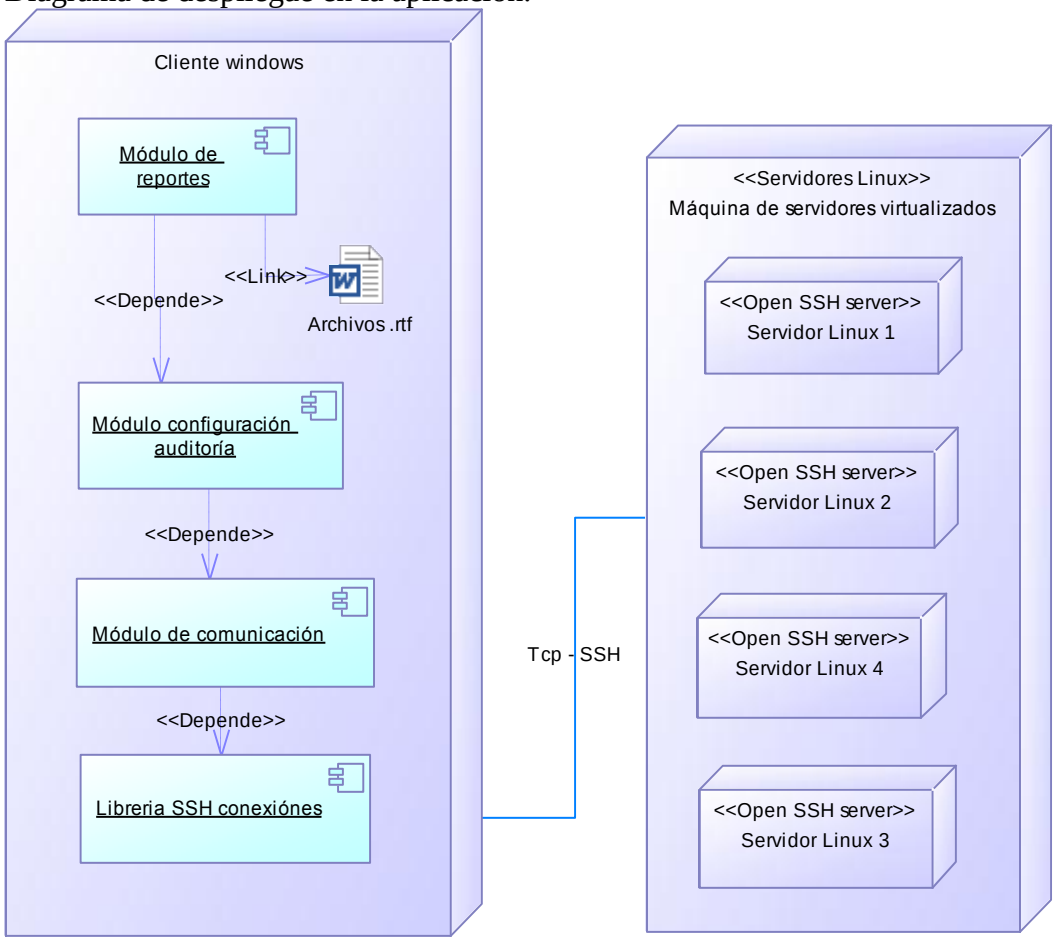


Figura 33. Elaborado por: José Benítez

3.1.4 Pruebas del producto

En esta parte se procede a realizar las pruebas de funcionalidad con el usuario final, para cada uno de los módulos desarrollados en este proyecto.

En la tabla 32 tenemos las descripciones de las pruebas realizadas al módulo de conexiones.

Tabla 32.
Pruebas módulo de conexiones ssh.

Módulo de conexiones SSH						
#	**Descripción del caso de prueba**	**Pre requisito**	**Resultado esperado**	**Resultado obtenido**	**Estado**	**Observaciones**
1	Se conecta exitosamente al servidor.	Usuario, contraseña, puerto, distribución, número de servidor	Conexión exitosa consola abierta.	Conexión exitosa consola abierta.	Exitosa	Ninguna
2	Si ingresa mal	Usuario, contraseña,	Clase: ClienteSSH_SYN |	Clase: ClienteSSH_SYN |	Exitosa	Ninguna

	parámetros controla excepción.	puerto, distribución, número de servidor	Sub: ConexionSYN_SSH | Parámetros: Ip: 192.168.1.12 User: root |Descripción: usuario o contraseña inválidos. Error: Auth fail.	Sub: ConexionSYN_SSH | Parametros: Ip: 192.168.1.12 User: root |Descripción: usuario o contraseña inválidos. Error: Auth fail.		
3	Inicializa hilos de lectura y escritura.	Conexión exitosa	Envía comandos y recibe respuestas imprime síncronamente.	Envía comandos y recibe respuestas imprime síncronamente.	Exitosa	Ninguna

Nota. Elaborado por: José Benítez

Para las pruebas del módulo configuración de auditoria se puede ver en la tabla 33.

Tabla 33.
Pruebas módulo configuración de auditoría.

Módulo de configuración de auditoría						
#	**Descripción del caso de prueba**	**Pre requisito**	**Resultado esperado**	**Resultado obtenido**	**Estado**	**Observaciones**
1	4 pestañas con 4 componentes iguales por separado	Conexión exitosa	Conexión exitosa consola abierta.	Conexión exitosa consola abierta.	Exitosa	Ninguna
2	Botón para seleccionar todos los tipos de análisis y envía a ejecutarse en un vector.	Conexión exitosa y servidor escuchando	Empieza la ejecución de todos los comandos uno a uno, imprime simultáneamente los resultados conforme se va obteniendo.	Empieza la ejecución de todos los comandos uno a uno, imprime simultáneamente los resultados conforme se va obteniendo.	Exitosa	Ninguna
3	Botón ejecutar solo tipos de análisis escogidos.	Conexión exitosa y servidor escuchando	Empieza la ejecución de todos los comandos uno a uno, imprime simultáneamente los resultados conforme se va obteniendo.	Empieza la ejecución de todos los comandos uno a uno, imprime simultáneamente los resultados conforme se va obteniendo.	Exitosa	Ninguna

Nota. Elaborado por: José Benítez

Tenemos el detalle de las pruebas realizadas al módulo de comunicación, en la tabla 34 que se encuentra en la siguiente página.

Tabla 34.
Pruebas módulo de comunicación.

Módulo de comunicación						
#	**Descripción del caso de prueba**	**Pre requisito**	**Resultado esperado**	**Resultado obtenido**	**Estado**	**Observaciones**
1	Carga al lado izquierdo la imagen de la distribución y obtiene detalles de conexión ssh	Conexión exitosa	Carga la imagen de la distribución captura detalles conexión	Carga la imagen de la distribución captura detalles conexión	Exitosa	Ninguna
2	Cotones para desconectar cambian de imagen según el servidor conectado.	Conexión exitosa	Cambia de imagen del botón para desconectar.	Cambia de imagen del botón para desconectar.	Exitosa	
3	Enlaza al módulo de configuración de auditoría	Conexión exitosa	Abre el formulario para configurar la auditoría y carga pestaña según consola conectada.	Abre el formulario para configurar la auditoría y carga pestaña según consola conectada.	Exitosa	Ninguna
4	Carga en listwiew tipos de análisis seleccionados y vincula con el módulo de reportes	Configuración auditoría exitosa y obtención de información finalizada	Abre el formulario para configurar la auditoría y carga pestaña según consola conectada.	Abre el formulario para configurar la auditoría y carga pestaña según consola conectada.	Exitosa	Ninguna

Nota. Elaborado por: José Benítez

En la tabla 35 tenemos las descripciones de las pruebas al módulo de reportes.

Tabla 35.
Pruebas módulo de reportes.

Módulo de reportes						
#	**Descripción del caso de prueba**	**Pre requisito**	**Resultado esperado**	**Resultado obtenido**	**Estado**	**Observaciones**
1	Carga automáticamente el listview con el listado escogido en la configuración	Obtención de información terminada impresa en el cuadro de texto.	Imprime todos los tipos de análisis en el listview correspondiente al servidor conectado.	Imprime todos los tipos de análisis en el listview correspondiente al servidor conectado.	Exitosa	Ninguna
2	Filtra toda la información y lo almacena en un vector para la posterior lectura del mismo.	Resultados obtenidos, servidor, tipos de análisis seleccionados.	Obtiene vector con la información clasificada, analizada y filtrada.	Obtiene vector con la información clasificada, analizada y filtrada.	Exitosa	Ninguna

3	Selecciona tipo de análisis en listview y carga en cuadro de texto el archivo plano .rtf conjuntamente con el índice del vector lleno con la información filtrada.	Vector de resultados obtenidos lleno.	Imprime el resultado obtenido del servidor conjuntamente con las recomendaciones necesarias.	Imprime el resultado obtenido del servidor conjuntamente con las recomendacion es necesarias.	Exitosa	Ninguna
4	En el botón exportar genera un informe general con toda la información cargada de en el vector.	Vector de resultados obtenidos lleno.	Captura una plantilla y carga toda la información del vector de resultados en un archivo .docx abre el documento y lo guarda.	Captura una plantilla y carga toda la información del vector de resultados en un archivo .docx abre el documento y lo guarda.	Exitosa	Ninguna

Nota. Elaborado por: José Benítez

3.2 Fase de entrega

En esta fase se demostrará por medio de una pila de entrega, el uso del sprint por cada clase desarrollada para la integración de todas las clases, métodos y funciones.

Debe permitir ver los avances y cambios realizados en todo el tiempo que se ha tomado para este proyecto.

3.2.1 Entrega del producto

En esta fase se obtiene la pila de entregables donde se detallan el sprint de cada requerimiento como se puede ver en la tabla 36.

Tabla 36.
Pila de entregables de software SHL.

Sistema HARDENING Linux (SHL) - Pila de entregables			
Requerimiento	**Sprint**	**Encargado**	**Días**
Ambiente de desarrollo y pruebas	Análisis de la arquitectura.	José Benítez	1
	Virtualización y configuración Centos.	José Benítez	1
	Virtualización y configuración Ubuntu.	José Benítez	1
	Virtualización y configuración Fedora.	José Benítez	1
	Virtualización y configuración OpenSUSE.	José Benítez	1
Módulo de conexión	Estudio librería SharpSSH y sus componentes.	José Benítez	2
	Programación clase de conexión SSH.	José Benítez	10
	Creación de hilos.	José Benítez	2
	Vinculación con el módulo de comunicación.	José Benítez	2
	Desconexión y control de excepciones.	José Benítez	2
	Formulario de loggin al sistema.	José Benítez	2

Módulo de comunicación	Desarrollar bloque de consolas para 4 conexiones, enlazar los hilos de lectura.	José Benítez	10
	Bloque botones para desconexiones de las conexiones abiertas.	José Benítez	2
	Bloque para información de la conexión, cambia imágenes de botones, e imprime detalles conexión ssh.	José Benítez	2
	Bloque para reportes y recomendaciones.	José Benítez	2
	Inicia la conexión enlaza hilos y obtiene promt.	José Benítez	2
	Botón salir e ingresar al módulo de configuración de la auditoría.	José Benítez	2
Recomendaciones de HARDENING	Lectura de libros de HARDENING de servidores y resumen.	José Benítez	15
	Pruebas de ejecución en los servidores montados, identificación de diferencias entre distribuciones.	José Benítez	3
	Selección de los comandos a ejecutar y clasificación de los tipos de análisis.	José Benítez	9
	Redacción de archivos .rtf que contienen las recomendaciones de HARDENING.	José Benítez	3
Módulo de configuración de auditoría	Formulario con 4 pestañas, se selecciona la pestaña del último servidor conectado.	José Benítez	5
	CheckList con los tipos de auditoría a ejecutarse 4 iguales, carga y almacena en un vector.	José Benítez	3
	4 Botones para ejecutar todo, no seleccionar ninguno y ejecutar seleccionados.	José Benítez	7
	Método para rellenar el arraylist y enviar al módulo de comunicación a ejecutar.	José Benítez	5
Módulo de reportes	Formulario con paneles y ListView con dos pestañas para reportes y alarmas.	José Benítez	1
	Método para filtrado de la información sin analizar y almacenamiento en un vector.	José Benítez	10
	Método para la impresión del vector y análisis de su información según algoritmos de análisis personalizados.	José Benítez	10
	Rellenar el ListView con los tipos de análisis escogidos, imprime el vector lleno según opción la escogida.	José Benítez	3
	Selección del documento .rtf que se imprime en la sección de recomendaciones.	José Benítez	3
	Botón para exportar el resultado en un informe general en un archivo .docx correspondiente a Microsoft Word.	José Benítez	3
Pruebas	Pruebas de funcionalidad módulo de conexiones	José Benítez	3
	Pruebas de funcionalidad módulo de comunicación.	José Benítez	5
	Pruebas de funcionalidad módulo de configuración de auditoría.	José Benítez	2
	Pruebas de funcionalidad módulo de reportes.	José Benítez	5
	Pruebas de caja negra y caja chica del software desarrollado.	José Benítez	5
Documentación	Documentación capítulo 1 resumen teórico de HARDENING.	José Benítez	10
	Modelado de diagramas correspondientes al capítulo 2	José Benítez	10
	Modelado de diagramas y requerimientos de la metodología SCRUM del capítulo 3	José Benítez	10
	Modelado de diagramas tablas y manuales del capítulo 4	José Benítez	10
	Corrección de errores ortográficos de fondo y forma y acoplamiento a las normas APA	José Benítez	10

Nota. Elaborado por: José Benítez

CAPÍTULO 4

FASE DE SOPORTE Y CIERRE DEL PROYECTO

4.1 Fase de soporte y mantenimiento

Las características para las configuraciones de los servidores que se han establecido para este proyecto se detallan en esta fase así como también las diferentes soluciones que se pueden dar ante los posibles problemas con la escalabilidad en las distribuciones escogidas para este proyecto de titulación.

No necesariamente en el ámbito empresarial este escenario se repite porque las distribuciones linux podrían estar instaladas en *servidores blade* o directamente en equipos físicos como también podrían estar montados en máquinas virtuales con otras herramientas que nos ayudan a virtualizar, sin embargo todo este capítulo se lo hace referencia al escenario que se ha delimitado en el alcance de software.

Para tener una idea clara del escenario que planteamos para el desarrollo y pruebas se ha creado una ilustración que se puede ver en la figura 34.

Gráfica de la arquitectura para la aplicación.

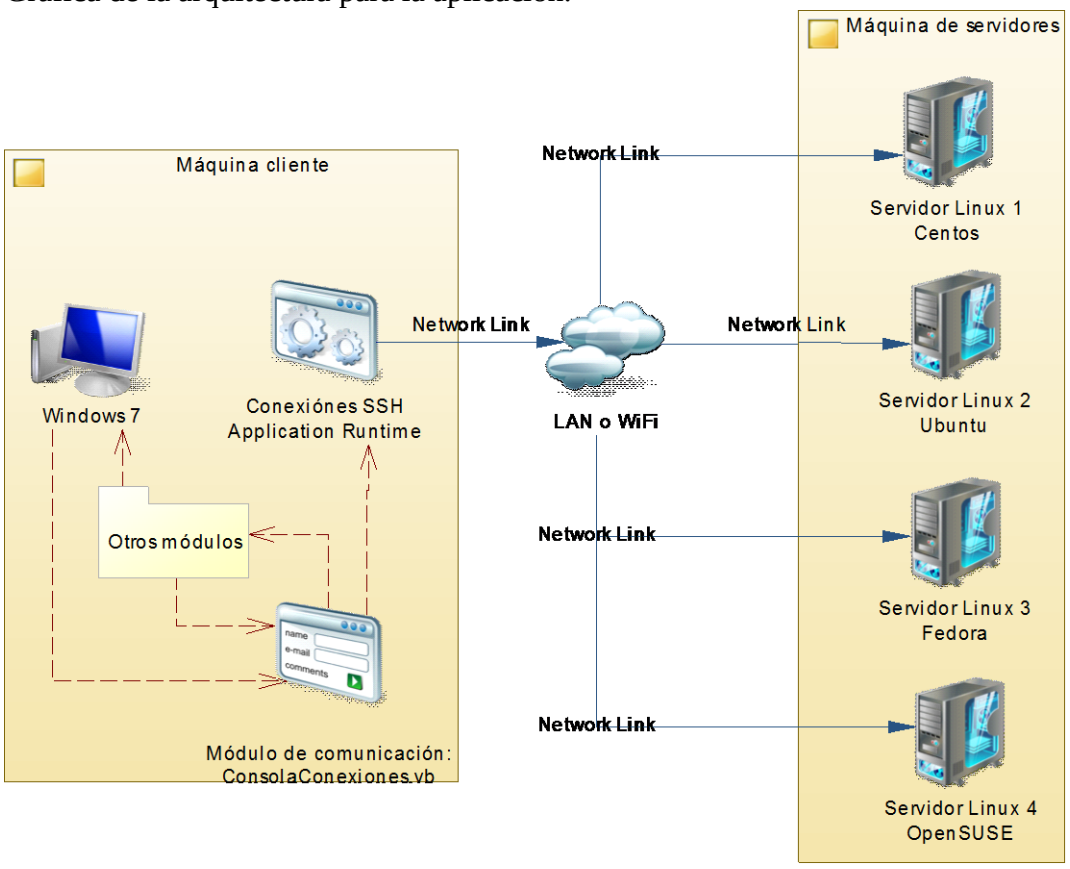


Figura 34. Elaborado por: José Benítez

4.1.1 Requerimientos del software

En esta fase se especifica los requerimientos básicos que se deben considerar para que la aplicación SHL tenga un correcto funcionamiento.

En este proyecto no se utilizan bases de datos porque los reportes finales son documentos de Microsoft Word por lo tanto es necesario que en la máquina cliente se tenga instalado lo siguiente:

- Microsoft office 2010 en adelante
- Net Framework 4.0
- Windows 7 en adelante

En la máquina de los servidores se recomienda tener instalado:

- Herramienta de virtualización *VMware® Workstation* 10 en adelante.
- Máquinas virtuales instaladas las distribuciones especificadas con anterioridad, esto aplica en este proyecto no necesariamente se repite en todos.

4.1.2 Configuración de ambiente de desarrollo

Para el desarrollo y pruebas de esta aplicación se ha instalado máquinas virtuales con las siguientes características especificadas en la siguiente tabla.

Tabla 37.
Máquinas virtuales detalles.

Característica	Recomendación
Herramienta para virtualización	VMware® Workstation 10
Tipo instalación	Personalizada
Compatibilidad mínima de hardware	Workstation 8.0
Tipo de disco	SCSI
Adaptador de red	Bridge Automatic
Servicio indispensable	OpenSSH
Memoria RAM	2GB

Nota. Elaborado por: José Benítez

Estas recomendaciones para la configuración del ambiente de desarrollo aplican en el caso que se desee ejecutar y probar el software, pero no necesariamente siempre el mismo caso. La aplicación es funcional también con cualquier distribución perteneciente a las familias de las distribuciones escogidas, el único requisito indispensable es que tengan el servicio ssh levantado.

El software está desarrollado específicamente para 4 distribuciones pero existen distribuciones de linux que nacieron a partir de las escogidas, es decir que tiene escalabilidad debido a que se conservan los nombres de los archivos y directorios de los ficheros, en la tabla 38 se detallan las distribuciones compatibles.

Tabla 38.
Compatibilidad de la aplicación

Familia	Distribución seleccionada	Distribuciones compatibles	Versiones
Debian	Ubuntu	Kubuntu, Zentyal, Edubuntu, Fluxbuntu, Newtoos, Lubuntu, Mythbuntu, Xubuntu	Superiores al 2013
Slackware linux	OpenSUSE	Caixa Mágica, SLES, Astaro, SLED	Superiores al 2013
RedHat	Fedora	Momonga, MythDora, Ekaaty, Synergy, Fusion, Parsidora, Fuduntu, MeeGo	Superiores al 2013
RedHat	Centos	Elastix, BlueOnyx, Trixbox	Superiores al 2013

Nota. Elaborado por: José Benítez

4.1.3 Configuraciones de máquinas virtuales

Para establecer la conexión y poder realizar las pruebas necesarias en función de los tipos de análisis escogidos, para este software son necesarios 3 servicios adicionales instalados, configurados y en ejecución éstos en algunos casos no vienen por defecto instalados, y tomando en cuenta que en cada distribución los comandos se diferencian a excepción de Fedora y Centos que son los mismos. Se detallan los comandos necesarios en cada distribución en la tabla 39:

Tabla 39.
Preparación ambiente desarrollo

Servicio	Acción	Fedora y centos	OpenSUSE	Ubuntu
OpenSSH	**Instalación**	yum -y install openssh openssh-server openssh-clients	yast -i openssh	sudo apt-get install openssh-server
	Archivo de configuración	/etc/ssh/sshd_config	/etc/ssh/sshd_config	/etc/ssh/sshd_config
	Iniciar	service sshd start	rcsshd start	sudo /etc/init.d/ssh start
	Parar	service sshd stop	service sshd stop	sudo /etc/init.d/ssh stop
	Reiniciar	service sshd restart	rcsshd restart	sudo /etc/init.d/ssh restart
FTP	**Instalación**	yum install vsftpd	zypper in vsftpd	sudo apt-get install vsftpd
	Archivo de configuración	/etc/vsftpd/vsftpd.conf	/etc/vsftpd.conf	/etc/vsftpd.conf
	Iniciar	systemctl start vsftpd.service	systemctl enable vsftpd	service vsftpd start

	Parar	systemctl stop vsftpd.service	systemctl stop vsftpd	service vsftpd stop
	Reiniciar	systemctl restart vsftpd.service	systemctl restart vsftpd	service vsftpd restart
NFS	**Instalación**	yum -y install nfs-utils	yast2 -i nfs -kernel-server	apt-get install nfs -common nfs -kernel-server
	Archivo de configuración	/etc/exports	/etc/exports	/etc/exports
	Iniciar	systemctl start nfs-server	rcnfsserver start	sudo service nfs -kernel-server start
	Parar	systemctl stop nfs -server	rcnfsserver stop	sudo service nfs -kernel-server stop
	Reiniciar	systemctl restart nfs -server	rcnfsserver restart	sudo service nfs -kernel-server reload

Nota. Comandos a ejecutar para instalar y controlar los servicios en cada distribución.
Elaborado por: José Benítez

4.2 Fase de cierre del producto

En esta fase se especifican los detalles del documento a presentarse con el historial de los avances, donde se ha detallado las fechas de culminación y aprobación de los capítulos de este proyecto de titulación, conjuntamente con los detalles del documento y finalmente los datos del comité aprobador.

4.2.1 Detalles del documento

En la tabla 40 tenemos la información general del documento elaborado para el desarrollo de este proyecto de titulación.

Tabla 40.
Detalles documento final.

Información del documento	
Identificación del documento	Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS
Responsable	José Benítez
Fecha de emisión	05/03/2015
Fecha última modificación	30/04/2015
Nombre del archivo	Trabajo de titulación José Benítez

Nota. Elaborado por: José Benítez

4.2.2 Historial del documento

En la tabla 41 está el histórico del documento hasta llegar a la culminación del presente proyecto.

Tabla 41.
Detalles fechas avances documentación

Avances	Fecha de culminación	Fecha aprobación
Capítulo 1	30/09/2014	04/10/2014
Capítulo 2	04/12/2012	13/12/2014
Capítulo 3	25/01/2015	31/01/2015
Capítulo 4	05/02/2015	07/02/2015
Conclusiones y recomendaciones	13/02/2015	14/02/2015
Referencia glosario y anexos	20/02/2015	21/02/2015
Borrador finalizado	26/02/2015	28/02/2015
Documento final corregido	-------	---------

Nota. Fechas relacionadas al avance del documento de tesis no relacionados al software.
Elaborado por: José Benítez

4.2.3 Aprobación

En la tabla 42 esta las fechas de aprobación de las personas involucradas.

Tabla 42.
Aprobación del documento.

Rol	Nombre	Aprobación	Fecha
Director del trabajo de titulación	Ing. Xavier Calderón	SI	28/02/2015
Editor y desarrollador	José Benítez	SI	

Nota. La fecha corresponde a la aprobación definitiva del comité aprobador.
Elaborado por: José Benítez

CONCLUSIONES

- El software se puede analizar e identificar las vulnerabilidades del sistema operativo analizado y consecuentemente genera un reporte con las recomendaciones y el resultado obtenido.

- El software contiene un módulo de conexiones que consta de 4 clases cada una contiene el mismo código y no se aplica el polimorfismo de programación porque cada una establece un canal de comunicación con el servidor conectado.

- Durante toda la investigación realizada se ha identificado muchos cambios en las nuevas distribuciones por lo tanto el análisis protección de LOGS y en el listado del uso del comando *su* se utiliza JOURNALCRL.

- El sistema contempla los aspectos generales que deberían tener todo servidor, no se considera servicios específicos como bases de datos, servidor web o de dominios así como también el firewall o las *ACL (Access Control List)* que puedan tener porque eso depende de la red en la que se encuentre.

- El software está dirigido a administradores de sistemas operativos y en consecuencia se omite criterios de seguridad como por ejemplo: cambiar los puertos por defecto de los servicios ssh, FTP, telnet, mysql.

- No existe servidor invulnerable porque siempre existirán riesgos, pero con el software si podemos identificar que tan seguro o expuesto a un ataque esta nuestro sistema operativo.

- La aplicación permite realizar el análisis de los 4 servidores en paralelo pero para acceder al módulo de reportes la visualización es unitaria para cada servidor.

- Las 4 versiones de Linux que se han escogido son en base a la estabilidad que ofrecen cada una de ellas y basándose en las distribuciones preferidas por las empresas, por lo tanto es compatible con otras versiones derivadas de las que se han escogido para este proyecto.

RECOMENDACIONES

- Para analizar otras distribuciones se recomienda verificar la familia de la que provienen para saber la compatibilidad del software, se deberá ingresar el promt que corresponde a la distribución que deseamos conectar.

- Desde el módulo de comunicación después de tener una conexión abierta se recomienda no abrir aplicaciones desde el software como el editor vi por ejemplo, debido a que no es una consola propiamente dicha, por la arquitectura en la que se desarrolló se envían los comandos desde un cuadro de texto.

- En el caso de ejecutar algún comando que demore en responder o abre una aplicación propia del sistema como el *editor vi* se recomienda ejecutar el comando *"parar"* para salir de la ejecución del mismo.

- Se recomienda la lectura del manual de usuario, antes de proceder a la manipulación del software que también está disponible en la aplicación en ejecución. será de mucha utilidad para tener una adaptación breve a la aplicación.

- Se recomienda utilizar esta aplicación para servidores con sistemas operativos Linux de versiones no inferiores al año 2013.

- Se recomienda en los servidores que se desea auditar tener instalado y levantado el servicio de SSH porque es el único protocolo por el cual el software accede a los servidores.

- Si se desea auditar nuevas distribuciones de las que se han seleccionado para este proyecto se recomienda revisar si existen diferencias en los archivos de configuración o en los directorios.

GLOSARIO DE TÉRMINOS

HARDENING: Conjunto de conceptos y técnicas, políticas empresariales, para mejorar las seguridades en servidores, es el proceso de asegurar un sistema mediante la reducción de vulnerabilidades en el mismo, para garantizar la integridad de la información.

TI: Tecnologías de la información, término utilizado para referirse a las nuevas tecnologías utilizadas para el procesamiento de la información y la comunicación.

SCRUM: Es una metodología de desarrollo que combina las características de ser ágil y adoptar una estrategia de desarrollo incremental, se basa en la adaptación continua de las cambios en la evolución del proyecto y en los tiempos estimados, interactúa mucho más con los clientes, y al final se entrega la documentación de cada interacción o también llamados sprints.

SUDOERS: En linux se denomina al grupo de usuarios que tienen privilegios de administrador cuyo listado se encuentra en el archivo */etc/sudoers*.

LOGS: También conocido como registros es un conjunto de reportes que contienen los mensajes generados por los servicios, programas, actividades de los usuarios y mensajes de las aplicaciones.

SCRIPTS: Es un programa de código por lotes usualmente en archivos de texto, su codificación es interpretada por el servidor que lo tiene en sus directorios pueden tener en sus primeras líneas de código; !/bin/Bash; #!/bin/ksh ; #!/bin/csh

GUID: Representa el ID o identificador único del grupo de usuarios en los sistemas operativos Linux.

SUID: Representa el ID o identificador del usuario en los sistemas operativos Linux.

NFS: (Network File System) conocido también como sistema de archivos de red y es un protocolo para compartir ficheros remotos.

CRON: Cron es el nombre del programa que permite a usuarios Linux/Unix ejecutar automáticamente comandos o scripts a una hora o fecha específica, o se puede entender como una aplicación para calendarizar tareas.

FTP: Es el protocolo de transferencia de archivos que permite compartir archivos desde un servidor a un cliente o viceversa, no dispone de buenas seguridades.

AWK: Es un lenguaje de programación diseñado para procesar datos basados en texto, fue creado para el análisis de textos.

SYSTEMD: **"**Es un servicio o demonio de administración del sistema diseñado exclusivamente para Linux, fue desarrollado para reemplazar el sistema de inicio init heredado de los sistemas operativos" (Wikipedia®, 2015)

JOURNALCRL: Servicio que lleva un registro diario en el que se almacena la información de todas las acciones realizadas en el servidor.

LISTADO DE REFERENCIAS

Álvarez Martín & Gonzales Pérez, P. (2013). *Hardening de servidores GNU/Linux.* Madrid, España: 0xWORD.

Ferran Pichel, L. (Marzo de 2011). *Internet security auditors.* Obtenido de ISec Lab #13 Hardening básico de Linux permisos y configuraciónes: http://www.isecauditors.com/sites/default/files//files/iseclab13-hardening_basico_linux_permisos_y_configuraciones.pdf

Gigena, M. (26 de 10 de 2012). *Todo Java*. Obtenido de Implementación en JAVA (Sockets) : http://labojava.blogspot.com/2012/10/implementacion-en-java-sockets.html

INTECO Instituto Nacional de Tecnologías y la comunicación. (Marzo de 2009). *Instituto nacional de ciberseguridad.* Obtenido de Ingeniería del software: metodologías y ciclos de vida: https://www.incibe.es/file/N85W1ZWFHifRgUc_oY8_Xg

Palacio, J. (2006). *Navegapolis.* Obtenido de El modelo scrum: http://www.navegapolis.net/files/s/NST-010_01.pdf

Peña, T. F. (27 de 11 de 2014). *Programa de Administración de Sistemas e Redes*. Obtenido de Instalación de NFS en Debian: http://persoal.citius.usc.es/tf.pena/ASR/Tema_4html/node7.html

Wikipedia®. (26 de Noviembre de 2014). *Wikipedia la enciclopedia libre*. Obtenido de Virtualización: http://es.wikipedia.org/wiki/Virtualizaci%C3%B3n

Wikipedia®. (18 de 02 de 2015). *Wikipedia la enciclopedia libre*. Obtenido de systemd: http://es.wikipedia.org/wiki/Systemd

ANEXOS

Anexo 1. Manual de usuario

Contenido

1. Introducción

El manual de usuario describe paso a paso el funcionamiento del software desarrollado para llegar al objetivo final que es obtener un informe general de todos los análisis realizados al servidor escogido.

Es importante destacar que el software es dirigido a un administrador de sistemas operativos y la interfaz gráfica está basado en las últimas tendencias de aplicaciones de escritorio con iconos más grandes y colores llanos para la optimización de los nuevos equipos táctiles, es por esta razón que difiere la acostumbrada navegación web con las barras de menús cambiando el diseño.

Para obtener el informe general el usuario debe seguir en orden paso a paso cada instrucción.

2. Requisitos

Máquina cliente se recomienda que tenga instalado lo siguiente:

- Microsoft office 2010 en adelante
- Net Framework 4.0
- Windows 7 en adelante

En la máquina de servidor se recomienda tener instalado:

- Servicio OpenSSH para poder conectar la aplicación.

3. Navegación de interfaz

Se describen los formularios principales involucrados en este proyecto.

3.1. Login

Formulario de ingreso

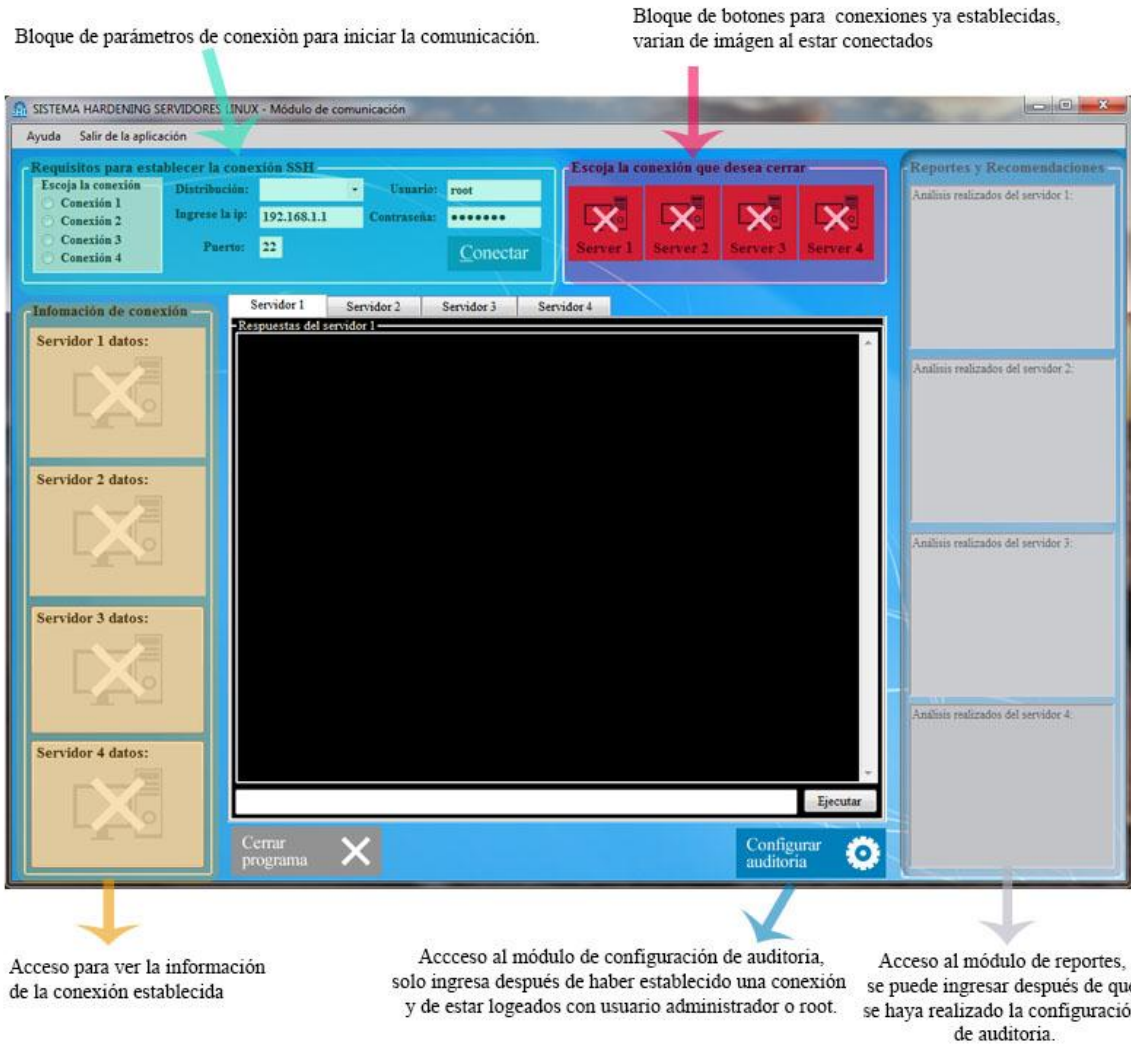
Elaborado por: José Benítez

Para ingresar al sistema se dispone de una ventana donde deberá ingresar la contraseña y solo tiene 3 intentos, pasados estos 3 intentos la aplicación se cierra, a continuación en la figura se puede visualizar el usuario y contraseña del sistema desarrollado.

3.2. Módulo de comunicación

La siguiente pantalla en aparecer es el formulario que pertenece al módulo de comunicación y es donde se accede al resto de módulos de la aplicación en la siguiente figura se pude visualizar sus bloques de funcionalidades.

Formulario del módulo de comunicación



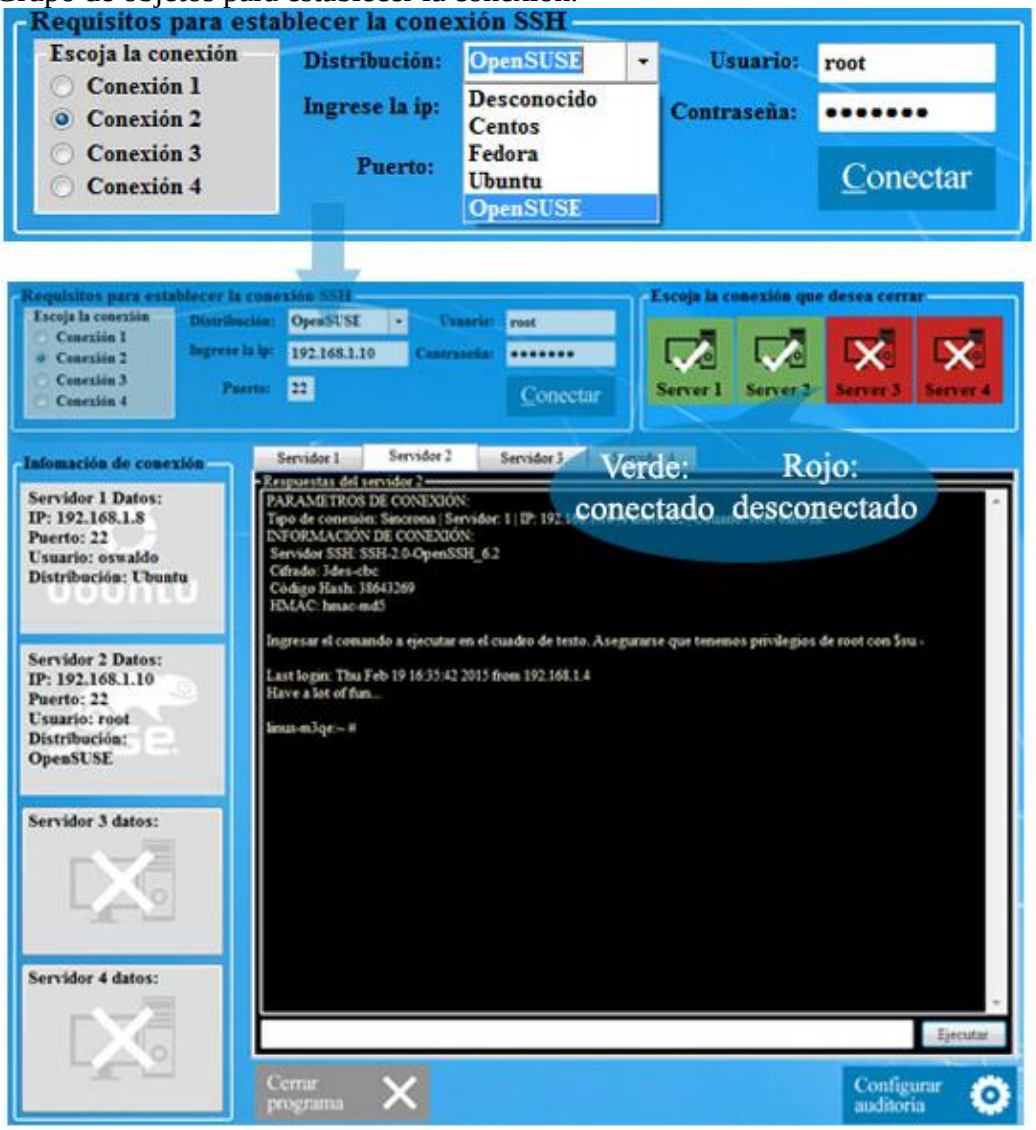
Elaborado por: José Benítez

3.3. Pasos para la obtención del informe general

3.3.1. Paso 1: Establecer la conexión al servidor

Primero se debe llenar los datos para poder iniciar la conexión en el bloque que dice “Requisitos para establecer la conexión SSH” en la figura se puede ver a continuación.

Grupo de objetos para establecer la conexión.

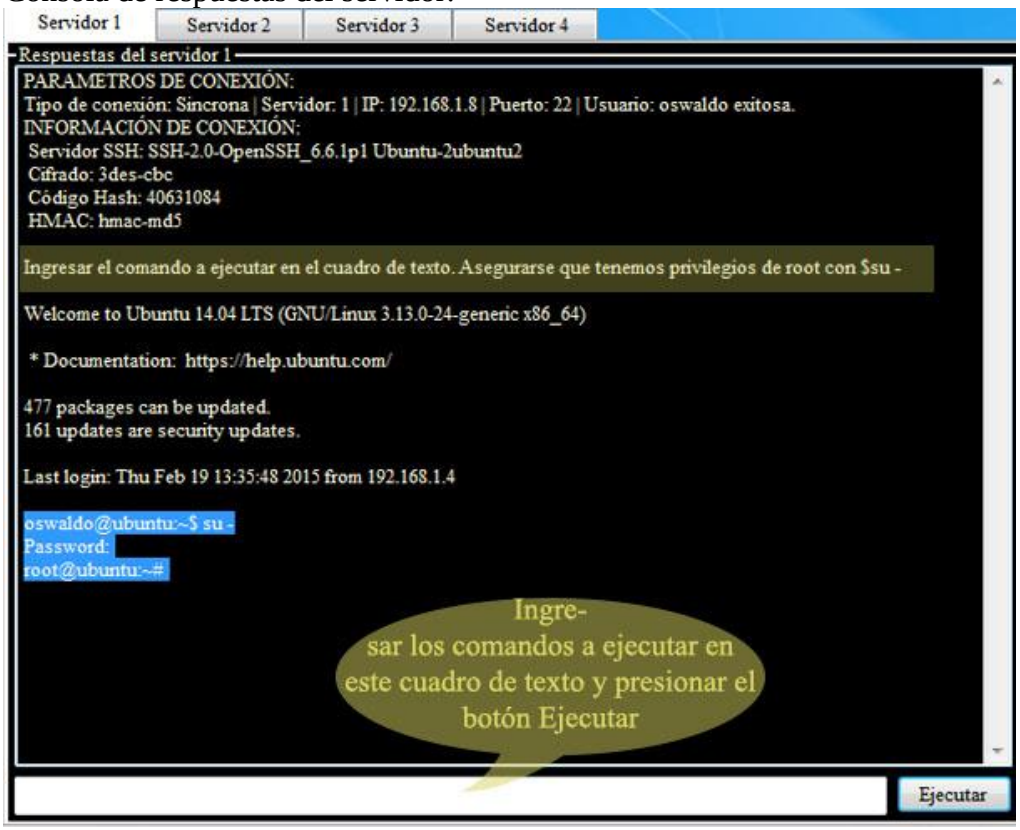


Elaborado por: José Benítez

3.3.2 Paso 2: Asegurarse de tener los privilegios de administrador

Después de haber establecido la conexión al servidor nos muestra un mensaje de recomendación que nos dice "Asegurarse que tenemos privilegios de root con *su* –"

Consola de respuestas del servidor.



Elaborado por: José Benítez

3.3.3 Paso 3: Configurar la auditoría seleccionando lo que se necesita

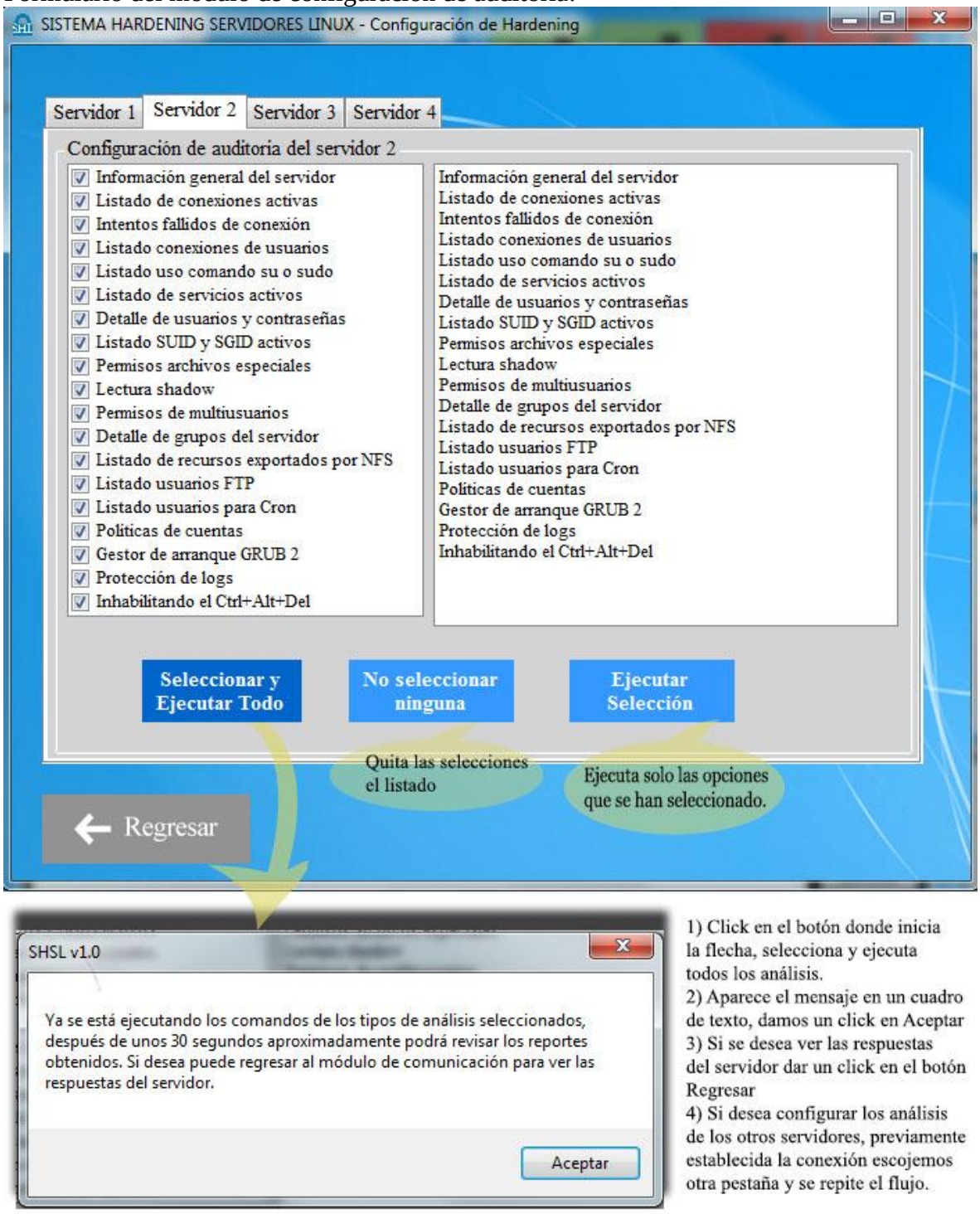
En este formulario tenemos 3 opciones:

- Seleccionar y Ejecutar Todo: Selecciona todos los tipos de análisis y envía los comandos a ejecutarse.
- No seleccionar ninguna: Quita la selección y limpia el listado
- Ejecutar Selección: Selecciona y ejecuta los que estén seleccionados.

En la siguiente figura se puede ver los eventos que siguen después de escoger la primera opción "Seleccionar y Ejecutar Todo"

Como se puede ver en la ventana tenemos pestañas al seleccionar, en cada una visualizaremos la misma ventana debido a que cada servidor tiene sus propios objetos y funciones que son similares.

Formulario del módulo de configuración de auditoría.



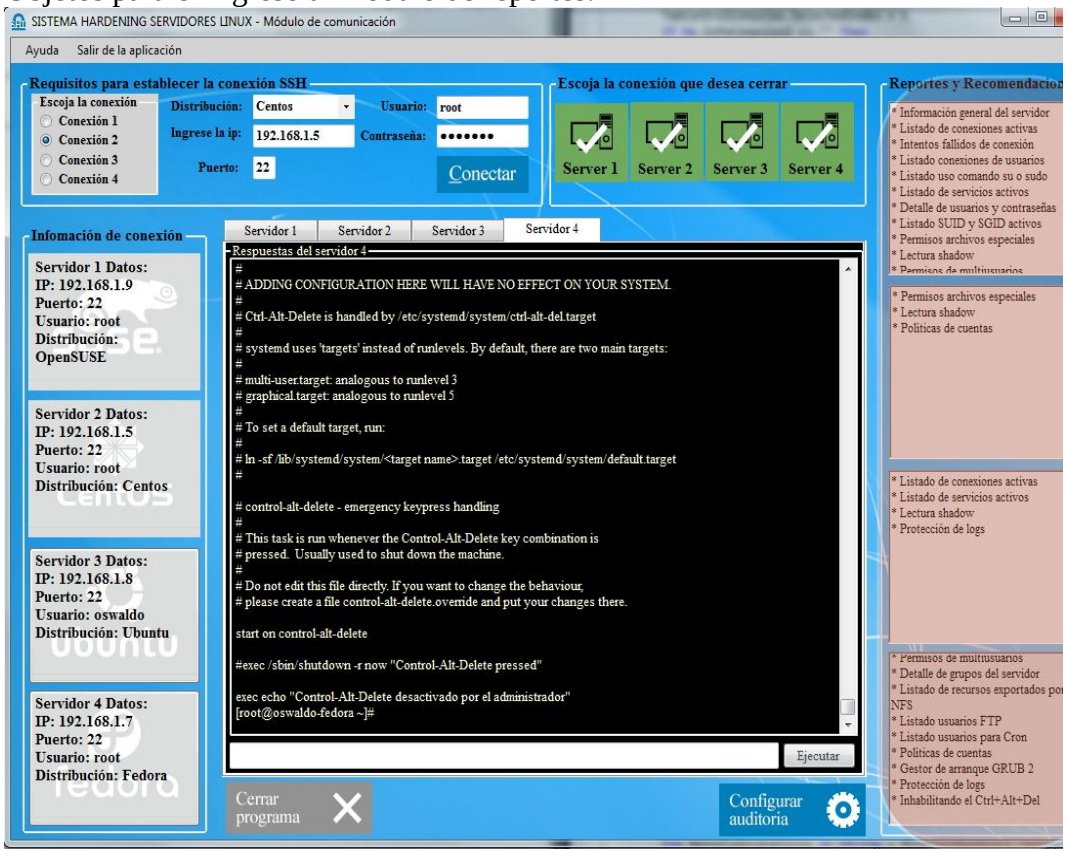
Elaborado por: José Benítez

3.3.4 Paso 4: Ingresar al módulo de reportes.

En el módulo de comunicación aparece el listado de los tipos de análisis ejecutados, y esto implica que se debe acceder después de haber establecido una conexión y obtener los permisos de root y posteriormente haber configurado la auditoría.

Se ingresa a este módulo después de que haya terminado la obtención de información, y se cambie de color el cuadro en la parte derecha.

Objetos para el ingreso al módulo de reportes.



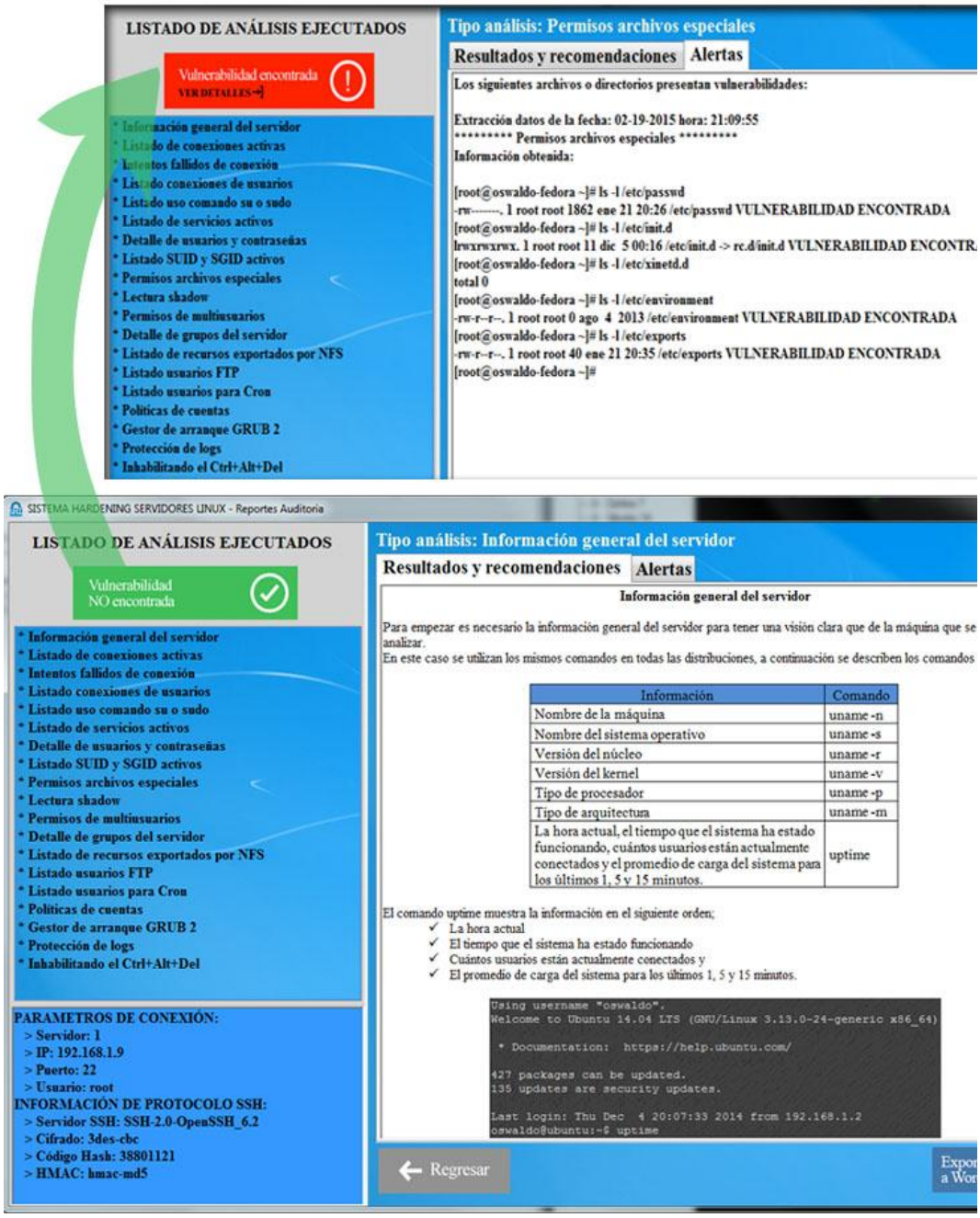
Elaborado por: José Benítez

3.3.5 Paso 5: Revisar los resultados y alarmas.

Abriendo el módulo de reportes se carga toda la información enviada por el servidor y que está impresa en el cuadro de texto con fondo negro para mediante algoritmos filtrar y analizar las respuestas y al final guardarlas en un vector que será leído cada vez que de un clic en la ventana, en la posición derecha se cargan en la pestaña "Resultados y recomendaciones" las recomendaciones conjuntamente con el resultado de cada análisis.

Cerca de la misma ubicación tenemos un botón de color verde que cada vez que detecte una vulnerabilidad se cambia la imagen del botón por una de fondo rojo, al dar clic en el botón mencionado nos muestra la pestaña "Alertas" para la visualización de los resultados, en la figura se puede ver lo explicado con anterioridad.

Formularios del módulo de reportes.



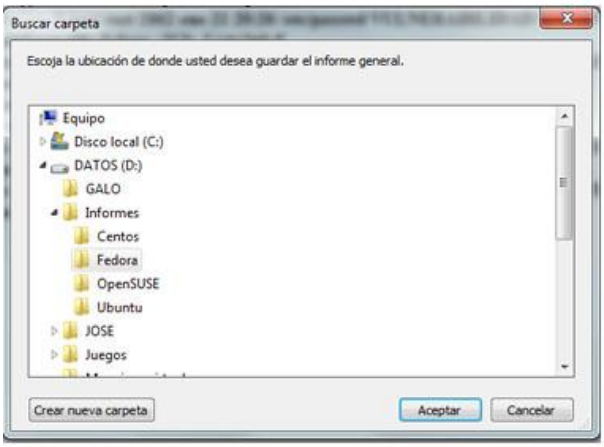
Elaborado por: José Benítez

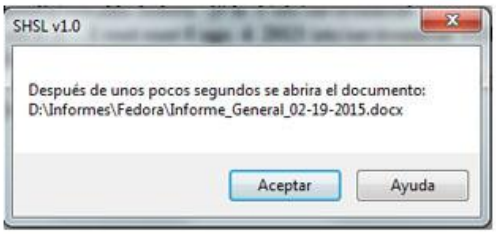
3.3.6 Paso 6: Exportar a un documento de Microsoft Word

Para exportar toda la información en un documento general que sea de fácil acceso y muy versátil en su manejo el botón “Exportar a Word” extrae un archivo .docx de formato que está incluido en las fuentes del software, lo copia y crea uno nuevo con el formato que tiene marcadores o posiciones marcadas, y añade las respuestas de los servidores después de las recomendaciones de cada análisis, en la siguiente figura se puede ver las actividades.

Pasos para exportar el informe de la auditoría.

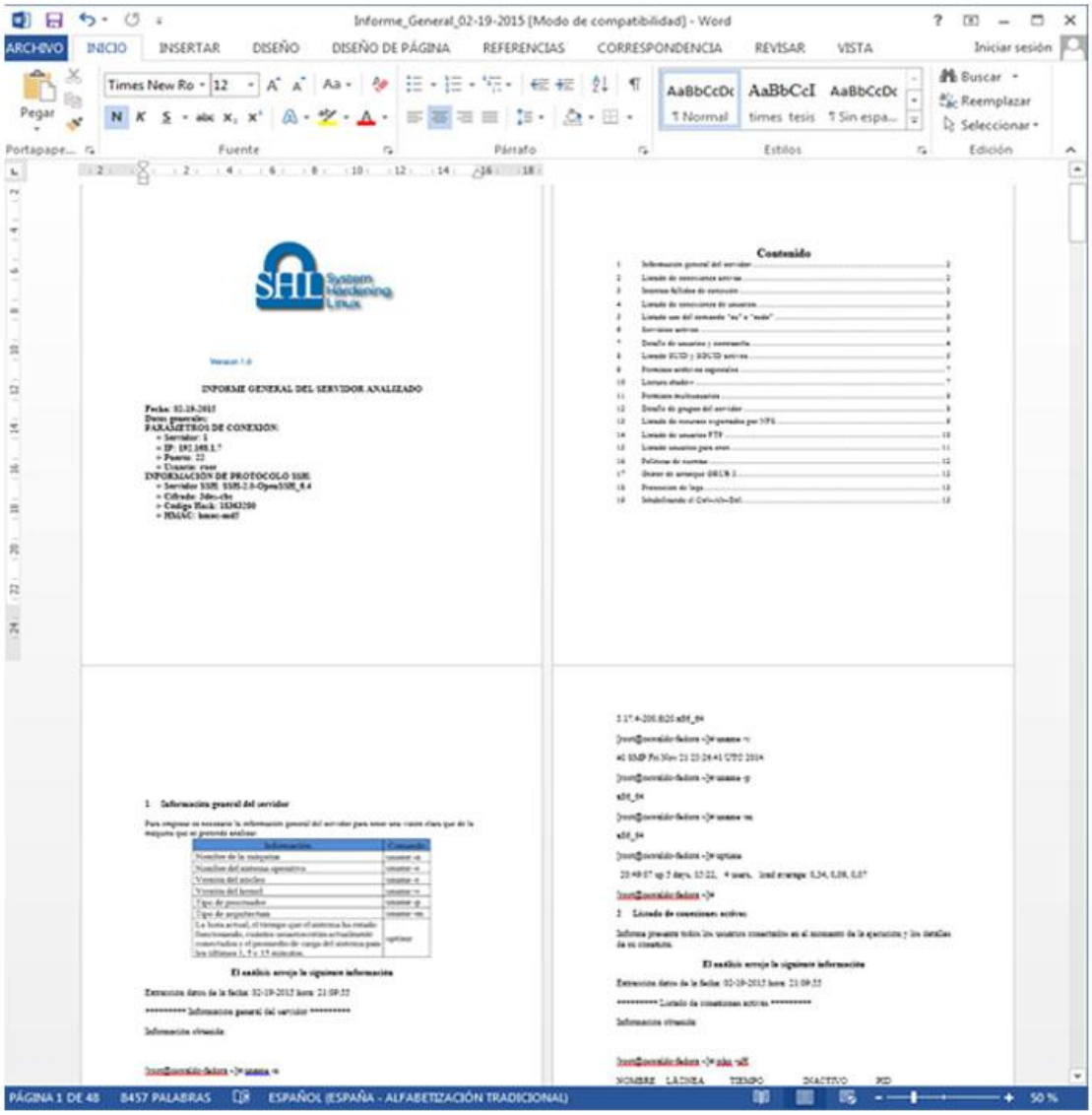
1) Damos un click en el botón .





3) Damos un click en Aceptar y esperamos a que se abra el informe general.

2) Seleccionamos el directorio donde deseamos guardar el informe general.



Elaborado por: José Benítez

3.3.7 Paso 7: Cerrar las conexiones abiertas.

En el grupo de objetos que se titula “Escoja la conexión que desea cerrar” estan 4 botones que cambian de color según el estado de la conexión de cada servidor.

Tiene dos estados estos botones;

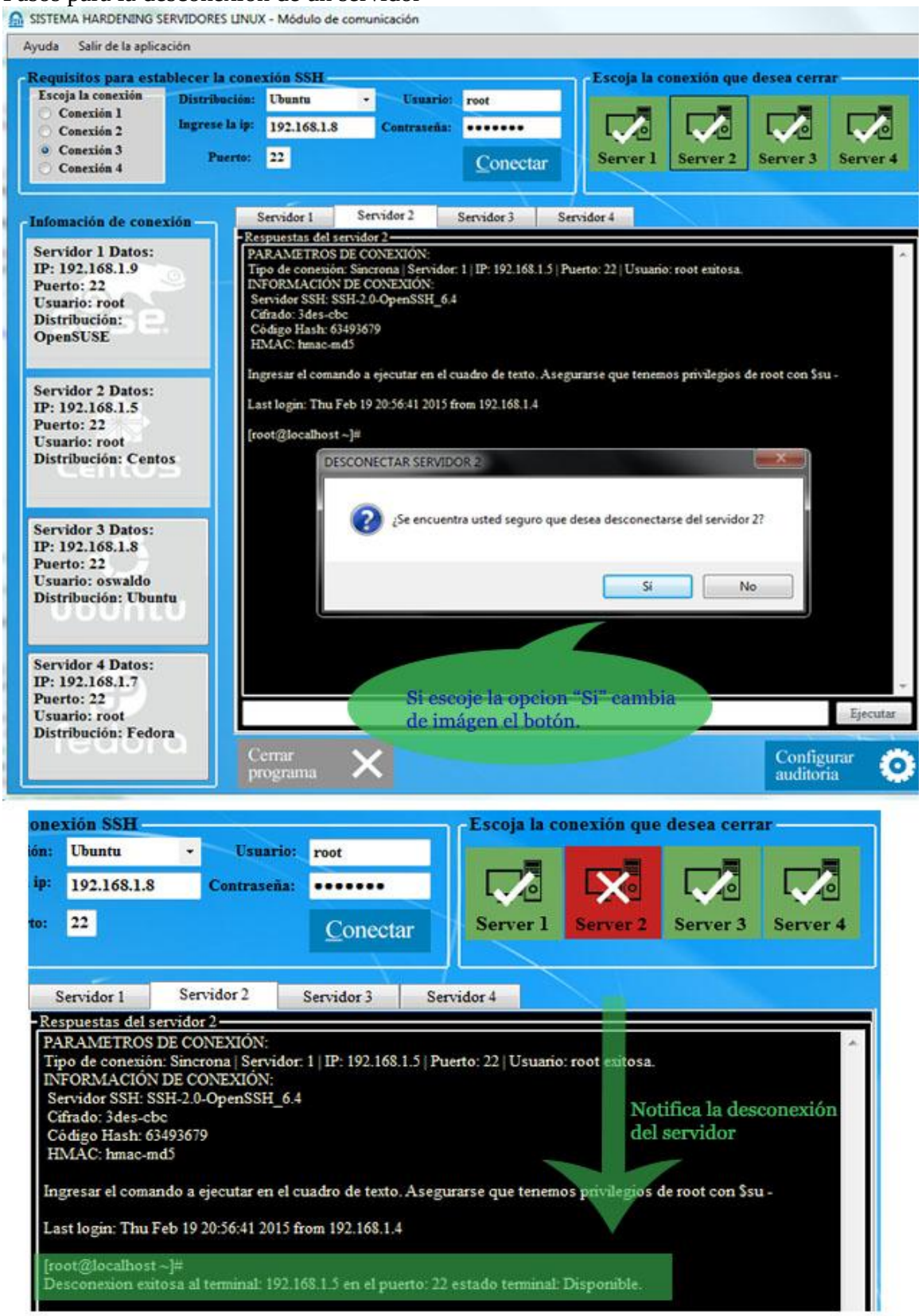
- Rojo = Desconectado
- Verde = Conectado

Si se encuentra conectado solicita la confirmación de la actividad que se pretende realizar con un cuadro con dos opciones SI NO.

Si se ha escogido la opción desconectar en el cuadro de texto nos aparece un mensaje de notificación de la desconexión exitosa.

En la siguiente figura podemos ver las acciones narradas con anterioridad.

Pasos para la desconexión de un servidor



Elaborado por: José Benítez