UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

BUENAS PRÁCTICAS DE SEGURIDAD ALINEADAS AL ISO/IEC 27002 PARA EL ASEGURAMIENTO DE EQUIPOS LINUX-DEBIAN PERTENECIENTES A UN CERT

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERÍA EN COMPUTACIÓN

P R E S E N T A N:

PERLA XÓCHITL DÍAZ FLORES

Y

ALEJANDRO REYES PLATA

DIRECTOR DE TESIS:
M.C. CINTIA QUEZADA REYES

MÉXICO D.F. 2015

Agradecimientos

Queremos comenzar por agradecer a nuestros respectivos padres Jova Flores Castro y Roberto Díaz Torres, Luz Claudia Plata Hernández y Esteban Reyes Galicia, a quienes les debemos estar aquí, fueron ellos, con su amor, esfuerzo, paciencia y apoyo incondicional, el combustible que suministró a nuestros cuerpos y almas las fuerzas necesarias para concluir esta importante etapa de nuestras vidas.

Asimismo, agradecemos con gran sentimiento a nuestra máxima casa de estudios la UNAM, a la Facultad de Ingeniería y a la DGTIC, y a todos nuestros maestros que en diversas etapas de nuestra vida, contribuyeron a forjar y moldear con sus enseñanzas a personas bien preparadas, listas para afrontar los grandes retos que depara esta vida.

Especialmente, agradecemos a nuestra asesora, M.C. Cintia Quezada Reyes, por su excelente apoyo, amistad, paciencia, dedicación y esfuerzo para que pudiéramos concluir esta tesis.

¡Goya! ¡Goya! ¡Cachún, cachún, ra, rá! ¡Cachún, cachún, ra, rá! ¡Goya!
¡UNIVERSIDAD!

Perla y Alejandro

Mamita, que estás en el cielo te agradezco todo el ánimo, confianza y apoyo que siempre me diste; Papito, gracias por ser siempre mi maestro en casa de 24 hrs. y mi inspiración para ser una profesionista; Hermanos, que con su ejemplo y constancia, me ayudaron a estar convencida de lo que tenía y quería hacer de mi vida profesional; y especialmente a ti Manuel, que dejabas a mi mano tus resistencias, capacitores y computadora para jugar.

Alex, mi amor y compañero de vida, siempre estuviste a mi lado en las buenas y en las malas, gracias por ser mi amigo y estar conmigo cumpliendo un reto más a tu lado, te amo cielo.

Perla Díaz

A ti, por tu inmenso amor, tu paciencia, comprensión, entusiasmo, por inspirarme a ser mejor cada día, por apoyarme a cumplir mis sueños, por estar conmigo en las buenas y en las malas, por ser mi amiga, mi confidente y mi amor, gracias por estar siempre a mi lado Perla.

A todas las personas que se cruzaron en vida y me apoyaron para que pudiera lograr mis objetivos, por motivarme y tenderme una mano cuando el camino se tornaba largo y pesado, a ustedes por siempre mi corazón y mi agradecimiento.

Alejandro Reyes

Índice

Índice de Figuras

Índice de Tablas

Introducción

En la actualidad la información es uno de los activos más importantes para cualquier organización, es por ello que en los últimos años se ha tenido un incremento de Equipos de Respuesta a Incidentes en Cómputo (CERT - Computer Emergency Response Teams), los cuales tienen como principal objetivo brindar servicios de seguridad de la información e implementar procesos de respuesta ante incidentes de seguridad en cómputo que pongan en riesgo la información crítica de su comunidad objetivo.

La información que un CERT maneja durante los procesos de respuesta a incidentes de seguridad, corresponde a información de carácter altamente confidencial y crítico, por tanto, la información debe ser manejada con base en estándares internacionales de seguridad de la información.

En este sentido, la información es un activo muy importante para este tipo de organizaciones, por lo que es necesario que los equipos de cómputo utilizados para el manejo de esta información estén correctamente asegurados y administrados por personal capacitado. La combinación de estos aspectos puede garantizar la tríada de la seguridad; la confidencialidad, integridad y disponibilidad de la información.

Los CERTs son comúnmente blanco de ataques cibernéticos constantes por parte de personas malintencionadas que obedecen a diversos motivos. Por tanto, no solo es importante proteger la información relacionada con los incidentes atendidos, sino también los procesos internos de los CERTs, de lo contrario se vería afectada la reputación e imagen de los equipos de respuesta a incidentes como los líderes o expertos en materia de seguridad.

Son numerosas las consecuencias de los diversos ataques a las organizaciones, por lo que un CERT debe tener especial cuidado en cada uno de sus procesos, ya que si bien forma parte del flujo de atención y respuesta a incidentes, también es objeto de ataques, de ahí la decisión de desarrollar este trabajo de tesis, que tiene como objetivo principal describir cómo realizar el aseguramiento de los equipos de cómputo asignados a estaciones de trabajo, con sistema operativo Linux-Debian, pertenecientes a CERTs, alineándolos al estándar de buenas prácticas de seguridad de la información (ISO/IEC 27002), ya que sirve como soporte y facilita la certificación de los procesos de manejo de información crítica en el estándar ISO/IEC 27001.

El hecho de que un CERT tenga una certificación ISO/IEC 27001 en el proceso de gestión de incidentes de seguridad de la información, le proporciona confianza y credibilidad ante su comunidad objetivo (Constituency), facilitando el cumplimiento de las metas propuestas por cada CERT sin importar su circunscripción.

Cabe destacar que una buena práctica se puede definir como una técnica, método, proceso o conjunto de acciones que se consideran efectivas para entregar un resultado en particular. Para el desarrollo de esta tesis se considerarán como las acciones que permitan reducir el riesgo de que la información resultante del proceso de manejo de incidentes de seguridad informática se vea afectada en cualquiera de los tres principios de seguridad; integridad, disponibilidad y confidencialidad.

De ahí radica la importancia de definir y justificar el conjunto de buenas prácticas que se considerarán para el aseguramiento de dichos equipos de cómputo, ya que éstas influirán de manera directa para salvaguardar la información contenida y procesada por las estaciones de trabajo.

Las organizaciones o equipos de respuesta a incidentes que pretendan certificar sus procesos con base en el estándar *ISO/IEC 27001*, podrían tomar como referencia la implementación de buenas prácticas de seguridad descritas en este trabajo, las cuales serán de gran utilidad para cumplir con algunos de los controles descritos en el estándar *ISO/IEC 27001:2013,* los cuales están relacionados al aseguramiento de equipos de cómputo asignados a estaciones de trabajo. Sin embargo, aunque las organizaciones no tengan contemplada la certificación de sus procesos críticos, la adopción de las buenas prácticas de seguridad propuestas en este trabajo, minimizará el riesgo de posibles ataques informáticos enfocados a comprometer la información de las organizaciones, todo ello sin la necesidad de invertir grandes cantidades de recursos económicos y humanos para salvaguardar dicha información.

Es importante mencionar que lo propuesto en esta tesis, no contempla la adquisición de software comercial o algún dispositivo físico (hardware) para proteger el entorno de información de las estaciones de trabajo. Aunque es importante aclarar que la adopción de buenas prácticas de seguridad representan un subconjunto de todas las acciones que se pueden emplear para proteger y garantizar la seguridad de la información.

De este forma, el capítulo 1 inicia con una descripción de conceptos básicos de seguridad; amenazas, vulnerabilidades, servicios de seguridad, tipos de ataques y niveles de seguridad informática.

Posteriormente el capítulo 2 explica el concepto de buenas prácticas, la importancia de los estándares, normas y políticas de seguridad, en particular se describe el ISO/IEC 27001:2013; sus beneficios, modelo y estructura, lo cual sirve de base para realizar un mapeo de los controles del ISO/IEC 27002 que pueden ser aplicables a equipos de cómputo.

Enseguida, el capítulo 3 trata del origen de los CERTs, sus funciones y servicios principales, el proceso general de gestión de incidentes de seguridad y los diversos modelos organizacionales para un CERT, lo cual brinda un panorama amplio de lo importante que es para un CERT, contar con una guía de buenas prácticas de seguridad implementada sobre sus estaciones de trabajo.

Después, en el capítulo 4 se exponen las características y ventajas de emplear sistemas operativos Linux-Debian en los procesos de gestión de incidentes de los CERTs, lo cual sirvió como preámbulo para definir las buenas prácticas de seguridad del ISO/IEC 27002 que serán implementadas en un equipo de cómputo con sistema operativo Linux-Debian, es decir, el mapeo de buenas prácticas de seguridad se realizó procurando que las ventajas de usar equipos Linux-Debian en el proceso de gestión incidentes no se vieran disminuidas y al mismo tiempo se brindara un nivel de

seguridad adecuado al entorno de trabajo, actividades y amenazas cibernéticas a las cuales están expuestas las estaciones de trabajo Linux-Debian al interior de un CERT. Lo anterior representó el mayor reto de este trabajo de tesis, debido a que ejemplifica perfectamente el eterno paradigma; a mayor seguridad, menor funcionalidad y viceversa.

Finalmente en el capítulo 5, las buenas prácticas de seguridad alineadas al ISO/IEC 27002 se aplicaron de forma exitosa sobre un equipo Linux-Debian, estas buenas prácticas fueron aplicadas considerando el proceso de antes y durante la instalación del sistema operativo, la eliminación de aplicaciones innecesarias, usuarios y permisos, limitación y control del usuario "root", control en las conexiones de red, mantenimiento del sistema y herramientas de seguridad para sistemas Linux-Debian.

Por último se presentan las conclusiones a las que se llegaron en el presente trabajo, basadas completamente en los resultados obtenidos.

Objetivos

La presente tesis tiene un objetivo general y cuatro objetivos específicos. El objetivo general es realizar una matriz de controles y recomendaciones de buenas prácticas para el aseguramiento de equipos de cómputo asignados a estaciones de trabajo, con el sistema operativo Linux con la distribución Debian, matriz que se fundamenta en las recomendaciones y controles de seguridad definidos en el ISO/IEC 27002, en el contexto de equipos de cómputo pertenecientes a un CERT (Computer Emergency Response Team - Equipo de Respuesta a Incidentes en Cómputo).

Para alcanzar el objetivo general, se persiguen cuatro objetivos específicos que son necesarios para la consecución del objetivo general de esta tesis.

El primer objetivo específico busca describir los conceptos básicos, los enfoques de seguridad informática, los diferentes tipos de ataques a los que se está expuesto, los servicios a considerar en la seguridad informática y los estándares existentes, en especial los que consideran la seguridad informática.

El segundo objetivo específico consiste en analizar y esquematizar el ISO/IEC 27002 a fin de identificar las buenas prácticas que sean factibles de implementar en una estación de trabajo con sistema operativo Linux-Debian para su correcto aseguramiento (hardening). Es decir en este objetivo define el ¿Qué? debe ser implementado.

El tercero objetivo específico, es realizar el análisis del contexto del caso de estudio, en primer instancia mencionar las organizaciones que existen para la investigación, implementación, desarrollo y gestión de la seguridad informática, a nivel nacional e internacional, así como describir las actividades principales que se desarrollan en un CERT, como la criticidad, sensibilidad e importancia de la información que se puede manejar en una organización de este tipo.

En el cuarto objetivo específico se establece el ¿Cómo?, es decir, las buenas prácticas de seguridad previamente identificadas en el segundo objetivo específico, en este objetivo, se implementan en una estación de trabajo con sistema operativo Linux-Debian y se describen los resultados obtenidos.

Una vez alcanzados los cuatro objetivos específicos se consigue el objetivo general, en es generar una matriz de controles con buenas prácticas de seguridad aplicadas a un sistema operativo Linux-Debian para equipos de cómputo asignados a estaciones de trabajo pertenecientes a un CERT, que a su vez se alinean ISO/IEC 27002:2013. Lo anterior con la finalidad de alcanzar un nivel de seguridad adecuado y óptimo en los equipos de cómputo, para así garantizar la confidencialidad, integridad y disponibilidad de la información en organizaciones que gestionan y atienden incidentes de seguridad en cómputo.

Justificación

Un CERT (*Computer Emergency Response Team* – Equipo de Respuesta a Incidentes en Cómputo) es un equipo de respuesta a incidentes de seguridad informática, el primero de ellos fue creado en 1988 por la DARPA (*Defense Advanced Research Projects Agency*) una agencia del departamento de defensa de los Estados Unidos de esa época. Con el tiempo, el número de equipos de respuesta a incidentes alrededor del mundo creció considerablemente, cada uno con propósitos propios y específicos pero sin desviarse de una línea base de seguridad informática.

La comunicación e interacción entre equipos de respuesta a incidentes experimentó diversas adversidades como el idioma, los medios de comunicación, los protocolos, las zonas horarias y la inexistencia de estándares internacionales. En el año 1989 a raíz de un incidente de impacto mundial, provocado por el gusano Wank, dejó ver la necesidad de una organización que coordinara la interacción entre los diversos CERTs, por lo que fue creado en el año 1990 el FIRST (*Forum for Incident Response and Security Teams*).

El FIRST es un foro, que como sus siglas lo describen, tiene presencia a nivel mundial. El FIRST permite a sus integrantes responder más rápida y efectivamente a los incidentes de seguridad en el ámbito informático, debido a la implementación de una serie de buenas prácticas, herramientas y relaciones más estrechas y confiables entre sus distintos integrantes alrededor del mundo. Lo que propicia el intercambio de ideas y soluciones a problemas comunes, que permiten generar contribuciones a soluciones globales.

En la página oficial del FIRST, http://www.first.org/members/map/, se indican todos los CERTs miembros a nivel mundial. Actualmente existen en México 4 equipos de respuesta a Incidentes acreditados ante el FIRST, el UNAM-CERT ubicado en la DGTIC (Dirección General de Cómputo y Tecnologías de la Información y Comunicaciones) en la UNAM (Universidad Nacional Autónoma de México), el Mnemo-CERT, Scitum-CSIRT y el CERT-MX ubicado en las instalaciones de la Policía Federal de Comisión Nacional de Seguridad de México.

El UNAM-CERT fue fundado en el año 2000, con ello se tiene el primer equipo de respuesta a incidentes en México con reconocimiento mundial. El cual atiende incidentes de seguridad, dentro de la competencia de la UNAM, difunde boletines, noticias, información actualizada sobre vulnerabilidades y promueve una cultura de seguridad a través de programas de divulgación y capacitación en seguridad de la información. Organiza periódicamente en México eventos como la RENASEC (Reunión Nacional de Seguridad en Cómputo), el Admin-UNAM, la reunión internacional del proyecto Honeynet, el DISC (Día Internacional de la Seguridad en Cómputo) y el Congreso de Seguridad en Cómputo, la mayoría realizados una vez por año.

El CERT-MX perteneciente a la Policía Federal recibió la acreditación oficial de FIRST en Julio del 2011, acreditación que lo reconoce como un equipo de respuesta a incidentes de seguridad nacional, lo que lo convierte en el primer CERT gubernamental del país, su misión es proporcionar soporte en la respuesta y defensa

ante los incidentes de seguridad de la información ocurridos en el dominio .MX y en las infraestructuras de tecnología de información y comunicación críticas de México.

La información que el CERT-MX maneja durante la respuesta a incidentes de seguridad corresponde a información de carácter altamente confidencial y crítico, por tanto, la información debe ser manejada con base en estándares internacionales de seguridad de la información.

En este sentido, la información es un activo muy importante para este tipo de organizaciones, por lo que es necesario que los equipos de cómputo utilizados para el manejo de esta información estén correctamente asegurados y administrados por personal capacitado. La combinación de estos aspectos puede garantizar la tríada de la seguridad; la confidencialidad, integridad y disponibilidad de la información.

Los CERTs son comúnmente blanco de ataques constantes por parte de personas malintencionadas que obedecen a diversos motivos. Por tanto, no solo es importante proteger la información relacionada con los incidentes atendidos, sino también los procesos internos de los CERTs, de lo contrario se vería afectada la reputación e imagen de los equipos de respuesta a incidentes como los líderes o expertos en materia de seguridad.

Son numerosas las consecuencias de los diversos ataques a las organizaciones, por lo que un CERT debe tener especial cuidado en cada uno de sus procesos, ya que si bien forma parte del flujo de atención y respuesta a incidentes, también es objeto de ataques, de ahí la decisión de desarrollar este trabajo de tesis que sirva de referencia a las personas responsables de asegurar y administrar adecuadamente la seguridad de los equipos de cómputo pertenecientes a CERTs.

Una vez definido el contexto del desarrollo de esta tesis, se procederá a limitar el tema de estudio al aseguramiento de equipos de cómputo, en especial a estaciones de trabajo, es decir, no se contemplarán equipos de cómputo de tipo servidor. Lo anterior debido a que los equipos de cómputo asignados a las estaciones de trabajo son la primera herramienta tecnológica con la que se da atención a los incidentes de seguridad en cómputo. Otro de los puntos a limitar en el desarrollo de esta tesis es el tipo de sistema operativo que se asegurará, en este caso es el sistema operativo Linux-Debian.

Finalmente, el aseguramiento de los equipos de cómputo asignados a estaciones de trabajo, con sistema operativo Linux-Debian, pertenecientes a CERTs, estarán alineados al estándar de buenas prácticas de seguridad de la información (ISO/IEC 27002), ya que sirve como soporte y facilita la certificación de los procesos de manejo de información crítica en el estándar ISO/IEC 27001.

El hecho de que un CERTs tenga una certificación ISO/IEC 27001 en el proceso de gestión de incidentes de seguridad de la información le proporciona confianza y credibilidad ante su comunidad objetivo (Constituency), facilitando el cumplimiento de las metas propuestas por cada CERT sin importar su circunscripción.

Cabe destacar, que una buena práctica se puede definir como una técnica, método, proceso o conjunto de acciones que se consideran efectivas para entregar un resultado en particular. Para el desarrollo de esta tesis se considerarán como las acciones que permitan reducir el riesgo de que la información resultante del proceso de manejo de incidentes de seguridad informática, se vea afectada en cualquiera de los tres principios de seguridad; integridad, disponibilidad y confidencialidad.

De ahí radica la importancia de definir y justificar el conjunto de buenas prácticas que se considerarán para el aseguramiento de dichos equipos de cómputo, ya que éstas influirán de manera directa para salvaguardar la información contenida y procesada por las estaciones de trabajo.

Las organizaciones o equipos de respuesta a incidentes que pretendan certificar sus procesos con base en el estándar *ISO/IEC 27001*, podrían tomar como referencia la implementación de buenas prácticas de seguridad descritas en este trabajo, las cuales serán de gran utilidad para cumplir con algunos de los 114 controles descritos en el estándar *ISO/IEC 27002:2013,* los cuales están relacionados al aseguramiento de equipos de cómputo asignados a estaciones de trabajo. Sin embargo, aunque las organizaciones no tengan contemplada la certificación de sus procesos críticos, la adopción de las buenas prácticas de seguridad propuestas en este trabajo, minimizará el riesgo de posibles ataques informáticos enfocados a comprometer la información de las organizaciones, todo ello sin la necesidad de invertir grandes cantidades de recursos económicos y humanos para salvaguardar dicha información.

Es importante mencionar que lo propuesto en esta tesis, no contempla la adquisición de software comercial o algún dispositivo físico (hardware) para proteger el entorno de información de las estaciones de trabajo. Aunque es importante aclarar que la adopción de buenas prácticas de seguridad representan un subconjunto de todas las acciones que se pueden emplear para proteger y garantizar la seguridad de la información.

Metodología

La información que se considera para la realización de esta tesis, es aquella que permite en primera instancia comprender y conocer:

- Los conceptos generales de seguridad en cómputo.
- Los estándares relacionados con la seguridad en cómputo.
- Los objetivos y funciones principales de un CERT.
- Las buenas prácticas y controles propuestos en el ISO/IEC 27002

Lo anterior para generar una matriz en relación a recomendaciones y controles, con las configuraciones seguras del sistema operativo Linux-Debian, de equipos de cómputo asignados a estaciones de trabajo, pertenecientes a un CERT. La metodología para el desarrollo de este trabajo se divide en tres secciones; definición, análisis e implementación:

Metodología (Definición, Análisis e Implementación)

I. Definición del objetivo general.

Antes de definir una matriz de controles y recomendaciones de configuraciones seguras, partiendo del análisis de las buenas prácticas de seguridad descritas en el ISO/IEC 27002:2013, es necesario plantear los siguientes cuestionamientos que permiten definir el objetivo general de esta tesis:

- ¿Qué se quiere proteger?
- ¿Por qué se quiere proteger?
- ¿De qué o quién se quiere proteger?
- ¿Cómo se va a proteger?

Con relación a la pregunta ¿Qué se quiere proteger?, el alcance es para equipos de cómputo, en especial a estaciones de trabajo pertenecientes a CERTs.

Es importante mencionar el ¿Por qué se quiere proteger? a equipos de cómputo pertenecientes a un CERT, esto es debido a que las estaciones de trabajo son utilizadas para brindar servicios de respuesta a incidentes tienen alto riesgo de ser

comprometidas por un intruso o bien manejan información altamente confidencial que sólo debería ser accedida por personal autorizado.

En consecuencia, la respuesta a la pregunta ¿De qué o de quién se quieren proteger los equipos de cómputo pertenecientes a un CERT?, es en primera instancia de todo aquel software malicioso que pretenda comprometer la seguridad de dichas estaciones de trabajo y en segunda instancia de cualquier intruso con intenciones malintencionadas sobre la infraestructura de TI (Tecnologías de la información) de algún CERT.

Por último, ¿Cómo se va a proteger el equipo de cómputo?, se hace referencia a un conjunto de recomendaciones y buenas prácticas de seguridad de la información contempladas en el ISO/IEC 27002:2013.

II. Describir el marco de referencia.

Cabe destacar que se describe el marco de referencia en el que se sustentan las buenas prácticas de configuración segura para un equipo de cómputo, que en este caso esta basado en las recomendaciones y controles propuestos en el ISO/IEC 27002:2013.

III. Identificación del contexto y alcance.

Una vez resueltas las cuestiones anteriores se procede con la definición e identificación del contexto en el que se desarrollará esta tesis, que será equipos de cómputo asignados a estaciones de trabajo de un CERT.

IV. Descripción y viabilidad del sistema a asegurar.

Se menciona la definición de Hardening, se describe y revisa la seguridad gestionada por Linux-Debian y el por qué de utilizar Linux-Debian, para lo cual fue necesario investigar las características y potencialidades de dicho sistema con el propósito de determinar la efectividad del aseguramiento.

V. Desarrollo de una matriz en la que se relaciona el marco de referencia, amenazas y vulnerabilidades.

El planteamiento y análisis de los puntos anteriores permitie el desarrollo de una matriz de implementación de controles para estaciones de trabajo de equipos de cómputo pertenecientes a un CERT con la distribución del sistema operativo Linux-Debian.

VI. Descripción e implementación de recomendaciones de controles y configuraciones seguras.

En este punto se describe la implementación del control y la configuración considerada en la matriz de buenas prácticas alineadas al ISO 27002, para el hardening de un

equipo de cómputo, asignado a una estación de trabajo de un CERT, con el sistema operativo Debian.

Capítulo 1

Conceptos Básicos

1.1 Definiciones

El concepto de seguridad proviene del latín “securitas”, se refiere al conjunto de medidas y acciones enfocadas a la protección de un ente contra determinados riesgos a los que puede estar en peligro.

Por otro lado, el dato es considerado como un conjunto de símbolos que representan hechos, situaciones o valores, mientras que la información se refiere al conjunto de datos organizados de forma congruente y útil. Es decir, los datos son la materia prima que al ser procesada se convierte en información, la cual debe ser: precisa, oportuna, íntegra y significativa (ver Figura 1.1).

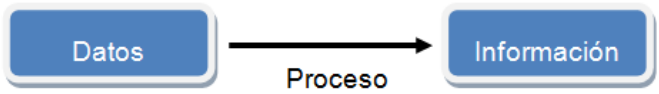


Figura 1.1 - Tratamiento de los datos a información

El término de informática proviene del acrónimo **infor**mación y auto**mática**. Se define como el conjunto de conocimientos científicos y técnicos, que se ocupan del tratamiento de la información por medios automáticos, principalmente mediante dispositivos electrónicos de procesamiento y almacenamiento de datos.

La seguridad de la información es la disciplina que habla de los riesgos, las amenazas, las buenas prácticas y esquemas normativos que exigen niveles de aseguramiento de procesos y tecnologías para elevar el nivel de confianza en la creación, uso, almacenamiento, transmisión, recuperación y disposición final de la información[1].

De acuerdo con las definiciones anteriores, se determina el término de seguridad informática, como el nombre dado a la colección de herramientas diseñadas para la protección de los sistemas de cómputo a fin de evitar amenazas de confidencialidad, integridad o disponibilidad [2] (ver Figura 1.2).

Figura 1.2- Diagrama de la Seguridad de la Información

[1] La Gerencia de la Seguridad de la Información, disponible en http://www.isaca.org/Journal/Past-Issues/2011/Volume-5/Pages/JOnline-La-Gerencia-de-la-Seguridad-de-la-Informacion-Evolucion-y-Retos-Emergentes.aspx (2011), consultado 04-02-2013.

[2] María J. López Barrientos, Cintia Quezada Reyes. Fudamentos de seguridad informática. p 23.

1.1.1 Principios de seguridad de la información

La seguridad de la información busca proteger la confidencialidad de la información contra accesos no autorizados, evitar alteraciones indebidas que pongan en peligro la integridad de la información y garantizar la disponibilidad de la misma. Por tal razón, la seguridad de la información se basa en los siguientes principios: confidencialidad, integridad y disponibilidad.

a) Confidencialidad

Es una cualidad de la información en la que se garantiza que es accesible únicamente por el personal autorizado a dicha información.

En consecuencia, la información catalogada como privada o confidencial deberá ser accesible sólo por las personas autorizadas. El objetivo de la confidencialidad es prevenir la divulgación no autorizada de la información, manteniéndola oculta o en secreto.

b) Integridad

La información tiene integridad cuando es oportuna, precisa, completa y coherente. Sin embargo, las computadoras son incapaces de proporcionar o proteger todas éstas cualidades de la información. Por tanto, en el campo de la seguridad de la información la integridad se define con base en dos aspectos: integridad de los datos y la integridad del sistema. [3]

La integridad de datos es un requisito para que la información y los programas sólo puedan ser alterados por una entidad específica y autorizada para ello.

La integridad del sistema es el requisito de que un sistema se comporte con base en lo esperado, libre de manipulaciones no autorizadas, deliberadas o imprevistas en el sistema.

c) Disponibilidad

Es la cualidad que garantiza que la información sea proporcionada por los sistemas de forma rápida y que el servicio esté siempre accesible por los usuarios autorizados para acceder al recurso. [4]

Los tres aspectos anteriores componen la tríada de la seguridad de la información, lo que significa que un sistema es seguro o fiable si garantiza la confidencialidad, integridad y disponibilidad (ver Figura 1.3).

[3] NIST. An Introduction to Computer Security: The NIST Handbook, Special Publication 800-12, p. 5-6.

[4] Ibídem, p. 7.

Figura 1.3 - Tríada de la seguridad de la información.

1.1.2 Enfoques de seguridad

En la seguridad de la información se consideran los enfoques de seguridad física y lógica de los sistemas de información, cabe destacar que la seguridad lógica es uno de los aspectos más vulnerados, la mayoría de los ataques son dirigidos especialmente hacia los principios de la seguridad informática (confidencialidad, integridad y disponibilidad), los cuales tienen como propósito proteger la información almacenada y procesada por los sistemas de información. Por lo general, la información es uno de los activos más importantes para cualquier organización, por lo tanto, deben existir estrategias y técnicas que la protejan, considerando aspectos de seguridad física y seguridad lógica.

La seguridad física es uno de los aspectos menos contemplados en el diseño de un sistema de información. Para un atacante puede ser más fácil acceder y copiar información confidencial directamente del sistema que la contiene, a través de un dispositivo *USB* (*Universal Serial Bus* – Bus Universal en Serie), que intenta acceder vía lógica al sistema para substraer dicha información.

La seguridad física consiste en la aplicación de barreras físicas, procedimientos y mecanismos de control de acceso, para reducir el riesgo de que las amenazas se materialicen sobre los recursos e información.

Es importante resaltar que cada sistema es único y por tanto la política de seguridad a implementar no debe ser única. Este enfoque de seguridad física, debe contemplar amenazas ocasionadas por el hombre y por la propia naturaleza del medio físico en el que se encuentran ubicados los sistemas a proteger.

Las principales amenazas que se prevén en la seguridad física son:

- Desastres naturales: incendios, tormentas, terremotos, inundaciones entre otros.
- Amenazas ocasionadas por el hombre de forma no intencional: deficiente instalación eléctrica, condiciones inadecuadas de funcionamiento, etcétera.
- Disturbios, robo, fraude, sabotajes internos y externos.

Para evitar que las amenazas anteriores se materialicen sobre la información, se recomienda implementar controles de acceso físico que contemplen:

- Uso de guardias de seguridad.
- Detectores de metales y o dispositivos de almacenamiento (escaner corporal).

- Sistemas biométricos.
- Uso de animales (caninos).
- Protecciones físicas (rejas, candados, chapas, etcétera). [5]

La seguridad lógica consiste en la aplicación de barreras, procedimientos y mecanismos de control de acceso a la información, dentro de los sistemas y aplicaciones que sólo permitan a las personas autorizadas el acceso y manipulación de dicho activo.

La seguridad lógica debe garantizar como mínimo los siguientes aspectos para el aseguramiento de la información:

- Restringir el acceso a los archivos y software.
- Asegurar que los custodios de la información tengan los permisos necesarios para desempeñar sus funciones.
- Tener supervisión y registro de la interacción de los usuarios con los sistemas.
- Que la información transmitida sea enviada a través del medio adecuado para ello.
- Asegurar que la información y programas, sean manipulados por los usuarios correctos con los permisos adecuados (confidencialidad).
- Asegurar que el destinatario al cual se le haya enviado la información, sea quien la reciba y no alguien más.
- Que la información recibida sea la misma que ha sido transmitida (integridad).
- La existencia de sistemas secundarios (emergencia) para la transmisión de datos hacia y desde diferentes puntos (disponibilidad).

El primer paso para proteger la información de cualquier sistema es identificar los aspectos físicos y lógicos que se quieren proteger, para posteriormente reforzarlos con la implementación de mecanismos o controles. Lo anterior no garantiza que la información no sufrirá algún tipo de ataque sobre su integridad, disponibilidad o confidencialidad, sino que el riesgo de que ello suceda, disminuya considerablemente.

Al respecto *Gene Spafford* expresa que: “Un sistema se vuelve inseguro simplemente con el mero hecho de encenderlo. El único sistema totalmente seguro sería uno que estuviese apagado, desconectado de cualquier red, metido dentro de una caja fuerte de titanio, rodeado de gas y vigilado por unos guardias armados insobornables. Aun así yo no apostaría mi vida por él” [6].

1.1.3 Riesgo

El riesgo es la probabilidad de que algo desafortunado ocurra sobre un activo o recurso. Para contextualizar un riesgo en un entorno en particular, se debe tener una amenaza y una vulnerabilidad, por ejemplo, si se tiene una estructura hecha de madera y se prende fuego a unos cuantos centímetros de la misma, se puede identificar una amenaza (el fuego) y una vulnerabilidad (estructura de madera), la

5 Jason Andress. The Basics of Information Security, Understanding the Fundamentals of InfoSec in Theory and Practice, p11.

6 Implementación de redes privadas virtuales (VPN) utilizando el protocolo IPSEC (2002), disponible en http://www.redes-linux.com/manuales/vpn/trabajo.pdf, consultado 04-02-2013.

probabilidad de que la estructura de madera sea afectada por el fuego se le denomina riesgo. [7]

Como resumen de lo visto hasta este punto, se muestra la Figura 1.4 que ejemplifica la relación entre diversos aspectos que pueden interactuar entre sí, para dar forma al concepto de seguridad de la información con base en los elementos del riesgo.

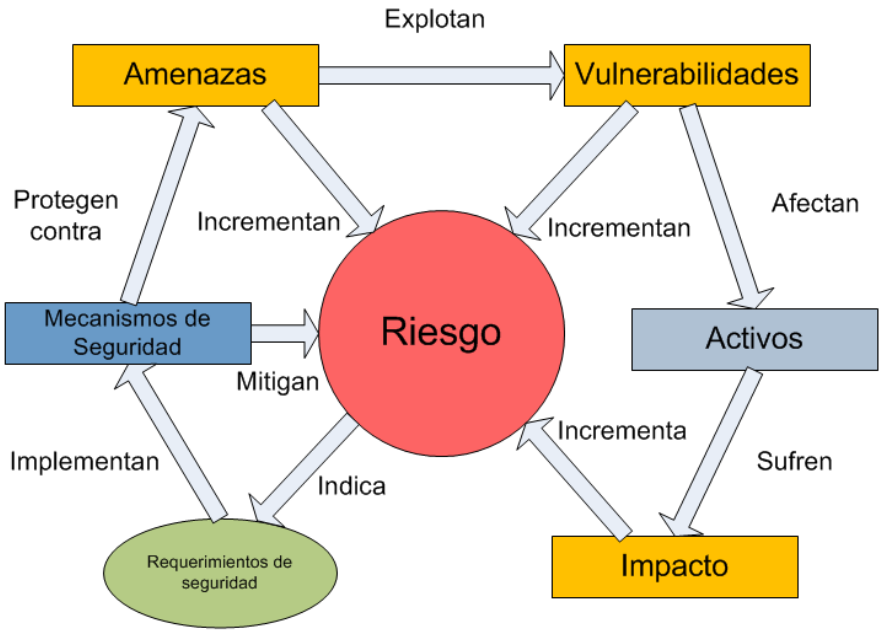


Figura 1.4 - Elementos del riesgo.

1.2 Amenazas

Las amenazas son todo aquello que tiene el potencial de hacer daño [8]. Dicho de otra forma una amenaza es una condición del entorno del sistema de información, que dada la oportunidad, podría ocurrir una violación a la seguridad. No se omite mencionar, que existen amenazas intencionales, las cuales buscan deliberadamente causar un daño y las no intencionales, las cuales son producto del entorno, acciones u omisiones no malintencionadas. En ambos casos, la presencia de la amenaza representa un riesgo potencial de que ésta se materialice, sobre algún activo crítico de la organización.

1.2.1 Tipos de amenazas

En términos informáticos, la materialización de una amenaza no es más que la ejecución de un ataque, el cual consiste en aprovechar las vulnerabilidades de los sistemas informáticos para comprometer la confidencialidad, integridad y disponibilidad de la información. Estos ataques se dividen en dos, los pasivos y los activos, y a su vez se clasifican en cuatro categorías: intercepción, interrupción, modificación, y fabricación. Cada categoría puede afectar uno o más principios de la tríada de la información como se muestra en la Figura 1.5.

7 Jason Andress. Op. Cit., p 10.

8 Ibídem, p 10.

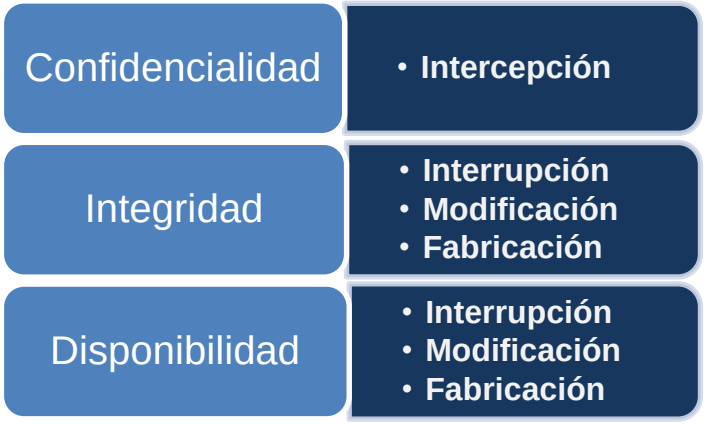


Figura 1.5 - Tipos de ataques.

Los tipos de amenazas son:

1. Amenazas de software

Estas amenazas refieren a fallas asociadas al software instalado por defecto en los sistemas operativos, software desarrollado, sofware mal implementado y al software malicioso diseñado para afectar directamente al sistema. Las cuales según sus características podrían ser agrupadas en alguna de las cuatro categorías mencionadas en la Figura 1.5.

A continuación se describen de forma general las amenazas más comunes que afectan a los sistemas de información y comunicación.

a) **Sofware desarrollado:** Es el software creado por el propio usuario del sistema, el cual puede tener errores o huecos de seguridad que pasaron desapercibidos por el programador durante la etapa de desarrollo, un intruso podría aprovechar de estas deficiencias para comprometer la seguridad de los sistemas donde sea ejecutado dicho software.
b) **Software de aplicación**: Este software no fue desarrollado con fines maliciosos, pero por sus características y funcionalidades puede ser utilizado por un usuario para realizar ataques a los sistemas de información.
c) **Cookies:** Las cookies no son un archivo malicioso como tal, son archivos de texto que los navegadores de Internet utilizan para facilitar la conexión de un equipo informático con un sitio Web, sin embargo, representan una amenaza para la privacidad de la información compartida por los usuarios en Internet.
d) **Phishing:** Consiste en la suplantación de identidad de sitios Web, con la intención de engañar y estafar a usuarios legítimos para que proporcionen datos confidenciales como: usuarios, contraseñas, números de tarjetas de crédito, etcétera. Estas amenazas no sólo afectan a la víctima, sino la imagen y reputación de la organización suplantada.

e) Spam: Se refiere a la acción de envío de correo electrónico a quien no lo solicita y de forma masiva. Estos correos representan uno de los principales focos de infección en la red (ver Figura 1.6). [9]

Figura 1.6 - Spam.

f) Scams: Son los correos electrónicos que pretenden obtener datos confidenciales de los usuarios con la intención de estafarlos económicamente u obtener un beneficio no legal de ello.

g) Malware: Proviene de la contracción de las palabras en inglés ***Mal****icious Soft****ware,*** es todo aquel software diseñado y programado para afectar uno o más de los aspectos de la seguridad de la información. Las siguientes descripciones son referentes a las amenazas más comunes englobadas dentro de la categoría de *Malware:*

- **Virus:** Son programas informáticos diseñados para infectar archivos y equipos de cómputo, no requiere de la colaboración del usuario para su propagación, debido a que es código que se replica uniéndose a otro objeto generalmente sin consentimiento ni conocimiento del usuario, algunos toman el control de los programas de correo electrónico, otros adquieren control de los recursos del sistema infectado. Incluso algunos virus pueden destruir o corromper archivos de datos, borrar programas o dañar el propio sistema operativo.

- **Gusanos:** Los gusanos son programas parecidos a los virus, sólo que éstos realizan copias de sí mismos aprovechándose de manera excesiva de los recursos del sistema comprometido, su objetivo es replicarse a través de las redes de datos y así causar el mayor impacto en los equipos de cómputo y redes de datos.

- **Caballos de Troya:** Conocidos como “troyanos”, es un tipo de malware que se hace pasar por un programa inofensivo para vulnerar la seguridad de los sistemas, posteriormente se activa de manera discreta y sin consentimiento del usuario cumpliendo así, su propósito nocivo para el que fue creado.

- **Adware:** Son programas con fines de publicidad, muestran anuncios no deseados en los equipos y sitios Web; muchos de estos anuncios redirigen

[9] Spam (2011), disponible en http://milcris.multiply.com/journal/item/218, consultado 06-04-2012.

a los usuarios a sitios maliciosos para robar sus datos confidenciales o infectar sus equipos.

- **Spyware**: Son programas espías que recopilan información de la actividad en equipos de cómputo y sitios Web sin el consentimiento ni conocimiento del usuario referente. Esta información es enviada de forma remota a un tercero.

- **Rootkits**: Son programas especialmente diseñados para ocultar información de procesos, archivos, conexiones red, etcétera. Comúnmente son instalados por los intrusos en sistemas de información donde quieren que su actividad maliciosa pase desapercibida.

- **Exploits:** Son programas diseñados para crear accesos (abrir puertas) en los sistemas, se aprovechan de huecos de seguridad existentes en los sistemas informáticos asociados a las aplicaciones instaladas.

- **Botnets**: Son un conjunto de equipos infectados por un código malicioso (*bots*) que son controlados por un servidor C&C (*Command & Control* – Comando y Control). Estas redes de bots reciben intrucciones de forma remota para infectar más equipos vulnerables en la red, y son usados para realizar ataques de denegación de servicio distribuido sobre sitios Web o a equipos conectados a Internet.

2. Amenazas humanas

Al hablar de amenazas, no se debe omitir que el factor humano es la principal fuente de éstas ya que existen en los sistemas de información y comunicación, en éste tipo de amenaza en el que se invierten más recursos para controlarlos y contrarrestar sus efectos. De aquí la famosa frase de que “el eslabón más débil en la seguridad informática es el humano”. Este tipo de amenazas pueden ser muy diversas, sin embargo, las más comunes se describen a continuación:

a) **Ingeniería social:** Se compone de todas aquellas estrategias y acciones a fin de obtener información confidencial mediante la manipulación de los custodios de la información, para que éstos la proporcionen de forma voluntaria. Esta amenaza se aprovecha de la debilidad natural de las personas que tienden a confiar en la palabra de otras.
b) **Sabotaje**: Es común que éstas acciones sean realizadas por personal de la propia organización, ya sea por un despido o inconformidad laboral o bien por un competidor directo de la organización. Las acciones van dirigidas a dañar físicamente o lógicamente los equipos informáticos críticos y causar así, una interrupción del servicio.
c) **Fraude:** Estas acciones se aprovechan de los privilegios y permisos que le permiten a una persona malintencionada obtener un beneficio propio, ajeno al objetivo de la organización a través del acceso a información y programas restringidos.
d) **Robo:** Son las actividades enfocadas a la extracción física o lógica de información mediante el empleo de dispositivos de almacenamiento o uso de cuentas de correo electrónico.

e) **Intrusos remunerados:** se trata de personas con un grado elevado de conocimientos de *Pen-testing,* los cuales son contratados para identificar las vulnerabilidades de la seguridad de los sistemas de información de una organización y realizar actividades malintencionadas como el copiado, borrado, modificación y creación de información.
f) **Curiosos:** Se trata de personas que vulneran la seguridad de sistemas, de los cuales no están autorizados para ingresar. Esto lo hacen por simple desafío a sus conocimientos y habilidades informáticas o por el deseo de conocer la información que resguardan los sistemas.

3. Amenazas de hardware

Aunado a las amenazas antes mencionadas, también existen las amenazas inherentes a fallas físicas que pueden ocurrir en alguno de los componentes que constituyen un equipo informático. De este tipo se mencionan las siguientes:

a) **De fabricación:** Ocurre cuando los componentes del hardware no son compatibles ni cumplen con los requerimientos para su correcto funcionamiento, por ejemplo, si se adquiere un módulo de memoria RAM y esta dañado, al colocarse en el equipo podría causar un daño físico irreversible en la tarjeta madre del sistema.

b) **Suministro de energía:** las variaciones de voltaje y no tener una conexión trifásica correctamente aterrizada, podría provocar daños en los dispositivos electrónicos. También al tener un voltaje de abastecimiento por debajo del requerido podría provocar daños a los circuitos de forma permanente.
c) **Uso o desgaste**: todos los componentes de un equipo de cómputo tienen una vida útil, si no se implementan mantenimientos preventivos al hardware, la disponibilidad de la información se podría ver afectada, por ejemplo, si la fuente de alimentación se dañara o si el disco duro sufriera un daño físico, el usuario se vería imposibilitado para acceder a la información contenida en el equipo.

1.3 **Vulnerabilidad**

Una vulnerabilidad se refiere a la debilidad existente en un sistema informático que permite que una entidad comprometa la confidencialidad, la integridad y la disponibilidad de los sistemas e información que procesan y almacenan.

Al hablar de vulnerabilidad es importante no perder de vista el concepto de riesgo, el cual representa la probabilidad de que una amenaza se materialice sobre un activo, explotando algún tipo vulnerabilidad. En un sistema informático se deben proteger los activos, es decir, todos aquellos recursos que conforman al sistema y se agrupan en activos de: hardware, software y datos. De los cuales, el más crítico de todos son los datos, los demás tienen una mayor probabilidad de restablecerse.

1.3.1 Tipos de vulnerabilidades

A continuación se muestra la clasificación general de vulnerabilidades que contempla seis categorías.

a) **Física:** Se relaciona con la posibilidad de acceso físico al área en donde se ubica el sistema. Las vulnerabilidades de este tipo son explotadas por falta o deficiencias en las políticas de acceso de personal no autorizado a áreas físicas restringidas.
b) **Natural**: Se refieren a las deficiencias para enfrentar desastres naturales relacionados con la humedad, polvo, agentes contaminantes, sismos, terremotos, inundaciones, incendios, etcétera. Las cuales se pueden ver reflejadas por ejemplo, en la falta de extintores de fuego, *no-breaks*, mal sistema de ventilación, entre otros.
c) **De hardware**: Este tipo de vulnerabilidades se relacionan con los defectos, fallas de fabricación, la no verificación de las especificaciones técnicas, falta de mantenimiento.
d) **De software**: Se relaciona con las fallas y debilidades que hacen posible el acceso indebido a los sistemas y en consecuencia a la información que éstos almacenan. Ejemplos: configuración e instalación indebida de programas, protocolos de comunicación carentes de seguridad, mal diseño y programación de una aplicación, etcétera.
e) **De red**: Al conectar un equipo a una red de datos se incrementa el riesgo de que sea comprometido, pues la cantidad de personas que pudieran interactuar con el sistema es mayor, sin olvidar el riesgo de que la información transmitida pueda ser interceptada por un ente no autorizado.
f) **Humana**: Esta vulnerabilidad se sustenta en el hecho de que el factor humano es el eslabón más débil de la seguridad y se ve reflejado en el daño que las personas pueden causar a la información y a los sistemas que la almacenan. Por ejemplo, las personas por su propia naturaleza son vulnerables a la ingeniería social, ingeniería social inversa, lo anterior aunado a la falta de capacitación, cansancio, aumentan el riesgo de explotación de esta vulnerabilidad.

Existe una clasificación más general de las vulnerabilidades la cual se menciona a continuación.

a) **De diseño:** Se presenta en el diseño de protocolos de red o comunicaciones, o bien en políticas de seguridad deficientes que fueron diseñadas para proteger los activos de la organización.
b) **De implementación**: Se presenta cuando existen errores de programación en las aplicaciones, por carencia o deficientes pruebas de “caja negra” y “caja blanca” en la fase de desarrollo y fase de pruebas.
c) **De uso:** Mala configuración asociada a la interacción que tienen las aplicaciones y sistemas operativos con el usuario.

Así mismo las vulnerabilidades se pueden clasificar en las siguientes dos categorías, esto en relación al tiempo de detección, características y posibles soluciones de la vulnerabilidad.

a) Vulnerabilidad conocida: son vulnerabilidades que se presentan en las aplicaciones y sistemas operativos de forma cotidiana debido a la naturaleza de su programación y funcionamiento. Su afectación en los sistemas está ampliamente documentada y existe más de una solución.

b) Vulnerabilidad de día cero: son todas aquellas vulnerabilidades que no tienen una forma de mitigación o erradicación conocida, pero sí se sabe cómo materializar una amenaza sobre ellas.

Existen sitios en Internet que se dedican a registrar y documentar las vulnerabilidades asociadas a versiones y variantes de sistemas operativos, así como de las aplicaciones que interactúan con los mismos. Estos sitios son útiles para identificar las vulnerabilidades asociadas a dicho software, así como las acciones que se pueden implementar para erradicar dicha vulnerabilidad de los sistemas, algunos de ellos se señalan a continuación en las Figuras 1.7-1.10.

a) NIST (*National Institute of Standards and Technology* – Instituto Nacional de Estándares y Tecnología, http://web.nvd.nist.gov/) (Figura 1.7)

Figura 1.7 - Base de datos de vulnerabilidades de la NIST.

b) OSVDB (*Open Source Vulnerability Database* – Base de Datos Abierta e Independiente de Vulnerabilidades, http://www.osvdb.org/) (Figura 1.8)

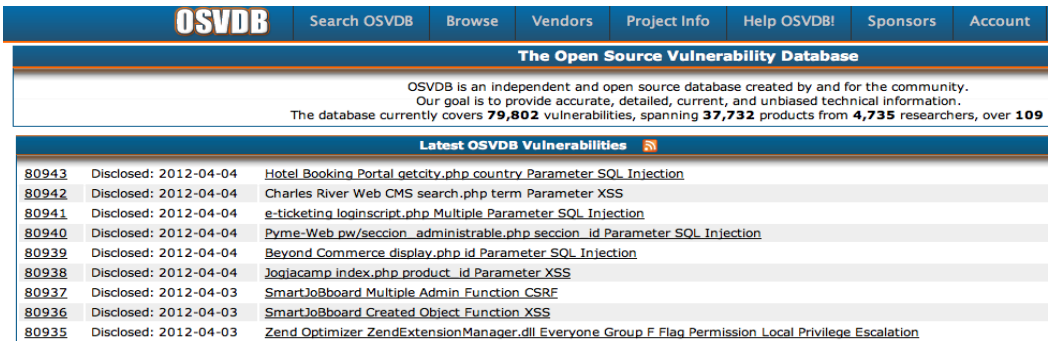


Figura 1.8 - Base de datos de vulnerabilidades de la OSVDB.

c) CVE (*Common Vulnerabilities and Exposures* – Vulnerabilidades y Exposiciones Comunes, http://cve.mitre.org/) (Figura 1.9)

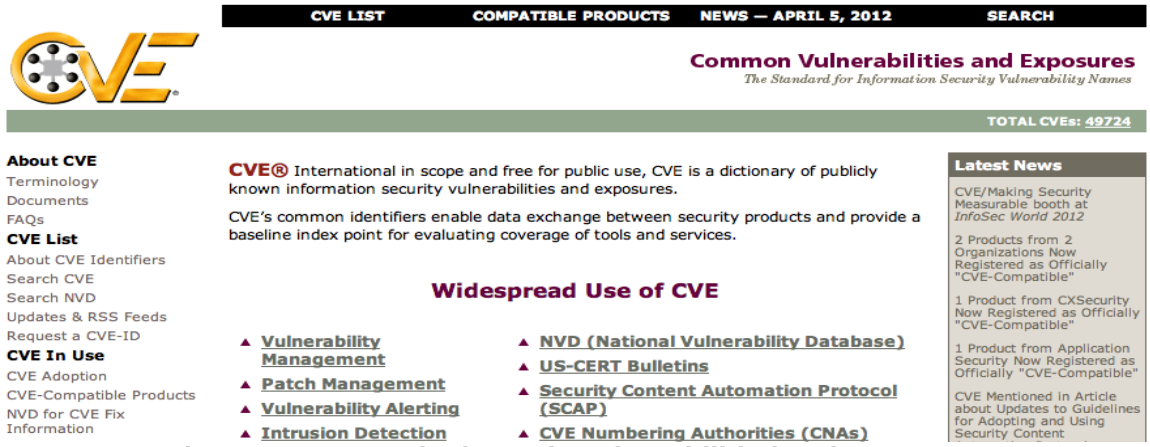


Figura 1.9 - Base de datos de vulnerabilidades de MITRE.

d) SecurityFocus (http://www.securityfocus.com) (Figura 1.10)

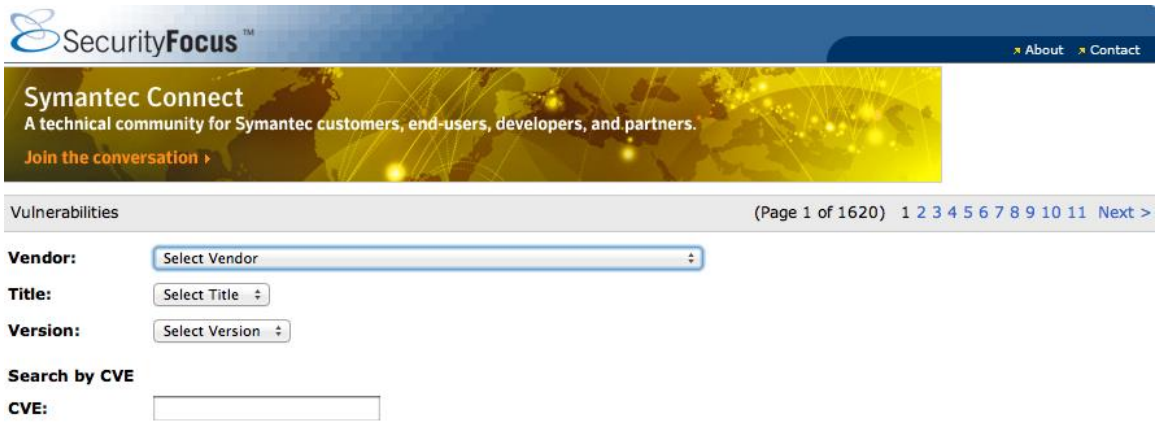


Figura 1.10 - Base de datos de vulnerabilidades de la Security Focus.

1.4 Servicios de seguridad informática

El objetivo de un servicio de seguridad es mejorar la funcionamiento de los sistemas de información y comunicación en las organizaciones. Los servicios de seguridad están diseñados para mitigar los ataques a la seguridad de la información y hacen uso de uno o más mecanismos de seguridad para proporcionar el servicio. Los servicios son seis y se detallan a continuación:

a) Confidencialidad

Este servicio de seguridad garantiza que la información no pueda estar disponible o accesible a personas, entidades o procesos no autorizados para que puedan leer, copiar o modificar la información.

Algunos de los métodos para garantizar la confidencialidad son:

- Uso de listas de control de acceso a los recursos críticos.

- Cifrado de la información almacenada y transmitida.

b) Autenticación

Servicio de seguridad que garantiza que la comunicación es auténtica, es decir, se encarga de verificar que el origen de los datos sea el correcto, quién los envía, así como comprobar que los datos se enviaron y recibieron de forma correcta.

Algunos métodos de autenticación son:

- Implementación de controles biométricos: huellas dactilares, retina, iris, geometría de mano, voz, etcétera.
- Tarjetas inteligentes (*smart card*) que guardan información del dueño de la misma.
- Uso de contraseñas robustas.

c) Integridad

Servicio de seguridad que garantiza que la información sea modificada, creada y borrada, sólo por el personal autorizado para ello. El principal problema de la integridad no se refiere a modificaciones malintencionadas, sino a los cambios accidentales.

Algunos métodos para detectar la pérdida de integridad son: Algoritmos hash (MD5, Sha-1, etcétera).

d) No repudio

El no repudio previene a los emisores o receptores de negar un mensaje transmitido o recibido. Cuando un mensaje es enviado, el receptor puede comprobar que el mensaje fue enviado por el presunto emisor. De la misma forma, cuando un mensaje es recibido, el remitente puede comprobar que el mensaje fue recibido por el receptor.

Los servicios de no repudio proporcionan evidencia que puede ser verificada por una tercera entidad. Los siguientes servicios son los que pueden ser proporcionados con infraestructura de llave pública y privada, en específico con el uso de firma electrónica:

- No repudio de origen: garantiza al receptor que el emisor no pueda negar haber enviado el mensaje.
- No repudio de envío: garantiza al emisor que el receptor no pueda negar haber recibido el mensaje.

e) Control de acceso

Servicio de seguridad que implementa controles de acceso, a fin de garantizar que un usuario sea identificado y autenticado de manera exitosa, para que entonces le sea permitido el acceso al activo o recurso.

f) Disponibilidad

Servicio de seguridad que garantiza que los usuarios autorizados tengan acceso a los activos y recursos del sistema, con base en los lineamientos de forma y tiempo definidos por el proveedor del servicio.

1.5 Ataques

En términos informáticos, un ataque es una técnica por la cual un intruso intenta tomar el control, dañar o perjudicar un sistema.

1.5.1 Tipo de Ataques

A continuación se detallan las cuatro categorías generales de ataques a la seguridad de la información. En las imágenes ilustrativas de cada uno de ellos, la entidad **A** funge como **Emisor**, la entidad **B** como **Receptor** y la **C** como **Intruso**.

a) Intercepción

La intercepción se presenta cuando los usuarios no autorizados acceden a datos o aplicaciones restringidas, éste es un ataque contra la confidencialidad de la información. La intercepción se puede materializar mediante la escucha de conversaciones, lectura de correo electrónico, lectura de archivos en formato físico o lógico, *sniffeo* del tráfico de red entre otros (ver Figura 1.11).

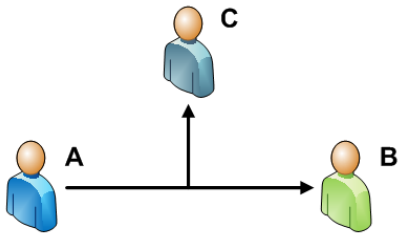


Figura 1.11 - Ataque de intercepción.

b) Interrupción

Los ataques de interrupción afectan la disponibilidad de los activos de forma temporal o permanente. La integridad de la información también se puede ver comprometida pues en el traslado de los datos, éstos podrían ser interceptados y modificados. Un ejemplo de estas amenazas son los ataques de Denegación de Servicio (*Denial of Service*) los cuales consisten en el envío masivo de peticiones de conexión hacia un servidor Web, en consecuencia, el servidor se ve imposibilitado para atender todas las peticiones de servicio de los clientes (ver Figura 1.12).

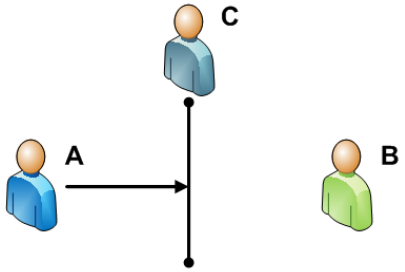


Figura 1.12 - Ataque de interrupción.

c) Modificación

En este tipo de ataques una entidad desconocida no sólo tiene acceso al recurso, también lo altera. Éste es un ataque en contra de la integridad de la información y representa un riesgo para la disponibilidad del servicio. Por ejemplo, al acceder a un archivo de configuración crítico del sistema (de servidor Web) y alterar su contenido, se corromperá la integridad del mismo y también la disponibilidad del servicio podría resultar comprometida si la configuración del archivo se modificara (ver Figura 1.13).

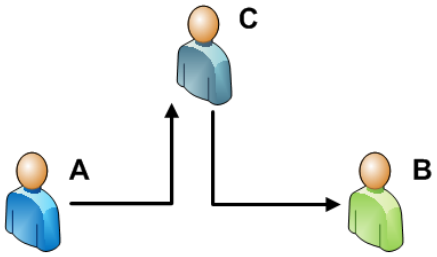


Figura 1.13 - Ataque de modificación.

d) Fabricación

Los ataques de fabricación implican la generación de datos, procesos, comunicaciones u otras actividades similares en un sistema informático. [10] Éste es un tipo de ataque contra la integridad y disponibilidad, por ejemplo, se pueden crear tramas de paquetes de red, con alteraciones en las cabeceras de los mismos, para después enviarlos al receptor para causar algún comportamiento no previsto en los sistemas, por ejemplo, el desencadenamiento de una falla de seguridad (ver Figura 1.14).

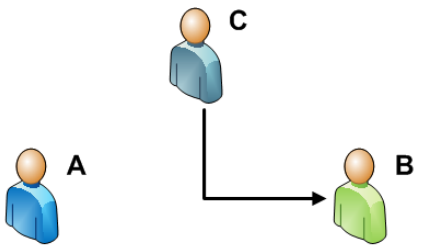


Figura 1.14 - Ataque de fabricación.

[10] Jason Andress. Op. Cit., p 9

Asimismo, cabe destacar que estos tipos de ataques se pueden clasificar dependiendo de sus efectos y acciones en las siguientes categorías:

a) Ataques pasivos

Estos ataques tienen como objetivo la intercepción de los datos, por tanto, no comprometen la integridad y disponibilidad de la información. Es decir, el intruso únicamente la observa, escucha, obtiene o monitorea mientras está siendo transmitida [11], por tal razón, estos ataques son los más difíciles de detectar ya que no realizan modificación alguna a los datos almacenados o transmitidos, sin embargo, es posible mitigarlos mediante la implementación de mecanismos de cifrado.

Los objetivos primordiales de este tipo de ataques son la intercepción de datos y el análisis de tráfico, los cuales, de tener éxito, podrían proporcionar datos del origen y destino de la comunicación, frecuencia, longitud de los mensajes y los tiempos en los cuales se genera determinado tipo de tráfico.

b) Ataque activos

En este ataque sí se altera la información, ya que implica la modificación del flujo de datos o la creación de flujos falsos, por ello estos ataques son difíciles de prevenir pues dependen de factores externos. La estrategia de mitigación ante estos ataques consiste en la detección y recuperación en el menor tiempo posible para reducir el impacto sobre los activos críticos. Los ataques de interrupción, modificación y fabricación pertenecen a esta categoría.

1.5.2 Fases de un ataque

Los ataques a los sistemas de cómputo según el autor Kimberly Graves[12], se pueden dividir en cinco fases (ver Figura 1.15), aunque existen otras fuentes que mencionan tres o cuatro, todo depende del autor y en este caso revisaremos las cinco fases, como se indica a continuación:

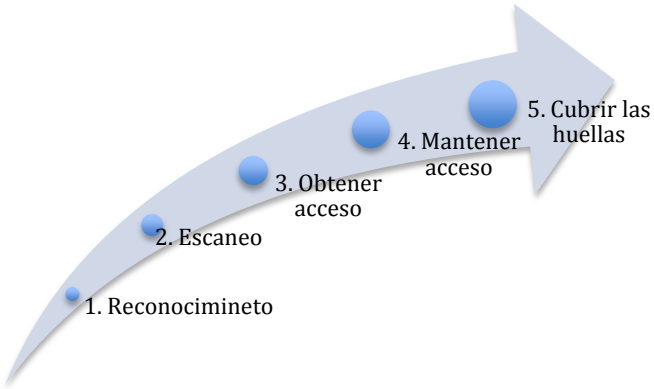


Figura 1.15 - Fases de un ataque.

[11] María J. López Barrientos, Cintia Quezada Reyes. Fudamentos de seguridad informática. p 105-106.

[12] Kimberly Graves, CEH Official Certified Ethical Hacker Review Guide, p 4-5.

Fase 1. Reconocimiento: esta fase puede hacer un reconocimiento pasivo o activo, todo depende de las herramientas empleadas. La meta principal de la fase es la obtención de información referente a la víctima potencial sin particularizar en un sistema. Se utiliza el *sniffing* de la red, ingeniería social y el reconocimiento activo que consiste en descubrir equipos vulnerables en la red.

Fase 2. Escaneo: esta fase requiere de la información obtenida en la fase anterior, para este punto el intruso ya seleccionó a él o los sistemas víctimas, y emplea escaneos de puertos, realiza mapeos de la red, *scanners* de vulnerabilidades, el resultado de esta fase le permitirá saber al intruso la forma y herramientas que se deberán emplear para vulnerar la seguridad de dicho sistema.

Fase 3. Obtener acceso: se lleva a cabo el ataque directo, con la información de las vulnerabilidades localizadas en las fases anteriores; se emplean las herramientas y técnicas para explotar dichas vulnerabilidades, para lo cual se pueden utilizar ataques basados en *buffer overflows*, malformación de paquetes, saturación del servicio, *SQL injection*, entre otros. Al término de esta fase, el intruso debe tener acceso al equipo víctima.

Fase 4. Mantener el acceso: Una vez que el intruso ha ganado acceso al sistema, intentará mantener abierta la puerta que le permitió el acceso o bien abrir alguna otra que no sea tan visible para el administrador del sistema. En algunas ocasiones, el intruso es quien aplica un *hardening* al propio sistema para evitar que otro intruso también vulnere dicho sistema. Se pueden utilizar backdoors, rootkits, troyanos o bien crear cuentas válidas en el sistema para acceder de forma remota por servicios como SSH (*Secure Shell* – Shell Seguro), FTP (*File Transfer Protocol* – Protocolo de Tranferencia de Archivos) o TELNET (*TELecommunication NETwork*).

Fase 5. Cubrir las huellas: Una vez que un intruso ha ganado acceso al sistema y puede mantener dicho acceso, él debe cubrir las huellas para evitar ser detectado por el personal de seguridad. Dependiendo del país y el contexto de dicho evento de seguridad, el intruso podría realizar esta fase para evitar acciones legales en su contra, es común el uso de esteganografía, borrado y alteración de bitácoras del sistema, ocultación de archivos, entre otros.

1.6 Niveles de seguridad informática de CC (Criterios Comunes)

1.6.1 El origen de los CC (Criterios Comunes)

Los TCSEC (*Trusted Computer System Evaluation Criteria* – Criterios de Evaluación de la Seguridad de los Sistemas de Computación) mejor conocidos como el “Libro Naranja”, fue desarrollado en el año de 1983 de acuerdo con las normas de seguridad en equipos de cómputo del Departamento de Defensa de los Estados Unidos. En este documento se definen cuatro niveles que describen cuatro divisiones jerárquicas de seguridad para la protección de la información, desde el mínimo grado de seguridad hasta el máximo. Los niveles de seguridad son protección mínima (D), discrecional (C), obligatoria (B) y controlada (A). Estos niveles han sido la base de desarrollo de estándares internacionales como el ISO/IEC (*International Organization for*

Standardization/International Electrotechnical Commission – Organización Internacional de Estandarización/Comisión Electrotécnica Internacional) la cual es una organización internacional no gubernamental que genera normas industriales y comerciales a nivel mundial.

De las cuatro divisiones mencionadas anteriormente, TCSEC define siete conjuntos de criterios de evaluación denominados como niveles D, C1, C2, B1, B2, B3 y A1, en donde el nivel D es el nivel más bajo y el nivel A1 el más alto. Además para implementar por ejemplo el nivel B2, en un sistema operativo de un equipo de cómputo, se establece que se requieren los niveles inferiores (D, C, C2 y B1). [13] Los niveles se definen de la siguiente forma:

- **Nivel D**. Protección mínima sin seguridad.
- **Nivel C1**. Limitaciones de accesos a datos.
- **Nivel C2**. Acceso controlado a los sistemas de información, archivos de registro y auditoría del sistema.
- **Nivel B1**. Equivalente al nivel C2, pero con una mayor protección individual por cada archivo.
- **Nivel B2**. Los sistemas deben estar diseñados para restringir el acceso de personas no autorizadas a información de carácter restringido o confidencial.
- **Nivel B3.** Dominios de seguridad. Los sistemas deben estar diseñados para ser altamente resistentes ante la interacción de personas no autorizadas.
- **Nivel A1.** Protección de seguridad verificada. [14]

Posteriormente, aparecieron el ITSEC (*Information Technology Security Evaluation Criteria* – Criterios de Evaluación de la Seguridad de las Tecnologías de Información) creados en el año de 1992 en colaboración de Francia, Alemania, Países Bajos y Reino Unido [15]. Este documento proponía más niveles que el TCSEC pero en general ambos eran similares y presentaban las mismas deficiencias con respecto a las evaluaciones de los productos. En la Figura 1.16, se observa la equivalencia entre ambos criterios, el ITSEC contempla más clases, sin embargo todas ellas son cubiertas por su antecesor el TCSEC, lo cual no representa una importante mejora en el criterio, pero si una clasificación más particular.

Clases ITSEC		Clases TCSEC
E0	<-->	D
F-C1, E1	<-->	C1
F-C2, E2	<-->	C2
F-B1, E3	<-->	B1
F-B2, E4	<-->	B2
F-B3, E5	<-->	B3
F-B3, E6	<-->	A1

Figura 1.16 - Comparación entre ITSEC y TCSEC

[13] Donald C. Latham. Department of Defense Trusted Computer System Evaluation Criteria, 26 de diciembre de 1985, p 11.

[14] Glosario y abreviaturas (2011), disponible en https://www.ccn-cert.cni.es/publico/seriesCCN-STIC/series/400-Guias_Generales/401-Glosario_y_abreviaturas/401/index.html, consultado 11-04.2011.

[15] Information Technology Security Evaluation Criteria (ITSEC) Provisional Harmonised Criteria, 28 de Junio de 1991, COM(90) 314, p 14.

1.6.2 CCITSE (*Common Criteria for Information Technology Security* - Criterios Comunes para la Seguridad de las Tecnologias de Información)

Este nuevo esquema de evaluación de seguridad, mejor conocido como CC (*Common Criteria* – Criterios Comunes) reemplazó a (ITSEC y TCSEC). El esquema fue aprobado en 1999 y adoptado por ISO como criterios de evaluación para la Seguridad de Tecnologías de la Información (ISO/IEC 15408).

En el año 2000 se firmó un importante Acuerdo de Reconocimiento Mutuo (MRA por sus siglas en inglés) que incluía a los países de Canadá, Francia, Alemania, Reino Unido, Australia, Nueva Zelanda, Finlandia, Grecia, Italia, Holanda, Noruega y España. El acuerdo indicaba que un certificado CC obtenido en un país, es reconocido por los demás países firmantes [16].

Los CC son un esfuerzo para alinear las Tecnologías de la Información (TI) a criterios de seguridad comunes, que han sido concensuados y aceptados por varios países. Los objetivos de los Criterios Comunes son mejorar la seguridad de TI en el desarrollo y evaluación del producto, a fin de proteger las inversiones hechas en productos de seguridad de TI y de esta forma facilitar la adopción internacional.

De acuerdo con las autoras del libro Fudamentos de seguridad informática [17]los CC son una norma internacional utilizada para evaluar la seguridad de los productos de tecnología de la información basados en criterios comunes, en esta evaluación existen tres grupos de interés: consumidores, desarrolladores y evaluadores.

Las clases de productos se evalúan mediante Perfiles de Protección (PP por sus siglas en inglés) que describen los requerimientos de seguridad que los productos de seguridad deben cumplir. Estos PP se aplican a *firewall*s, tarjetas inteligentes, IDS, IPS, y otros productos que deberían tener requerimientos de seguridad. Los CC también definen los niveles de confianza (*Evaluation Assurance Level* – Evaluación del Nivel de Garantía), para los productos evaluados, si un producto obtiene un alto EAL significa que tiene un alto nivel de confianza con respecto a las funciones de seguridad, las cuales serán implementadas efectivamente.

De esta forma los consumidores podrán apoyarse en los CC para saber si el producto que desea adquirir cumple o no con sus expectativas. En el caso de los desarrolladores de productos de TI, los CC les proporcionan información para dirigir sus desarrollos a las necesidades de los consumidores y para los evaluadores, los CC les proveen aspectos de seguridad en TI que deben ser implementados por los desarrolladores de tal forma que la evaluación sea repetible, independiente y confiable para asegurarles a los consumidores que los productos de seguridad adquiridos cumplen con los requerimientos que dicen cubrir los desarrolladores.

---

[16] María J. López Barrientos, Cintia Quezada Reyes, Op. Cit. p 28.

[17] Ibídem, p 28.

Una descripción detallada de los CC se sale del objetivo de este trabajo de tesis, por tanto se mencionan de forma general las partes que componen la norma.

Parte 1. Introducción y modelo general: según las autoras del libro Fudamentos de seguridad informática [18], esta sección está destinada a las personas con pocos conocimientos referentes a la evaluación de seguridad. En ella se describe cómo se establecieron los CC y hacia quiénes están dirigidos.

Parte 2. Requerimientos de seguridad funcional: sección dirigida a usuarios y desarrolladores, en ella se expresan los requerimientos de seguridad funcional para los productos de TI, estos requeriemientos están organizados en 11 clases (clases de requerimientos funcionales) cada una de ellas enfocada a un área particular de la seguridad, a su vez cada clase se descompone en familias funcionales, donde cada una trata diferentes aspectos de la seguridad referentes a su clase. Las clases son:

1. Clase FAU: Auditoría de seguridad.
2. Clase FCO: Comunicación.
3. Case FCS: Soporte de cifrado.
4. Clase FDP: Protección de datos de usuario.
5. Clase FIA: Identificación y autenticación.
6. Clase FMT: Administración de seguridad.
7. Clase FPR: Privacía.
8. Clase FPT: Protección de las funciones de seguridad del objeto de evaluación.
9. Clase FRU: Utilización de recursos.
10. Clase FTA: Acceso a la TOE (objeto de evaluación).
11. Clase FTP: Canales confiables.

Parte 3. Requerimientos de garantía de seguridad: esta sección es destinada a los desarrolladores de TI, pues en ella se define el criterio de confiabilidad usado para evaluar el desempeño de los desarrolladores y sus productos [19]. Implementa 7 niveles de evaluación de garantía (*Evaluation Assurance Level* – EAL) que definen la escala de los CC utilizados en la evaluación de los productos. Las clases de garantía son:

1. Clase ACM: Administración de la configuración.
2. Clase ADO: Distribución y operación.
3. Clase ADV: Desarrollo.
4. Clase AGD: Documentos guía.
5. Clase ALC: Soporte del ciclo de vida.
6. Clase ATE: Pruebas.
7. Clase AVA: Evaluación de la vulnerabilidad.

[18] Ibídem, p 30-44.

[19] Ibídem, p 44-52

Capítulo 2

Buenas Prácticas y Estándares de Seguridad Informática

2.1 **BUENAS PRÁCTICAS**

En la actualidad existe una gran variedad de normas y estándares de seguridad desarrollados por diversas organizaciones, públicas y privadas, que tienen como principal propósito la elaboración de normas y estándares aplicables a distintos ámbitos de la seguridad de la información. Cada organización que desee alinearse a ellas debe conocer a detalle las ventajas y requisitos de su implementación para poder determinar la factibilidad de su puesta en marcha.

Existen normas y estándares para la fabricación de productos, los cuales establecen lineamientos y requisitos que los fabricantes deben cumplir, de tal forma que sus productos puedan ser usados en todo el mundo. Estos estándares pretenden asegurar que las características de los productos sean consistentes, por ejemplo:

- Los productos que tienen el logotipo con las siglas *WiFi,* deben garantizar que cumplen con el estándar 802.11, relacionado a redes inalámbricas de área local, lo que significa que el producto es compatible con cualquier otro producto que cumpla con el mismo estándar, aunque éste haya sido fabricado en cualquier parte del mundo.
- "El uso de tarjetas de crédito en cualquier cajero automático, esto es posible ya que los fabricantes de las tarjetas de crédito se basan en el estándar *ISO 7819* en el cual, entre otros aspectos, define las dimensiones del plástico" [20].

Una característica fundamental de las normas y estándares es que son aplicables a diversos tipos de organizaciones (pequeñas, medianas, grandes, con pocos o muchos recursos, locales, internacionales, etcétera). Este capítulo pretende dar a conocer la importancia de la implementación de estándares para mitigar el riesgo de ataques a la seguridad informática contenida en sistemas de información y comunicación críticos pertenecientes a diversos Equipos de Respuesta a Incidentes que deseen mejorar la seguridad de la información en sus procesos.

Las buenas prácticas son un conjunto de acciones que han tenido un excelente desempeño y resultados en un determinado escenario y que se espera que en escenarios similares se obtengan resultados parecidos. Comúnmente las mejores prácticas son sólo el comienzo y sirven para acelerar la implementación de un servicio de mejoras en los procesos de las instituciones.

Una definición más formal se puede encontrar en el sitio european-microfinance.org [21], el cual dice que las buenas prácticas son aquellas estrategias, planes, tácticas, procesos, metodologías, actividades y enfoques documentados que son eficaces, pertinentes y ampliamente aceptados, además de que son desarrollados por organizaciones profesionales e implementados por personal capacitado. Estas buenas

[20] Estándares de comunicaciones (1999), disponible en http://www.eveliux.com/mx/estandares-de-telecomunicaciones.php, consultado 20-04-2011.

[21] Jeffrey Steve Borbón Sanabria. ¿Qué son las buenas prácticas? Revista de seguridad de la UNAM (n.d), disponible en http://www.european-microfinance.org/good-practice-definition_es.php, consultado 21-01-13.

prácticas previamente debieron haber sido puestas en marcha mediante la investigación y experiencia, la cual debió dejar ver que son fácilmente modificables y mejoradas según el contexto.

Los criterios de selección que toda buena práctica debe cumplir son:

- Estar documentada.
- Ser accesible.
- Basada en procesos y metodologías.
- Implementada.
- Capaz de establecer objetivos.
- Transferible.
- Sostenible: La relación entre costo y beneficio debe ser favorable.
- Encontrarse en un proceso de evaluación y de mejora continua.
- Eficiente.
- Eficaz.

Resumiendo lo anterior, una buena práctica es simplemente un proceso o metodología que representa la forma más efectiva de lograr un objetivo específico, por lo consiguiente, las buenas prácticas de seguridad de la información están formadas por políticas y normas específicas que en la mayoría de los casos provienen de aquellas que se aplican a la administración de empresas. Por ejemplo, una buena práctica de tipo general consiste en difundir ampliamente las normas de manera que todos los participantes en una actividad las entiendan y las acepten.

En la actualidad existe una gran variedad de estándares internacionales, guías y manuales de buenas prácticas que son ampliamente empleados para buscar el aseguramiento de la información dado que es el activo más valioso y preciado de toda organización. Para resguardar la información, es necesario implementar procesos ligados a buenas prácticas que tengan como propósito la protección de la integridad, disponibilidad y confidencialidad de la información, por lo tanto la buena práctica debe ser una acción positiva que debe tener éxito, ser innovadora y sostenible.

2.2 **Estándares**

2.2.1 Definición de Estándar

La *ISO* define los estándares como “los acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías o definiciones de características para asegurar que los materiales, productos, procesos y servicios cumplan con su propósito” [22]. Los cuales deberán ser documentados, difundidos y adoptados de la misma forma por todas las entidades que los adopten.

[22] Estándares de comunicaciones, Op. Cit.

El diccionario de la Real Academia de la Lengua dice que un estándar es lo "que sirve como tipo, modelo, norma, patrón o referencia" [23]. En el campo técnico, la estandarización es el proceso por el cual se establecen normas comúnmente aceptadas, que permiten la cooperación de diferentes empresas o instituciones sin menospreciar su posibilidad de competir. Los estándares contienen especificaciones técnicas que deben reunir los productos o servicios que requieran cumplir satisfactoriamente con las funciones para los que fueron creados, para lo cual existen diversos tipos de estándares de acuerdo con su alcance y cobertura, los cuales se explican a continuación.

a) **Estándar Internacional:** "Son todos aquellos estándares normalizados o adoptados oficialmente por organismos de estandarización internacionales, conformados por los representantes legales de cada país" [24]. Este tipo de estándares también son conocidos como de *jure,* los cuales son promulgados por grupos de gente de diversas áreas de conocimiento que contribuyen a la definición y desarrollo del estándar.

b) **Estándar Nacional:** Son estándares nacionales normalizados por los cuerpos de estandarización nacionales de cada país. En México la Dirección General de Normas de la Secretaría de Economía es la representante oficial de ISO en el país.

c) **Estándar Local:** Son estándares promovidos, generados y adoptados por organizaciones industriales, las cuales realizan ciertos productos en un país y no los pueden hacer de la misma forma en otro país.

Existen diversos tipos de estándares definidos con base en el alcance y comunidad a la que van dirigidos. Es importante no confundir el concepto de norma y estándar, la norma es una serie de requisitos que cierto producto o servicio debe cumplir con base en una ley establecida. En cambio, el estándar mide la calidad y los requisitos que debe cumplir dicho producto o servicio para su aceptación y adopción. Sin embargo, para efectos de este trabajo se manejará el término de normas y estándares en el mismo sentido y con base a la siguiente definición:

"Los estándares y normas son descripciones técnicas detalladas, elaboradas con el fin de garantizar la interoperabilidad entre elementos construídos independientemente, así como la capacidad de replicar un mismo elemento de manera sistemática" [25].

---

[23] Real Academia Española: Estándar (n.d), disponible en http://buscon.rae.es/ consultado 21-04-212.

[24] Opentia, Estudio sobre Estándares informáticos tipos y caracterizaciones (2007), disponible en http://people.ffii.org/~abarrio/estandares/OPENTIA-estudioTiposDeEstandares-20070129.pdf , consultado 21-04-2012.

[25] Normas y Estándares (n.d), disponible en http://es.wikitel.info/wiki/Normas_y_estandares, consultado 21-04-212.

2.2.2 Historia de los Estándares

Definir en qué año o época se implementó el concepto de estandarización, es una labor complicada debido a que la normalización de las cosas, es una acción que muchas de las personas aplican a diario sin darse cuenta, sin embargo, como explica Evelio Martínez[26], formalmente la historia de la estandarización comienza en el siglo XIX con el invento del telégrafo, evento que hizo evidente la necesidad de las personas de comunicarse de forma rápida y eficiente, este evento ayudó a que en 1865 se fundara la *ITU* (*International Telegraph Union* – Unión Internacional de Telegrafía), que fue la primera organización separada del gobierno y de carácter internacional, la cual representó el primer esfuerzo para estandarizar las comunicaciones en varios países.

Posteriormente en 1884 se funda la *IEEE* (*Institute of Electrical and Electronics Engineers* – Instituto de Ingenieros Eléctricos y Electrónicos), un organismo encargado de la promulgación de estándares para redes de comunicaciones, está conformada por ingenieros eléctricos, en sistemas y telecomunicaciones.

En 1906 se funda la *IEC* (*International Electrotechnical Commission* – Comisión Electrotécnica Internacional), organismo que realiza y promueve estándares para el área de la Ingeniería Eléctrica y Electrónica.

En el año 1918, un grupo de organizaciones enfocadas a la rama de la ingeniería fundaron el Comité Estadounidense de Estándares para la Ingeniería, el *AESC* (*American Engineering Standards Committee*), el cual en 1928 se convierte en la Asociación de Estándares Estadounidense (*ASA - American Standards Association*).

En 1947 es fundada la ISO (*International Organization for Standardization* – Organización Internacional para la Normalización), la cual se ocupa en definir y promulgar estándares, solo que a una diversidad más amplia de sectores.

Posteriormente en “1966 ASA se convierte en el Instituto de Estándares de los Estados Unidos de América (*USASI – The United States of America Standards* Institute), y finalmente en el año de 1969 se le denomina formalmente como *ANSI* (*American National Standards Institute* – Instituto Nacional Americano de Estándares), el cual forma parte de la *ISO* y de la *IEC* (*International Electrotechnical Commission* – Comisión Electrotécnica Internacional)” [27].

2.2.3 Estándares y normas de seguridad

Conforme se comienza a adentrar en el mundo de la seguridad de la información, se incrementa la necesidad de concientizar a los usuarios en el manejo correcto de la información y los sistemas que interactúan con la misma, sin embargo, en ambientes

26 Ibídem. Historia de la estandarización.

27 ANSI Historical Overview (n.d), disponible en http://www.ansi.org/about_ansi/introduction/history.aspx consultado 20-04-2014.

laborales o empresariales no es tan simple poder sugerirle a un compañero qué antivirus emplear o cómo debería proteger la información almacenada en el teléfono celular, esto debido a que existen lineamientos preestablecidos por los administradores de la seguridad, los cuales basan y justifican su actuar en "…guías y documentos que describen cómo abordar la seguridad de una forma responsable, procedimental y orientada al cumplimiento de los estándares mínimos requeridos para la tecnología actual…"[28].

Lo anterior deja ver por qué se ha popularizado el uso de normas y estándares en la actualidad y en consecuencia que cada vez más procesos y actividades de las organizaciones sean repetibles, organizadas y estructuradas. Es por ello que entidades como *ISO y IEEE* entre otras, proponen documentos de este tipo, los cuales se crean a partir de la experiencia de diferentes grupos que participan durante el proceso de definición y culminar con documentos que serán puestos a disposición del público, algunos por un precio y otros gratuitos.

Como ejemplo de estándares o normas de seguridad se tienen el *ISO/IEC 27002, ISO/IEC 27005, CoBIT, ITIL, NIST SP 800-30, BS 25999,* etcétera.

Lograr una alineación entre los documentos antes indicados requiere un arduo trabajo y colaboración entre diversas áreas de una organización, pero ofrece la oportunidad de crear un conjunto de métodos y procedimientos complementarios encaminados al aseguramiento de los activos, procesos y recursos tecnológicos de una organización, además de fomentar la conciencia de los usuarios en términos de seguridad de la información.

2.2.4 Norma Mexicana

En México existe el "Sistema Mexicano de Evaluación de la Conformidad, que comprende la certificación obligatoria (Normas Oficiales Mexicanas, NOM) o voluntaria (Normas Mexicanas, *NMX*) de las cuales sólo las *NOM* son de uso obligatorio en su alcance y las *NMX* expresan recomendaciones o parámetros" [29].

Figura 2.1 - Logotipo de la Norma Mexicana.[30]

[28] Buenas prácticas, estándares y normas (2011), disponible en http://revista.seguridad.unam.mx/numero-11/buenas-prácticas-estándares-y-normas, consultado 22-01-2013.

[29] Normalización en México (n.d), disponible en http://www.mitecnologico.com/Main/Normalizaci%F3nEnM%E9xico, consultado 21-04-2012.

[30] NOM Certification, (2014), disponible en http://www.e-switch.com/Portals/0/NOM.png, consultado 21-04-2012.

La Figura 2.1 representa el logotipo basado en "la marca que pueden colocar los fabricantes que cumplieron con la acreditación de un producto en la conformidad de alguna norma de tipo *NOM*, el formato es el siguiente NOM-NNN-LLLL-NNNN N=dígito, L=letra" [31].

2.2.5 Familia de Normas ISO/IEC 27000

Antes de entrar de lleno a la definición de la familia de normas *ISO/IEC* es conveniente explicar que el objetivo primordial de la *Organización Internacional para la Normalización* es consolidarse como la entidad responsable de la normalización a nivel mundial. Todo ello mediante la promoción del "desarrollo de la normalización en aras de fortalecer el intercambio de bienes y servicios que favorezcan las actividades de cooperación económicas, intelectuales, científicas y tecnológicas" [32], de esta forma se concilian los intereses de fabricantes, usuarios y gobiernos para la elaboración de normas técnicas internacionales.

El alcance de *ISO* "abarca la normalización de todos los campos, salvo los referidos a normas sobre tecnología eléctrica y electrónica, de los cuales es responsable la Comisión Internacional de Electrotecnia (*IEC – International Electrotechnical Commission*)" [33], por tal razón es común ver en ciertas normas la terminación *ISO/IEC*.

ISO es una red de 163 institutos nacionales de normalización, un miembro por país (con 25,572 colaboradores en todo el mundo), tiene una Secretaría Central en Ginebra (Suiza), la cual desde su fundación en febrero de 1947 ha publicado más de 19,000 normas, tal como se puede ver en la Tabla 2.1, donde también se muestra que la mayor parte de los Estándares (5,242) pertenecen a la categoría de *Engineering technologies* que representan el 26.7% del total.

Tabla 2.1 Producción de Estándares por sector. [34]

Sectores (basados en la Clasificación Internacional para Estándares [ICS])	Elementos de trabajo		Estándares Internacionales			
	Nuevos	Totales	Nuevos	No. páginas	Total de estándares	Total de páginas
Generalidades, infraestructuras y ciencias	**72**	**746**	**103**	**6556**	**1703**	**72691**
Salud, seguridad y medio ambiente	**82**	**193**	**58**	**2642**	**773**	**32883**
Tecnologías de Ingeniería	**378**	**958**	**352**	**14900**	**5242**	**226283**
Electrónica, tecnologías de información y comunicación	**313**	**698**	**268**	**36811**	**3186**	**231970**
Transporte y distribución de mercancías	**167**	**422**	**89**	**3497**	**1988**	**57700**
Agricultura y tecnología alimentaria	**66**	**150**	**50**	**1427**	**1094**	**28283**
Tecnología de materiales	**275**	**671**	**242**	**7628**	**4460**	**124058**
Construcción	**58**	**143**	**35**	**1459**	**423**	**16479**
Tecnologías especiales	**8**	**26**	**11**	**241**	**154**	**3862**
TOTAL	**1419**	**4007**	**1208**	**75161**	**19023**	**794209**

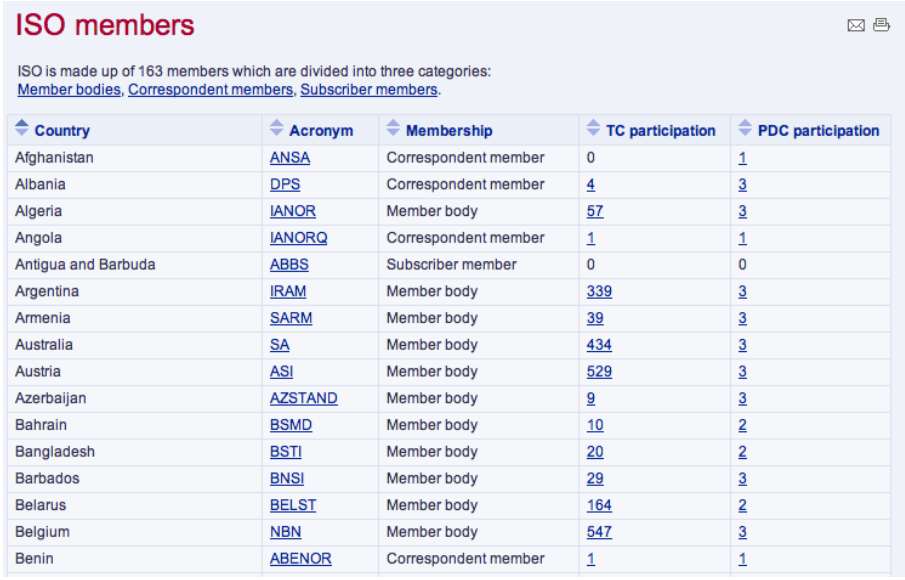
31 Normatividad Mexicana (2015), disponible en http://es.wikipedia.org/wiki/Normatividad_Mexicana consultado 21-04-2014.

32 ISO (n.d), disponible en http://www.unit.org.uy/miembros/iso.php, consultado 21-04-2012.

33 Ibídem, ISO.

34 ISO in figures (2011), disponible en http://www.iso.org/iso/iso-in-figures_2011.pdf, consultado 18-04-2012.

La Figura 2.2 muestra el número de miembros de *ISO* por cada país.



ISO members

ISO is made up of 163 members which are divided into three categories:
Member bodies, Correspondent members, Subscriber members.

Country	Acronym	Membership	TC participation	PDC participation
Afghanistan	ANSA	Correspondent member	0	1
Albania	DPS	Correspondent member	4	3
Algeria	IANOR	Member body	57	3
Angola	IANORQ	Correspondent member	1	1
Antigua and Barbuda	ABBS	Subscriber member	0	0
Argentina	IRAM	Member body	339	3
Armenia	SARM	Member body	39	3
Australia	SA	Member body	434	3
Austria	ASI	Member body	529	3
Azerbaijan	AZSTAND	Member body	9	3
Bahrain	BSMD	Member body	10	2
Bangladesh	BSTI	Member body	20	2
Barbados	BNSI	Member body	29	3
Belarus	BELST	Member body	164	2
Belgium	NBN	Member body	547	3
Benin	ABENOR	Correspondent member	1	1

Figura 2.2 - Miembros del ISO.[35]

El *IEC* (*International Electrotechnical Commission* – Comisión Electrotécnica Internacional), es una organización sin fines de lucro y no gubernamental. Su principal objetivo es preparar y publicar estándares internacionales para todas las tecnologías eléctricas o bien relacionadas a la electrónica.

IEC surgió en la ciudad de Londres (Reino Unido) en el año de 1906, posteriormente en el año de 1948 cambia de sede a Geneva (Suiza), ciudad en la que también se encuentra la sede de *ISO*.

"Actualmente tiene 63 miembros, está conformado por 174 comités y subcomités, además cuenta con 532 grupos de trabajo encargados de producir normas de seguridad, desempeño, construcción y equipamiento eléctrico, y servicios para sectores de productos específicos" [36].

Las organizaciones ISO e *IEC* (ver Figura 2.3) han establecido un comité técnico en conjunto denominado *ISO/IEC JTC1* (*ISO/IEC Joint Technical Committee*). Este comité se enfoca en los asuntos relacionados con tecnologías de la información. La mayoría del trabajo de *ISO/IEC JTC1* se realiza por subcomités que tratan con un campo o área en particular. Específicamente el subcomité *SC 27* es el que se encarga de las técnicas de seguridad en tecnologías de información. [37] Dicho subcomité ha desarrollado la familia de estándares internacionales para el Sistema de Gestión y Seguridad de la Información (*SGSI*). La familia incluye estándares internacionales

35 ISO members (2012), disponible en http://www.iso.org/iso/about/iso_members.htm, consultado en 18-04-2012.

36 IEC (n.d), disponible en http://www.unit.org.uy/miembros/iec.php, consultado 19-04-2014.

37 ISO/IEC JTC1 disponible en http://www.iso27000.es/otros.html#section4a, consultado 10-04-2014.

sobre requerimientos, gestión de riesgos, métricas, medición y lineamientos de implementación del *SGSI*. Esta familia adoptó el esquema de numeración utilizando las series del número 27000 en secuencia, por lo que a partir de julio de 2007 ediciones como el *ISO/IEC 17799* ahora se encuentran bajo el esquema de numeración con el nombre *ISO/IEC 27002*:2013.

Figura 2.3 - Logotipo de ISO e IEC.[38]

El objetivo principal de la *ISO/IEC 27000* es la mejora continua de los procesos relacionados con el manejo de la información. Asimismo, pretende que las organizaciones certificadas con esta norma demuestren la eficacia en los mecanismos y procesos que garantizan la confidencialidad, integridad y disponibilidad de la información, además de asegurar la continuidad del negocio y la afectación mínima de los activos ante diversos incidentes.

La familia de normas *ISO/IEC 27000* proporciona un marco para la gestión de la seguridad. Estas normas especifican los requisitos para implementar y mantener un *SGSI* (Sistema de Gestión de Seguridad de la Información) aplicable a cualquier tipo de organización, pública o privada, grande o pequeña. A continuación se describen de forma general las normas de la familia *ISO/IEC 27000* que más se relacionan con la seguridad de la información (ver Figura 2.4).

a) ISO/IEC 27000: "Establece el vocabulario estándar a emplearse en el SGSI, a fin de evitar distintas interpretaciones de conceptos técnicos y gestión"[39].

b) ISO/IEC 27001: Especifica los requisitos a cumplir para implantar el SGSI certificable, es la norma más importante de la familia:
- Define el *SGSI*, su gestión y las responsabilidades de los participantes.
- Se basa en el modelo de *PDCA* (*Plan-Do-Check-Act* – Planificar-Hacer-Verificar-Actuar).
- Puntos clave: Gestión de riesgos y mejora continua de procesos.

[38] Logotipo de ISO, disponible en http://www.bitcompany.biz/wordpress/wp-content/uploads/2011/07/iso-iec-logos.jpg, consultado 11-11-2014.

[39] ISO 27000 (n.d), disponible en http://www.iso27000.es/download/doc_iso27000_all.pdf, consultado 19-04-2014.

c) ISO/IEC 27002: Define el código de buenas prácticas, no certificable, su objetivo es la gestión de la seguridad de la información.

- Provee recomendaciones sobre las medidas que las organizaciones pueden implementar para asegurar su información y procesos relacionados.
- No es un estándar certificable.
- Hasta antes el 1 de julio de 2007, era conocida como *ISO 17799:2005*.

d) ISO/IEC 27003: Establece la guía para la "implementación de SGSI e información acerca del uso del modelo PDCA"[40] y de los requerimientos de cada una de sus cuatro fases. Tiene su origen en el Anexo B de la Norma BS7799-2 y en la serie de documentos publicados por la BSI (*British Standards Institution* – Institución de Estándares Británicos).

e) ISO/IEC 27004: Establece la forma de medir la eficacia del *SGSI* y de los controles mediante métricas y técnicas, las cuales se usan fundamentalmente en la medición de los componentes de la fase *Do (*Hacer*)* del ciclo *PDCA*.

f) ISO/IEC 27005: Trata acerca de la gestión de riesgos en seguridad de la información definido en la norma *ISO/IEC 27001*. Provee recomendaciones, métodos y técnicas de evaluación de riesgos de Seguridad.

Figura 2.4 - Familia ISO 27000.

Las normas restantes de la familia *ISO/IEC 27000* se salen del alcance de este trabajo de tesis por lo tanto no fueron descritas en los incisos anteriores. Cabe destacar que para definir las buenas prácticas de seguridad que serán implementadas en el aseguramiento de sistemas Linux-Debian se tomará como referencia el *ISO/IEC 27002*:2013.

[40] Ibídem, ISO 27000.

2.3 **Política de seguridad**

2.3.1 Definición de una Política

Antes de presentar el concepto de política de seguridad, se muestran los seis puntos que para el autor del libro *Manual para Elaborar Manuales de Políticas y Procedimientos,* es y debe ser una **política**.

a) "La decisión unitaria que se aplica a todas las situaciones similares.
b) Una orientación clara de hacia dónde deben dirigirse todas las actividades de un mismo tipo.
c) La manera consistente de cómo tratar a la gente.
d) Un lineamiento que facilita la toma de decisiones en actividades rutinarias.
e) Lo que la dirección desea que se haga en cada situación definida.
f) Aplicable al 90-95% de los casos. Las excepciones sólo podrán ser aplicables por alguien de un nivel inmediato superior". [*41*]

Con relación a los aspectos descritos anteriormente, en los que el autor Martín describe una política, se puede definir el concepto de política de seguridad como: las directrices y objetivos generales de una organización relativos a la seguridad de la información, los cuales deben alinearse a la misión y visión de la organización. Es decir, la política de seguridad regula la forma en la que una organización previene, protege y gestiona los riesgos identificados.

De acuerdo con el libro *Fundamentos de Seguridad Informática* una política de seguridad "*es un conjunto de leyes, reglas y prácticas que regulan la manera de dirigir, proteger y distribuir recursos en una organización para llevar a cabo los objetivos de seguridad informática dentro de la misma*" [42].

En este sentido, las políticas de seguridad surgen como una herramienta de control institucional para la concientización del personal acerca de la importancia y criticidad que tienen para la organización sus activos críticos. Es por ello que proponer y establecer políticas de seguridad, requiere de un compromiso serio por parte de la propia organización. [43]

Antes de implementar políticas de seguridad en una organización se debe realizar un análisis de las amenazas que podrían comprometer la seguridad de los activos críticos, así como el impacto y probabilidad de que esto ocurra, todo ello mediante la relación de los aspectos sensibles, físicos y lógicos que intervienen en la organización. Una vez concluida la lista de activos críticos a proteger, se pondera cada uno de ellos con un valor específico y se calcula la probabilidad de que sea vulnerado, esto servirá para definir qué activos son más críticos y por ende cuáles deberían ser los controles a implementar en cada uno de ellos.

---

[41] Martín G. Álvarez Torres. Manual para Elaborar Manuales de Politicas y Procedimientos. p 80.

[42] María J. López Barrientos, Cintia Quezada Reyes. Fudamentos de seguridad informática. p 129.

[43] Políticas de Seguridad (n.d). disponible en http://auditoriasistemas.com/auditoria-informatica/politicas-de-seguridad/, consultado 03-04-2014.

Con base en lo anterior, previo a definir una política se deben responder las siguientes cuestiones que servirán de guía para definir una política de seguridad.

1. ¿Qué se quiere proteger?
2. ¿Contra quién?
3. ¿Cómo?

También se debe definir la directriz que adoptará la organización con respecto a las políticas de seguridad, estas directrices son:

a) Todo lo que no está prohibido, está permitido.
b) Todo lo que no está permitido, está prohibido.

Las políticas de seguridad de la información deben considerar principalmente los siguientes elementos:

a) Alcance de las políticas.
b) Objetivos de la política.
c) Responsabilidades de los usuarios con respecto a los activos que tienen acceso.
d) Requerimientos para su aplicación en los sistemas de información.
e) Sanciones por el no cumplimiento de las políticas.

Es necesario resaltar que las políticas de seguridad de la información por sí solas no constituyen una garantía para la seguridad de la organización, ellas son sólo un complemento de los mecanismos de seguridad que las organizaciones deberían implementar para salvaguardar sus activos críticos.

2.3.2 El ISO/IEC 27001

Este *ISO* es una norma de gestión de seguridad de la información específica para telecomunicaciones, elaborada en conjunto con la *ITU* (*International Telecommunication Union* – Unión Internacional de Telecomunicaciones). La Norma determina los requisitos para establecer, implantar, poner en funcionamiento, controlar, revisar, mantener y mejorar un *SGSI,* contextualizado con base en las amenazas y riesgos identificados para cada organización. En otras palabras, esta norma específica los requisitos para la implantación de los controles de seguridad que satisfagan las necesidades de las organizaciones en materia de seguridad de la información.

La última actualización de esta norma fue “publicada en Octubre de 2013, es la norma principal de la serie *ISO/IEC 27000*, contiene los requisitos para poner en funcionamiento el sistema de gestión de seguridad de la información (SGSI). Tiene su origen en la *BS 7799-2:2002* y es la norma con arreglo a la cual se certifican por auditores externos los SGSI de las organizaciones. Sustituye a la BS 7799-2,

habiéndose establecido unas condiciones de transición para aquellas empresas certificadas en esta última" [44].

Beneficios de la Norma ISO/IEC 27001:2013

El beneficio principal de la aplicación de esta norma radica en que las organizaciones certificadas demuestran un amplio compromiso con la gestión de la seguridad de la información, en consecuencia, las organizaciones adquieren diversas ventajas, las cuales se explican a continuación.

a) Ventaja competitiva: cada vez son más las compañías que requieren la certificación según la norma *ISO/IEC 27001* como un requisito indispensable para proveer un servicio. Contar con esta certificación permite hacer una declaración pública de la capacidad que tiene la organización para gestionar la información.

b) Ventaja propia: La minimización de riesgos es un factor tangible en las organizaciones que adoptan esta norma, pues en el proceso de certificación se implementan mecanismos, procesos y acciones que reducen significativamente el riesgo de materialización de las amenazas sobre los activos de la organización.

Otros beneficios inherentes a la adopción de la norma son:

- Inventario real de los activos disponibles en la organización.
- Aumento de la confianza de la comunidad objetivo.
- Mejora la imagen de la organización, favorece la competitividad.
- "Demuestra el compromiso de la directiva con la seguridad de la información" [45].
- Reglas claras para el personal que integra la organización.
- Reducción de costos, mejora de los procesos y servicio.
- Disposición de planes de contingencia ante incidentes de diversos orígenes.
- Facilidad para la integración con otras normas, como la ISO 90001.

Modelo PDCA

"El concepto de ciclo *PDCA* fue inicialmente desarrollado por *Walter Shewhart,* un pionero de la estadística, en la década de 1930 desarrolló el Control de Proceso Estadístico en los Laboratorios *Bell* de *EE.UU*. (ver Figura 2.5). A este ciclo a menudo se le conoce como el ciclo de *Shewhart*, fue asumido y promovido de manera eficaz a partir de 1950 por *W. Edwards Deming* en aquellos tiempos una autoridad en materia

44 La serie 27000, disponible en http://www.iso27000.es/download/doc_iso27000_all.pdf, consultado 19-04-2014.

45 Beneficios del ISO/IEC 27001, disponible en http://www.bsigroup.es/certificacion-y-auditoria/Sistemas-de-gestion/estandares-esquemas/Seguridad-de-la-Informacion-ISOIEC27001/, consultado 21-04-2014.

de gestión de calidad, en consecuencia el ciclo fue más conocido como el ciclo o modelo *Deming*" [46].

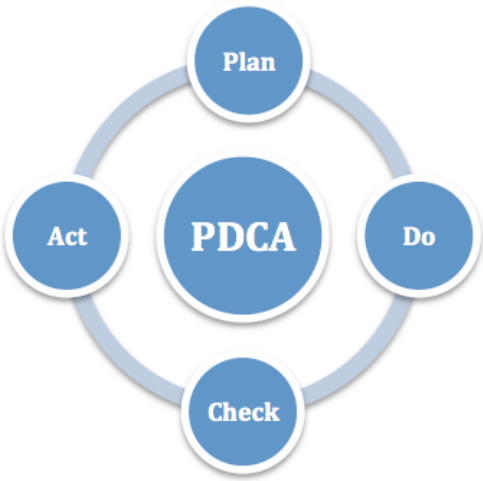


Figura 2.5 - Ciclo PDCA.

El modelo *Plan-Do-Check-Act*, ampliamente conocido como ciclo de *Deming*, constituye la base para todos los procesos de mejora continua. El ciclo es parte fundamental del *SGSI,* se compone de cuatro fases, las cuales para su correcta implementación requieren un compromiso serio de la dirección, una adecuada planificación, definición de fechas y responsabilidades.

a) Fase de Planificación (*Plan*)

En esta fase se debe estudiar y analizar el proceso de gestión de la información a fin de definir qué aspectos pueden cambiarse y mejorarse. En esencia se deben identificar los recursos y controles necesarios para reducir al máximo los riesgos, sin embargo, hay riesgos residuales que tendrán que ser aceptados por la organización.

Las actividades generales que se deben realizar en esta fase son:

- Definir el alcance del *SGSI*.
- "Identificar el proceso que se quiere mejorar". [47]
- Definir la política de seguridad.
- Definir metodología de análisis de riesgos.
- Análisis y evaluación de riesgos.
- Recopilar los datos: es necesario investigar todos los factores involucrados en el proceso que se quiere mejorar.
- Inventario de activos.

---

[46] PDCA Cycle (n.d), disponible en http://www.hci.com.au/hcisite3/toolkit/pdcacycl.htm, consultado 22-04-2014.

[47] Círculo de Deming PDCA (2012), disponible en http://www.empresasandalucia.com/circulo-de-deming-pdca-plan-do-check-act-planificar-hacer-verificar-actuar/ , consultado 22-04-2014.

- Identificar las amenazas y vulnerabilidades a la información.
- Identificar impactos sobre la información.
- Seleccionar los controles que serán implementados en el *SGSI*.
- Elaborar los pronósticos con base en la información recabada en el punto anterior, lo cual debe permitir la predicción de resultados ante diversos factores involucrados en el proceso.
- Planificar los cambios. Se deben definir y planificar las acciones o cambios a fin de mejorar los puntos débiles de la organización.

b) Fase de Ejecución (*Do*)

En esta fase se efectúan las acciones previstas y dispuestas para el cambio, según lo dispuesto en la fase de planificación. En esta fase ya se implementa y gestiona el *SGSI* con base en las políticas, controles, procesos y procedimientos definidos.

Las actividades generales que se deben realizar en esta fase son:

- Definir el plan de tratamiento de riesgos.
- Formación, capacitación y concientización del personal a quien van dirigidas. las soluciones y acciones definidas a favor de la mejora.
- "Implementación de controles de seguridad seleccionados en la fase anterior". [48]
- Verificar las acciones correctivas definidas en la fase de Planeación.
- Operar el *SGSI.*

c) Fase de Verificación (*Check*)

Una vez que se llevaron a cabo todos aquellos cambios planificados en el proceso, éstos deben ser verificados y comparar los resultados obtenidos con los previstos, es decir en "esta fase se evalúa la eficacia de los controles que se implementaron" [49]. Para lo cual es necesario tener bien definido qué se requiere controlar, con qué periodicidad y cómo se piensa realizar, todo ello con la finalidad de tener registros (evidencia) que permita verificar el correcto o deficiente funcionamiento del *SGSI*.

Las actividades generales que se deben realizar en esta fase son:

- Revisar el *SGSI.*
- Medir eficacia de los controles.
- Revisar riesgos residuales.
- Realización de auditorías internas para saber el estado actual del *SGSI.*
- Realizar el registro de acciones y eventos.

48 INTENCO (CERT), Modelo PDCA (n.d), disponible en http://cert.inteco.es/Formacion/SGSI/Conceptos_Basicos/Modelos_PDCA_SGSI/., consultado 18-02-2012

49 Ibídem, Modelo PDCA.

d) Fase de Actuar (*Act*)

Las modificaciones detectadas en la fase anterior se efectúan en esta fase a fin de implementar mejoras al proceso. Es necesario mencionar que una vez cumplidos los objetivos para el proceso, se debe comenzar otro ciclo *PDCA*, debido a que la seguridad de la información es también un proceso continuo.

Las actividades generales que se deben realizar en esta fase son:

- Implementar mejoras al *SGSI.*
- Implementar acciones correctivas y preventivas.
- Verificar eficacia de las acciones anteriores.

2.3.3 El ISO/IEC 27002

"Es un estándar para la seguridad de la información, liberado por primera vez en el año 2000 como *ISO/IEC 17799:2000* con el nombre de *Information Technology – Security Techniques – Code of Practice for Information Security Management*, posteriormente fue revisado y actualizado en el año del 2013" [50]. Cabe aclarar que el *ISO/IEC 17799:2000* tiene su origen en el *BSI* (*British Standards Institution* – Institución de Estándares Británica*)* 7799-1 liberado en 1995.

El *ISO/IEC 27002:2013* contiene una guía de buenas prácticas que describe los controles recomendables en cuanto a seguridad de la información dirigido a las organizaciones interesadas o responsables de implementar un *SGSI*, por tanto no es un estándar certificable.

2.3.3.1 Estructura

El estándar internacional *ISO/IEC 27002* va orientado a la seguridad de la información en las empresas u organizaciones, de modo que las probabilidades de ser afectados por robo, daño o pérdida de información, se minimicen al máximo. El hecho de que una organización adopte la totalidad de los controles recomendados en este estándar, no garantiza que la organización sufra de algún incidente relacionado con la seguridad de la información, lo posible es mitigar las probabilidades de que las amenazas se materialicen sobre los activos.

La guía se basa en 114 controles agrupados en 14 dominios. A continuación se describen cada uno de los 14 dominios obtenidos de la Norma *ISO/IEC 27002:2013*[51]:

I. Política de seguridad.

50 IT news, ISO/IEC 17799 (n.d), disponible en http://itnews.ec/marco/000130.aspx, consultado 22-04-2014.

51 ISO/IEC 17799. Information technology – Security techniques – Code of practice for information security management, Second edition 2005-06-15.

Su propósito es proveer a la gerencia un documento de la política de seguridad de la información, el cual debe contener políticas organizacionales claras y bien definidas que le permitan el control sobre las acciones realizadas en materia de seguridad de la información.

II. Organización de la seguridad de la información.

La organización de la seguridad de la información se divide en 2 aspectos: organización interna y organización externa. La primera tiene como objetivo manejar la seguridad de la información al interior de la organización. El segundo es el aspecto que se enfoca en la organización de la seguridad por terceros (clientes, proveedores, colaboradores, etcétera).

III. Seguridad de recursos humanos.

El objetivo es concientizar a empleados y terceros de su responsabilidad para con los activos a su resguardo. Para lo cual es necesario definir roles y responsabilidades de cada empleado durante el empleo y al término, así como los límites de cada uno con respecto al acceso y manipulación de la información.

IV. Gestión de activos.

Este dominio contempla la asignación de responsabilidades y clasificación de cada uno de los activos de la organización, por tanto, se debe tener un inventario actualizado de todos los activos. También se debe clasificar la información con base en su criticidad, por ejemplo, en restringida, secreta, ultra secreta.

V. Control de acceso.

Se deben implementar medidas y mecanismos que controlen el acceso a la información y sistemas de información y comunicación. Se deben implementar controles a la red, sistemas operativos, aplicaciones, y para ello deben existir registros y bitácoras de acceso.

VI. Cifrado.

Garantizar un uso adecuado y eficaz de la criptografía para proteger la confidencialidad, autenticidad e integridad de la información.

VII. Seguridad física y ambiental.

Contempla la seguridad en las áreas y la seguridad en los equipos. Se debe tener un perímetro de seguridad física que cuente con los mecanismos de seguridad necesarios para salvaguardar la información y equipos de procesamiento de datos. En cuanto a la seguridad ambiental se debe controlar la temperatura de los equipos de información y comunicación de modo que no se ponga en riesgo la seguridad de la información.

VIII. Seguridad de las operaciones.

Asegurar la operación correcta y segura de cada uno de los procesos, equipos de información y comunicación. En este dominio también se contempla la gestión de todos los cambios relacionados con documentación referente a los procesos y procedimientos de la organización.

IX. Seguridad en las telecomunicaciones.

Garantizar la protección de la información en las redes de datos y la facilitación del procesamiento de la misma.

X. Adquisición, desarrollo y mantenimiento de los sistemas de información.

Consiste en la implementación de mecanismos y medidas seguridad para adquirir nuevos equipos y sistemas de información.

XI. Relaciones con los proveedores.

Proteger los activos de la organización a los cuales tienen acceso los proveedores.

XII. Gestión de incidentes en la seguridad de la información.

Se encarga de que sea aplicado un enfoque consistente y efectivo acerca de cómo la organización dará gestión de los incidentes en la seguridad de la información.

XIII. Aspectos de seguridad de la información en la gestión de la continuidad del negocio.

Contempla el desarrollo del Plan de Continuidad del Negocio que garantice el restablecimiento de las operaciones básicas de la organización en el menor tiempo posible ante un incidente de seguridad. Estos planes no deben ser estáticos, sino sometidos a pruebas, mantenimiento y reevaluaciones a fin de mantenerlos actualizados y en mejora continua.

XIV. Cumplimiento.

Se deben definir explícitamente, documentar y actualizar todos los requisitos legales para cada sistema de información y para la organización en general, tales como derechos de propiedad intelectual, confidencialidad de la información, control de auditorías. También se contemplan aspectos relacionados con el cumplimiento de las políticas y estándares de seguridad de la información.

2.4 Mapeo de controles del ISO 27002 a equipos de cómputo

A continuación se listan los 114 controles de los 14 dominios descritos en el *ISO/IEC* 27002:2013 que se deben tomar en cuenta, para certificar los procesos críticos de las instituciones que así lo requieran. De acuerdo al alcance de esta tesis, se consideraran 36 controles para que se implementen en el *hardening* sobre el sistema operativo de un equipo de cómputo.

En la Tabla 2.2 en la última columna se indica en color rojo y con las siglas NA (No aplica), los controles que dada la definición y objetivo del control no pueden ser cubiertos con un *hardening* sobre un sistema operativo de un equipo de cómputo. Los controles marcados con color verde y con las siglas AP (Aplica) son aquellos controles que sí pueden ser cubiertos por dicho *hardening*.

No aplica = NA Aplica = AP

Tabla 2.2 Controles ISO/IEC 27002

CONTROLES ISO/IEC 27002

A.5 Políticas de seguridad de la información		
A. 5.1 Orientación de la dirección para la gestión de seguridad de la información.	A.5.1.1 Políticas para la seguridad de la información.	NA
	A.5.1.2 Revisión de las políticas para la seguridad de la información.	NA
A.6 Organización de la seguridad de la información		
A.6.1 Organización interna	A.6.1.1 Roles y responsabilidades para la seguridad de la información.	NA
	A.6.1.2 Separación de tareas.	NA
	A.6.1.3 Contacto con las autoridades.	NA
	A.6.1.4 Contacto con grupos de interés especial.	NA
	A.6.1.5 Seguridad de la información en la gestión de proyectos.	NA
A.6.2 Dispositivos móviles y teletrabajo	A.6.2.1 Política para dispositivos móviles.	NA
	A.6.2.2 Teletrabajo.	NA
A.7 Seguridad de los recursos humanos		
A.7.1 Antes de la contratación	A.7.1.1 Investigación de antecedentes.	NA
	A.7.1.2 Términos y condiciones de contratación.	NA
A.7.2 Durante la contratación	A.7.2.1 Responsabilidades de la dirección.	NA
	A.7.2.2 Concientización educación y capacitación en seguridad de la información.	NA
	A.7.2.3 Proceso disciplinario.	NA
A.7.3 Cese o cambio de puesto de trabajo	7.3.1 Cese o cambio de puesto de trabajo.	NA
A.8 Gestión de Activos		
A.8.1 Responsabilidad sobre los activos	A.8.1.1 Inventario de activos.	NA
	A.8.1.2 Propiedad de los activos.	NA
	A.8.1.3 Uso aceptable de los activos.	NA
	A.8.1.4 Devolución de los activos.	AP
A.8.2 Clasificación de la información	A.8.2.1 Directrices de clasificación.	NA
	A.8.2.2 Etiquetado y manipulación de la información.	NA
	A.8.2.3 Manipulación de los activos.	NA
A.8.3 Manejo de los medios de almacenamientos	A.8.3.1 Gestión de medios removibles.	NA
	A.8.3.2 Eliminación de medios.	AP
	A.8.3.3 Medios físicos en tránsito.	AP
A.9 Control de accesos		
A.9.1 Requisitos de negocio para el control de accesos	A.9.1.1 Política de control de accesos.	NA
	A.9.1.2 Control de acceso a las redes y servicios asociados.	AP
A.9.2 Gestión de acceso de usuario	A.9.2.1 Gestión de altas/bajas en el registro de usuarios.	AP
	A.9.2.2 Gestión de los derechos de acceso asignados a usuarios.	AP
	A.9.2.3 Gestión de los derechos de	AP

	acceso con privilegios especiales.	
	A.9.2.4 Gestión de información confidencial de autenticación de usuarios.	AP
	A.9.2.5 Revisión de los derechos de acceso de los usuarios.	NA
	A.9.2.6 Retirada o adaptación de los derechos de acceso.	AP
A.9.3 Responsabilidades del usuario	A.9.3.1 Uso de información confidencial para la autenticación.	AP
A.9.4 Control de acceso a sistemas y aplicaciones	9.4.1 Restricción del acceso a la información.	AP
	9.4.2 Procedimientos seguros de inicio de sesión.	AP
	9.4.3 Gestión de contraseñas de usuario.	AP
	9.4.4 Uso de herramientas de administración de sistemas.	AP
	9.4.5 Control de acceso al código fuente de los programas.	AP
A.10 Cifrado		
A.10.1 Controles criptográficos	A.10.1.1 Política de uso de los controles criptográficos.	NA
	A.10.1.2 Gestión de claves.	NA
A.11 Seguridad Física y ambiental		
A.11.1 Áreas seguras	A.11.1.1 Perímetro de seguridad física.	NA
	A.11.1.2 Controles de acceso físico.	NA
	A.11.1.3 Seguridad de oficinas, recintos e instalaciones.	NA
	A. 11.1.4 Protección contra las amenazas externas y ambientales.	NA
	A. 11.1.5 El trabajo en áreas seguras.	NA
	A. 11.1.6 Áreas de acceso público, carga y descarga.	NA
A.11.2 Seguridad de los equipos	A.11.2.1 Emplazamiento y protección de equipos.	NA
	A.11.2.2 Instalaciones de suministro.	NA
	A.11.2.3 Seguridad del cableado.	NA
	A.11.2.4 Mantenimiento de los equipos.	AP
	A.11.2.5 Salida de activos fuera de las dependencias de la empresa.	AP
	A.11.2.6 Seguridad de los equipos y activos fuera de las instalaciones.	AP
	A.11.2.7 Reutilización o retirada segura de dispositivos de almacenamiento.	AP
	A.11.2.8 Equipo informático de usuario desatendido.	AP
	A.11.2.9 Política de escritorio pantalla limpia.	AP
A.12 Seguridad de las operaciones		
A.12.1 Responsabilidades y procedimientos de operación	A.12.1.1 Documentación de procedimientos de operación.	NA

	A.12.1.2 Gestión de cambios.	NA
	A.12.1.3 Gestión de capacidades.	AP
	A.12.1.4 Separación de entornos de desarrollo, prueba y producción.	NA
A.12.2 Protección contra código malicioso	A.12.2.1 Controles contra código malicioso.	AP
A.12.3 Copias de seguridad	A.12.3.1 Copias de seguridad de la información.	AP
A.12.4 Registro de actividad y supervisión	A.12.4.1 Registro y gestión de eventos de actividad.	AP
	A.12.4.2 Protección de los registros de información.	AP
	A.12.4 3 Registros de actividad del administrador y operador del sistema.	AP
	A.12.4 4 Sincronización de relojes.	AP
A.12.5 Control de software operacional	A.12.5.1 Instalación del software en sistemas en producción.	AP
A.12.6 Gestión de la vulnerabilidad técnica	A.12.6.1 Gestión de las vulnerabilidades técnicas.	AP
	A. 12.6.2 Restricciones en la instalación de software.	AP
A.12.7 Consideraciones de las auditorías de los sistemas de información	A. 12.7.1 Controles de auditoría de los sistemas de información.	AP
A.13 Seguridad en las telecomunicaciones		
A.13.1 Gestión de la seguridad en las redes	A.13.1.1 Controles de red.	NA
	A.13.1.2 Mecanismos de seguridad asociados a servicios en red.	NA
	A.13.1.3 Segregación de redes.	NA
A.13.2 Intercambio de información con partes externas	A.13.2.1 Políticas y procedimientos de intercambio de información.	AP
	A.13.2.2 Acuerdos de intercambio.	NA
	A.13.2.3 Mensajería electrónica.	NA
	A.13.2.4 Acuerdos de confidencialidad o de no divulgación.	NA
A.14 Adquisición desarrollo y mantenimiento de los sistemas de información		
A.14.1 Requisitos de seguridad de los sistemas de información	A.14.1.1 Análisis y especificación de los requisitos de seguridad de la información.	NA
	A.14.1.2 Seguridad de las comunicaciones en servicios accesibles por redes públicas.	NA
	A.14.1.3 Protección de las transacciones por redes telemáticas.	NA
A.14.2 Seguridad en los procesos de desarrollo y soporte	A.14.2.1 Política de desarrollo seguro de software.	NA
	A.14.2.2 Procedimientos de control de cambios en los sistemas.	NA
	A.14.2.3 Revisión técnica de las aplicaciones tras efectuar cambios en el sistema operativo.	NA
	A.14.2.4 Restricciones a los cambios en los paquetes de software.	NA

	A.14.2.5 Principios de construcción de los sistemas seguros.	NA
	A.14.2.6 Seguridad en entornos de desarrollo.	AP
	A.14.2.7 Desarrollo de software por terceros.	NA
	A.14.2.8 Pruebas de funcionalidad durante el desarrollo de los sistemas.	NA
	A.14.2.9 Pruebas de aceptación.	NA
A.14.3.Datos de prueba	A.14.3.1 Protección de los datos utilizados en pruebas.	NA
A.15 Relaciones con los proveedores		
A.15.1 Seguridad de la información en las relaciones con proveedores	A.15.1.1 Política de seguridad de la información para proveedores.	NA
	A.15.1.2 Tratamiento de seguridad dentro de acuerdos con proveedores.	NA
	A.15.1.3 Cadena de suministro en tecnologías de la información y comunicaciones.	NA
A.15.2 Gestión de la prestación del servicio por proveedores	A.15.2.1 Supervisión y revisión de los servicios prestados por terceros.	NA
	A.15.2.2 Gestión de cambios en los servicios prestados por terceros.	NA
A.16 Gestión de incidentes en la seguridad de la información		
A.16.1 Gestión de incidentes de seguridad de la información y mejoras	A.16.1.1 Responsabilidades y procedimientos.	NA
	A.16.1.2 Notificación de los eventos de seguridad de la información.	AP
	A.16.1.3 Notificación de puntos débiles de la seguridad.	AP
	A.16.1.4 Valoración de eventos de seguridad de la información y toma de decisiones.	NA
	A.16.1.5 Respuesta a los incidentes de seguridad.	NA
	A.16.1.6 Aprendizaje de los incidentes de seguridad de la información.	NA
	A.16.1.7 Recopilación de evidencias.	NA
A.17 Aspectos de seguridad de la información en la gestión de la continuidad del negocio		
A.17.1. Continuidad de la seguridad de la información	A.17.1.1 Planificación de la continuidad de la seguridad de la información.	NA
	A.17.1.2 Implantación de la continuidad de la seguridad de la información.	NA
	A.17.1.3 Verificación, revisión y evaluación de la continuidad de la seguridad de la información.	NA
A.17.2. Redundancias	A.17.2.1 Disponibilidad de instalaciones para el procesamiento de la información.	NA
A.18 Cumplimiento		
A.18.1 Cumplimiento de los requisitos legales y contractuales	A.18.1.1 Identificación de la legislación aplicable.	NA
	A.18.1.2 Derechos de propiedad	NA

	intelectual (DPI).	
	A.18.1.3 Protección de los registros de la organización.	NA
	A.18.1.4 Protección de datos y privacidad de la información personal.	NA
	A.18.1.5 Regulación de los controles criptográficos.	NA
A.18.2.Revisiones de la seguridad de la información	A.18.2.1 Revisión independiente de la seguridad de la información.	NA
	A.18.2.2 Cumplimiento de las políticas y normas de seguridad.	NA
	A.18.2.3 Comprobación del cumplimiento.	NA

Es importante mencionar que no se debe confundir el *ISO/IEC 27001:2013* y el *ISO/IEC 27002*:2013, a pesar de que ambos estándares refieren a dichos controles de seguridad, la diferencia radica en el alcance de cada uno.

Por un lado el *ISO/IEC 27002:2013* es un estándar de buenas prácticas, que expone los objetivos de seguridad, las consideraciones o controles que se deben tener en cuenta, así como sugerencias, sin considerar algún tipo de priorización entre sus controles.

En cambio el *ISO/IEC 27001:2013* contiene las especificaciones a seguir por parte de las organizaciones que deseen certificarse bajo este estándar, así como el modelo para implementar un Sistema de Gestión de Seguridad de la Información.

Es importante mencionar que el listado de objetivos de control y los controles del estándar 27002 están incluidos como anexo en el 27001 y no aparecen en el cuerpo del estándar 27001 ya que no son el objetivo del mismo.

El objetivo principal del *ISO/IEC 27001:2013,* es que los riesgos sean analizados y gestionados, que exista una planificación, implementación, revisión y mejora continua de la seguridad. Por tanto, para implementar un *SGSI* no es necesario implementar los 114 controles dispuestos en el *ISO/IEC 27002:2013*, es decir, se deben implementar sólo los controles con base en las posibilidades y necesidades obtenidas una vez hecho el análisis de riesgos de la organización dispuesto en el *ISO/IEC 27001:2013*.

Capítulo 3

Equipo de Respuesta a Incidentes de Cómputo (*Computer Emergency Response Team* - CERT)

3.1 Definición de CERT

Un *CERT*, por sus siglas en inglés, es un Equipo de Respuesta ante Emergencias en Cómputo (*Computer Emergency Response Team*), mejor conocido como Equipo de Respuesta a Incidentes de Cómputo, cabe destacar que también se consideran otros dispositivos o herramientas tecnológicas, es decir, no se basan únicamente en incidentes de cómputo, por tal motivo también es conocido como Equipo de Respuesta a Incidentes Informáticos.

Está conformado por un equipo de profesionales en tecnologías de la información que brindan servicios de respuesta a incidentes y de gestión de seguridad en cómputo, su propósito es utilizar sus conocimientos tecnológicos para dar respuesta a problemas de seguridad en informática que afecten a los sistemas críticos de información de su circunscripción, así como implementar campañas y mecanismos de difusión que eleven la conciencia colectiva en materia de seguridad informática, con el propósito de mitigar el riesgo de ataques cibernéticos que afecten a la comunidad objetivo (*Constituency*) a la cual se le provee el servicio de protección.

Cualquier grupo que se autodenomina un CSIRT debe reaccionar a incidentes de seguridad reportados, así como a las amenazas informáticas de su comunidad[52]*.*

Es importante aclarar que un *CERT* y un *CSIRT* tiene las mismas funciones, ambos son un Equipo de Respuesta a Incidentes de Seguridad Informática, la diferencia radica en que las siglas de *CERT* son una marca registrada y si un equipo de respuesta a incidentes desea usarlas debe pagar una membresía anual al dueño de la marca, ante esta situación muchos equipos de respuesta a incidentes optaron por usar las siglas de *CSIRT* (*Computer Security Incident Response Team* – Equipo de Respuesta a Incidentes de Seguridad en Cómputo) que no son una marca registrada.

3.1.1 Orígenes del CERT

El primer ataque que afectó a Internet sucedió el 2 de noviembre de 1988, cuando un *malware* de tipo gusano conocido como "*Morris",* afectó a cerca del 10% de todos los servidores de Internet de esa época. La reacción a este incidente fue aislada y sin coordinación, además dejó ver la falta de mecanismos de seguridad en Internet y la carencia de entidades dedicadas a recibir los reportes de incidentes y dar respuesta a fin de salvaguardar la infraestructura crítica de información, es así como surge el primer *CERT* en la *Universidad de Carnegie Mellon* en Estados Unidos, ver Figura 3.1.

Ante el crecimiento acelerado de Internet, pronto fue claro que un solo *CERT* no sería suficiente para dar respuesta a todos los incidentes generados, por lo que en los años subsecuentes aumentó el número de equipos de respuesta a incidentes, de manera que surgió la necesidad de coordinar a los diferentes *CERTs* o *CSIRTs* de todo el mundo para evitar la respuesta aislada y la duplicación de esfuerzos, es así como se crean foros de equipos de respuesta a incidentes a través de los cuales se comparte

52 Proyecto Amparo, Manual básico de Gestión de Incidentes de Seguridad Informática, (2011), p 15.

información sobre vulnerabilidades, ataques informáticos de impacto global, herramientas y técnicas utilizadas para la contención y erradicación de los incidentes.

Figura 3.1 - Logo del primer CERT.

Un *CERT* o *CSIRT* debe establecer a qué comunidad objetivo (*Constituency*) pertenece, difundir los servicios de seguridad que ofrece y bajo qué condiciones, es importante que cada miembro de la comunidad comprenda los alcances y posibilidades de su equipo de respuesta, para saber qué se espera recibir ante un incidente de seguridad, de la misma forma los miembros de la comunidad objetivo, deben conocer qué se espera de ellos para que puedan recibir los servicios de seguridad de parte de su equipo de respuesta.

La mayoría de los incidentes de seguridad informática se originan fuera de los límites de la comunidad objetivo y afectan a sistemas de información en el interior, otros se originan dentro del perímetro de la comunidad y afectan a sistemas de información ubicados fuera del perímetro de protección, por tanto es vital y fundamental que cada *CERT* difunda a qué comunidad proveerá los servicios de seguridad informática.

Tal como lo expresa el Manual del Proyecto Amparo "Un equipo de respuesta a incidentes, tiene como principal objetivo proteger infraestructuras críticas de la información, su alcance se limita por el tipo de servicio el cual le fue confinado para proteger" [53]. Un *CERT* o *CSIRT* puede brindar servicios de seguridad a cualquiera de las infraestructuras siguientes:

- Gobierno.
- Telecomunicaciones.
- Finanzas.
- Industrias.
- Transporte.
- Energía.
- Suministros de Agua.
- Salud Pública.

Cada una de las infraestructuras críticas mencionadas, utiliza servicios de información las cuales se pueden dividir en cuatro categorías generales:

- Internet: servicios Web, alojamiento, correo electrónico, etcétera (ver Figura 3.2).
- Sistemas de control: *SCADA*, *PCS* y *DCS*.
- Hardware: servidores, equipos personales, teléfonos IP, equipos y elementos de red.
- Software: sistemas operativos, aplicaciones, etcétera.

[53] Proyecto Amparo (2011), Op. cit., p 16.

Figura 3.2 - Servicios por Internet.

3.2 Foros de Equipos de Respuesta a Incidentes

3.2.1 Foros de Respuesta

En 1989, el número de equipos de respuesta a incidentes seguía creciendo, cada uno con propósitos diferentes pero con una meta en común, proveer respuesta ante incidentes de seguridad informática. La interacción entre los equipos tuvo dificultades debido a las diferencias de idioma, zona horaria, normas, leyes, convenios internacionales y la creación de un nuevo gusano llamado "*Wank*" que expuso los problemas de coordinación y comunicación entre los diversos equipos de respuesta a incidentes.

Ante estas dificultades fue creado en 1990 el *FIRST* (*Forum for Incident Response and Security Teams*), un Foro de Equipos de Respuesta a Incidentes y de Seguridad, desde entonces el número de equipos de respuesta a incidentes ha crecido constantemente. Actualmente son 324 equipos distribuidos en 70 países, en la imagen (Figura 3.3) se observan en color verde los países que cuentan con al menos un *CSIRT* establecido, cabe destacar que África es el continente que tiene menos equipos de respuesta a incidentes. [54]

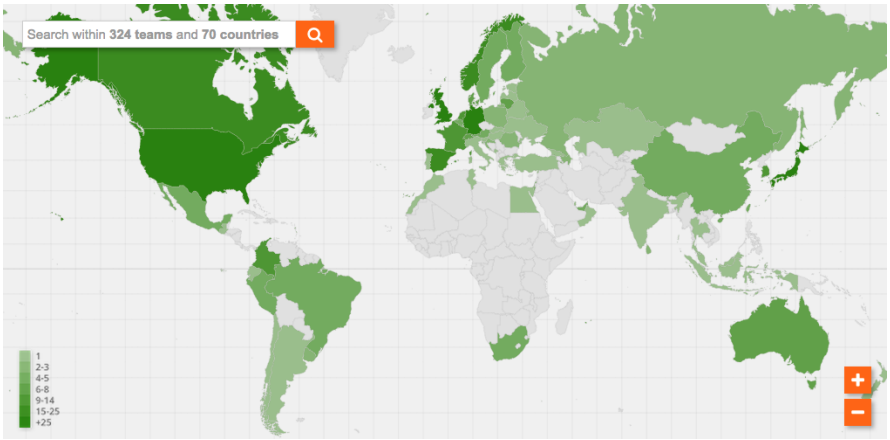


Figura 3.3 - Países que pertenecen a FIRST.

El objetivo de esta organización es fomentar la cooperación y coordinación en la prevención de incidentes, promoviendo el intercambio de información y la colaboración entre todos los equipos de respuesta a incidentes a nivel mundial. En la gráfica se

[54] Members around the world (2015), disponible en http://www.first.org/members/map, consultado 30-01-2015.

puede apreciar que Estados Unidos y Europa representan una tasa de crecimiento mayor respecto al número de *CERTs*[55], ver Figura 3.4.

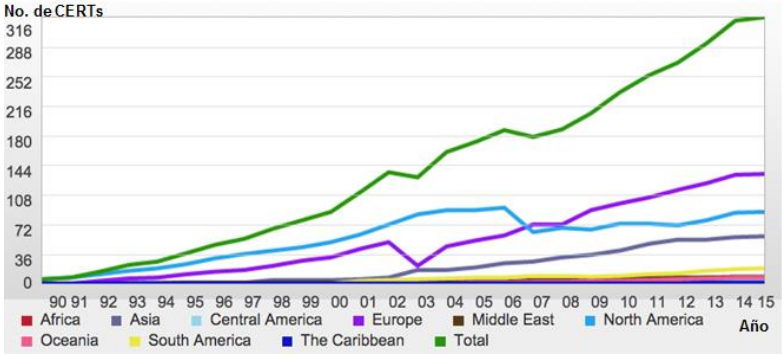


Figura 3.4 - Crecimiento de miembros de FIRST.

Año

3.2.2 Organizaciones de Respuesta a Incidentes

Existen otras organizaciones a nivel mundial que coordinan las acciones de los equipos de respuesta a incidentes: Asociación Trans-Europea de Investigación y Educación de Redes (*Trans-European Research and Education Networking Association*) y el grupo de *CERTs* gubernamentales de Europa (*EGC Group; European Government CERTs – Grupo de CERTs de Gobiernos Europeos*) (Ver tabla 3.1).

Tabla 3.1 Foros de respuesta.

Nombre de la organización	Descripción de la Organización
www.first.org	**FIRST (***Forum for Incident Response and Security Teams - Foro de Equipos de Respuesta a Incidentes y de Seguridad***):** Desde su creación sus miembros han trabajado de forma casi continua en problemas de seguridad relacionados con ataques e incidentes, incluyendo el manejo de vulnerabilidades de seguridad que afectan a la gran mayoría de los millones de sistemas y redes de todo el mundo que están conectados a Internet. Es un punto de reunión para el intercambio de técnicas, buenas prácticas y experiencias relacionadas con la seguridad informática, los miembros de *FIRST* tienen la oportunidad de dar forma a los nuevos estándares desarrollados en *ISO* e *ITU-T*.
www.terena.org	**TERENA:** En colaboración con el *TF-CSIRT* (*Task Force - Collaboration of Incident Response Teams / Fuerza Especial - Colaboración de Respuesta a Incidentes*) ofrece un foro para colaborar, innovar y compartir el conocimiento con el propósito de fomentar la seguridad y desarrollo de la tecnología de Internet, infraestructura y servicios de *TI* (Tecnologías de la Información), a través del intercambio de experiencias y conocimientos, además promueve estándares y procedimientos para responder a incidentes de seguridad informática mediante la reducción de los tiempos y el uso eficiente de los recursos empleados. Coordina a diversos *CERTs* de diferentes países y colabora con 10 empresas transnacionales enfocadas a redes y seguridad: CISCO *Systems*, *Juniper Networks*, *Nokia Siemens Networks*, *ADVA Optical Networking*, entre otros.
www.apcert.org	**APCERT**: por sus siglas en inglés *Asia Pacific Computer*

[55] FISRT History (2015), disponible en http://www.first.org/about/history, consultado 30-01-2015.

APCERT Asia Pacific Computer Emergency Response Team	*Emergency Response Team (*Asia Pacífico - Equipo de Respuesta a Incidentes de Emergencia en Cómputo*),* está conformado por 28 *CSIRTs* de la región de Asia-Pacífico dedicados al manejo y respuesta de los incidentes de seguridad informática. Estos equipos adoptan un enfoque de colaboración para hacer frente a las amenazas electrónicas y ataques informáticos que representen un riesgo para las economías e infraestructuras críticas de la región. Para lo cual sus integrantes intercambian información y tecnología para la elaboración de políticas y mecanismos que permitan mitigar el impacto de los incidentes de seguridad a gran escala en la red. Este comité tiene a su cargo la responsabilidad de: • Mantener una red de contactos con expertos de seguridad informática. • Informar a todos los equipos de la región sobre incidentes de seguridad. • Facilitar el intercambio de información tecnología entre sus miembros.
www.egc-group.org EGC group	**EGC (*Grupo de CERTs de Gobiernos Europeos*):** es un grupo informal de *CSIRTs* gubernamentales que facilita la cooperación en materia de respuesta a incidentes entre sus miembros, basándose en las similitudes de problemas de seguridad que comparten los *CSIRTs* gubernamentales en Europa. Actualmente consta de 13 miembros de diferentes países.

Además de los foros de respuesta a incidentes ya mencionados, también existen otras organizaciones dedicadas al intercambio de información y apoyo en labores de contención de incidentes de seguridad informática. Ver tabla 3.2.

Tabla 3.2 Otras organizaciones de seguridad a nivel mundial

Nombre de la organización	Descripción de la Organización
www.isaccouncil.org MULTI-STATE ISAC	**ISAC** (*Information Sharing and Analysis Centre - Centro de Intercambio y Análisis de Información)***:** Proporciona un foro en el que se comenta e intercambia información relacionada con infraestructura crítica como de telecomunicaciones, energía, banca, finanzas, transporte, sistemas de agua, servicios de emergencia, etcétera.
puck.nether.net/mailman/listinfo/nsp-security nsp europe uncompromised safety and securit	**NSP-Security Forum:** Foro de seguridad conformado por Proveedores del Servicio de Red (*Network Service Providers*) de todo el mundo, en su mayoría está constituido por los responsables de gestionar los incidentes de seguridad en las redes de los *NSP* casi en tiempo real, junto con los responsables de los equipos de cómputo implementados en sus entornos locales, los miembros de este foro pueden colaborar con otros miembros de cualquier otra parte del mundo en la mitigación de los incidentes en un perímetro muy cercano al origen del incidente.
http://www.team-cymru.org/ TEAM CYMRU COMMUNITY SERVICES	**CYMRU (nombre en tributo a historia de Gales):** Este grupo está dedicado a hacer que Internet sea más seguro, ayuda a las organizaciones a identificar y erradicar los problemas asociados a sus redes de datos.

Los beneficios de pertenecer al foro del *FIRST* son los siguientes:

- Disponibilidad de comunicación a través del foro entre los miembros de los diferentes *CSIRTs* a nivel mundial, lo que facilita la respuesta ante diferentes incidentes de seguridad informática, proporciona diferentes mecanismos de asistencia y comunicación de manera segura como el uso de llaves *PGP* (*Pretty Good Privacy*).
- Anualmente organiza la conferencia para el manejo de incidentes de seguridad y conferencias que tienen como punto central el análisis de las últimas estrategias utilizadas en la prevención y respuesta de incidentes, análisis de vulnerabilidades y aspectos relacionados al tema de seguridad informática.
- Para los equipos que pertenecen al *FIRST* se establecen reuniones cada dos años que tienen como objetivo compartir información sobre el uso de herramientas y aspectos que afectan las operaciones de respuestas a incidentes.
- Los equipos que pertenecen al *FIRST* reciben información fidedigna y de forma pronta respecto a incidentes, además todos sus integrantes reciben total apoyo del *FIRST* para gestionar los incidentes de seguridad, brindan colaboración e intercambio de ideas.

3.3 Funciones principales de un CERT

3.3.1 Catálogo de servicios

El catálogo de servicios que un equipo de respuesta a incidentes puede ofrecer es muy amplio. Los servicios que ofrece un *CERT* o *CSIRT* se pueden clasificar como servicios proactivos, reactivos y de gestión de calidad de la seguridad, todos encaminados a proteger la infraestructura de información de una comunidad objetivo[56], sin embargo, la realidad es que ningún CSIRT brinda todos los servicios, debido a que cada *CSIRT* se encarga de una comunidad objetivo con características, propósitos, alcances y metas diferentes, cada equipo elige sus servicios de acuerdo con las necesidades de su comunidad.

3.3.2 Servicios Proactivos

Este tipo de servicios generalmente proporcionan elementos que ayudan a prevenir y anticipar eventos desfavorables que afecten la seguridad de la información[57]. Estos servicios reducen el número de incidentes a mediano y largo plazo, es decir, si los servicios proactivos son deficientes y limitados, los servicios reactivos irán a la alza, estos dos servicios tienen una relación inversa, algunos de estos servicios pueden ser:

- Monitoreo de riesgos y amenazas.
- Monitoreo y evaluación de la seguridad.
- Auditorías de seguridad y evaluaciones.

---

56 Richard A., Julia H., Pamela D., David W., Lisa R., [CMU/SEI-2010-TR-012].CERT Resilience Management Model versión 1.0, p 20.

57 Proyecto Amparo (2011), Op. cit., p 25.

- Mantenimiento de la seguridad en infraestructura crítica.
- Desarrollo de herramientas y aplicaciones de seguridad.
- Detección de Intrusos.
- Difusión de vulnerabilidades y amenazas relacionadas con la seguridad de la información.
- Concientización del personal en materia de seguridad informática.

3.3.3 Servicios Reactivos

Se realizan cuando un evento desfavorable ocurrió en un sistema de información; un equipo comprometido, robo o pérdida de información, infección por código malicioso (*malware*), vulnerabilidad de software, etcétera. Los servicios reactivos representan la tarea más sustancial e importante de un *CSIRT*.

- Emisión de alertas relacionadas con seguridad de la información.
- Gestión de incidentes de seguridad informática; Detección, *Triage*, Contención, Erradicación, Recuperación y Retroalimentación.
- Análisis de vulnerabilidades.
- Análisis de artefactos.

3.3.4 Servicios de gestión de calidad de la seguridad

Estos servicios por lo general son realizados por el área de Tecnología de la Información (TI) de una organización, basándose en la experiencia y conocimientos del personal del *CSIRT* el cual apoya en el diseño e implementación de estos servicios, así como dar una visión más clara que ayude a mejorar la seguridad general de la organización e identificar riesgos, amenazas y debilidades sobre la infraestructura crítica de información. Estos servicios podrían entrar en la categoría de servicios proactivos debido a que contribuyen indirectamente en la mitigación de riesgos de seguridad.

- Análisis de riesgos.
- Elaboración de Plan de Continuidad del Negocio (*BCP - Business Continuity Planning*) y Plan de Recuperación de Desastres (*DRP - Disaster Recovery Plan*).
- Implementación de estándares de seguridad (*ISO/IEC 27001*).
- Educación y entrenamiento.
- Evaluación o certificación de productos.

3.3.5 AAA (Autenticación, Autorización y Auditoría)

Los servicios descritos anteriormente deben alinearse a la estructura de cada *CSIRT* y a su vez dividir sus tareas en tres grupos: Autenticación, Autorización y Auditoría (AAA).

a) Autenticación

Es un proceso que asegura que algo o alguien es quien dice ser. Las personas o entidades que acrediten su identidad pueden tener acceso a los recursos de cómputo e información que posee un *CSIRT*.

Los sistemas que habitualmente utilizan los humanos para identificar a una persona, como el aspecto físico o la forma de hablar, son demasiado complejos para una computadora; el objetivo de los sistemas de identificación de usuarios no suele ser identificar a una persona, sino autenticar que esa persona es quien dice ser realmente. [58]

En la Tabla 3.3 se ilustra el comparativo de los métodos biométricos más comunes, el estudio fue realizado por el *CERT* de España (*RedIRIS*), también miembro del *FIRST*.

Tabla 3.3 Comparación de métodos biométricos

Comparación de métodos biométricos					
Aspecto	**Ojo (Iris)**	**Ojo (Retina)**	**Huellas dactilares**	**Geometría de la mano**	**Voz**
Fiabilidad	Muy Alta	Muy Alta	Alta	Alta	Alta
Facilidad de uso	Media	Baja	Alta	Alta	Alta
Prevención de ataques	Muy Alta	Muy Alta	Alta	Alta	Media
Aceptación	Media	Media	Media	Alta	Alta
Estabilidad	Alta	Alta	Alta	Alta	Media
Identificación / Autenticación	Ambas	Ambas	Ambas	Autenticación	Autenticación
Estándares	-	-	ANSI / NIST, FBI	-	SVAPI
Interferencias	Uso de anteojos	Irritaciones	Suciedad, heridas	Artritis o reumatismo	Ruido ambiental o resfriados

b) Autorización

Es el proceso de verificación que algo o alguien tiene la autoridad para realizar cierta operación. Las personas o entidades autorizadas pueden tener acceso únicamente a los recursos de cómputo e información necesarios para sus áreas y dominios de trabajo, en definitiva la autorización es un proceso en el cual se relacionan los recursos (bases de datos, páginas, archivos, etcétera) a los que se tiene acceso, con el tipo de acceso o permiso (lectura, modificación, eliminación, creación) sobre ese recurso. En definitiva, determina el conjunto de privilegios de acceso que se tienen como usuario. La autorización se implementa de muchas maneras, una de las más habituales es en el proceso de una aplicación que relaciona los recursos a los que se tiene acceso. En este proceso es común la aplicación de Listas de Control de Acceso *ACL* (*Access Control Lists*) esenciales en la gestión de cualquier política para controlar el acceso a los recursos del sistema.

c) Auditoría

En este proceso se registran los accesos autorizados y no autorizados a los recursos por los usuarios. Por lo general este proceso se implementa en dispositivos de red (*router*) o en sistemas operativos, cabe destacar que esta información deberá estar en continua vigilancia, con esta información se podrán generar estadísticas de acceso, de uso, lo que ayudará a identificar mejoras e implementar políticas para el acceso físico a los recursos e información que poseen los dispositivos o áreas objeto de auditoría de un *CSIRT*.

58 Rediris, (n.d). Autenticación, disponible en http://www.rediris.es/CERT/doc/unixsec/node14.html, consultado el 15-09-2013

3.4 Gestión de incidentes

Los equipos de respuesta a incidentes que se auto nombren como *CSIRT (*equivalente a CERT*)* deben proporcionar como mínimo el servicio de gestión de incidentes de seguridad, debido a que es la tarea más sustancial para cualquier comunidad objetivo.

La gestión de incidentes de seguridad tiene como objetivo atender y resolver cualquier incidente que cause una interrupción en el servicio, de la manera más eficaz y rápida posible[59]. La gestión de incidentes no debe confundirse con la gestión de problemas, pues a diferencia de la última, no se preocupa de encontrar y analizar las causas subyacentes a un determinado incidente sino exclusivamente a restaurar el servicio.

En la respuesta a incidentes existen diversas etapas las cuales se muestran en la Figura 3.5

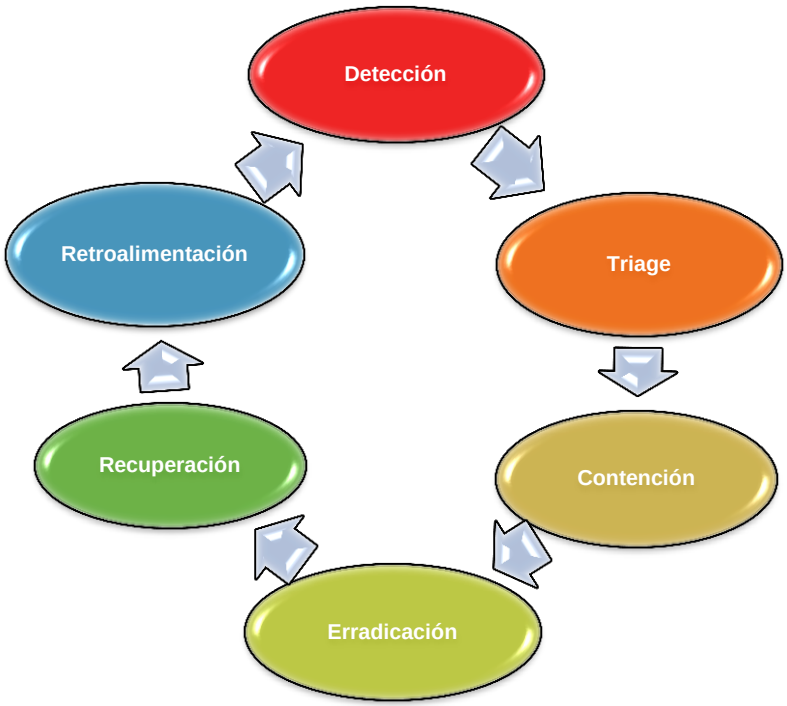


Figura 3.5 - Proceso de atención y respuesta a incidentes de seguridad informática.

a) Detección

Este punto del proceso tiene como objetivo la identificación del incidente, en el que se cerciorará que efectivamente se trata de un incidente de seguridad y no de una mala configuración, falla lógica o física de un sistema de información o comunicación. Es decir, se decide si es sólo un evento o un incidente. Es común que al detectar que un equipo de cómputo o sistema no funciona como se espera que lo haga, se suela decir: “tiene un virus”, “hackearon el equipo”, “la red ya no funciona”, y en consecuencia se emita una alerta sobre un posible incidente de seguridad.

[59] Proyecto Amparo (2011), Op. cit., p 37.

Por lo que en esta etapa se debe seleccionar al personal adecuado para manejar el incidente, quien debe tener conocimiento técnico y tener claras las políticas de manejo de incidentes de la organización, ya que un mal manejo de la información del incidente puede resultar en la pérdida de la evidencia, lo que dificultaría el proceso de manejo del incidente.

Existen diversas señales que se pueden observar y que podrían ser indicativos de que un incidente está ocurriendo o que ya ocurrió, tales como los que se listan a continuación.

- Intentos de acceso al sistema fallidos.
- Modificación de datos sin alguna explicación.
- Nuevas cuentas de usuarios sin explicación.
- Nuevos archivos con nombres extraños.
- Modificaciones inexplicables a archivos propios del sistema.
- Intentos de escritura o cambios en el sistema.
- Alguna alarma de sistemas de detección de intrusos.
- Negación de servicio.
- Caídas del sistema.
- Desempeño pobre del sistema.
- Operación no autorizada de un *sniffer* en la red.
- Intentos de obtención de información por medio de ingeniería social.
- Uso inusual del sistema.
- La mayoría de los incidentes de seguridad ocurren fuera de los horarios de trabajo.

Es posible que en esta etapa se recolecte evidencia de los hechos, sin embargo, se debe tener especial cuidado para no alterar los datos recolectados.

b) Triage

El *triage* es un término en inglés ampliamente utilizado para referirse a las acciones que definen un proceso y los elementos que lo involucran. También involucra las acciones de categorización, correlación, priorización y asignación. Ver Figura 3.6.

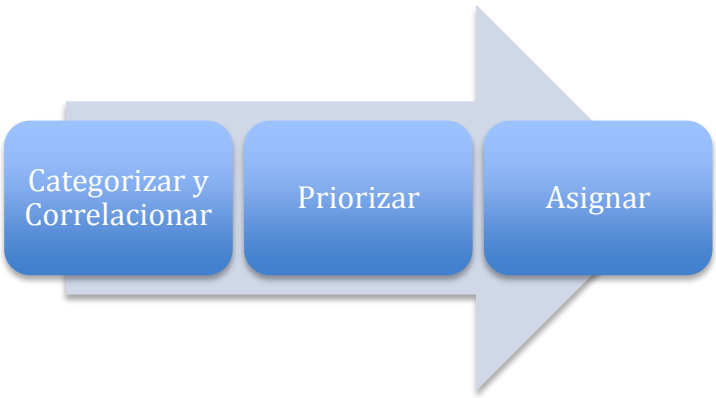


Figura 3.6 - Proceso de Triage.

"El objetivo del triage es asegurar que toda la información destinada al servicio de manejo de incidentes sea canalizada a través de un punto de contacto único, independientemente del método por el cual haya llegado el reporte (por ejemplo,

correo electrónico, fax, teléfono o correo postal), con la finalidad de tener una redistribución y manejo del servicio adecuado"[60].

En esta etapa se analiza toda la evidencia recolectada con la intención identificar el alcance e impacto del incidente, si ocurrió el borrado, la modificación, el copiado o la eliminación de la información, lo que permitirá saber si el incidente se llevó a cabo (vulnerabilidad explotada), es decir, qué pasó y cuáles son las consecuencias de dicho incidente.

En otras palabras, todo incidente detectado primero debe ser categorizado de acuerdo con el tipo de incidente y se debe correlacionar al incidente con otros eventos, es decir, se determina si es un nuevo incidente o es parte de otro, después se debe priorizar, de acuerdo con el impacto sobre los sistemas de información y finalmente se debe asignar un responsable para el seguimiento y resolución del mismo.

Existen dos tipos de *triage*:

- Táctico: Se enfoca en agrupar, categorizar y asignar los reportes con base en un criterio establecido.
- Estratégico: Todo incidente es evaluado de forma minuciosa, de acuerdo con el entorno e impacto en la continuidad del negocio de la instancia afectada.

Es posible que ocurran múltiples incidentes que afecten a una organización, por lo que es necesario determinar un nivel de prioridad para la resolución de cada uno de ellos. El nivel de prioridad se basa esencialmente en tres parámetros: [61]

- Impacto: determina la importancia del incidente dependiendo de cómo éste afecta a los procesos de negocio y del número de usuarios afectados.
- Urgencia: depende del tiempo máximo de demora que acepte la organización afectada para la resolución del problema y el nivel de servicio.
- Criticidad: medida con base en los recursos afectados de forma directa y colateral.

En la Figura 3.7 se muestra el ejemplo de un diagrama de prioridades en función de la urgencia e impacto del incidente [62]. Es preciso decir que cada organización determina el nivel de impacto de acuerdo con el tipo de incidente.

Un ejemplo relacionado con la Figura 3.7, una organización deberá resolver un incidente catalogado como crítico en menos de 2 horas, por lo tanto el incidente será catalogado como alto ya que deberá ser resuelto en menos de 8 horas y así sucesivamente.

---

[60] Moira J., Don S., Klaus-Peter, Georgia K., Robin R., Mark Z. [CMU/SEI-2003-HB-002].Handbook for Computer Security Incident Response Teams (CSIRTs), p 60.

[61] Ibídem. p 95,124, 125, 127.

[62] Proyecto Amparo (2011), Op. cit., p 40.

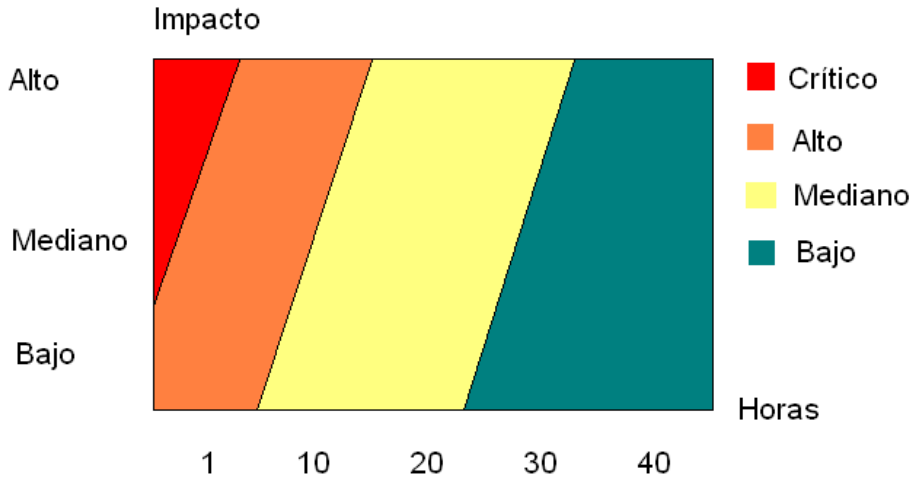


Figura 3.7 - Impacto contra tiempo.

c) Contención

En esta etapa se cruza la línea y se interactúa con el sistema o sistemas involucrados en el incidente. La contención del incidente está en función del análisis previo que se haya realizado (*triage*), este análisis dará la pauta para las acciones implementadas a fin de evitar que el incidente tenga un mayor impacto en la continuidad del negocio y la comunidad objetivo de la organización afectada.

Con frecuencia cuando se gestiona un incidente de seguridad en cómputo, se busca mitigar el impacto o daño en los sistemas de información en el menor tiempo posible, esto puede incluir la desconexión de la fuente eléctrica y de acceso a la red de los equipos involucrados, sin embargo, se debe considerar la criticidad de los servicios que proveen estos equipos y el impacto que causarían al ser desconectados por unos minutos de la red, en algunas ocasiones estas acciones pueden desencadenar conflictos más graves que los que inicialmente el incidente causó.

Cuando el incidente de seguridad tiene repercusiones legales en las organizaciones afectadas, por lo general se retrasan las acciones en la fase de la contención del incidente, esto para reunir evidencias necesarias que permitan iniciar las acciones legales correspondientes contra los responsables del incidente, procurando que la evidencia conserve su integridad y disponibilidad.

Con base en lo descrito en párrafos anteriores, la fase de contención se puede implementar de acuerdo con los siguientes tres escenarios:

- Estrategia de contención: consiste en actuar rápido y con eficacia para limitar al máximo el impacto sobre la continuidad del negocio, cada organización debe definir un riesgo tolerable y aceptable para cada tipo de incidente.
- Contención a corto plazo: se basa en impedir que un incidente genere un daño mayor, sin realizar ningún cambio en el sistema; cambiar el equipo a otra red, *firewall*s, configuraciones de *DNS*.

- Contención a largo plazo: Se enfoca en obtener una imagen bit a bit (idéntica) de los sistemas de almacenamiento involucrados en el incidente, posteriormente se realiza un análisis forense sobre la imagen y no sobre los sistemas, de esta forma se garantiza la disponibilidad de la evidencia.

d) Erradicación

En esta fase se llevan a cabo todas las actividades necesarias para eliminar los archivos o elementos causantes del incidente, se limpia el sistema de artefactos del ataque, en este punto debe ser claro el objetivo del ataque y el vector de infección para lo cual es necesario realizar un análisis en sitio y en los laboratorios del CERT a fin de identificar las características, artefactos, técnicas y herramientas utilizadas para el ataque y de esta forma implementar mecanismos de erradicación eficaces sobre los sistemas afectados.

Ejemplos de artefactos: puertas traseras *shells*, *rootkits* u otros códigos maliciosos (*malware*), además de contenidos maliciosos como kits de *phishing*, cuentas de usuario creadas por los intrusos en el sistema, etcétera. En caso de que el sistema ya no sea confiable y se dude de la eficiencia de las acciones realizadas para limpiar el equipo, será necesario acudir a la copia de seguridad más reciente o a reinstalar el sistema operativo y sus paqueterías.

En esta fase el *hardening* de los sistemas empleados por un CERT para implementar la erradicación se ponen a prueba. Durante esta fase es posible que se tenga que realizar la incautación de los equipos afectados o muestras de código malicioso extraídas de equipos de cómputo afectados por un incidente de seguridad.

En esta fase, el equipo de respuesta a incidentes de seguridad en cómputo (*CERT*) debe utilizar un laboratorio de análisis para recrear el escenario del ataque o bien para analizar algún artefacto en un ambiente controlado a fin de identificar el impacto, comportamiento, interacción con el sistema de archivos, conexiones de red. Si bien se deben tomar precauciones para evitar que las acciones de erradicación no comprometan la seguridad de los sistemas utilizados para el análisis del incidente y a su vez estos sistemas sean usados para afectar la infraestructura tecnológica de un *CERT*, estas precauciones sólo reducen el riesgo de que pueda ocurrir un suceso desafortunado en la red interna de un *CERT*.

En el capítulo 5 se definirán las acciones mínimas que un Equipo de Respuesta a Incidentes debe implementar en el *hardening* de sus equipos con sistema operativo Linux utilizados tanto para la operación como para las investigaciones y análisis de artefactos. A continuación en la figura 3.8 se muestra una propuesta del Proyecto Amparo[63] sobre un diagrama ideal para un *CERT* con respecto a una red segura separada de la organización.

[63] Proyecto Amparo (2011), Op. cit., p 72

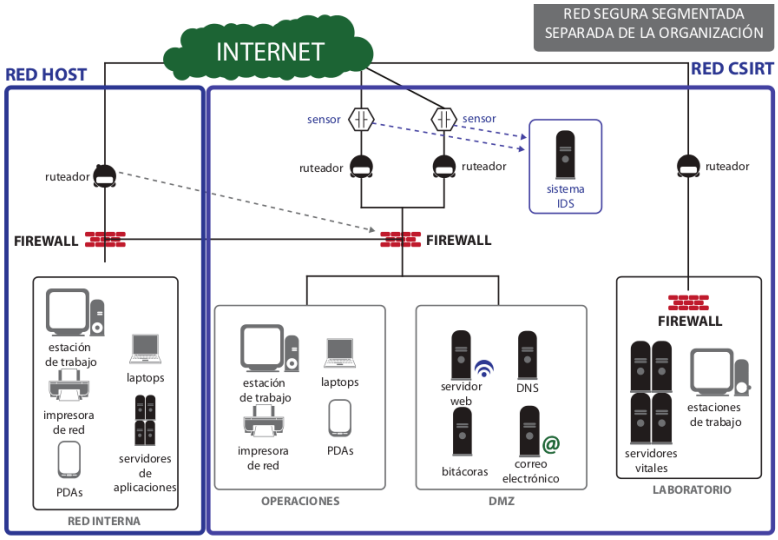


Figura 3.8 - Diagrama de seguridad ideal.

Como se puede ver en el esquema de la Figura 3.8, el laboratorio dispuesto para el análisis de los incidentes de seguridad está totalmente separado de la organización, lo cual reduce el riesgo de que la red interna, *DMZ* o el área de operaciones sea vulnerada. Sin embargo, este diagrama de red es el ideal, ya que no siempre se tienen los recursos y apoyo de la alta dirección para el financiamiento de una infraestructura como la que se plantea. Es por ello que el *hardening* de los sistemas Linux empleados en los *CERTs* es un mecanismo de seguridad compensatorio en estos casos y por tanto es más común que los *CERTs* implementen esquemas como el mostrado en la Figura 3.9[64], en donde el *hardening* de sus equipos empleados en el laboratorio representa un factor clave para evitar la posible propagación de un ciber-ataque en contra de la red interna y servicios proporcionados por el *CERT*.

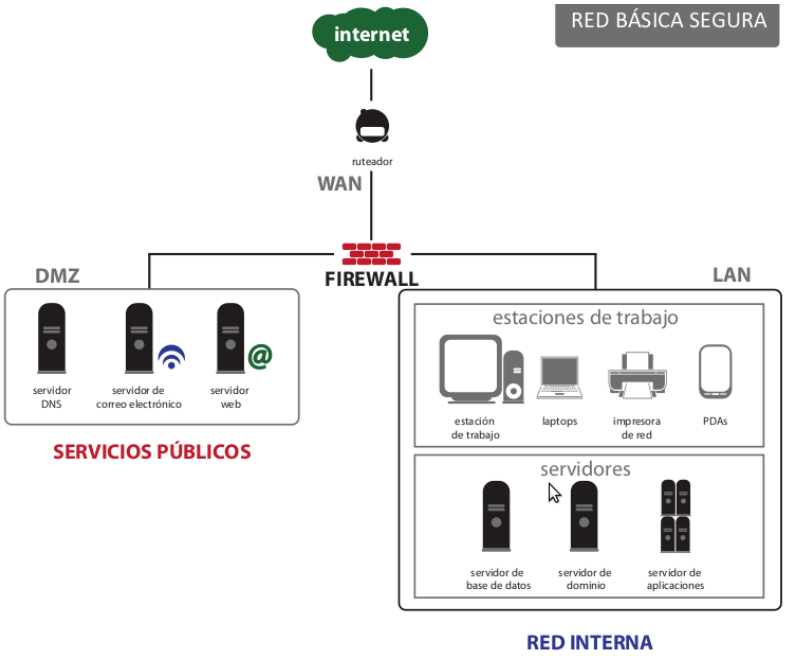


Figura 3.9 - Diagrama básico de seguridad.

64 Proyecto Amparo (2011), Op. cit., p 69

e) Recuperación

En esta fase se procura que los sistemas vuelvan a su funcionamiento normal, es en esta etapa donde entra en operación el *hardening* de los equipos y redes de datos a fin de robustecer la seguridad de los equipos y mitigar el riesgo de futuros incidentes de seguridad.

Los respaldos del sistema realizados con anterioridad serán de gran utilidad para regresar el sistema afectado a un estado confiable y seguro para la *continuidad del negocio* y seguridad de la información. En algunos casos se implementarán mecanismos de seguridad como: *Firewalls*, *Intrusion Prevention System* (*IPS* - Sistema de Prevención de Intrusos), instalación y aplicación de actualizaciones de seguridad, cambios de contraseñas, en casos extremos será necesaria la migración del sistema operativo a una versión más segura y confiable.

Por último, se deben realizar pruebas de *Pen-test* (*Penetration Test*), estas pruebas de penetración son un conjunto de técnicas utilizadas para evaluar la seguridad de redes, sistemas de computación y aplicaciones relacionadas con las tecnologías de la información.

Una vez que el sistema ha vuelto a sus actividades normales, se debe continuar con las actividades de monitoreo constante para identificar anomalías en bitácoras, *IPS, IDS*, verificación de integridad de archivos, cuentas del sistema, configuraciones, procesos, etcétera.

f) Retroalimentación

Esta fase se refiere a las lecciones aprendidas en el ciclo de vida del incidente, es decir, a la retroalimentación de los incidentes ocurridos. Para ello, es indispensable que una vez resuelto el incidente, se documente de forma clara y precisa cada uno de los pasos implementados para su gestión.

La ventaja de documentar las acciones ejecutadas en la resolución satisfactoria del incidente de seguridad, sirve de apoyo para la atención de incidentes similares en un futuro, el personal a cargo del manejo de incidentes puede guiarse de las acciones realizadas con anterioridad que dieron resultados y enfocar sus esfuerzos para reducir los tiempos de respuesta.

La información que se debe documentar y registrar en la gestión de los incidentes es:

- Descripción del tipo de incidente.
- Eventos registrados por los sistemas.
- Daños producidos en los sistemas.
- Acciones para contener y erradicar el incidente.
- Comunicaciones realizadas con terceros.
- Lista de evidencias obtenidas durante las fases de análisis y la detección.
- Comentarios del personal involucrado.
- Acciones para reforzar la seguridad y evitar futuros incidentes similares.

La documentación del incidente facilitará el análisis realizado sobre las causas y consecuencias de los incidentes sobre los sistemas informáticos de la organización.

El informe final sobre el incidente, debe contemplar los siguientes puntos:

- Revisión minuciosa de los registros de actividad (*logs*) de los equipos y dispositivos afectados directamente e indirectamente por el incidente.
- Las pérdidas económicas asociadas al incidente de seguridad.
- El daño a la imagen y reputación de la organización afectada.
- Análisis de las posibles consecuencias con terceros (daño colateral).
- Revisión del intercambio de información con otras empresas e instituciones.
- Posibles acciones legales emprendidas contra los responsables del incidente.
- Analizar la eficacia y rapidez de las acciones y decisiones tomadas.
- Valoración de los procedimientos que no hayan sido adecuados.
- Adquisición de hardware, software y recursos necesarios para reforzar la seguridad de los sistemas y redes de datos ante futuros incidentes de seguridad.
- Revisión de las políticas de seguridad de la información.

Para resumir lo visto de gestión de incidentes, se muestra el siguiente diagrama que ejemplifica las partes generales del ciclo de vida de un incidente. En el cual toda gestión de incidentes inicia con el *triage*, donde un incidente puede ser clasificado inicialmente o bien identificado como un nuevo evento o como parte de algún incidente ya existente, al cual se le está dando seguimiento. Como bien indica el libro *CSIRTs* del Software *Engineering Institute*[65] una vez que se asigna un *ticket* (identificador de caso) a un incidente, es posible que un incidente pase por diversas transiciones de estado hasta que complete su ciclo y sea concluida su gestión.

El cierre de un incidente, normalmente se produce cuando ninguna de las partes involucradas en el incidente tiene la capacidad para identificar la causa del reporte o no existe información sustancial para dar seguimiento formal a las acciones de resolución. En el mejor de los casos se logran aplicar las etapas de la gestión de incidentes adecuadamente y el incidente es concluido, ver Figura 3.10.

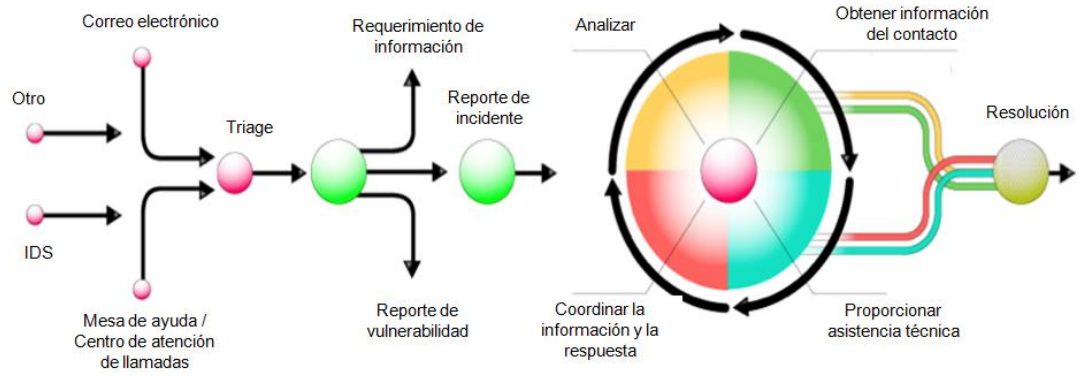


Figura 3.10 - Ciclo de vida de un incidente.

65 Moira J., Don S., Klaus-Peter, Georgia K., Robin R., Mark Z. [CMU/SEI-2003-HB-002]. Op. cit., p 77.

3.5 Modelos organizacionales de los CSIRTs

Existen diversos modelos organizacionales en los que un *CSIRTs* se puede desempeñar, todo depende de la misión y objetivos del equipo, de la comunidad objetivo que les interesa proteger y de los servicios que se brindan a la organización. [66]

3.5.1 Tipos de modelos organizacionales

a) CSIRT de Equipo de seguridad

Este modelo no es un modelo muy común para un *CSIRT*. Propiamente es un área centralizada con la responsabilidad de proporcionar o coordinar las acciones de manejo de incidentes. Estas tareas son realizadas por los administradores de las áreas de sistemas, redes y otros expertos de seguridad que tienen la misión de mantener, configurar y proteger los equipos y redes de la organización.

Generalmente este modelo se establece cuando no existe un *CSIRT* constituido dentro de la organización. Así mismo, no existe una autoridad central en la organización que proporcione la respuesta a incidentes de seguridad en acciones como la recopilación, análisis de datos de evidencia o en la aplicación de medidas de recuperación y mitigación. No hay una asignación formal de responsabilidades respecto a los incidentes de seguridad, es decir, el personal que desempeña actividades de tecnologías de la información es también el responsable de los eventos de seguridad ocurridos en la organización, por tanto este modelo es reactivo y no favorece la aplicación de servicios proactivos.

b) CSIRT Interno Distribuido

Es una estructura central pequeña, la cual cuenta con un responsable de la seguridad que supervisa y coordina al personal distribuido en diferentes áreas de la organización. A este personal se le asignan responsabilidades específicas relacionadas con la seguridad informática a las cuales se les dedican tiempo y esfuerzo. Este modelo es muy eficiente en organizaciones grandes en las que un equipo centralizado no es suficiente[67]. En este modelo sí existe un *CSIRT* reconocido como una entidad formal, con la responsabilidad del manejo de todas las actividades de respuesta a incidentes ocurridos dentro de la institución, para establecer este modelo, en la organización deben existir *procedimientos*, *procesos* y políticas para el manejo de incidentes, además de un método establecido para la comunicación de estrategias de seguridad.

c) CSIRT Interno Centralizado

Este modelo implementa un equipo centralizado, conformado por personal técnico especializado sobre el que recae toda la responsabilidad de reportar, analizar y dar respuesta a los incidentes. Es frecuente que el personal del *CSIRT* se dedique

[66] Proyecto Amparo (2011), Op. cit., p 49.

[67] Carnegie Mellon University [F19628- 05-C-0003].Creating and Managing Computer Security Incident Handling Teams (CSIRTs) (2008) Organizational Models, p 3-7.

únicamente a las actividades de respuesta a incidentes, por tal razón el *CSIRT* que implemente este modelo debe tener la autoridad y capacidad para recolectar información de una amplia variedad de fuentes, en poco tiempo sintetizarla y discriminarla para su posterior reporte a las instancias correspondientes, también el equipo puede participar en el análisis de vulnerabilidades y pruebas de seguridad en los sistemas afectados a fin de coadyuvar esfuerzos en actividades de contención y erradicación de los incidentes.

d) CSIRT Combinado

Es un modelo combinado entre el distribuido y el centralizado, en él se establece una interacción con un equipo central y otros miembros del equipo distribuidos en distintas ubicaciones geográficas dentro de la organización. El equipo central tiene la responsabilidad de ofrecer un análisis de alto nivel, proveer estrategias de mitigación y recomendaciones para la recuperación de los sistemas afectados. Por otro lado, los miembros del equipo distribuido implementan estrategias de seguridad y brindan su experiencia y conocimiento en las áreas asignadas. Este modelo reduce la carga de trabajo del equipo central lo que les permite dedicar más tiempo y recursos a la compresión y análisis de las amenazas de seguridad y actividad maliciosa que afecta a su comunidad objetivo.

e) CSIRT Coordinador

El modelo coordinador tiene un nivel de autoridad diverso, debido a que tiene a su cargo un conjunto de *CSIRTs* que pertenecen a la misma organización, este modelo cumple con el principal enfoque de un *CSIRT* que es coordinar y coadyuvar esfuerzos en el manejo de incidentes y vulnerabilidades sobre una comunidad objetivo, que se compone de diversas entidades las cuales pueden estar ubicadas en lugares geográficamente muy distantes. [68] Este modelo facilita el intercambio de información y provee estrategias de mitigación y recomendaciones para la respuesta y recuperación ante incidentes de seguridad, además provee recursos y referencias para la gestión de los incidentes mediante la difusión de bases de datos de vulnerabilidades, herramientas de seguridad, servicios de alertas y detección de incidentes, etcétera.

Puede existir un tipo de *CSIRT* con un modelo de coordinador donde su comunidad objetivo está compuesta por diversas instituciones de gobierno de un país. En este caso, el *CSIRT* sí puede tener autoridad total sobre todos los miembros de su circunscripción.

f) CSIRT Nacional

También existe otro tipo de *CSIRT* con modelo de coordinador en el cual la comunidad objetivo puede ser un país entero, sin embargo, no necesariamente se tiene plena autoridad sobre la comunidad objetivo.

---

[68] Paul C., Tom M., Tim G., Karen S., Computer Security Incident Handling Guide [800-61] (2012), NIST, p 13.

En México además del *CERT* académico de la *UNAM* (*UNAM-CERT*), existe el *Cyber Security Incident Response Team* de la empresa SCITUM, el CERT de la empresa Mnemo y el Centro Especializado en Respuesta Tecnológica (*CERT-MX*) de la Policía Federal, formalmente establecido el 1 de junio del 2010, ver Figura 3.11, el CERT-MX diferencia de los anteriores, estableció un modelo de tipo coordinador, debido a que atiende y da respuesta a los incidentes de seguridad informática que ponen en riesgo la infraestructura de información y comunicación críticas del país, coadyuvando esfuerzos con instituciones de las tres órdenes de gobierno; federal, estatal y municipal.

Team information	
Short team name	CERT-MX
Official team name	Centro Especializado en Respuesta Tecnologica de Mexico
Membership type	Full Member
Team host organization	National Comission of Security (CNS)
Country of team	Mexico
Date of establishment	2010-06-01
Public WWW server	http://www.cns.gob.mx
Constituency	
Type of constituency	Government, Private and Public sectors
Description of constituency	mexican national critical infrastructure, federal government and citizens
Internet domain address	.mx, .gob.mx
Country of constituency	Mexico

Figura 3.11 - Información de CERT-MX.[69]

3.5.2 Tipos de servicios que ofrece un CSIRT

Los modelos descritos anteriormente pueden brindar servicios básicos o adicionales dependiendo de las necesidades, infraestructura tecnológica y recursos humanos del CSIRT que circunscribe en cada organización.

Servicios básicos:

- Alertas y advertencias.
- Análisis de incidentes.
- Respuesta a incidentes en sitio.
- Soporte a incidentes vía telefónica y por correo electrónico.
- Coordinación de respuesta a incidentes.
- Coordinación de respuesta a vulnerabilidades.
- Configuración y mantenimiento de herramientas de seguridad.
- Coordinación de respuesta a artefactos.
- Servicios de detección de intrusiones.
- Observatorio tecnológico.
- Difusión de información relacionada con la seguridad.

[69] Sitio Web: http://www.first.org/members/teams/cert-mx, consultado 20-04-2015

- Concientización.

Servicios adicionales:

- Auditorías o evaluaciones de seguridad.
- Desarrollo de herramientas.
- Análisis de riesgo.
- Planificación de la continuidad del negocio.
- Consultoría de seguridad.
- Educación y capacitación.
- Evaluación y certificación de productos.

3.5.3 Tipo de organización

Para implementar un *CSIRT* es necesario hacer un estudio a fondo de la organización, de esta forma se definirá la estructura que mejor se adapte a su operación. Es necesario clasificar a la organización con base en los criterios de:

- Finalidad: Con fin de lucro o sin fin de lucro.
- Estructura: Formal o informal.
- Tamaño: grande, mediana, pequeña micro.
- Localización: multinacional, nacional, local, regional.
- Producción: bienes y servicios.
- Propiedad: pública, privada, mixta.
- Grado de integración: total o parcialmente integrada.
- Actitud ante cambios: rígido o flexible.
- Ambiente Externo: la interacción con terceros tales como proveedores, clientes, socios, etcétera.
- Ambiente Interno: todo lo relacionado con la organización donde se encuentra el *CSIRT*.

Capítulo 4

Buenas Prácticas de Seguridad Informática en el Sistema Linux-Debian

4.1 El sistema operativo UNIX

Las raíces de UNIX (ver Figura 4.1) se remontan a mediados de los años 60's, cuando de acuerdo con Eric Steven Raymond (2003) [70], *General Electric,* los laboratorios Bell de AT&T y el Instituto Tecnológico de Massachusetts iniciaron un proyecto llamado MULTICS (*MULTiplexed Information and Computing Service – Servicio de Información y Computación Multiplexada*). El proyecto fue patrocinado principalmente por la ~~a~~ Agencia de Proyectos de Investigación Avanzada del Departamento de Defensa de los Estados Unidos (ARPA – *Advanced Research Projects Agency*, también conocida como DARPA – *Defense Advanced Research Projects Agency*).

Figura 4.1 - Logotipo de la marca registrada de UNIX.[71]

MULTICS fue un sistema modular construido para sistemas de bancos de procesadores, memoria y equipo de comunicaciones de alta velocidad, además fue diseñado para soportar el concepto de seguridad multinivel (MLS - *Multi-Level Security*), mediante la implementación de ACL's (*Access Control List* – Listas de Control de Acceso) para prevenir que la información que hubiera sido clasificada en un determinado nivel de seguridad, llegara a las manos de alguien que no hubiera sido explícitamente autorizado para ello.

El proyecto fue abandonado por AT&T en 1969, mismo año en que dos investigadores que habían trabajado en el proyecto de MULTICS: Ken Thompson y Dennis Ritchie, crearon la primera versión de UNIX, aunque tal como se comenta en el sitio Web Linux-os (http://www.linux-os.com.ar/linuxos/category/multics/), el proyecto fue bautizado inicialmente con el nombre de UNICS, como acrónimo *Uniplexed Information and Computing Service (*Servicio de Información y Computación Uniplexada*)* en oposición al concepto "Multiplexado" de Multics, resaltándola como una versión miniatura de Multics.

UNIX fue diseñado como un sistema que adoptaba la filosofía de hacer una sola acción, pero hacerla bien mediante el empleo de programas compactos llamadas herramientas, en donde cada una de ellas tenía un propósito específico, de manera que juntando las herramientas, los programadores podían hacer cosas muy complicadas, cambiando así la forma de pensar de muchos que desarrollaron más herramientas, como el procesamiento de textos. Thompson y Ritchie lograron cumplir con la solicitud de agregar herramientas que permitieran el procesamiento de textos a UNIX en una máquina PDP11/20 (ver Figura 4.2), y como consecuencia de ello

70 Eric Steven Raymond. ESR (2003). *The Art of Unix Programming, Origins and History of Unix,* 1969-1995. Obtenida el 05 de septiembre de 2012, de http://www.faqs.org/docs/artu/ch02s01.html.

71 UNIX Tutorial Home page. (n.d), Obtenida el 05 de septiembre de 2012, de http://people.rit.edu/~dqn8613/409/Unix/index.xhtml.

consiguieron el apoyo económico de los laboratorios Bell. Fue así como por vez primera, en 1970, se habla oficialmente del sistema operativo UNIX.

Figura 4.2 - Fotografía de la microcomputadora PDP-7. [72]

En 1972 se tomó la decisión de escribir nuevamente UNIX, pero esta vez en el lenguaje de programación C. "Inicialmente Unix fue escrito en lenguaje ensamblador y las aplicaciones desarrolladas en una mezcla de ensamblador y un lenguaje interpretado llamado B, que tenía la característica de ser lo suficientemente pequeño como para ejecutarse en un PDP-7 pero B no era lo suficientemente poderoso como para la programación de sistemas, por lo que en 1973 Thompson y Ritchie finalmente terminaron de reescribir Unix en el nuevo lenguaje C". [73]

El lenguaje de programación C fue diseñado para ser un lenguaje sencillo y portable, la mayoría de los programas escritos en C podrían ser movidos entre distintos tipos de computadoras pero no siempre corrían adecuadamente, ya que cada sistema operativo realizaba entradas y salidas de formas ligeramente distintas, no obstante UNIX se había convertido en un sistema operativo popular en muchas universidades y estaba siendo comercializado por varias compañías.

La universidad de *Berkeley* adquirió una copia de UNIX, pero no sólo lo usó, sino que dos de sus estudiantes; Bill Joy y ChuckHaley, comenzaron a realizar modificaciones significativas al software, creando así en 1978 el *Berkeley Software Distribution* (BSD), debido a estas aportaciones se desarrollaron dos estándares de UNIX, el AT&T *System* V y el UNIX de *Berkley*. La mayoría de las compañías adoptaron el de AT&T, excepto Sun Microsystems (SunOS) y Digital (Ultrix) que eligieron el UNIX de *Berkeley*, que era el preferido en ambientes de desarrollo y académicos.

72 Eric Steven Raymond. ESR (2003). *The Art of Unix Programming, Origins and History of Unix,* 1969-1995. Obtenida el 05 de septiembre de 2012, de http://www.faqs.org/docs/artu/ch02s01.html.

73 Ibídem.

En la Figura 4.3 - Historia de UNIX del año de 1969 a 2010, se ilustra la evolución de UNIX en el tiempo, así como sus diferentes vertientes entre las que destacan, Free BSD, OpenBSD, Sun OS, HP/UK, Solaris, AIX, MAC OSX y LINUX.

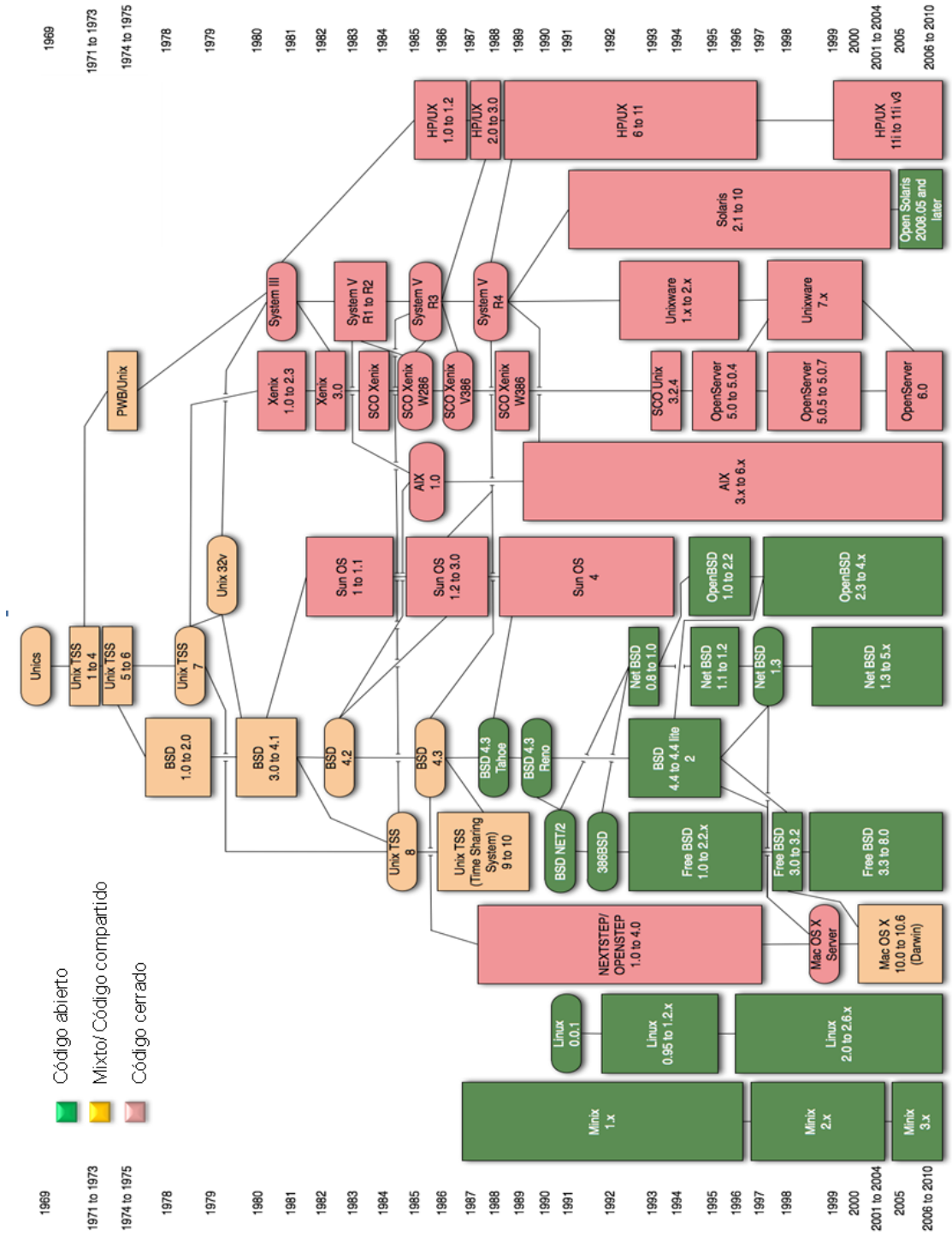


Figura 4.3 - Historia de UNIX del año de 1969 a 2010. [74]

[74] Southwest Florida GNU/Linux Users Group. (2010). Obtenida el 06 de septiembre de 2012 de http://files.meetup.com/95734/Unix_history-simple.svg.

4.2 GNU, software libre y código abierto

A finales de la década de 1970, Richard Stallman trabajaba en el laboratorio de Inteligencia Artificial (IA) en el Instituto Tecnológico de Massachusetts, al mismo tiempo participaba en una comunidad cuyo propósito era desarrollar un sistema operativo de tiempo compartido (ITS - *Compatible Time-SharingSystem*), donde el trabajo de Richard Stallman se limitaba a mejorar el sistema. Compartir el software era algo normal, si otra universidad o empresa se interesaba en algún programa realizado en el laboratorio, se le entregaba el código fuente de dicho programa para que pudiera modificarlo, tomar partes del mismo o únicamente leerlo, como se explica en el sitio Web de gnu.org [75], aunque estos programas no se consideraban como software libre, porque no existía el concepto, en realidad lo eran.

Posteriormente los cambios en la industria finalmente llegaron al laboratorio de IA, muchos de sus integrantes se fueron del laboratorio, al mismo tiempo que se adquirían equipos con sistema operativo de marca registrada, en los que no se permitía compartir el código, al presentarse esta situación Richard Stallman tomó la decisión de renunciar al laboratorio y comenzar el proyecto GNU (GNU isNot Unix – GNU no es un Unix) (ver Figura 4.4) junto con la Fundación de Software Libre (FSF - *Free Software Foundation*), el objetivo del proyecto GNU era "desarrollar un sistema y aplicaciones completamente libres y abiertas, donde nunca se impidiera a la gente hackear o compartir sus cambios" [76].

Figura 4.4 - Logotipo del proyecto GNU. [77]

Además Richard Stallman creó la Licencia Pública General GNU la cual dice:

> *Los códigos pueden ser copiados y modificados sin ninguna restricción y ambas copias y trabajos derivados deben ser distribuidas bajo la misma licencia que el original, sin poner restricciones adicionales* [78]*.*

Lo anterior impulsó el nacimiento del concepto de software libre, el cual nació formalmente en 1984 como una iniciativa de Richard Stallman de compartir el código

75 La primera comunidad que comparte software (2012). Obtenida el 21 de septiembre de 2012, de http://www.gnu.org/gnu/thegnuproject.es.html.

76 Fogel Karl. FK (2007). Producir Software de Código Abierto – Libre vs Abierto, p5. Obtenida el 23 de septiembre de 2012, de http://producingoss.com/es/producingoss.pdf.

77 GNU, Proyecto GNU (2008). Logo de GNU (n.d.). Obtenida el 21 de septiembre de 2012 de http://lacasadetux.wordpress.com/2008/09/06/historia-del-proyecto-gnu/.

78 Fogel Karl. FK (2007). Producir Software de Código Abierto – Libre vs Abierto, p5. Obtenida el 23 de septiembre de 2012, de http://producingoss.com/es/producingoss.pdf.

fuente de los programas de computadoras a la comunidad de programadores que se pudieran interesar en él. Según el sitio Web del GNU el software libre significa:

> *El software respeta la libertad de los usuarios y la comunidad. En términos generales, los usuarios tienen la libertad de copiar, distribuir, estudiar, modificar y mejorar el software. Con estas libertades, los usuarios (tanto individualmente como en forma colectiva) controlan el programa y lo que hace.* [79]

Con base a la definición anterior, Debian es un sistema operativo de "software libre", por tanto, quien lo usa adquiere cuatro derechos o libertades sobre él:

- libertad 0: la libertad de usar el programa, con cualquier propósito.
- libertad 1: la libertad de estudiar cómo funciona el programa y adaptarlo a las necesidades correspondientes.
- libertad 2: la libertad de distribuir copias de éste software.
- libertad 3: la libertad de mejorar el programa y hacer públicas las mejoras, modificando directamente el código fuente.

Posteriormente, al auge del software libre, surgió la confusión de si un software es libre entonces debe ser gratis. La confusión radicaba en la traducción del término "*free software*" el cual puede interpretarse como un software gratuito o libre dependiendo del contexto, en esencia todo software libre es gratuito pero no todo software gratuito es libre, esta definición causó gran controversia para las corporaciones, ya que ellas no querían pagar por un software "libre" porque lo asociaban con un costo nulo. Ante esta situación Bruce Perens y Eric S. Raymond lanzan el término de *Open Source* o Código Abierto, como una estrategia de mercado para introducir el software libre a las grandes empresas tal como lo expresa Karl Fogel [80] referente al proyecto de *opensource* (ver Figura 4.5):

> *…La Iniciativa por el Código Abierto es un programa de mercado para el software libre. Significa fundar el "software libre" sobre bases sólidas y prácticas más que en una discusión acalorada. La sustancia ganadora no ha cambiado, sí en cambio la actitud de perdedores y su simbolismo…*

Figura 4.5 - Logotipo de opensource. [81]

[79] La definición de software libre (2012). Obtenida el 21 de septiembre de 2012, de http://www.gnu.org/philosophy/free-sw.es.html.

[80] Fogel Karl. FK (2007). Producir Software de Código Abierto – Libre vs Abierto, p7. Obtenida el 23 de septiembre de 2012, de http://producingoss.com/es/producingoss.pdf.

[81] Open SourceIniative (n.d.). Obtenida el 23 de septiembre de 2012 de http://opensource.org/files/osi_symbol.png.

La definición completa de lo que es el código abierto se puede consultar en el sitio Web: http://opensource.org/docs/definition.php, la cual se basa en los siguientes 10 puntos:

1. Libre redistribución: puede ser brindado gratuitamente o vendido.
2. Código fuente: debe ser incluido o debe poder ser obtenido libremente.
3. Trabajos derivados: debe permitirse la redistribución de modificaciones.
4. Integridad del código fuente del autor: las licencias deben requerir que las modificaciones deban ser redistribuidas sólo como parches.
5. No discriminación contra personas o grupos.
6. No discriminación contra determinados campos de comportamiento: los usuarios comerciales no deben ser excluidos.
7. Distribución de la licencia: los derechos vinculados al programa deben aplicarse a todos hacia los cuales el programa es redistribuido, sin la necesidad de la ejecución de una licencia adicional por parte de éstos.
8. La licencia no debe ser específica de determinado producto: el programa no puede ser licenciado sólo como parte de una distribución mayor.
9. La licencia no debe restringir otro software: la licencia no puede insistir que cualquier otro software con el cual es distribuido debe ser también de código abierto.
10. La licencia debe ser tecnológicamente neutral.

La existencia de estos dos movimientos (Software Libre y Código Abierto) reconoce el mismo conjunto de licencias y desde el punto de vista práctico, son equivalentes puesto que ambos movimientos trabajan en el desarrollo de proyectos comunes. Por ejemplo, el proyecto Debian (Software Libre) tiene como objetivo ofrecer un entorno computacional 100% libre, pero también no tiene inconveniente alguno para integrarse con código no libre y cooperar con programadores de código abierto que comparten la ideología del software libre.

Con el amplio apoyo del software libre, el código abierto y la aceptación del proyecto GNU de nuevos programadores que compartían la ideología de Stallman, se produjo en 1990, un sistema operativo “libre”, excepto por el núcleo (kernel) -pieza indispensable para la gestión de memoria-, el disco y los recursos del sistema. Esta pieza faltante fue proporcionado por Linus Torvalds un estudiante de computación finlandés.

4.3 El sistema operativo LINUX

El 3 de julio de 1991 un finlandés Linus Torvalds envió un mensaje a través de un foro de MINIX (comp.os.minix), en el cual anunciaba que estaba trabajando en un nuevo proyecto:

> *Hola a todos aquellos que usan Minix.*
>
> *Estoy creando un sistema operativo (libre) (por puro hobby, no será tan grande ni profesional como GNU) para clones AT 386(486). Llevo trabajando en ello desde abril y ya empieza a estar listo. Me gustaría recibir comentarios sobre lo que a la gente le gusta/disgusta de minix, ya que mi SO se le parece un poco (misma disposición física del sistema de archivos (por motivos prácticos) entre otras cosas).*

LinusBenedictTorvalds [82]

Torvalds comenzó escribiendo un emulador de terminal para conectar su computadora con los servidores UNIX de la universidad de *Helsinky* partiendo del sistema experimental *Minix*, con el paso de los meses se dio cuenta que estaba escribiendo el núcleo de un nuevo sistema operativo el cual bautizó con el nombre de Linux por la similitud con su apellido.

En el año de 1991 el *kernel* del Linus Torvalds era sólo eso un *kernel*, aún le faltaba mucho trabajo para conformar un sistema operativo, le faltaba un compilador, librerías, utilidades, sistemas de bases de datos, interfaz gráfica, etcétera. Tal como se comenta en el artículo electrónico "Una breve historia de Linux", [83] el proyecto GNU tenía todo ello y viceversa al proyecto GNU, le falta un *kernel* que fuera compatible, ambos eran la combinación perfecta, afortunadamente esta unión se concretó y tuvo su primer resultado el 5 de octubre de 1991, año en el que Linus anunció la primera versión "Oficial" de Linux, la versión 0.02. En esta versión se pudo ejecutar *Bash* (GNU *Bourne Again Shell*) y gcc (Compilador GNU de C).

Cabe aclarar que el nombre Linux se refiere estrictamente al núcleo Linux, pero es comúnmente utilizado para describir al sistema operativo tipo Unix (que implementa el estándar POSIX), el cual está formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU. Posteriormente con la adopción de la licencia GPL (*General Public License* – Licencia Pública General) de GNU, se agregaron libertades al software de Linux impulsando así el nacimiento del sistema operativo completamente libre, con su propio núcleo, y utilidades que cualquiera podría modificar, mejorar y redistribuir, obteniendo así el apoyo de muchos desarrolladores que se sumaron al proyecto y que lo veían como una seria alternativa ante el software restrictivo de Microsoft, Apple e IBM.

Al mismo tiempo del crecimiento de Linux, surgió una disputa entre Torvalds y Stallman por el nombre que debería llevar el proyecto (promotor del proyecto GNU) éste último pedía que se diese más reconocimiento a su labor y que el proyecto se llamara "GNU/Linux", muchos decían que el núcleo lo es todo y otros que un núcleo sin utilidades es inútil. Al final, Torvalds concluyó que quien quisiera usar GNU/Linux está en total libertad de hacerlo pero que renombrar Linux no era una opción. Aun con la libertad de referirse a este Sistema Operativo de las dos formas, los usuarios adoptaron el nombre de "Linux" por ser más corto y en consecuencia fácil de recordar.

En el año de 1994 se liberó la versión 1.0 estable de Linux (ver Figura 4.6) en la que participaron cientos de desarrolladores, ya en 1995 Linux era capaz de ejecutarse en plataformas de hardware como DEC (obsoleto) y Sun, además podría correr tanto en computadoras personales como grandes servidores con múltiples procesadores,

82 Benedict Torvalds L. Linux's History (n.d.). Obtenido el 07 de septiembre de 2012, de http://www.cs.cmu.edu/~awb/linux.history.html.

83 Alvy, Una breve historia de Linux (2009), Obtenida el 06 de septiembre de 2012, de http://noticias.lainformacion.com/ciencia-y-tecnologia/ciencias-informaticas/una-breve-historia-de-linux_Zj2ge6h8QOZgWu0h8eyqM6/.

haciéndolo más universal, un buen ejemplo de ello es lo que se comenta en el artículo "Una breve historia de Linux", en la actualidad empresas como Amazon y Google utilizan Linux, por lo tanto todos somos usuarios de Linux en cierto modo. [84]

Figura 4.6 - Logotipo del Sistema Operativo LINUX, un pingüino llamado Tux. [85]

En la Figura 4.7, se observan las distintas distribuciones de Linux que han surgido formalmente desde 1992 a la fecha, se observa que varias distribuciones se basan en Debian.

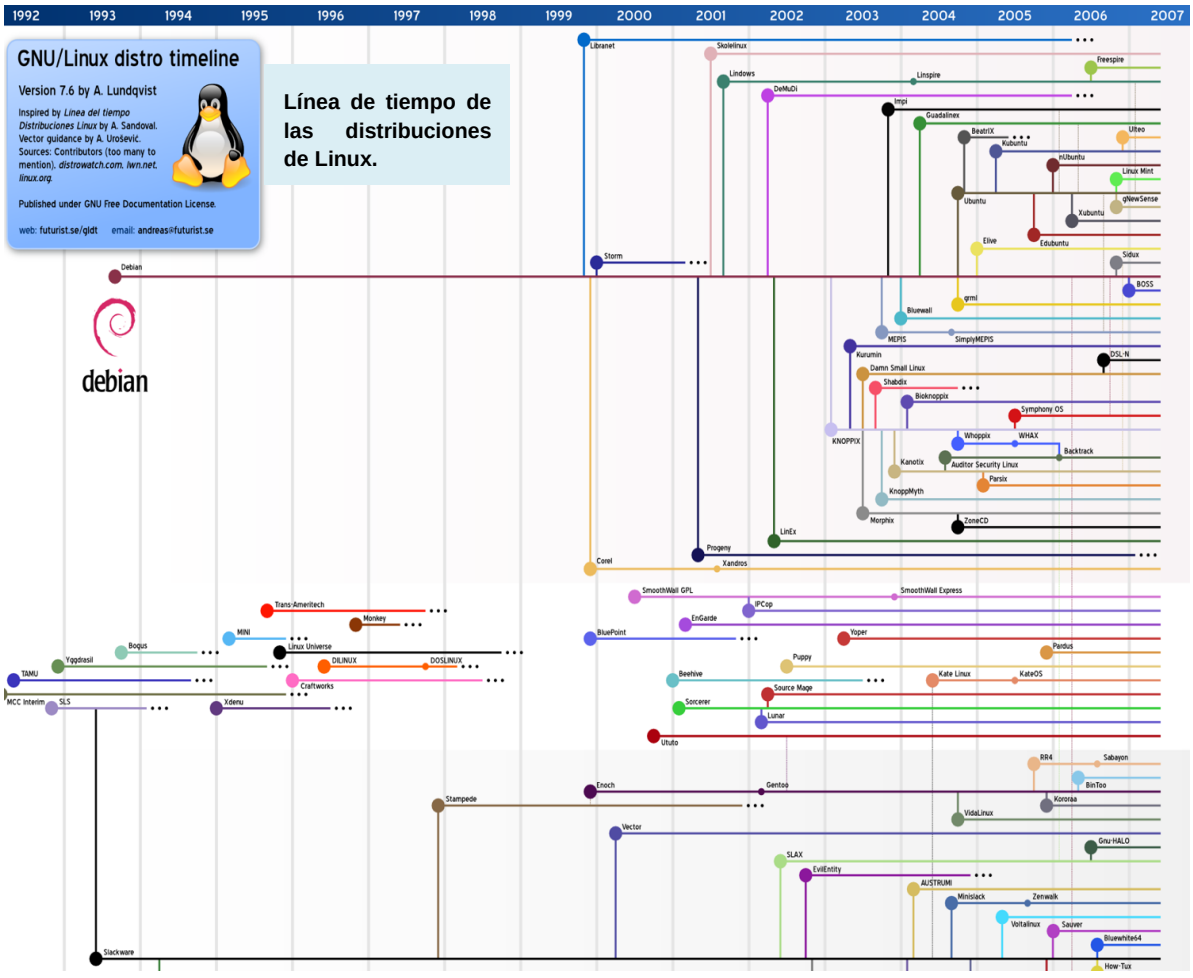


Figura 4.7 - Historia de distribuciones de Linux. [86]

84 Ibídem

85 Las 14 mejores distribuciones de Linux (2010), Obtenida el 06 de septiembre de 2012, de http://mediterrahosting.com/wp-content/uploads/2012/07/linux-logo-full.jpg.

La Figura 4.8, es la continuación de la Figura 4.7, ambas figuras se leen de arriba hacia abajo.

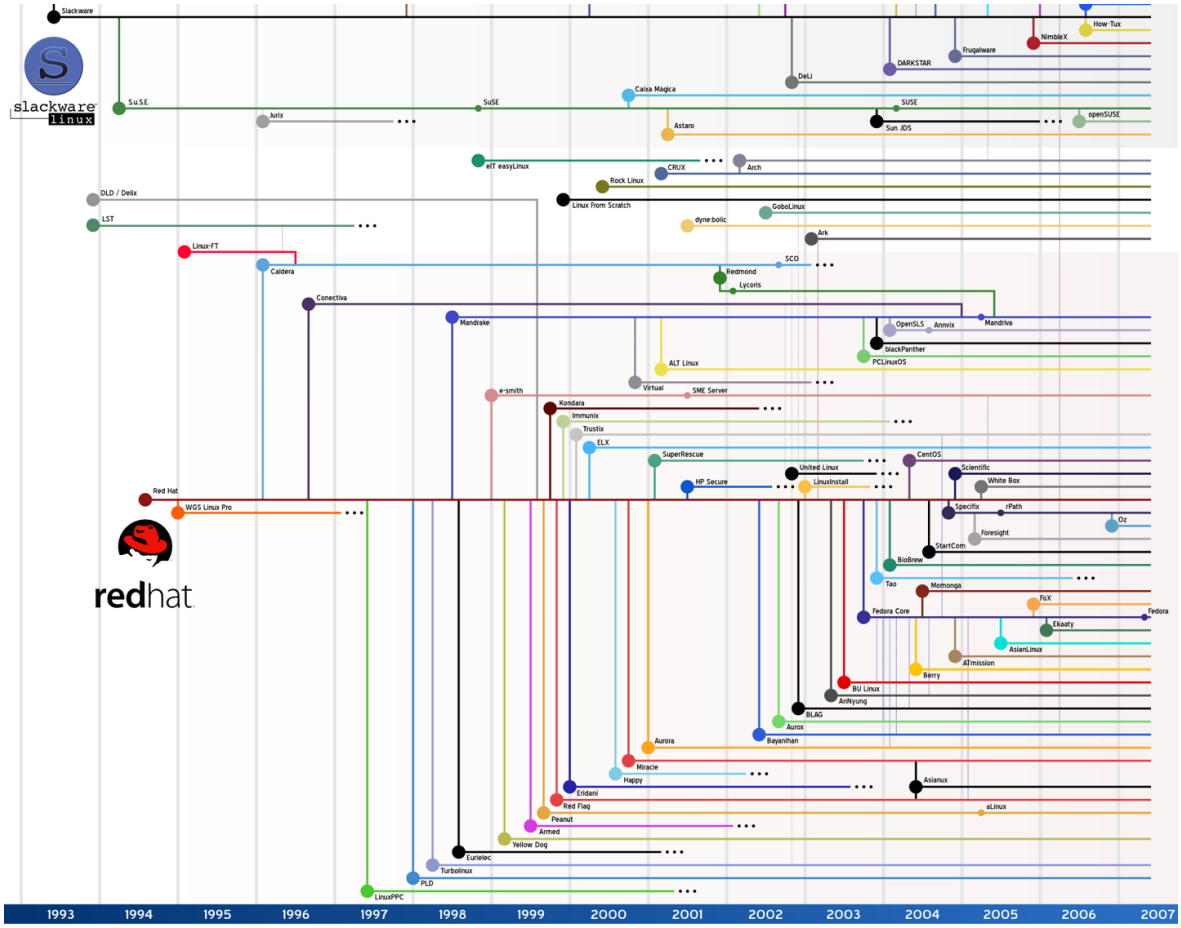


Figura 4.8 - Historia de distribuciones de Linux. [87]

4.4 Distribución Debian del sistema operativo Linux

Debian es una distribución GNU/Linux y GNU/KFreeBSD (la letra "K" proviene del hecho que sólo se usa el *kernel* de FreeBSD), la cual es un sistema operativo completo, que incluye software y sistemas para la instalación o administración, todos basados ya sea en Linux o en el *kernel* de FreeBSD. Es de notar que el presente trabajo sólo cubrirá sistemas GNU/Linux.

Formalmente el 16 de agosto de 1993 nace el proyecto Debian, desarrollado por Ian Murdock, el nombre se basa en la fusión del nombre de su esposa y el suyo (ver Figura 4.8), en sus inicios contó con el apoyo de Richard Stallman creador del proyecto GNU, desarrollado con la intención de tener un sistema operativo totalmente libre.

---

[86] Línea de tiempo de las distribuciones de Linux (n.d.). Obtenida el 18 de septiembre de 2012, de http://robertbriones.files.wordpress.com/2007/07/linea-tiempo-linux.png.

[87] Ibídem.

Debra + **Ian = Debian**

Figura 4.9 - Logotipo del proyecto Debian. [88]

Murdock también escribió el Manifiesto de Debian que utilizó como base para la creación de la distribución Linux Debian, este manifiesto explica qué es el proyecto, por qué se desarrolló y los beneficios de tener un software libre y gratuito que se distribuya bajo los términos de la Licencia GNU.

> *El proyecto Debian (http://www.debian.org/) es un grupo mundial de voluntarios que se esfuerzan por producir una distribución de sistema operativo que esté compuesta enteramente de software libre. El producto principal del proyecto a la fecha es la distribución de software Debian GNU/Linux, la cual incluye a Linux como núcleo del sistema operativo, así como miles de aplicaciones pre-empaquetadas* [89]

En este manifiesto Murdock, también plasmó claramente los objetivos de Debian. La calidad; Debian debe desarrollarse con el mayor cuidado para ser digna del *Kernel* de Linux, además debe ser una distribución no comercial lo suficientemente fiable para competir con las principales distribuciones comerciales. Esto le ha permitido a Debian tener mucho éxito y en la actualidad ser capaz de soportar 13 arquitecturas, alrededor de 17,300 paquetes y software que satisface casi cualquier necesidad que un usuario de casa o empresarial podría tener [90].

Dado el hecho de que la versión completa de Debian ocupa alrededor de 50 CD-ROM, se considera a Debian más como una meta-distribución de la que se extraen distribuciones específicas enfocadas a un público en particular, las cuales son: [91]

a) **Debian-Junior**: Ofrece a los niños una atractiva y fácil forma de usar Debian.
b) **Debian-Edu**: Se enfoca en la creación de una distribución especializada para el mundo académico.
c) **DebianMed**: Dedicada al campo de la medicina.
d) **Debian-Multimedia**: De los creadores de AGNULA (Distribución de Linux orientada al audio) pero con un enfoque de multimedia.
e) **Debian-Desktop**: Se enfoca en un ambiente de Escritorio.
f) **Debian-Ham**: Creada por Bruce Perens, dirigida a los radios aficionados.

---

88 Logotipo de Debian (2012), Obtenida el 24 de septiembre de 2012, de http://www.debian.org/Pics/openlogo-50.png.

89 Una breve historia de Debian (2004). Obtenida el 24 de septiembre de 2012 de http://www.debian.org/doc/manuals/project-history/ch-intro.es.html.

90 Hertzog Raphael y Mas Roland. (2012). The Debian Administrator's Handbook – Lifecycle of a Release p3. Obtenida el 25 de septiembre de 2013, de http://debian-handbook.info/download/stable/debian-handbook.pdf.

91 Ibídem, p 17.

g) **Debian-NP** (Non-Profit – No lucro): Es para organizaciones sin fines de lucro.
h) **Debian-Lex**: Destinada para el trabajo en el campo legal.

4.4.1 Versiones y paquetes de Debian

Tal como se expresa en el sitio Web del proyecto (www.debian.org), Debian es la única distribución abierta a contribuciones como desarrolladores y usuarios que tengan el deseo de ser parte del proyecto, también es la única distribución de Linux (relevante) que no es parte de una entidad comercial, lo cual ha permitido que sea una distribución basada en las necesidades y deseos de los usuarios. A continuación se explican cada una de ellas resaltando que el proyecto puede tener simultáneamente 3, 4 o 5 versiones; la experimental, inestable, prueba y estable [92].

a) **Experimental**: Se trata de un grupo de paquetes de Debian correspondientes al actual software en desarrollo, los cuales no necesariamente están completos. Esta versión alberga importantes modificaciones a los paquetes "base", cuya integración en la versión inestable causaría importantes repercusiones, por tanto, es una versión completamente aislada sin la posibilidad de migrar sus paquetes a otra versión.
b) **Inestable:** Es la versión utilizada por los desarrolladores del proyecto, la cual por lo regular tiene el nombre clave de "sid". En esta versión se encuentran paquetes de interés para los desarrolladores aun cuando representan un riesgo para la estabilidad del sistema, la versión se caracteriza por implementar un ciclo de mejora continua hasta que los desarrolladores ya no encuentren errores críticos en cada uno de los paquetes.
c) **Prueba:** En esta versión se encuentras paquetes que han estado previamente en la versión inestable que no han tenido modificaciones recientes y que ya se les han eliminado los fallos críticos detectados, además estos paquetes se deben poder instalar en todas las arquitecturas para las cuales fueron desarrolladas.
d) **Congelada:** cuando la versión de pruebas llega a un nivel de operatividad y aceptable de fallos entonces se "congela", lo cual significa que no se aceptan nuevos paquetes de la versión inestable y se comienza un arduo trabajo para depurar la mayor cantidad de errores de la versión, la cual no será puesta como estable hasta que no se alcance un nivel aceptable entre sus desarrolladores.
e) **Estable:** Se refiere a la versión estabilizada de la distribución la cual es recomendada para su uso y distribución. Cuando se llega a esta versión únicamente se aprueban actualizaciones que corrijan errores y sólo pueden hacerse por los líderes del proyecto.

En la Tabla 4.1, se muestran las versiones estables de Debian las cuales son 12 hasta el primer cuarto del 2013. Es importante resaltar que el nombre de las versiones de Debían fue tomado de los personajes de la película de Toy Story [93].

[92] Ibídem, p 22-25

[93] Historia del proyecto Debian (2003). Obtenida el 23 de septiembre de 2012, de http://debianitas.net/doc/minicomos/Historia%20y%20FAQ%20sobre%20Debian/pdf/historia-julio2003.pdf.

Tabla 4.1 -Historia de versiones de Debian. [94]

Personaje	Versión	Nombre	Fecha	Arquitecturas	No. Paquetes
	1.1	Buzz	17-06-1996	1	474
	1.2	Rex	12-12-1996	1	848
	1.3	Bo	02-06-1997	1	974
	2.0	Hamm	24-07-1998	2	1,500
	2.1	Slink	09-03-1999	4	2,250
	2.2	Potato	15-08-2000	6	3,900
	3.0	Woody	19-07-2002	11	8,500
	3.1	Sarge	06-06-2005	11	15,400
	4.0	Etch	08-04-2007	11	18,000
	5.0	Lenny	14-02-2009	12	23,000
	6.0	Squeeze	06-02-2011	9	29,000
	7.8	Wheezy	10-01-2015	13	36,000

La Figura 4.10 muestra la línea de tiempo de las versiones Hamm hasta Lenny de Debian, en la cual se puede apreciar que es común que al mismo tiempo que se está liberando una versión del tipo “estable”, el proyecto Debian simultáneamente está trabajando en la siguiente versión, lo cual refleja la constante evolución y mejora de este sistema operativo.

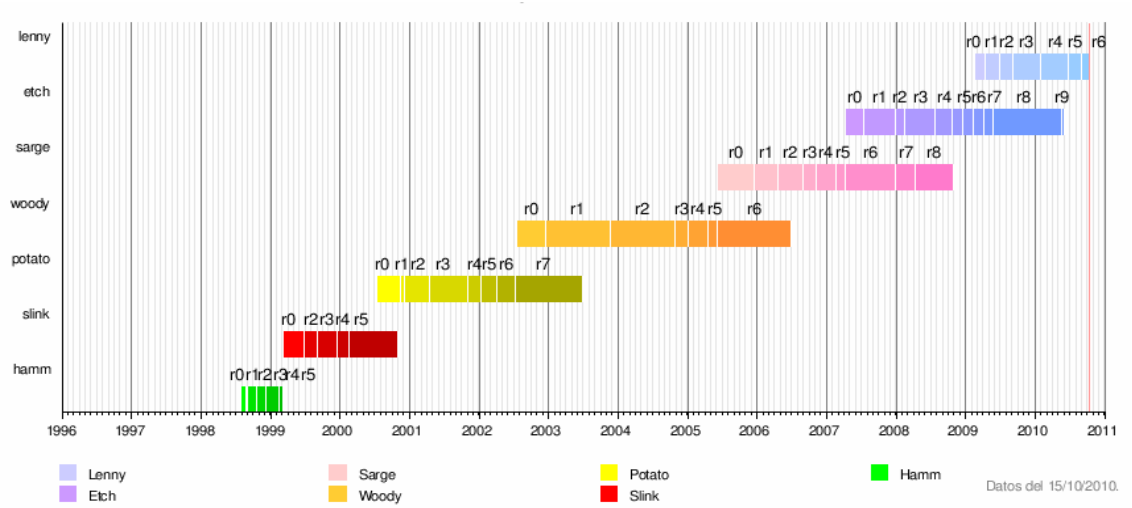


Figura 4.10 - Línea de tiempo de versiones de Debian. [95]

94 Publicaciones de Debian (2004), Obtenida el 24 de septiembre de 2012, de http://www.debian.org/doc/manuals/project-history/project-history.es.txt.

Los tipos de paquetes que constituyen una distribución Debian se pueden dividir en 3[96]:

a) **Tipo *main*:** únicamente los paquetes de esta área se consideran parte de la distribución. Ninguno de los paquetes de tipo *main* requiere software que no sea de esta sección, lo cual faculta que cualquier persona puede usar, compartir, modificar y redistribuir los paquetes de esta sección. Cada paquete debe cumplir con el DFSG (*Debian Free Software Guidelines*) el cual es un conjunto de directrices que el proyecto Debian utiliza para definir si una licencia de software es libre.
b) **Tipo *contrib*:** es la sección que contiene paquetes complementarios destinados a trabajar con la distribución Debian, pero que requieren de software no incluido en la distribución. Cada paquete en *contrib* debe cumplir con el DFSG.
c) **Tipo *non-freee*:** esta sección contiene paquetes complementarios destinados a trabajar con la distribución Debian que no cumplen con la DFSG o tienen otros problemas que hacen que su distribución sea problemática.

4.4.2 Seguridad en Debian

Tal como lo dice Javier Fernández [97] asegurar un sistema Debian no se diferencia mucho de asegurar algún otro sistema Linux, el sistema puede ser tan seguro como el administrador sea capaz de hacerlo. La instalación predeterminada de Debian trata de ser segura, pero no es tan rigurosa y paranoica como la de otros sistemas operativos que instalan todos los servicios deshabilitados de forma predeterminada. Debido a ello no existe mucha diferencia entre las distribuciones de Linux, salvo por la instalación base y el sistema de gestión de paquetes, la mayoría de las distribuciones tienen en común varias aplicaciones pero con diferentes versiones.

Además, existe mucha colaboración de parte de los equipos de seguridad de cada una de las distribuciones más importantes de Linux, la información de las vulnerabilidades no se esconde, al contrario se difunde abiertamente por lo que su solución se desarrolla de forma coordinada y se libera al mismo tiempo para todas las distribuciones, en términos generales se puede decir que la seguridad es relativamente la misma entre todas las distribuciones, el por qué se considera a la distribución Debian más segura se explica de aquí en adelante.

La ventaja sustancial de Debian en cuanto a seguridad se basa en la facilidad del sistema para la actualización de sus paquetes mediante el comando “apt”. Además Debian a diferencia de otras distribuciones pone a disposición muchas herramientas

95 Línea de tiempo de Debian GNU/Linux (n.d.). Obtenida el 21 de septiembre de 2012, de http://linuxgnublog.org/imgblog/lintem.png.

96 Archive areas (2012). Obtenida el 25 de septiembre de 2012, de http://www.debian.org/doc/debian-policy/ch-archive.html.

97 Javier Fernández-Sanguino Peña. (2005). Manual de Seguridad de Debian, Capitulo 2. Obtenido el 12 de octubre de 2012, de http://www.linux-cd.com.ar/manuales/debian-seguridad/ch2.es.html.

de seguridad como: *nessus, raccess, nikto, bass, satan, nmap, xprobe, hping2, nbtscan, tiger, rats, flawfinder, freeswan, pptp, sanitizer, amavis-postfix*, tan sólo por mencionar algunas (consultar http://www.linux-cd.com.ar/manuales/debian-seguridad/ch-sec-tools.es.html. Por último Debian, tiene una versión estándar de instalación pequeña (con menor funcionalidad) en consecuencia más segura a diferencia de otras distribuciones que en el afán de darle más funcionalidad al sistema (más servicios predeterminados) y facilitarle el trabajo al usuario indirectamente hacen al sistema más inseguro.

A continuación se presentan los aspectos de seguridad más importantes de Debian GNU/Linux, empleados para proveer un sistema seguro en su conjunto. [98]

1. Debian, maneja abiertamente los problemas de seguridad, tal como lo dice la *Debian Social Contract* (http://www.debian. org/social_contract): *Nosotros **no** encubriremos los problemas, mantendremos el reporte de la base de datos abierto para el público todo el tiempo*. Estos temas de seguridad son discutidos en listas de correo como *debian-security*.
2. El equipo de seguridad de Debian sigue de cerca los temas de seguridad, vigila con regularidad las fuentes de información relacionadas como: Bugtraq (http: //www.securityfocus.com/cgi-bin/vulns.pl), en búsqueda constante de vulnerabilidades en los paquetes contenidos en Debian.
3. Las actualizaciones de seguridad son la prioridad número uno para Debian. Cuando se identifica un problema de seguridad en algún paquete, la actualización que lo corrige es liberada tan rápido como es posible y distribuida tanto en versiones estables como inestables.
4. La información de seguridad es centralizada en un punto (http:// security.debian.org).
5. Debian continuamente mejora la seguridad desde un enfoque de conjunto, por ello es común el desarrollo de nuevos proyectos como es el caso de los mecanismos de verificación automática de firmas de paquetes. El cual previene la instalación de paquetes que no hayan sido desarrollados o aprobados por el proyecto.
6. Debian promueve una cantidad considerable de herramientas de seguridad, las cuales se enfocan en: integridad de archivos, auditoría, seguridad perimetral, detección de intrusos, etcétera.
7. Dado que la seguridad es un tema importante para Debian, muchos de los servicios instalados de forma predeterminada son seguros por defecto, conservando un balance entre la funcionalidad y seguridad.

4.4.3 Ventajas de Debian

En el tema anterior se explicaron las ventajas de Debian en cuanto a seguridad, en las Tablas 4.2 y 4.3, se especifican las ventajas y desventajas generales de usar la distribución Debian, las cuales son mencionadas superficialmente, debido a que una mejor explicación se puede encontrar en el sitio oficial http://www.debian.org/intro/why_debian.es.html.

98 Ibídem, Capitulo 2, p 18-19

Tabla 4.2 - Ventajas de los sistemas Debian GNU/Linux.

Ventajas de usar Debian	Descripción
Es mantenida por sus usuarios	Si algo requiere ser arreglado o mejorado, simplemente se hace.
Soporte técnico incomparable	Cuando se envía una pregunta a las listas de correo, se obtiene respuesta en menos de 15 minutos. El grupo de desarrolladores está conformado por más de 500.
Muchas personas y organizaciones la usan	Instituciones educativas, comerciales, sin fines de lucro y organizaciones gubernamentales utilizan Debian (http://www.debian.org/users/).
El mejor sistema de empaquetamiento	El paquete dpkg se encarga de que los archivos viejos de software no causen conflictos al sistema al actualizarse o instalarse otra versión.
Instalación sencilla	Se ha mejorado la simplicidad y facilidad de instalación de Debian (ambiente gráfico).
29,000 elementos de software	Cada paquete es 100% libre, es decir, se puede modificar, redistribuir y no se infringe ninguna ley.
Buena integridad de los paquetes	Todos los paquetes se encuentran en un mismo sitio, se han corregido al máximo los problemas relacionados con complejas dependencias, además es la única distribución que se rige por normas y estándares de calidad.
Disponibilidad de código fuente y herramientas para su manejo	Los desarrolladores tienen a su disposición cientos de herramientas y lenguajes de desarrollo, además de millones de líneas de código fuente en el sistema base.
Fácil actualización	Tan sólo con ejecutar apt-getupdate; apt-getdist-upgrade; se puede actualizar el sistema de forma completa.
Sistema de seguimiento de errores	Es un sistema público, no se intentan esconder los problemas de seguridad, Debian responde a los problemas de forma clara, honesta y abierta.
Estabilidad	Existen muchos casos de máquinas con sistemas Debian que trabajan más de un año sin reiniciarse debido a la estabilidad de sus paquetes y actualizaciones.
Rápido y ligero en memoria	Al estar basado en GNU/Linux, Debian es ligero en casi todos los aspectos a diferencia de otros sistemas (Windows) que son más rápidos pero sólo en algunas áreas. Aunque ésta es una característica que la mayoría de los sistemas Linux cumple.
Controladores escritos por usuarios de GNU/Linux	Esto permite seguir con el soporte del hardware aun cuando el fabricante halla detenido su producción.
Buena seguridad del sistema	Debian y la comunidad del software libre se esmeran en corregir rápidamente los problemas de seguridad de la distribución.
Software de seguridad	Alta disponibilidad de herramientas de seguridad.

Tabla 4.3 - Desventajas de los sistemas Debian GNU/Linux

Desventajas de usar Debian	Descripción
Falta de software popular	Debian no dispone de algunos paquetes de software populares, debido a que no son gratis, e incluirlos significaría un costo para el usuario final
Debian es difícil de configurar	Si bien la instalación puede ser más fácil que la de Windows, configurar por ejemplo una impresora puede no ser una tarea trivial en Debian y los sistemas Linux en general
No todo el hardware está soportado	Sólo el hardware viejo, raro o muy reciente no está soportado, o bien aquel hardware complejo que el fabricante genera sólo para sistemas Windows

Cabe destacar otra ventaja muy importante, Debian es totalmente gratis. En la Figura 4.11, se las diversas distribuciones que han tomado como sistema base a Debian, entre las que destacan; Ubuntu, Knoppix y Linux Mint esto refleja la gran popularidad y confianza que la comunidad "Linux" tiene sobre esta distribución, como se observa en la Figura 4.12 - Línea de tiempo de distribuciones desprendidas de Debian GNU/Linux, existen 94 distribuciones oficiales que se desprenden de Debian. [99]

Figura 4.11 – Distribuciones que tienen como base Debian GNU/Linux.

99 Distribuciones de Linux (2012). Obtenido el 26 de septiembre de 2012, de http://www.argentinawarez.com/linux-y-gnu/2407598-distribuciones-linux.html.

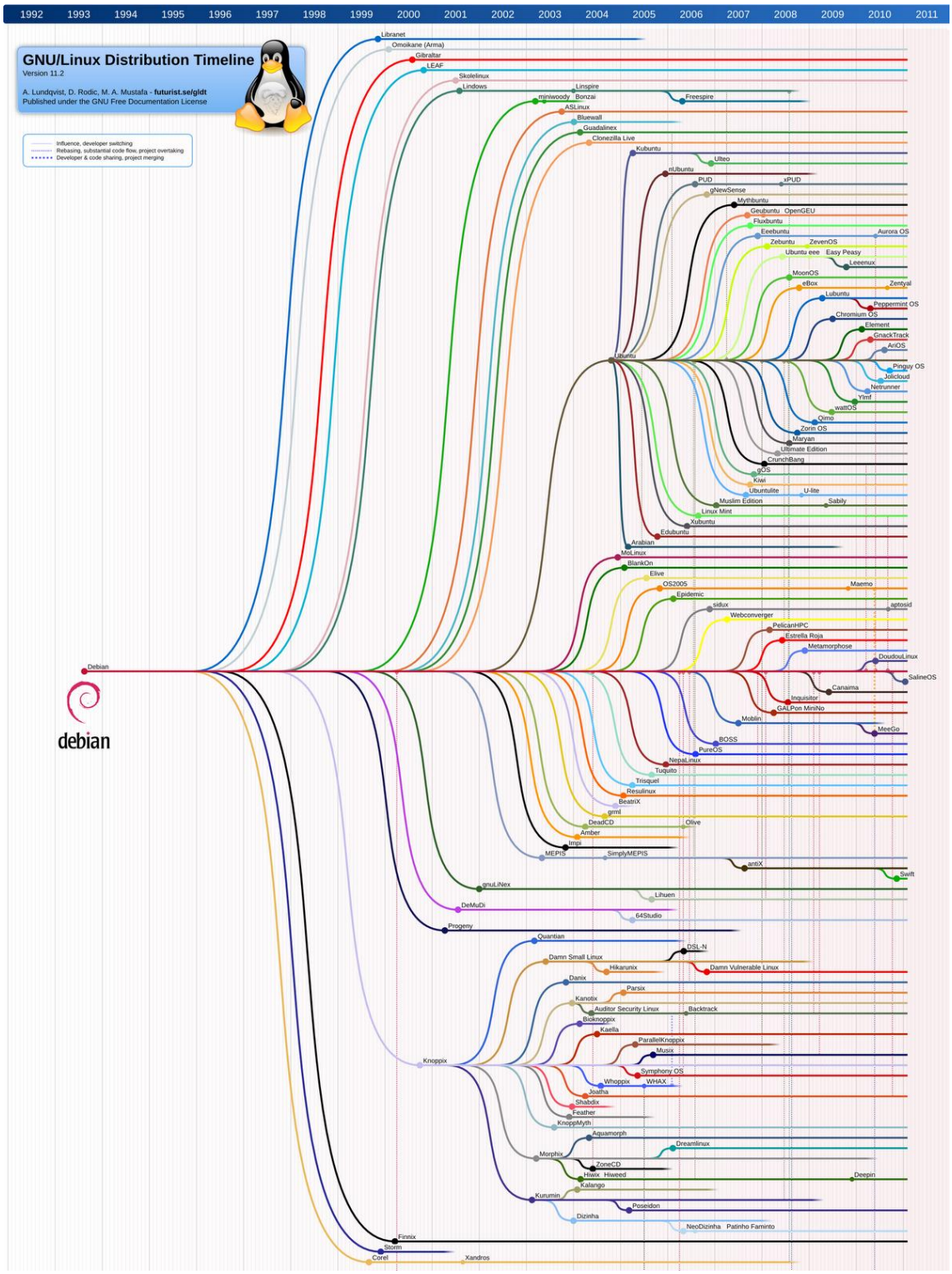


Figura 4.12 - Línea de tiempo de distribuciones desprendidas de Debian GNU/Linux. [100]

100 Línea de tiempo de Debian (n.d.). Obtenida el 26 de septiembre de 2012, de http://linuxgnublog.org/imgblog/relevante.png.

A fin de brindar un mejor panorama de la importancia y trascendencia del proyecto Debian, se presenta la Figura 4.13, la cual geo-localiza en un mapa mundial los desarrolladores oficiales del proyecto, se destaca el país de E.U. y países de Europa del Este. En México se cuenta con dos desarrolladores oficiales: **GunnarWolf** (gwolf@jabber.org) y **Raphael Geissert** (atomo64@gmail.com) más información en el sitio Web: http://db.debian.org.

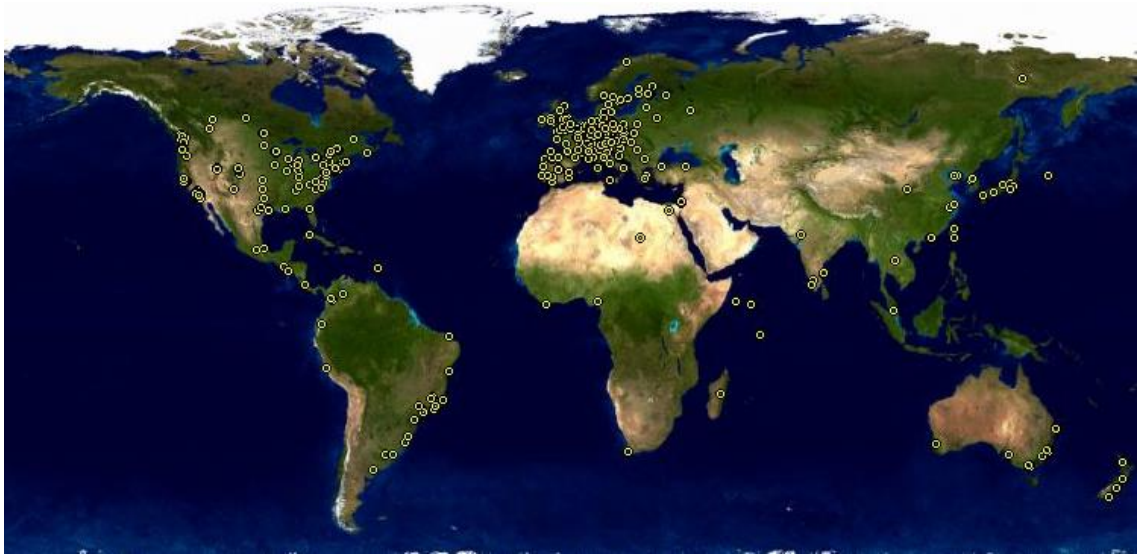


Figura 4.13 - Desarrolladores del proyecto Debian a nivel mundial. [101]

4.5 Hardening y buenas prácticas de seguridad

Hacerle la vida difícil a un intruso o atacante es el concepto que está detrás del *hardening*, es decir, su propósito es entorpecer al intruso y ganar tiempo para minimizar las consecuencias de un inminente incidente de seguridad y de ser posible impedir que se concrete.

Dicho lo anterior, el *hardening* es una acción compuesta por un conjunto de actividades realizadas por el administrador de un sistema operativo con la intención de fortalecer la seguridad de los equipos mediante la eliminación de software, servicios, usuarios innecesarios en el sistema, así como la aplicación de herramientas y técnicas que contribuyan a reducir las vulnerabilidades en los sistemas.

Es importante aclarar que el hardening es un proceso y no una tarea, lo cual significa que una vez iniciado no termina, como se comentó en el Capítulo 1, el equipo de cómputo más seguro es aquel que está apagado, desconectado de la red y protegido de accesos físicos, y aun así es inseguro. Desafortunadamente, lo anterior no es una solución práctica para las necesidades tecnológicas modernas, lo cual refleja la necesidad de resaltar que el *hardening* no necesariamente devolverá equipos "invulnerables" y 100% seguros, pues según el modelo de defensa en profundidad

[101] Desarrolladores a nivel mundial (2012). Obtenido el 27 de septiembre de 2012, de http://www.debian.org/devel/developers.loc.

(*Defense in Depth*) de Microsoft [102] la capa de *host* (ver Figura 4.14) es sólo una de tantas capas que deberían ser aseguradas en cualquier infraestructura tecnológica, a pesar de ello no serán cubiertas en este trabajo pues su explicación a detalle rebasaría el alcance del mismo.

Figura 4.14 - Descripción del Modelo Defensa en profundidad de Microsoft.

Es común que la mayoría de los sistemas informáticos que requieren ser configurados de manera segura, estén en ambientes de producción para implementar el *hardening,* en estos casos es similar a intentar reparar las brechas de una autopista y al mismo tiempo esquivar los autos que pasan a gran velocidad. La forma correcta de aplicarlo es hacerlo sobre un equipo de cómputo fuera de producción y es aquí donde surge la pregunta ¿hasta qué punto el *hardening* es una ayuda y no un obstáculo?, esta situación refleja uno de los principales paradigmas de la seguridad de la información y retos del *hardening*, concerniente al hecho de que la mayoría de los sistemas, protocolos y aplicaciones son desarrollados para funcionar y cumplir con un propósito específico, dejando en segundo plano la seguridad, es decir, a medida que se mejora la seguridad de los sistemas informáticos, la funcionalidad, flexibilidad, simplicidad y facilidad de manejo de los mismos se ve afectada, puesto que las decisiones que pueden tomar los usuarios se reducen significativamente al mínimo y de igual forma la probabilidad de que el usuario se equivoque y ponga en peligro la seguridad del sistema se reduce.

Un aspecto importante antes de aplicar el *hardening* de cualquier sistema, es conocer de quién se quiere proteger el sistema, si bien un sistema debe protegerse de todo, no siempre es práctico hacerlo de esta forma, para ello primero se debe analizar el sistema para entender qué se está protegiendo, por qué se está protegiendo, cuál es su criticidad y sus vulnerabilidades, de esta forma entender el papel que desempeña el sistema dentro de la organización y posteriormente identificar los actores que podrían comprometer la seguridad del sistema, las amenazas cibernéticas normalmente

[102] Comprender la defensa en seguridad (2008). Obtenido el 27 de octubre de 2012, de http://www.microsoft.com/latam/technet/articulos/articulos_seguridad/2008/junio/sm0608.mspx.

provienen de alguien con la motivación de obtener acceso no autorizado a la red organizacional, a continuación se muestran los tipos intrusos más comunes. [103]

1. **El curioso:** Este tipo de intruso, básicamente está interesado en averiguar información del sistema y los datos contenidos en él.

2. **El Malicioso:** El intruso trata de afectar la operatividad de los sistemas, o bien modificar el contenido de una página web (*defacement*), provocando con ello que la parte afectada pierda tiempo y dinero en recuperarse de los daños sufridos.

3. **El Intruso de Alto Perfil**: Este tipo de intruso trata de vulnerar la seguridad de un sistema para lograr popularidad y reconocimiento dentro del gremio.

4. **El intruso de competencia**: El cual está interesado en los datos que tiene el sistema. Podría ser alguien que pretenda obtener un beneficio financiero si consigue información de valor para un tercero.

5. **El Borrower**: Es un intruso que está interesado en comprometer al sistema para aprovecharse de sus recursos y emplearlos a favor del comercio que convenga a sus propios intereses.

6. **El Saltador:** Este tipo de intruso está interesado en utilizar el sistema como trampolín para vulnerar otros sistemas de la red, para lo cual se aprovecha de las relaciones de confianza que otros equipos tienen con éste.

Una vez que se tiene claro de quién(es) se quiere proteger el sistema, se debe tener en cuenta que el proceso de *hardening* es indispensable en la etapa de erradicación del proceso de manejo de incidentes (acción reactiva), y aún más importante como medida de prevención de incidentes, donde obviamente no debe ser considerado como el único mecanismo o técnica a emplear para salvaguardar la seguridad de los sistemas de información.

4.6 Mapeo de buenas prácticas de seguridad del ISO/IEC 27002 en el sistema Operativo Linux-Debian

En las Tablas desde la 4.4 hasta la 4.10 se muestra la extracción de los controles del ISO/IEC 27001 factibles para cumplir parcialmente o en su totalidad con el control de seguridad para el *hardening* en sistemas Debian GNU/Linux. En el capítulo 5, estos controles serán referenciados en cada uno de los puntos del *hardening*, según corresponda, con la finalidad de que todos sean cubiertos.

[103] Linux Security HOWTO, What Are You Trying to Protect? (2004). Obtenido el 03 de septiembre de 2012, de http://www.linuxdoc.org/HOWTO/Security-HOWTO/x82.html#AEN85.

Tabla 4.4 - Controles de ISO/IEC 27002 óptimos para el hardening en un equipo de cómputo. A.8 Gestión de Activos.

A.8 Gestión de Activos					
Objetivo de control	**Control de seguridad**	**Estado**	**Amenaza**	**Vulnerabilidad**	**Implementación del control de seguridad**
A.8.1 Responsabilidad sobre los activos	A.8.1.4 Devolución de los activos.	Aplica	-Técnicas forenses para la recuperación de información restringida.	-Falta de herramientas que permitan el borrado seguro de información restringida.	-Uso de herramientas para el borrado a bajo nivel.
A.8.3 Manejo de los medios de almacenamientos	A.8.3.2 Eliminación de medios.	Aplica	-Técnicas forenses para la recuperación de información restringida.	-Falta de herramientas que permitan el borrado seguro de información restringida.	-Uso de herramientas para el borrado a bajo nivel.
	A.8.3.3 Medios físicos en tránsito.	Aplica	-Divulgación o uso no autorizado de la información.	-Acceso a la información en claro.	-Implementación de mecanismos de cifrado de disco duro y USB.

Tabla 4.5 - Controles de ISO/IEC 27002 óptimos para el hardening en un equipo de cómputo.A.9 Control de accesos.

A.9 Control de accesos					
Objetivo de control	**Control de seguridad**	**Estado**	**Amenaza**	**Vulnerabilidad**	**Implementación del control de seguridad**
A. 9.1 Requisitos de negocio para el control de accesos	A.9.1.2 Control de acceso a las redes y servicios asociados.	Aplica	a) Acceso remoto de usuarios no autorizados. b) Conexión a redes distintas e inseguras que no pertenecen a la organización.	a) Fallas en la autenticación de usuarios remotos. b) Permitir al usuario la configuración de parámetros de conexión a la red.	a) Implementar mecanismo que validen que el cliente es quien dice ser, implementación de llaves públicas y privadas, así como listas de control de acceso. b) Eliminar la posibilidad de configuración del usuario normal de los parámetros de conexión a la red.

A.9.2 Gestión de acceso de usuario	A.9.2.1 Gestión de altas/bajas en el registro de usuarios.	Aplica	-Que existan cuentas de usuarios en el sistema sin justificación o que debieran de haberse dado de baja, lo que permitiría que se realice un mal uso del sistema y de la información.	-Falta de un control de cuentas de usuario para la creación, modificación y eliminación, así como la vigencia de las mismas en el sistema.	-Configuración de mensajes con las implicaciones del mal uso del sistema a los usuarios al iniciar sesión de forma remota o local. -Limitar y revisar los registros frecuentemente de la creación y eliminación de cuentas de usuario. -Implementar vigencia e inactividad de las cuentas de usuario.
	A.9.2.2 Gestión de los derechos de acceso asignados a usuarios.	Aplica	-Que usuarios a causa de mal uso de los privilegios asignados en el sistema tengan acceso de lectura, escritura o ejecución a archivos que no debieran tener acceso.	-Asignación incorrecta o genérica de permisos sobre archivos del sistema o aplicaciones. -Carencia de mecanismos para la implementación de políticas de acceso de las cuentas de usuario.	-Revisión y validación de permisos de las cuentas de usuario y de aplicación críticos del sistema. -Implementación de mecanismos en la gestión de cuentas de usuario y políticas asociadas a las cuentas.
	A.9.2.3 Gestión de los derechos de acceso con privilegios especiales.	Aplica	-Que usuarios a causa de mal uso de los privilegios asignados en el sistema tengan acceso de lectura, escritura o ejecución a archivos que no debieran tener acceso.	-Asignación incorrecta o genérica de permisos sobre archivos del sistema o aplicaciones. -Carencia de mecanismos para la implementación de políticas de acceso de las cuentas de usuario.	-Revisión y validación de permisos de las cuentas de usuario y de aplicación críticos del sistema. -Implementación de mecanismos en la gestión de cuentas de usuario y políticas asociadas a las cuentas.

	A.9.2.4 Gestión de información confidencial de autenticación de usuarios.	Aplica	-Un usuario no autorizado ingresa al sistema de forma remota para sabotearlo o comprometer la seguridad del mismo o de la información.	Falta de mecanismos que prohíban: -El uso de contraseñas débiles. -Reutilización de contraseñas. -El uso de contraseñas por más de 3 meses.	-Instalar o configurar herramientas que permitan utilizar contraseñas robustas, obligar al cambio de contraseñas periódico y habilitar el histórico de las mismas.
	A.9.2.6 Retirada o adaptación de los derechos de acceso.	Aplica	-Que existan cuentas de usuarios en el sistema sin justificación o que debieran de haberse dado de baja, lo que permitiría que se realice un mal uso del sistema y de la información.	-Falta de un control de cuentas de usuario para la creación, modificación y eliminación, así como la vigencia de las mismas en el sistema.	-Instalar o configurar parámetros del sistema que permitan la gestión de los derechos y permisos de los usuarios en el sistema
A.9.3 Responsabilidad del usuario	A.9.3.1 Uso de información confidencial para la autenticación.	Aplica	-Un intruso se aprovecha de las contraseñas débiles en el sistema.	-Falta de mecanismos de supervisión continua de la complejidad de las contraseñas utilizadas por los usuarios del sistema, así como para la expiración o desactivación.	-Instalar o configurar aplicaciones que impidan que los usuarios utilicen contraseñas con complejidad baja, las cuales puedan ser vulneradas por ataques de diccionario o fuerza bruta. Restricción de acceso a la BIOS y arranque del sistema.
A.9.4 Control de acceso a sistemas y aplicaciones	A.9.4.1 Restricción del acceso a la información.	Aplica	-Acceso no autorizado a la información restringida.	-Falta de controles de acceso a la información.	-Asignar permisos adecuados y necesarios para las actividades de los usuarios de acuerdo con sus roles. -Habilitar la caducidad de las cuentas de usuarios.
	A.9.4.2 Procedimientos seguros de inicio de sesión.	Aplica	a) Que un usuario no autorizado obtenga acceso al sistema por medio de un ataque de fuerza bruta o de diccionario.	a) Deficiencia de mecanismos para la identificación de intentos exitosos y fallidos de inicio de sesión o conexión remota.	a) Implementar mecanismos para: -Identificar y registrar intentos de inicios de sesión exitosos y fallidos.

			b) Un intruso podría aprovecharse de una sesión local o remota desprotegida. c) Un usuario válido ingresa al sistema fuera del horario laboral. d) Acceso de usuarios no autorizados al sistema.	b) Sesiones abiertas no se desactivan después de un tiempo determinado sin actividad. c) Falta del uso adecuado de las listas de control de acceso que restrinjan el ingreso en determinado tiempo. d) Cuentas de usuario y contraseñas compartidas entre usuarios.	-Bloqueo de cuentas y direcciones IP con más de 5 intentos fallidos. b) Las sesiones deberán ser desactivadas tras un periodo de inactividad definido, requiriendo que el usuario proporcione nuevamente las credenciales de acceso. c) Modificar archivos de configuración del sistemas para: -Restringir parcialmente el acceso a cierto tipo de contenido a ciertos horarios. -Limitar el tiempo que está disponible la pantalla de login para SSH. d) La creación de cada cuenta de usuario debe tener un identificador único para la autenticación adecuada y verificación de la identidad de los mismos.
	A.9.4.3 Gestión de contraseñas de usuario.	Aplica	-Ataques de fuerza bruta o diccionario.	-Uso de contraseñas débiles y carencia de mecanismos para gestionar la caducidad de las mismas.	-Instalar o configurar herramientas que permitan utilizar contraseñas robustas

					-Obligar al cambio de contraseñas periódico y habilitar el histórico de contraseñas.
	A.9.4.4 Uso de herramientas de administración de sistemas.	Aplica	a) Derivado de la instalación de software no permitido o autorizado, se podrían comprometer los recursos y el sistema. b) Un intruso podría aprovecharse de las consolas virtuales y puertos para ejecutar más de una acción a la vez en el sistema.	a) Permitir la instalación de software a cualquier usuario, sin limitar los privilegios de root. b) No proteger el acceso a las consolas virtuales y puertos con contraseña o bien no desactivarlas.	a) Limitar los servicios habilitados, los puertos locales abiertos y los servicios en ejecución, así como limitar los privilegios de ejecución de los usuarios. b) Cerrar las consolas físicas y desactivar los puertos que no se utilizan.
	A.9.4.5 Control de acceso al código fuente de los programas.	Aplica	-Códigos fuente de programas accesibles por usuarios no autorizados.	-Falta de mecanismos de protección y resguardo de códigos fuente críticos para el sistema o producto de la operación.	-Configuración adecuada de privilegios de acceso y uso de herramientas que prohíban la lectura, escritura y modificación de archivos críticos del sistema o la operación.

Tabla 4.6 - Controles de ISO/IEC 27002 óptimos para el hardening en un equipo de cómputo. A.11 Seguridad Física y ambiental.

A.11 Seguridad Física y ambiental					
Objetivo de control	**Control de seguridad**	**Estado**	**Amenaza**	**Vulnerabilidad**	**Implementación del control de seguridad**
A.11.2 Seguridad de los equipos	A.11.2.4 Mantenimiento de los equipos.	Aplica	-Fuga de información. -Acceso indebido al equipo de cómputo.	-Instalación de programas innecesarios. - Almacenamiento de información en registros de actividad en Internet.	-Eliminar programas y servicios innecesarios. -Borrar la memoria caché, historia de Internet y archivos temporales.

	A.11.2.5 Salida de activos fuera de las dependencias de la empresa.	Aplica	-Fuga de información.	-No se encuentra dentro del perímetro de seguridad.	-Listas de control de acceso y asignación de permisos restringidos, para impedir copiar información o software. -Cifrado de particiones que almacenan software restringido .
	A.11.2.6 Seguridad de los equipos y activos fuera de las instalaciones.	Aplica	-Fuga o robo de información. -Acceso indebido al equipo de cómputo.	-Exposición de puertos de conexión alámbricos e inalámbricos.	-Apagar conexiones bluetooth o inalámbricas. -No dejar los equipos desatendidos. -Protección de las consolas virtuales y físicas.
	A.11.2.7 Reutilización o retirada segura de dispositivos de almacenamiento .	Aplica	-Fuga de información.	Almacenamiento de la información en claro. -Formateo rápido (alto nivel) de los dispositivos de almacenamiento .	-Formateo de bajo nivel de los dispositivos de almacenamiento. Instalación de herramientas para el borrado seguro.
	A.11.2.8 Equipo informático desatendido de usuario.	Aplica	-Un intruso podría aprovechar un descuido para acceder de forma física a un sistema y su información.	-Falta de un procedimiento para el bloqueo de pantalla después de haber transcurrido determinado tiempo.	-Configurar el bloqueo automático de la pantalla o sesión (consola o terminal) tras haber transcurrido 3 minutos de inactividad. Así como proteger al sistema de reinicios no autorizados.
	A.11.2.9 Política de escritorio pantalla limpia.	Aplica	-Un intruso podría aprovechar un descuido por parte del usuario para observar los documentos ubicados en el “Escritorio” de la cuenta del usuario.	-El sistema permite que el usuario almacene archivos en el escritorio sin restricción alguna.	-Configurar archivos del sistema para que la carpeta de Escritorio de los usuarios tenga únicamente permisos de lectura.

Tabla 4.7 - Controles de ISO/IEC 27002 óptimos para el hardening en un equipo de cómputo. A.12 Seguridad de las operaciones.

A.12 Seguridad de las operaciones					
Objetivo de control	**Control de seguridad**	**Estado**	**Amenaza**	**Vulnerabilidad**	**Implementación del control de seguridad**
A.12.1 Responsabilidades y procedimientos de operación	A.12.1.3 Gestión de capacidades.	Aplica	-Consumo de los recursos del sistema.	-El uso excesivo de los recursos del sistema podría provocar una negación de los servicios.	-Herramientas que permitan monitorear los recursos y capacidades para el funcionamiento óptimo del sistema. Instalación del sistema en particiones separadas con capacidades de almacenamiento de acuerdo a las funciones.
A.12.2 Protección contra código malicioso	A.12.2.1 Controles contra código malicioso.	Aplica	-Comprometer la seguridad del sistema a causa de un código malicioso.	-Ausencia de mecanismos que permitan detectar y proteger el sistema de códigos maliciosos.	-Instalación y actualización de antivirus. -Mecanismos para limitar la instalación de software inseguro o malicioso. -Verificar la integridad de archivos críticos del sistema por medio de un *hash* bien conocido.
A.12.3 Copias de seguridad	A.12.3.1 Copias de seguridad de la información.	Aplica	-Que un usuario o alguna aplicación del sistema elimine o sobre escriba información crítica de forma intencional o no intencional.	-Ausencia de procedimientos periódicos de respaldo de información.	-Instalar una herramienta y realizar un procedimiento para el respaldo de información crítica del sistema.
A.12.4 Registro de actividad y supervisión	A.12.4.1 Registro y gestión de eventos de actividad.	Aplica	a) Borrado o modificación de los archivos que registran la actividad en el sistema (bitácoras).	a) Falta o mala asignación de permisos de lectura, escritura o ejecución a los archivos que registran los eventos en el sistema.	a) Habilitar el registro de eventos y asignar permisos a los archivos de registros, así como mantener el respaldo periódico de los mismos.

			b) Ataques al sistema que no son registrados, analizados y bloqueados. c) Fallos del sistema inesperados o provocados por un intruso.	-Se encuentra deshabilitado el registro de eventos en el sistema. b) Falta de mecanismos para la detección y revisión de actividades maliciosas ocurridas en el sistema. c) Inexistente mecanismo de supervisión para el correcto funcionamiento del sistema.	b) Instalación de un Sistema Detector de Intrusos (IDS) a nivel de *Host* que envíe alertas al administrador y en su defecto implemente acciones propias de un Sistema de Prevención de Intrusos (IPS). -Así como configuración de *firewall* interno. c) Instalación de herramientas que registren y notifiquen las fallas críticas del sistema o cambios en la integridad de archivos críticos para el sistema.
	A.12.4.2 Protección de los registros de información.	Aplica	-Accesos no autorizados.	-Falta de mecanismos para la protección de los registros del sistema.	-Modificación de permisos en los archivos de registro de actividad en el sistema.
	A.12.4.3 Registros de actividad del administrador y operador del sistema	Aplica	a) Accesos no autorizados. b) Acciones maliciosas en el sistema por parte de Administradores y operadores.	a) Falta de mecanismos para la protección de los registros del sistema. b) Falta de mecanismos para el monitoreo de actividades de los usuarios del sistema.	a) Modificación de permisos en los archivos de registro de actividad en el sistema. b) Implementación de mecanismos que registren la actividad del administrador y operadores del sistema.

	A.12.4.4 Sincronización de relojes.	Aplica	-Derivado de la inconsistencia de la hora del sistema se podría afectar el funcionamiento de aplicaciones y se dificultaría el rastreo respecto de la línea de tiempo de un intruso.	-Que los relojes de los sistemas no estén configurados para sincronizarse con una fuente acordada y confiable.	-Sincronización de los relojes con una fuente externa y única.
A. 12.5 Control del software en explotación	A. 12.5.1 Instalación del software en sistemas en producción.	Aplica	-Instalación de software malicioso desde fuentes no confiables.	-Carencia de selección de repositorios seguros para la instalación del software en el sistema.	-Configurar los repositorios oficiales y seguros del sistema para la instalación y actualización del software. Uso de máquinas virtuales para la instalación de software de pruebas.
A.12.6 Gestión de la vulnerabilidad técnica	A.12.6.1 Gestión de las vulnerabilidades técnicas.	Aplica	-Exploits de día cero, así como nuevas técnicas para comprometer sistemas de información.	-Falta de procedimientos para la instalación de parches y actualización de versiones de software vulnerable.	-Implementación de mecanismos para las actualizaciones de software, además de la gestión de parches para el sistema.
	A. 12.6.2 Restricciones en la instalación de software.	Aplica	-Un paquete o aplicación maliciosa puede instarse en el sistema y comprometer su seguridad.	-Deficiencia del sistema para validar que los repositorios de software sean confiables. -Carencia de herramientas o aplicaciones que realicen el proceso de instalación de software de forma más eficiente.	-Configuración y modificación de parámetros de seguridad para evitar que se instale software potencialmente malicioso proveniente de repositorios no confiables. -Instalación de herramientas para la gestión de paquetes y aplicaciones del sistema.

A.12.7 Consideraciones de las auditorías de los sistemas de información	A. 12.7.1 Controles de auditoría de los sistemas de información.	Aplica	-Que el proceso de auditoría interfiera la operación.	-Falta de planificación y requisitos para el proceso de auditoría en el sistema.	-Uso de mecanismos automatizados para proteger los accesos a las herramientas de auditoría a fin de prever cualquier posible mal uso.

Tabla 4.8 - Controles de ISO/IEC 27002 óptimos para el hardening en un equipo de cómputo. A.13 Seguridad en las telecomunicaciones

A.13 Seguridad en las telecomunicaciones					
Objetivo de control	**Control de seguridad**	**Estado**	**Amenaza**	**Vulnerabilidad**	**Implementación del control de seguridad**
A.13.1 Gestión de la seguridad en las redes	A.13.1.2 Mecanismos de seguridad asociados a servicios en red.	Aplica	-Ataques de reconocimiento y escaneo no conocidos en el sistema.	-Servicios de red sin configuración de seguridad.	-Configuración segura de servicios y kernel. -Desinstalación o desactivación de servicios que comprometan la: disponibilidad, confidencialidad e integridad de la información.

Tabla 4.9 - Controles de ISO/IEC 27002 óptimos para el hardening en un equipo de cómputo. A.14 Adquisición desarrollo y mantenimiento de los sistemas de información.

A.14 Adquisición desarrollo y mantenimiento de los sistemas de información					
Objetivo de control	**Control de seguridad**	**Estado**	**Amenaza**	**Vulnerabilidad**	**Implementación del control de seguridad**
A.14.2 Seguridad en los procesos de desarrollo y soporte	A.14.2.6 Seguridad en entornos de desarrollo.	Aplica	-Un usuario o intruso malintencionado podría modificar o eliminar parámetros de configuración del sistema y comprometer la seguridad del sistema.	-Falta de restricciones en el sistema que impidan que un usuario acceda a localidades del sistema no autorizadas ni necesarias para realizar sus actividades.	-Configuración de “jaulas” en el sistema para aislar los entornos de trabajo del resto de los archivos críticos del sistema.

Tabla 4.10 - Controles de ISO/IEC 27002 óptimos para el hardening en un equipo de cómputo. A.16 Gestión de incidentes en la seguridad de la información.

A.16 Gestión de incidentes en la seguridad de la información					
Objetivo de control	**Control de seguridad**	**Estado**	**Amenaza**	**Vulnerabilidad**	**Implementación del control de seguridad**
A.16.1 Gestión de incidentes de seguridad de la información y mejoras	A.16.1.2 Notificación de los eventos de seguridad de la información.	Aplica	-Ataques cibernéticos con intención de obtener acceso vía remota o local al sistema.	-Deficiencia de mecanismos para identificar y analizar los eventos de seguridad del sistema y poder identificar de forma eficientemente eventos que pudieran poner en riesgo la continuidad de la operación.	-Instalar o configurar herramientas para el monitoreo y seguimiento de logs de seguridad del sistema.
	A.16.1.3 Notificación de puntos débiles de la seguridad.	Aplica	-Código malicioso o Ataques cibernéticos de usuarios del sistema con intención de comprometer la seguridad del sistema.	-Carencia de mecanismos de supervisión de localidades criticas del sistema, indispensables para la operación y seguridad de la información procesada y almacenada.	-Instalación o configuración de herramientas para él envió de alertas en tiempo real sobre aspectos de seguridad que deban ser atendidos de forma inmediata.

Capítulo 5

Aseguramiento de sistemas LINUX-DEBIAN

Los controles del ISO/IEC 27002:2013 que aplican para el aseguramiento de equipos con sistema operativo Linux-Debian pertenecientes a un CERT, se mencionan en la Tabla 5.1.

Tabla 5.1 – Controles cubiertos en el *hardening*

Objetivo de control	Control de seguridad
A.8 Gestión de Activos	
A.8.1 Responsabilidad sobre los activos	A.8.1.4 Devolución de los activos
A.8.3 Manejo de los medios de almacenamientos	A.8.3.2 Eliminación de medios.
	A.8.3.3 Medios físicos en tránsito.
A.9 Control de accesos	
A. 9.1 Requisitos de negocio para el control de accesos	A.9.1.2 Control de acceso a las redes y servicios asociados
A.9.2 Gestión de acceso de usuario	A.9.2.1 Gestión de altas/bajas en el registro de usuarios.
	A.9.2.2 Gestión de los derechos de acceso asignados a usuarios.
	A.9.2.3 Gestión de los derechos de acceso con privilegios especiales.
	A.9.2.4 Gestión de información confidencial de autenticación de usuarios.
	A.9.2.6 Retirada o adaptación de los derechos de acceso.
A.9.3 Responsabilidades del usuario	A.9.3.1 Uso de información confidencial para la autenticación.
A.9.4 Control de acceso a sistemas y aplicaciones	A.9.4.1 Restricción del acceso a la información.
	A.9.4.2 Procedimientos seguros de inicio de sesión.
	A.9.4.3 Gestión de contraseñas de usuario.
	A.9.4.4 Uso de herramientas de administración de sistemas.
	A.9.4.5 Control de acceso al código fuente de los programas.
A.11 Seguridad Física y ambiental	
A.11.2 Seguridad de los equipos	A.11.2.4 Mantenimiento de los equipos.
A.11.2 Seguridad de los equipos	A.11.2.5 Salida de activos fuera de las dependencias de la empresa
	A.11.2.6 Seguridad de los equipos y activos fuera de las instalaciones.
	A.11.2.7 Reutilización o retirada segura de dispositivos de almacenamiento.
	A.11.2.8 Equipo informático desatendido de usuario.
	A.11.2.9 Política de escritorio pantalla limpia.
A.12 Seguridad de las operaciones	
A.12.1 Responsabilidades y procedimientos de operación	A.12.1.3 Gestión de capacidades.
A.12.2 Protección contra código malicioso	A.12.2.1 Controles contra código malicioso.
A.12.3 Copias de seguridad	A.12.3.1 Copias de seguridad de la información.
A.12.4 Registro de actividad y supervisión	A.12.4.1 Registro y gestión de eventos de actividad.
	A.12.4.2 Protección de los registros de información.
	A.12.4 3 Registros de actividad del administrador y operador del sistema
	A.12.4 4 Sincronización de relojes.
A. 12.5 Control del software en explotación	A. 12.5.1 Instalación del software en sistemas en producción
A.12.6 Gestión de la vulnerabilidad técnica	A.12.6.1 Gestión de las vulnerabilidades técnicas.
	A. 12.6.2 Restricciones en la instalación de software.
A.12.7 Consideraciones de las auditorías de los sistemas de información	A. 12.7.1 Controles de auditoría de los sistemas de información.
A.13 Seguridad en las telecomunicaciones	
A.13.1 Gestión de la seguridad en las redes	A.13.1.2 Mecanismos de seguridad asociados a servicios en red.
A.14 Adquisición desarrollo y mantenimiento de los sistemas de información	
A.14.2 Seguridad en los procesos de desarrollo y soporte	A.14.2.6 Seguridad en entornos de desarrollo.

A.16 Gestión de incidentes en la seguridad de la información	
A.16.1 Gestión de incidentes de seguridad de la información y mejoras	A.16.1.2 Notificación de los eventos de seguridad de la información.
	A.16.1.3 Notificación de puntos débiles de la seguridad.

5.1 Antes y durante la instalación del sistema

En este capítulo se mencionan las configuraciones para fortalecer la seguridad del sistema operativo Linux-Debian (*hardening),* las cuales se realizaron con la versión "debian-7.1.0-i386-xfce-CD.iso" descargada desde el sitio Web oficial de *Debian*: http://www.debian.org/CD/http-ftp/. La instalación se realizó en una máquina virtual utilizando el software de virtualización VMWare versión 5.0 para sistemas MAC OS X.

5.1.1 Proteger el Acceso al BIOS (Basic Input/Output System)

Cumple con A. 9.3.1 - Uso de información confidencial para la autenticación. [104]

En sistemas operativos *Linux* el flujo de control durante el arranque inicia en la *BIOS*, después pasa el control al gestor de arranque (*bootloader: Master Boot Record y GRUB*) y posteriormente al núcleo (*kernel*) [105] como se observa en la Figura 5.1.

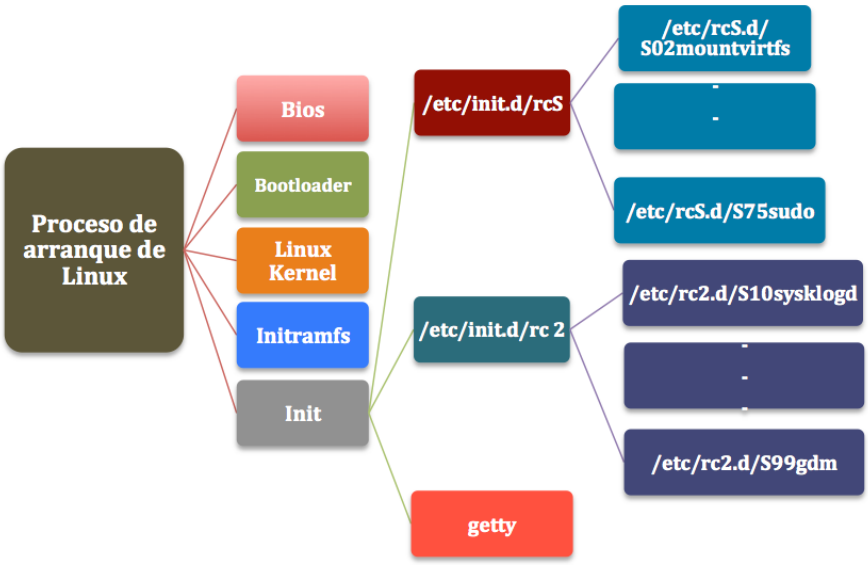


Figura 5.1 – Flujo del proceso de arranque del sistema operativo.

Para proteger el acceso y modificación del *BIOS,* se asigna una contraseña para limitar el acceso a las opciones de configuración y para el inicio con el gestor de arranque del equipo. Antes de que el sistema arranque, se accede a la configuración del *BIOS* (presionando la tecla F2), una vez dentro de la interfaz *PhoenixBIOS Setup Utility* en la pestaña de *Security* la opción *Set User Password,* permite habilitar una

104 El punto 5.1.1 se alinea con el control A.9.3.1 del ISO/IEC 27002:2013

105 El Proceso de Arranque en Linux. Obtenida el 14 de diciembre de 2013, de http://www.linuxyyo.es/el-proceso-de-arranque-en-linux

contraseña para limitar el inicio del gestor de arranque del sistema; la opción *Set Supervisor Password* (ver Figura 5.2), restringe el acceso a la configuración de parámetros de la *BIOS*.

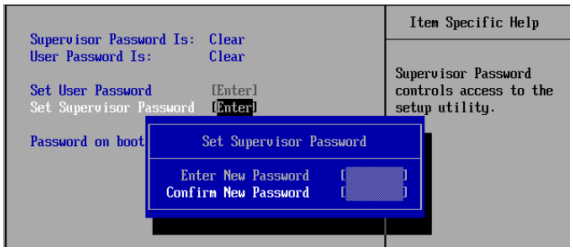


Figura 5.2 – Asignación de contraseña a la BIOS.

En la interfaz *PhoenixBIOS Setup Utility* de la *BIOS* también se pueden desactivar controladores de dispositivos de entrada/salida como el *Floppy* (ver Figura 5.3) que en la actualidad ya no es utilizado, sin embargo, tenerlo habilitado representa un punto de entrada que pudiera ser empleado por un intruso para comprometer al sistema o extraer información.

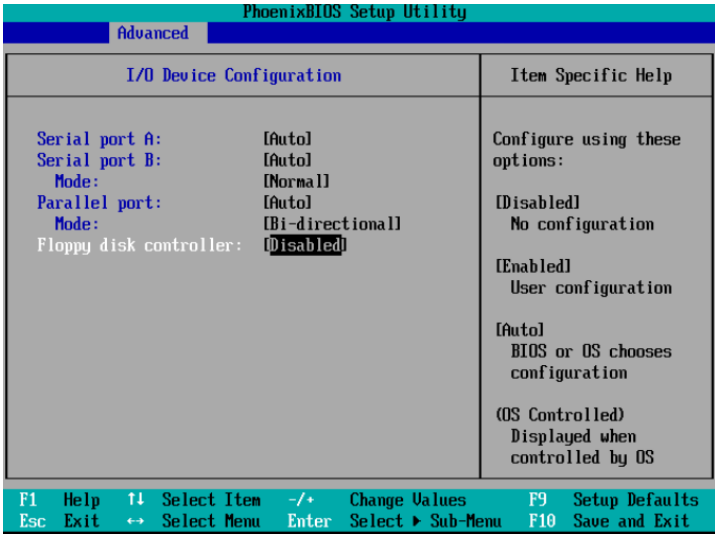


Figura 5.3 – Desactivar dispositivos no utilizados.

Una vez que se ha asignado una contraseña en la opción *Set User Password*, al iniciar el sistema solicita la contraseña para continuar con el arranque, de la misma forma si se desea entrar a la configuración de la BIOS se solicita la contraseña asignada en la opción *Set Supervisor Password* (ver Figura 5.4).

Figura 5.4 – Solicitud de contraseña para arrancar el sistema.

5.1.2 Instalación de LINUX-DEBIAN con particiones separadas

Cumple con A. 12.1.3 - Gestión de capacidades. [106]

Hay dos razones principales para dividir el sistema de archivos en particiones más pequeñas. La primera es el enfoque de seguridad, es decir si algo corrompiera el sistema de archivos solo se afectaría una partición, de esta forma únicamente se tiene que restaurar una porción del sistema y no reinstalarlo todo, por ejemplo, si el problema ocurre en la partición raíz (/) el directorio *home* quedaría intacto en caso de que se reinstale todo el sistema operativo.

La segunda razón se enfoca en la disponibilidad del sistema, si se comenzara a consumir el espacio de almacenamiento en el disco duro, podría provocar que el sistema se paralice por falta de espacio para el almacenamiento de archivos críticos del sistema. En el **Anexo A.I** se muestra el procedimiento que se implementó durante la instalación del sistema operativo *Debian,* con el objetivo de contar con un esquema de particiones separadas que favorece la seguridad del sistema. Una vez completado el procedimiento descrito en el **Anexo A.I**, se tiene un sistema operativo *Linux-Debian* con particiones separadas como se muestra en la Figura 5.5.

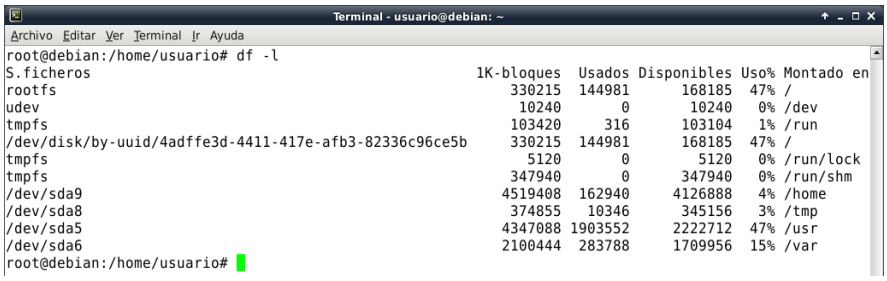


Figura 5.5 – Particiones separadas en Debian.

5.1.3 Asegurar el BootLoader

Cumple con A.8.3.3 - Soportes físicos en tránsito. [107]

Antes de comenzar la instalación de paquetes en *Debian* es necesario indicar a la herramienta APT (*Advanced Packaging Tool* – Herramienta Avanzada de Empaquetado) que no instale paquetes por defecto, de esta forma se brinda confianza al sistema en la instalación de paquetes, con el objeto de que sólo se instale lo necesario y lo proveniente de servidores confiables (ver Figura 5.6).

106 El punto 5.1.2 se alinea con el control A.12.1.3 del ISO/IEC 27002:2013

107 El punto 5.1.3 se alinea con el control A.8.3.3 del ISO/IEC 27002:2013

Figura 5.6 – Aseguramiento de APT.

Para fortalecer la seguridad del *BootLoader*, responsable de cargar el *kernel* en memoria para después ejecutarlo, se instala el paquete *grub-legacy*, después se calcula el *hash* MD5 de una contraseña en claro que será aplicada al gestor de arranque GRUB (*GNU Grand Unified Bootloader*) para restringir su edición, esto se puede realizar con la instrucción `grub-md5-crypt` como se muestra en la Figura 5.7.

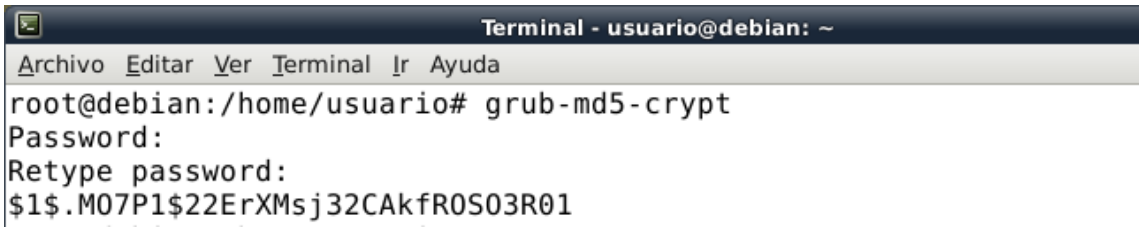


Figura 5.7 – Creación de hash MD5 de contraseña para GRUB.

Posteriormente, se actualiza el GRUB con el comando `update-grub`, en este paso se crea el archivo de configuración "/boot/grub/menu.lst", y se editará el archivo "/boot/grub/menu.lst" como se muestra en la Figura 5.8, agregando la línea "password --md5 hash" y el *hash* que se obtuvo con el comando grub-md5-crypt (ver Figura 5.7). Adicionalmente al final del archivo se agrega la línea: "password to topsecret" para impedir que la contraseña sea mostrada en claro al momento de ser tecleada por el usuario.

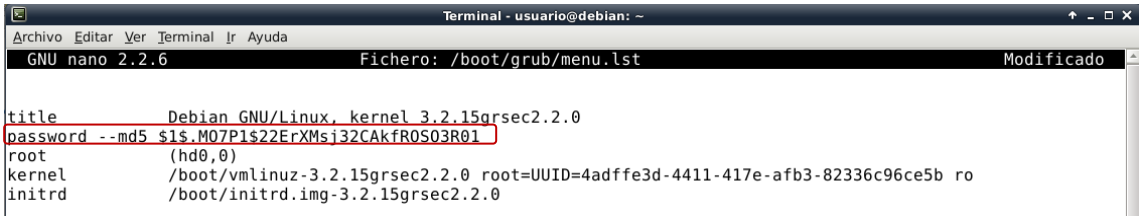


Figura 5.8 – Habilitar la secrecía de la contraseña de GRUB.

En este mismo archivo (menu.lst) se le quita el comentario a la línea que contiene el texto "lockalternative=true" para que la contraseña también se aplique cuando se requiera iniciar en modo "single user" también conocido como *Recovery Mode* (ver Figura 5.9).

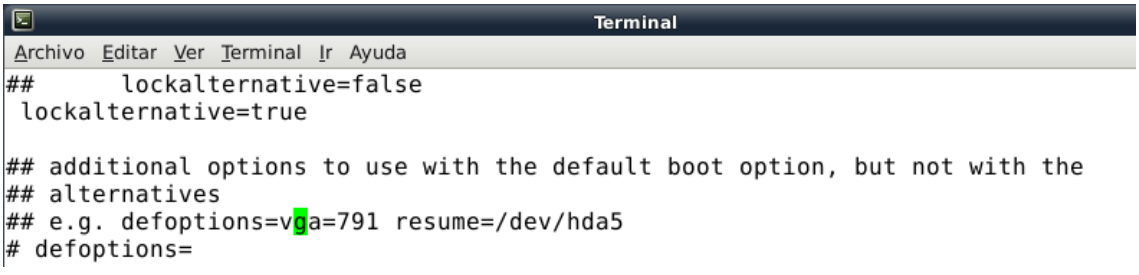


Figura 5.9 – Restricción de GRUB en Modo Recuperación.

Debido a que el archivo "menu.lst" contiene el *hash* de la contraseña de GRUB, es necesario modificar los permisos para que sólo el usuario "*root*" pueda modificar o leer su contenido, lo cual se realiza con el comando `chmod 640 /boot/grub/menu.lst`. En algunas versiones de *Debian* es necesario editar el archivo "/boot/grub/grub.cfg" y

agregar la contraseña para GRUB previamente creada con el comando `grub-mkpasswd-pbkdf2` como se muestra en la Figura 5.10.

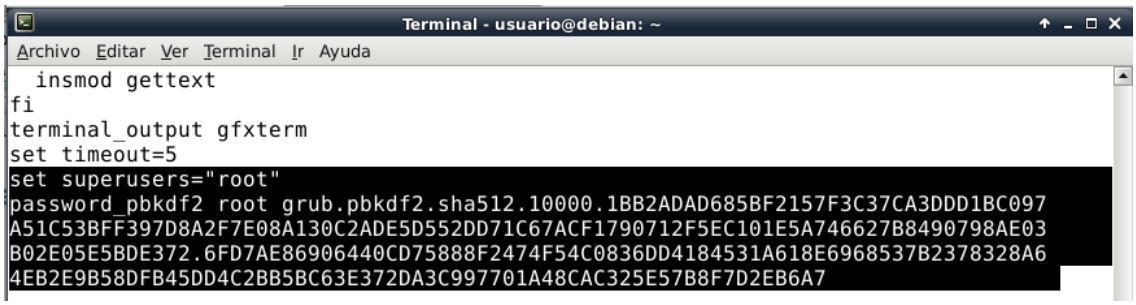


Figura 5.10 –Contraseña en grub.cfg.

La forma de verificar que el sistema solicita una contraseña para editar el GRUB, es reiniciar el sistema y teclear la letra “e” y a continuación el sistema debe pedir el *username* y *password* como se muestra en la Figura 5.11. Una vez que se introduce el usuario y contraseña correctos el sistema permite la edición de GRUB.

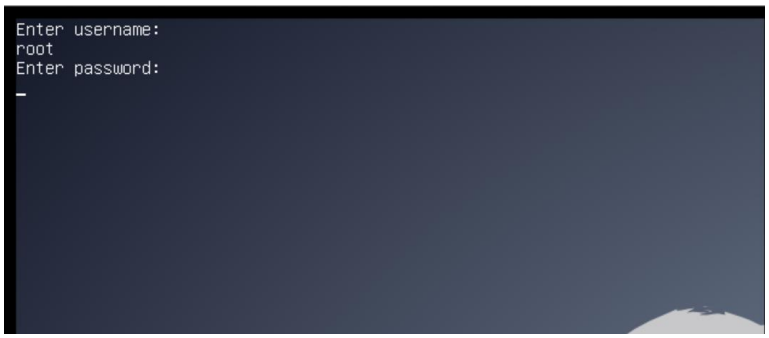


Figura 5.11 – Verificación de aseguramiento del GRUB.

5.1.4 Deshabilitar CTRL-ALT-Del (Reinicio del sistema)

Cumple con A.11.2.8 - Equipo informático de usuario desatendido. [108]

En los equipos *Linux* es muy común que al presionar las teclas *CTRL-ALT-Del* el equipo se reinicie, aunque el sistema esté bloqueado, para evitar que pase esto se debe editar el archivo “/etc/xdg/xfce4/xfconf/xfce-perchannel-xml/xfce4-keyboard-shortcuts.xml” (ver Figura 5.12) y sustituir el valor de *Ctrl-Alt-Delete* (xflock4) por “xflock4-taskmanager”, con ello al teclear *Ctrl-Alt-Delete,* el sistema lanza al administrador de tareas.

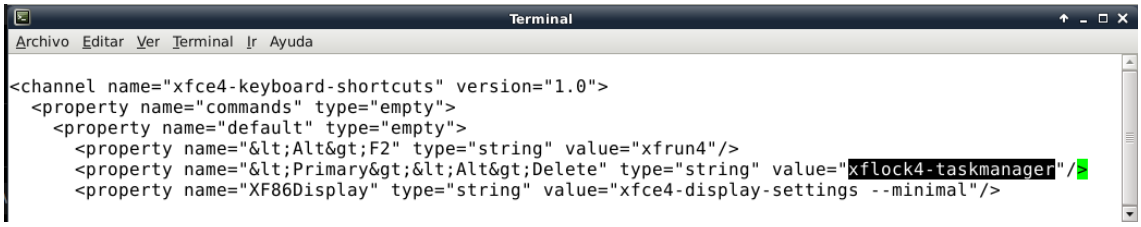


Figura 5.12 – Deshabilitar Ctrl-Alt-Delete.

108 El punto 5.1.4 se alinea con el control A.11.2.8 del ISO/IEC 27002:2013

Adicionalmente se puede comentar la línea “ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now” en el archivo “/etc/inittab”. Por último se deben cambiar los permisos de los ejecutables *halt* y *shutdown* para que sólo el administrador pueda ejecutarlos, esto se realiza con el comando: `chmod 550 /sbin/halt /sbin/shutdown`. Para que los cambios se realicen de forma inmediata se debe indicar al proceso "*init*" que lea el archivo de configuración “/etc/inittab”, esto se logra ejecutando el comando `init q` .

Aunado a lo anterior para que el protector de pantalla se active pasados 3 minutos de inactividad, la forma es dirigirse al “Menú de aplicaciones”->”Configuración” -> “Salvapantallas” y especificar 3 minutos para que sea lanzado el salvapantallas.

5.1.5 Asegurar el dispositivo de almacenamiento

1. Cifrado

Cumple con A.8.3.3 - Soportes físicos en tránsito.
Cumple con A.11.2.7 - Reutilización o retirada segura de dispositivos de almacenamiento.
Cumple con A.11.2.5 - Salida de activos fuera de las dependencias de la empresa.
[109]

En esta sección se cifra el directorio de trabajo de los usuarios “/home/usuario” con la finalidad de garantizar la confidencialidad de los documentos almacenados en él. El primer paso es identificar la partición sobre la cual está montado el “/home”. En la Figura 5.13 se muestra el resultado de la ejecución del comando `df -lh` donde se observa que la partición “/dev/sda9” de 4.4 GB es la partición que aloja el “/home”.

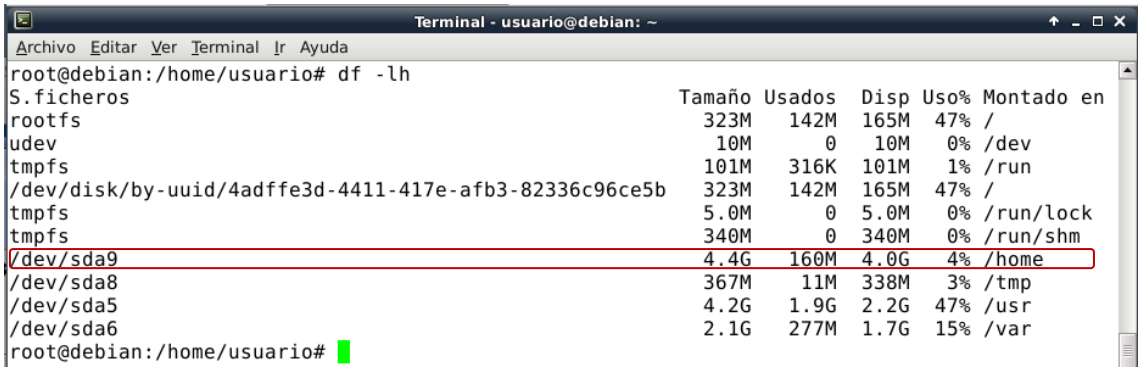


Figura 5.13 – Particiones del sistema Debian.

Para el cifrado de la partición es necesario instalar el paquete *cryptsetup* y sus librerías: *libpam-mount* y *initramfs.tools* (ver Figura 5.14).

109 El punto 5.1.5 – Cifrado, se alinea con los controles A.8.3.3, A.11.2.7 y A.11.2.5 del ISO/IEC 27002:2013

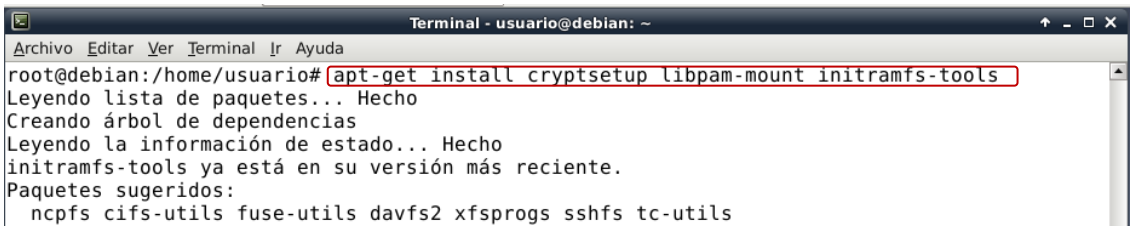


Figura 5.14 – Instalación de cryptsetup.

Después se respalda el directorio de trabajo del usuario junto con la ruta exacta de cada uno de sus archivos: `cp -pfr /home/usuario /tmp`. Posteriormente se cifra el directorio "/home/usuario" el cual será montado con el sistema de archivos *ecryptfs*.

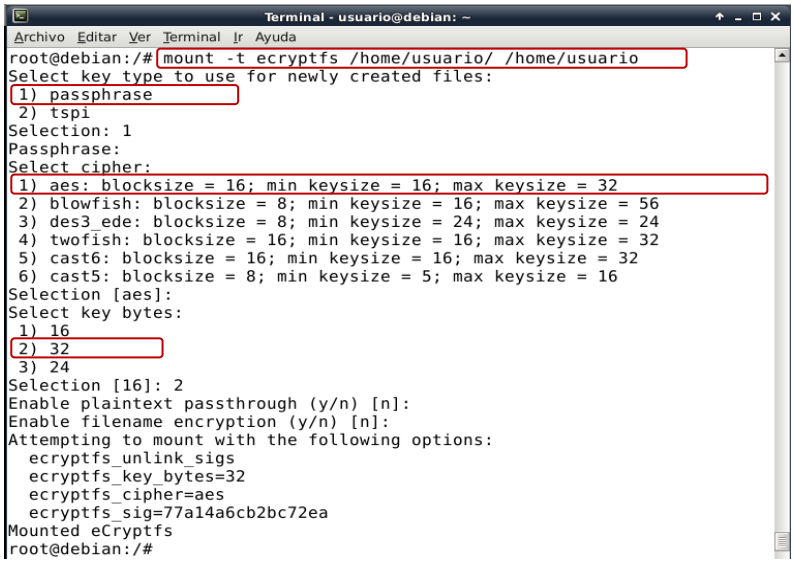


Figura 5.15 – Proceso de montaje del directorio cifrado "/home/usuario".

Lo primero que solicita el sistema es un *passphrase* con el que cifra al directorio, después solicita que se elija el algoritmo simétrico a utilizar, así como la longitud en *bytes* de la *key (clave)*, el resto de las preguntas se sugiere sean contestadas con la respuesta preestablecida como se muestra en la Figura 5.15.

Con el comando *mount* se comprueba que el directorio "/home/usuario" fue montado correctamente con una partición de tipo *ecryptfs* (ver Figura 5.16).

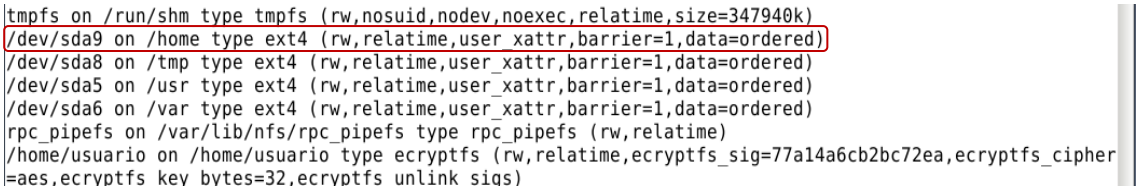


Figura 5.16 – Montaje de "/home/usuario" con ecryptfs.

Por último se restaurará el directorio del usuario a su origen: `cp -pfr /tmp/usuario /home; rm -rf /tmp/usuario`. Para verificar que el cifrado funciona adecuadamente, por ejemplo, se puede realizar una copia del archivo "/etc/passwd" en el *home* del usuario, después se valida que su contenido sea legible, se desmonta el directorio y

se repite el procedimiento para comprobar que el archivo sigue estando en el directorio, pero de forma cifrada, como se muestra en la Figura 5.17.

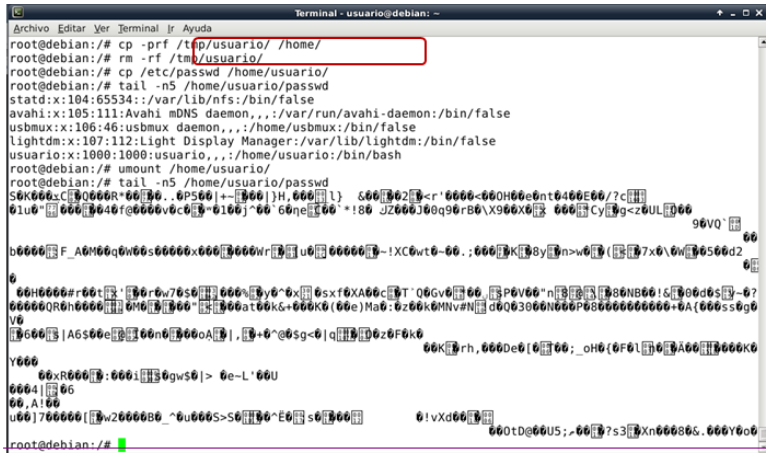


Figura 5.17 – Prueba de concepto del cifrado del directorio "/home/usuario".

En el **Anexo A.II** se automatizará el procedimiento anterior de cifrado y descifrado del directorio "/home/usuario" mediante un dispositivo USB con una *passphrase*.

2. Particiones

Cumple con A.12.1.3 - Gestión de capacidades. [110]

Particionar es la acción de dividir un disco duro en muchas unidades lógicas, por lo tanto una partición es un conjunto contiguo de bloques (en una unidad de almacenamiento) que se tratan como un disco independiente. Las principales particiones en Linux se describen brevemente en la Tabla 5.2.

Tabla 5.2 – Particiones comunes de Linux

Partición	Descripción
/bin	Binarios de los comandos esenciales.
/boot	Archivos estáticos del sector de arranque.
/dev	Archivos de dispositivos.
/etc	Configuración específica del sistema.
/lib	Librerías esenciales, compartidas y módulos de *kernel.*
/media	Punto de montaje para medios extraíbles.
/mnt	Punto de montaje para un sistema de archivos temporal.
/opt	Complemento en los paquetes de *software* de aplicación.
/sbin	Binarios esenciales del sistema.
/tmp	Archivos temporales.
/usr	Contiene una réplica de directorios orientados al usuario.
/var	Datos de variables por lo general para bitácoras.
/home	Directorio *home* de los usuarios.
/root	Directorio *home* de *root.*

110 El punto 5.1.5 – Particiones, se alinea con el control A.12.1.3 del ISO/IEC 27002:2013

El sistema operativo *Linux* (como muchos otros) se puede instalar en una sola partición primaria, sin embargo, si se opta por dividir el disco duro se adquieren ventajas importantes como las siguientes:

- Facilidad de uso: permite la recuperación rápida de sistemas de archivos dañados.
- Rendimiento: los sistemas de archivos más pequeños son más eficientes.
- Seguridad: la separación de los archivos del sistema operativo de los usuarios ofrece mayor seguridad y facilidad al momento de reinstalación de sistema operativo.
- Copia de seguridad y recuperación: fácil de respaldar y recuperar.
- Estabilidad y eficiencia: puede aumentar el espacio disponible en disco y definir tamaños de bloque diferentes, lo cual mejora la eficiencia.

Con el fin de complementar el punto 5.1.2 sobre la instalación del sistema con particiones separadas (durante la instalación), se describe por qué es importante dividir en particiones el disco duro y no dejar todas las particiones al interior de la raíz "/".

- Debido a que un mismo usuario tiene permisos de escritura en "/home" y "/tmp", se recomienda tener ambos directorios en particiones separadas, usando cuotas para reducir el riesgo de un ataque de DoS (Denegación de Servicio) intencional o no intencional, la elevación de privilegios, entre otros.

- Se recomienda que la partición del directorio "/var" esté en una partición separada y con un espacio más grande de lo normal, ya que los paquetes descargados por la herramienta APT en los sistemas *Debian*, son guardados en "/var/cache/apt/archives" y puede causar una Denegación de Servicio al sistema si se agota el espacio de almacenamiento destinado para tal propósito.

- Se recomienda separar la partición de "/usr" debido a que en está partición se instala todo aquel software no propio de la distribución *Debian*.

Aunado a lo anterior si no se cuenta con un esquema de particiones, los siguientes ataques pueden tener lugar en el sistema:

1. Ataques de Denegación de Servicio (DoS) en contra del espacio en disco duro.
2. Los usuarios pueden descargar o compilar programas con permisos SUID en el directorio "/tmp" o "/home".
3. El ajuste del rendimiento del sistema no es viable.

Aunque se haya particionado el sistema, aún queda un riesgo residual de que estos ataques puedan tener éxito, para mitigar aún más el riesgo se configura el archivo "/etc/fstab" con las siguientes características:

- **nosuid**: no coloca SUID/SGID en la partición especificada.
- **nodev**: no permite montar dispositivos especiales en esa partición.
- **noexec**: no permite la ejecución de binarios en esa partición.
- **ro**: permite montar sistema de archivos con permisos de sólo lectura.
- **quota**: habilitar cuotas en el disco.

Las opciones anteriores sólo se pueden implantar si se tienen particiones separadas. El archivo "/etc/fstab" contiene la lista de discos, particiones disponibles, puntos de montaje, tipos de partición, opciones, entre otros. Para asegurar las particiones del sistema se debe editar el archivo "/etc/fstab", agregando a la partición "/tmp" restricciones de *nosuid*, *noexec* y *nodev*; a la partición "/home" y "/usr" se deben agregar restricciones de *nodev;* y por último en la partición "/var" agregar restricciones *nosuid* como se muestra en la Figura 5.18.

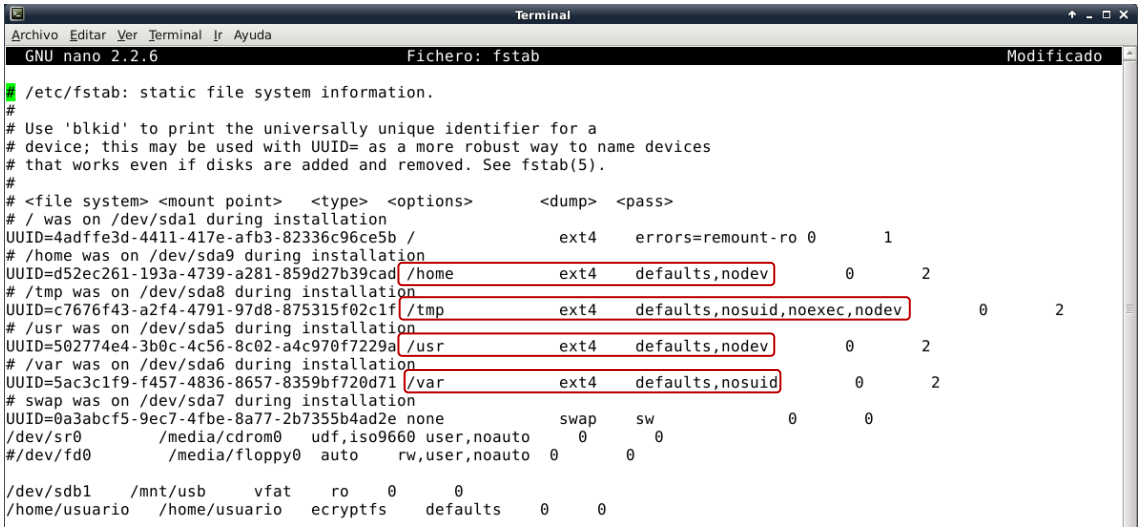


Figura 5.18 – Restricciones a las particiones mediante el archivo "/etc/fstab".

3. Uso de cuotas

Cumple con A.12.1.3 - Gestión de capacidades.
Cumple con A.14.2.6 - Seguridad en entornos de desarrollo.
Cumple con A.9.4.4 Uso de herramientas de administración de sistemas. [111]

El objetivo del uso de cuotas, es que los usuarios se vean obligados a mantenerse por debajo de su límite de consumo de disco, quitándoles la capacidad de consumir espacio ilimitado en el sistema. Para lo cual existen dos tipos de cuotas: una por ínodo y otra por bloque.

a) Ínodo: se indica el número de ínodos de un usuario o de un grupo de usuarios que puede poseer. Un ínodo es un número que actúa como apuntador para el sistema de archivos de Linux y le indica en qué bloques específicos del disco duro se encuentran los datos de un archivo (*Linux* trata todo como archivos).

b) Bloques: se indica el número de bloques de disco que pueden ser asignados a un usuario o a un grupo de usuarios. Un bloque corresponde por lo regular a 512 sectores y una cuota por bloques correspondería al total de bloques que un usuario puede utilizar en el sistema.

111 El punto 5.1.5 – Uso de cuotas, se alinea con los controles A.12.1.3, A.14.2.6 y A.9.4.4 del ISO/IEC 27002:2013

En el caso de cuotas por ínodos, el sistema indicaría el total de ínodos a los que el usuario tiene derecho, por simplicidad se puede hacer la analogía de que un ínodo corresponde a 1 archivo. Las cuotas por bloques o por ínodos, tienen límites de uso y son de dos tipos:

a) ***Hard:*** (duro) se establece (para bloques o ínodos), el límite máximo (absoluto) que no puede ser rebasado por el usuario.

b) ***Soft:*** (suave) este límite (para bloques o ínodos) es siempre menor al *Hard*, puede ser excedido por el usuario, pero será constantemente advertido de que el límite de uso para bloques o ínodos ha sido excedido y debe tomar medidas.

Cuando se usa el límite *Soft* dos situaciones se pueden presentar:

- Si NO se tiene establecido un tiempo de validez, el usuario podrá seguir usando bloques o ínodos hasta llegar al límite de *Hard* que será su límite absoluto.
- SI se tiene establecido el tiempo de validez, definido en términos de días, horas, minutos o segundos, en este caso, el usuario podrá seguir usando bloques o ínodos hasta que termine el tiempo de validez o llegue al límite *Hard* o cualquiera que ocurra primero.

Para implementar las cuotas en el sistema es necesario instalar los paquetes *quota* y *quotatool*. Después editar el archivo "/etc/fstab" agregando los campos de *usrquota* y *grpquota* en las opciones de la partición de "/home", tal como se muestra en la Figura 5.19.

```
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=4adffe3d-4411-417e-afb3-82336c96ce5b /               ext4    errors=remount-ro 0       1
# /home was on /dev/sda9 during installation
UUID=d52ec261-193a-4739-a281-859d27b39cad /home           ext4    defaults,nodev,usrquota,grpquota        0       2
```

Figura 5.19 – Asignación de parámetros de cuotas en el archivo "/etc/fstab".

El resto de procedimiento se detalla en el **Anexo A.II;** sin embargo, una vez que se hayan definido las cuotas los usuarios únicamente podrán ocupar 512 MB de espacio en un límite de tipo *Hard*. De esta forma se mitiga el riesgo de que a causa de agotamiento de espacio en el disco duro, el sistema colapse provocando una auto Denegación de Servicio (DoS).

4. Protección del Kernel

Cumple con A.13.1.2 - Mecanismos de seguridad asociados a servicios en red.
Cumple con A.9.4.5 - Control de acceso al código fuente de los programas. [112]

El núcleo de *Linux* que viene por defecto (*kernel*) no contiene funciones avanzadas para prevenir o contener ciertos tipos de ataques maliciosos. Para fortalecer la

112 El punto 5.1.5 – Protección del kernel, se alinea con los controles A.13.1.2 y A.9.4.5 del ISO/IEC 27002:2013

seguridad del kernel es necesario modificar el archivo "/etc/sysctl.conf" y habilitar las siguientes características:

- Protección contra ataques de *spoof*, sólo se acepta un paquete si la dirección IP origen es alcanzable desde la interfaz desde la cual se ingresó.
- Ignorar paquetes ICMP *broadcasts*.
- Ignorar paquetes ICMP que informan errores de paquetes no trasmitidos por el sistema.
- No aceptar redirecciones de paquetes ICMP.
- No enviar paquetes direccionados nuevamente de tipo ICMP.
- No aceptar información proveniente de *routes*, puesto que este equipo no es un router.

Las características previamente descritas, se deben aplicar en el archivo "/etc/sysctl.conf" como se muestra en la Figura 5.20.

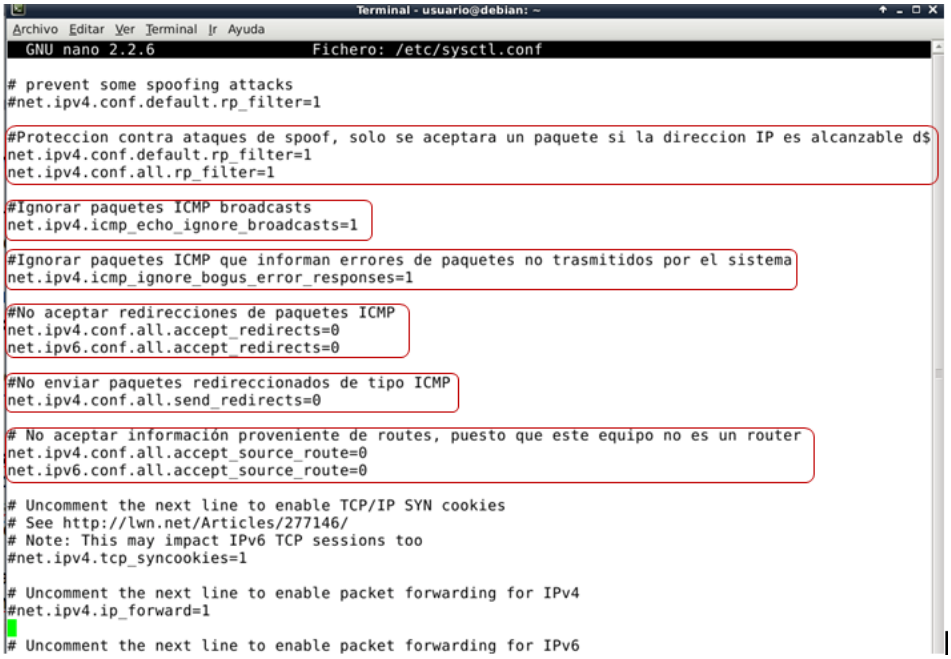


Figura 5.20 – Aseguramiento del kernel mediante "/etc/sysctl.conf".

Para proteger el sistema de ataques de *SYN Floods* que atentan contra la disponibilidad del sistema, se agregaran los parámetros mostrados en la Figura 5.21 en el recuadro rojo. Estos parámetros establecen que el sistema únicamente puede recibir 2,048 conexiones y el tiempo que mantendrá este tipo de conexiones es de 30 segundos. El resto de las configuraciones para asegurar el kernel se muestran en el **Anexo A.II**.

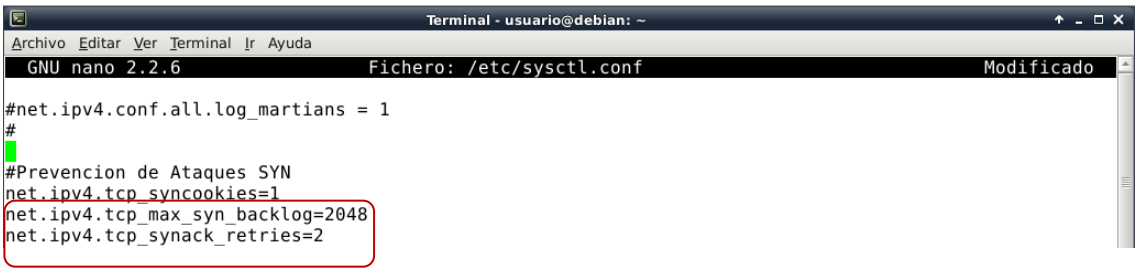


Figura 5.21 – Protección contra ataques DoS hacia el kernel.

5. Parche de seguridad para el Kernel

Cumple con A.12.2.1 - Controles contra el código malicioso.
Cumple con A.9.2.2 - Gestión de los derechos de acceso asignados a usuarios.
Cumple con A.9.2.3 - Gestión de los derechos de acceso con privilegios especiales.[113]

Se implementan dos enfoques para fortalecer la seguridad o hardenizar el *kernel* estándar de *Linux*: El primero consiste en la protección en las direcciones de espacio (memoria) y el segundo se refiere al control de acceso avanzado. Existen diversos parches disponibles para robustecer la seguridad de los sistemas *Debian*, las características más importantes que agregan al *kernel* son:

- Detección de Intrusos en *Linux* (incorporado en el paquete *lids-2.2.19*): protege procesos y archivos críticos incluso del usuario *root*.
- Confianza en *Linux* (dentro del paquete *trustees*): agrega permisos avanzados para la administración del *kernel* e implementa restricciones al *stack* (pila).
- *kernel-patch-2.2.18-openwall*: implementa restricciones en FIFOs y directorio "/proc".
- *kernel-patch-2.4-grsecurity*: implementa Listas de Control de Acceso (ACLs por sus siglas en inglés), protección contra ataques de *buffer overflow* y dificulta ataques de *fingerprinting*.

Para este *hardening* el parche más conveniente es el de *Grsecurity* debido a que proporciona protección al espacio de direcciones y un sistema de control de acceso avanzado. Este parche puede ayudar a prevenir el impacto o anular el éxito de ataques de desbordamientos de búfer (*Buffer Overflow*), las condiciones de carrera (*race conditions*) y el procesamiento de caracteres especiales que permiten a un atacante obtener privilegios de súper usuario en el sistema, este tipo de ataques y vulnerabilidades se explican a continuación.

- Los ataques de *Buffer Overflows* son el resultado fallido de un programa al no comprobar correctamente los límites de entrada proporcionados por el usuario. Si estos datos rebasan la cantidad de memoria destinada para la ejecución de un programa, los datos se sobreponen a los datos de *buffers* adyacentes lo cual le puede dar a un atacante control sobre la aplicación vulnerable y en algunos casos escalar privilegios en el sistema.

- *Race Conditions* es una vulnerabilidad en los programas, se presenta cuando dos o más procesos o aplicaciones pueden acceder a un mismo segmento de datos e intentan modificarlo al mismo tiempo, lo que podría generar un error de corrupción de datos, permitiendo que por medio de *exploits* locales consiguieran vulnerar los sistemas.

[113] El punto 5.1.5 – Parche de seguridad para kernel, se alinea con los controles A.12.2.1, A.9.2.2 y A.9.2.3 del ISO/IEC 27002:2013

- La vulnerabilidad relacionada con el procesamiento de caracteres especiales permite a un atacante insertar caracteres especiales en las URLs (víctimas) relacionadas con *scripts* que los formularios Web procesan. Estas secuencias de caracteres podrían permitir a los atacantes ejecutar código arbitrario en el servidor Web. Aunque la mayoría de servidores Web están configurados para ejecutarse como usuarios sin privilegios, este tipo de ataques pueden permitir la entrada no autorizada al sistema en caso de que no se tomen las medidas de seguridad pertinentes.

El parche de seguridad *Grsecurity* es un enfoque innovador para la seguridad en la que se implementa la detección de varias capas, la prevención, y el modelo de contención. Este parche es licenciado bajo la licencia GPL y ofrece las siguientes características:

- Funciones de Control de Acceso permitiendo al sistema generar políticas de menor privilegio.
- Prevención de *Race Conditions* en el directorio "/tmp".
- Auditoría amplia de los sucesos ocurridos en el sistema.
- Prevención de la ejecución de código arbitrario en el *kernel*.
- Asignación aleatoria del *stack* para mitigar los ataques de *buffer overflow*.
- Reducción de pérdida de información confidencial que podría ser filtrada por la lectura de los errores realizada por el *kernel*.
- Restricciones para que el usuario vea sólo sus procesos.

Para la instalación del *kernel* de seguridad *Grsecurity* es necesario descargar el paquete "linux-image-3.2.15-grsec.deb" y posteriormente instalarlo con el comando `dpkg` como se observa en la Figura 5.22.

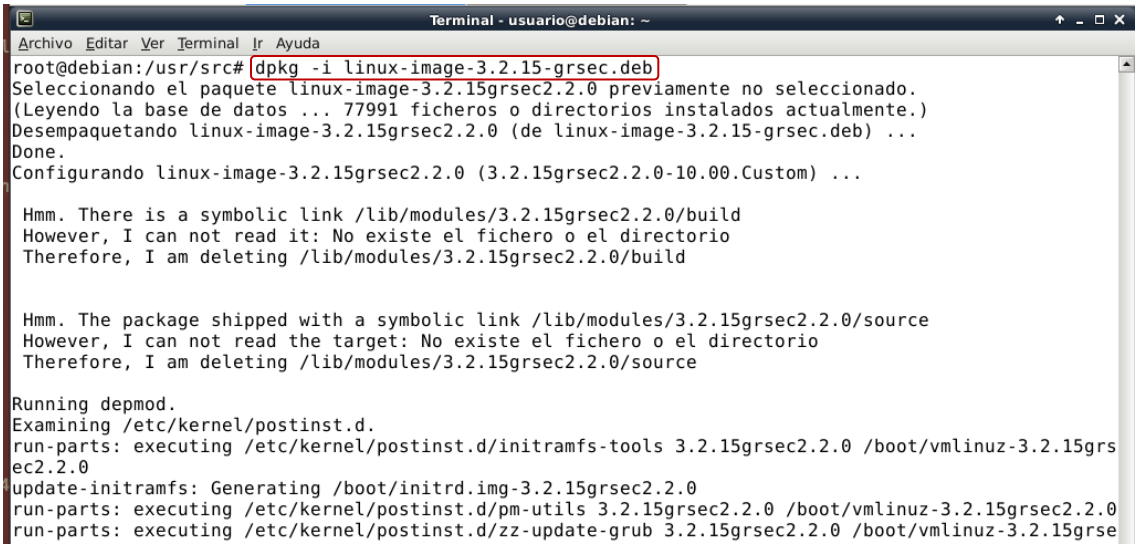


Figura 5.22 – Instalación del parche de seguridad Grsecurity.

En el **Anexo A.II,** se detalla la configuración del *kernel* para aplicar las características de seguridad mencionadas anteriormente.

5.2 Eliminación de aplicaciones innecesarias

5.2.1 Servicios y conexiones de red innecesarios

Cumple con A.13.1.2 - Mecanismos de seguridad asociados a servicios en red.
Cumple con A.8.3.2 - Eliminación de soportes.
Cumple con A.8.1.4 - Devolución de activos.
Cumple con A.11.2.9 - Política de puesto de trabajo despejado y bloqueo de pantalla. [114]

Los servicios son programas o aplicaciones que se ejecutan en el equipo, tales como los servicios de FTP, servicios de SSH, servicios de impresión, etc. Éstos por lo regular se encuentran a la espera de conexiones entrantes que requieren el servicio para conectarse o interactuar con el sistema. Algunos servicios pueden ser vulnerables y por lo tanto representan un riesgo de seguridad para el sistema, es por ello que una buena práctica de seguridad es no instalar servicios innecesarios o deshabilitar los que se encuentren sin uso.

Cuando es necesario instalar un nuevo servicio relacionado con la red (por ejemplo un demonio), en el sistema *Linux-Debian* se puede habilitar de dos formas: a través del súper-demonio de *inetd* (una línea será agregada al archivo /etc/inetd.conf) o a través de un programa automático (*script*) que se habilita a sí mismo al iniciar el sistema. Estos *scripts* son controlados a través de archivos ubicados en el directorio "/etc/init.d", los cuales son llamados, cuando entran en operación los niveles de ejecución del sistema, los llamados *SysV*. Este último proceso es llevado a cabo por un programa llamado *init*, cuyo objetivo es dar inicio a un determinado nivel de ejecución (*runlevel*) sobre el cual trabaja el sistema.

En este trabajo se analizaron los servicios que inician en el sistema de cómputo, se identificaron cuáles no son necesarios, para posteriormente desactivar su inicio automático en el sistema. Para lo cual, fue necesario comprender el proceso de arranque del sistema operativo *GNU/Linux*, el cual se desarrolla con base en la consecución de diversas fases o etapas, tales como búsqueda del *MBR* (*Master Boot Record* – Registro de Arranque Maestro), la carga del *kernel* y por último la iniciación de servicios.

En el **Anexo A.III** se describen los pasos necesarios para la desactivación de servicios y conexiones de red innecesarios en el sistema, así como la implementación de escritorio limpio y la instalación de herramientas para la eliminación de medios a bajo nivel (borrado seguro).

[114] El punto 5.2.1 se alinea con los controles A.13.1.2, A.8.3.2, A.8.1.4 y A.11.2.9 del ISO/IEC 27002:2013

5.2.2 Arranque del sistema y paquetería instalada

Cumple con A.11.2.4 - Mantenimiento de los equipos. [115]

Los scripts del directorio "/etc/init.d" se dividen en dos categorías:

- *Scripts* llamados directamente por *init*: Éstos sólo entran en operación en el arranque y apagado instantáneo del sistema (en caso de un corte del suministro eléctrico o por pulsar el usuario la combinación de teclas *Ctrl* + *Alt* + *Supr*).
- *Scripts* llamados indirectamente por *init*: Esto ocurre en el caso de un cambio del nivel de ejecución; aquí generalmente se ejecuta el *script* superior "/etc/init.d/rc", el cual se encarga de que los *scripts* correspondientes sean ejecutados en orden correcto.

Los *runlevels* son distintos estados en los cuales puede iniciar un sistema operativo GNU/Linux. En la actualidad existen siete niveles de ejecución sobre los cuales puede operar o trabajar un sistema como se puede apreciar en la Tabla 5.3.

Tabla 5.3 – Niveles de ejecución del sistema.

Nivel de ejecución	Descripción	Directorio
0	Nivel de ejecución conocido como *halt* se encarga de detener todos los procesos activos en el sistema, con el objetivo de llevar a cabo un correcto apagado del sistema.	/etc/rc0.d
1	Nivel de ejecución conocido como mono usuario o *single user*, permite la sesión de un único usuario por defecto inicia con usuarios *root*. Este nivel de ejecución es empleado para tareas de mantenimientos del sistema.	/etc/rc1.d
2	Nivel de ejecución multiusuario, sin soporte para red.	/etc/rc2.d
3	Nivel de ejecución multiusuario, con soporte para red.	/etc/rc3.d
4	Indefinido o sin uso.	/etc/rc4.d
5	Nivel de ejecución multiusuario, con capacidad gráfica (X *Windows*).	/etc/rc5.d
6	Nivel de ejecución de reinicio del sistema.	/etc/rc6.d

Cada directorio perteneciente a los distintos niveles de ejecución posee distintos enlaces simbólicos a archivos, los cuales son utilizados para el inicio y paro de cada uno de los procesos definidos en cada nivel de ejecución del sistema. La creación de este tipo de enlaces se realiza a partir de archivos creados dentro del directorio "/etc/init.d", es decir cada proceso o servicio que pretenda iniciar en un nivel de ejecución debe poseer su archivo (*script*) correspondiente dentro de este directorio.

Para forzar que el sistema interprete que un archivo debe iniciar en algún nivel de ejecución se debe agregar al inicio del nombre del enlace una letra mayúscula "S", seguida de un valor numérico entero que indica el nivel de prioridad para ejecutar el servicio cuyo valor oscila de -20 a 20 y por último se agrega el nombre relativo al

115 El punto 5.2.2 se alinea con el control A.11.2.4 del ISO/IEC 27002:2013

servicio. Cabe aclarar que los valores de -20 a 0 para el nivel de prioridad son exclusivos para los procesos pertenecientes al súper usuario (*root*). Por otro lado si el nombre del enlace inicia con la letra K significa que el servicio será detenido, además cada uno de los *scripts* pueden ser ejecutados como un *script* de arranque o de paro, para lo cual pueden admitir parámetros como: *start, stop, restart, reload, force-reload* y *status*, cuyo significado se explica en la Tabla 5.4.

Tabla 5.4 – Descripción de las opciones de los scripts de inicio

Opción	Significado del parámetro
start	Iniciar el servicio.
stop	Parar el servicio.
restart	Con el servicio en ejecución, pararlo y reiniciarlo; en caso contrario, iniciarlo.
reload	Leer la configuración del servicio nuevamente sin parada y reinicio del servicio.
force-reload	Leer nuevamente la configuración del servicio si éste lo soporta; en caso contrario igual que *restart.*
status	Mostrar estado actual.

En el **Anexo A.III** se describen los pasos necesarios para la desactivación de servicios innecesarios en el sistema desde los *runlevels*. Para saber la paquetería instalada en el sistema se puede emplear el comando: `dpkg -i` el cual listará los paquetes y versiones del software instalados actualmente en el sistema. Esto sirve para la búsqueda de vulnerabilidades relacionadas con las versiones del software en páginas Web dedicadas a la difusión de vulnerabilidades como: http://cve.mitre.org/ y http://nvd.nist.gov/.

5.2.3 Otros paquetes no utilizados

Cumple con A.11.2.4 - Mantenimiento de los equipos. [116]

Un paquete huérfano son aquellas librerías que ya no son necesarias, es decir, que ninguno de los paquetes instalados la indica como dependencia. Para realizar el rastreo en el sistema de este tipo paquetes se puede utilizar la herramienta *deborphan* la cual se puede instalar de la siguiente forma: *aptitude install deborphan*. Una vez que se ejecutó en el sistema, el resultado fue nulo para este caso (ver Figura 5.23), esto significa que dado que el sistema está prácticamente recién instalado, de momento no cuenta con paquetes huérfanos, sin embargo, para sistemas que no sean instalados desde cero y que se desee localizar los paquetes huérfanos, la herramienta *deborphan* puede resultar muy eficiente.

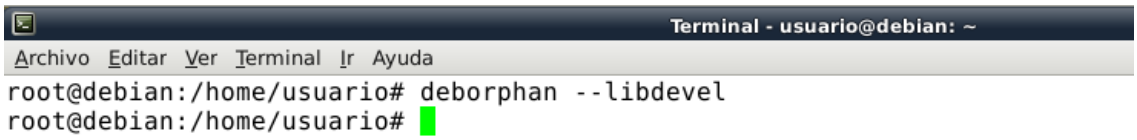


Figura 5.23 – Ejecución de herramienta deborphan.

116 El punto 5.2.3 se alinea con el control A.11.2.4 del ISO/IEC 27002:2013

Para más detalle de la desactivación de paquetes no utilizados en el sistema, consultar el **Anexo A.IV**.

5.3 Usuarios y permisos

5.3.1 El usuario normal y el súper-usuario

Cumple con A.9.2.2 - Gestión de los derechos de acceso asignados a usuarios.

Cumple con A.9.2.3 - Gestión de los derechos de acceso con privilegios especiales. [117]

Linux es un sistema operativo multiusuario y multitarea, es decir que más de un usuario puede trabajar en el sistema de forma simultánea con otros usuarios pudiendo ejecutar una o más tareas a la vez. Para ingresar al sistema cada usuario debe ingresar su *login* (nombre que lo identifica) y una contraseña de tipo unidireccional la cual no es almacenada como texto, es cifrada y almacenada en el archivo "/etc/shadow".

Existen tres tipos de usuarios:

a) **Usuario normal:** se refiere a cualquier usuario con posibilidad de autenticarse en el sistema, con los privilegios asignados para hacer uso de los recursos del sistema. Como indicador en el *prompt* utiliza el símbolo "$".
b) **Usuarios de sistema:** son usuarios propios del sistema y aplicativos vinculados a las tareas que debe realizar el sistema operativo, este tipo de usuarios no pueden ingresar al sistema con un *login* y contraseña, como ejemplo de este tipo de usuarios están; *mail, game, ftp, bin, sys, proxy*, etcétera. A estos usuarios también se les conoce como usuarios sin *login*.
c) **Usuario root (súper-usuario):** todo sistema operativo *GNU/Linux* cuenta con un usuario de tipo "súper-usuario" con los máximos privilegios (todos los privilegios), que le permitirán efectuar cualquier operación sobre el sistema, su existencia es imprescindible, ya que se utiliza para la administración del sistema, puesto que puede utilizar todos los programas, manipular cualquier archivo e interactuar sin "ninguna restricción" con el sistema de archivos, en resumen puede hacer cualquier "cosa" dentro del sistema. Como indicador en el *prompt* utiliza el símbolo "#".

La cuenta súper-usuario normalmente llamada *root*, viene pre-configurada para facilitar la administración del sistema, y no debería ser utilizada para tareas cotidianas como enviar o recibir correo, exploración general del sistema o programación de *scripts* o aplicaciones. El súper-usuario, a diferencia de las cuentas de usuario de sistema o normal, puede operar sin límites y un mal uso de la misma puede causar desastres irreparables en el sistema. Es más complicado corromper el sistema con cuentas de usuario normal a causa de algún error involuntario, por ello se recomienda utilizar cuentas de usuarios normales en la medida de lo posible y utilizar el comando *sudo* o *su* únicamente cuando las necesidades de interacción así lo demanden.

[117] El punto 5.3.1 se alinea con los controles A.9.2.3 y A.9.2.3 del ISO/IEC 27002:2013

El administrador del sistema (comúnmente conocido como súper-usuario) tiene la responsabilidad total de asegurarse que los usuarios tengan un UID (*User* ID) y GID (*Group* ID) adecuado para garantizar que los archivos sensibles (que puede incluir los archivos críticos del sistema), permanezcan bloqueados para usuarios normales. Por tal razón, una buena práctica de seguridad es que el usuario *root* sea el único que tenga asignado el ID de usuario igual a 0 y un ID de grupo también igual 0, de esta forma se minimiza el riesgo de que otros usuarios con permisos de *root* puedan hacer uso indebido de sus privilegios dentro del sistema.

5.3.2 Archivo passwd, group y shadow

Cumple con A.9.2.4 - Gestión de información confidencial de autenticación de usuarios.
Cumple con A.9.4.2 - Procedimientos seguros de inicio de sesión. [118]

La información de los usuarios, grupos y contraseñas se guardan en los siguientes archivos respectivamente.

a) **/etc/passwd:** tiene información de todos los usuarios registrados en el sistema como: nombre, UID, GID, información, directorio *home* y el *Shell* asignado.
b) **/etc/group:** almacena la información sobre los grupos existentes en el sistema como: el nombre del grupo, GID y la lista de usuarios que pertenecen a ese grupo.
c) **/etc/shadow:** contiene las contraseñas cifradas (*hash* MD5) de los usuarios además de otros datos para su validación.

En los sistemas Unix y Linux se les asigna un único UID (*User ID*) a las cuentas de usuario, como se vio anteriormente el UID “0” está reservado para el súper-usuario (*root*); así como se crea un UID para los usuarios también se tiene un GID (*Group ID*) utilizado para englobar a los usuarios en roles específicos.

En cada distribución de *Linux* y *Unix* se tienen grupos de números reservados, en sistemas operativos *RedHat* se reserva del número 1 al 499, en *Debian* los reservados van del número 1 a 999, por tal razón en los sistemas *Debian* la asignación de los usuarios creados inician a partir del UID 1000 y “nunca” en un número inferior.

Con el comando *id* se puede visualizar el UID y los GID a los que pertenece el usuario que ejecuta dicho comando en el sistema, cabe precisar que con la opción “-g” se puede visualizar el GID primario, donde el grupo primario es el grupo que se crea al mismo tiempo que se crea un usuario en el sistema. En la Figura 5.24 se puede ver la salida de la ejecución del comando “id”.

118 El punto 5.3.2 se alinea con los controles A.9.2.4 y A.9.4.2 del ISO/IEC 27002:2013

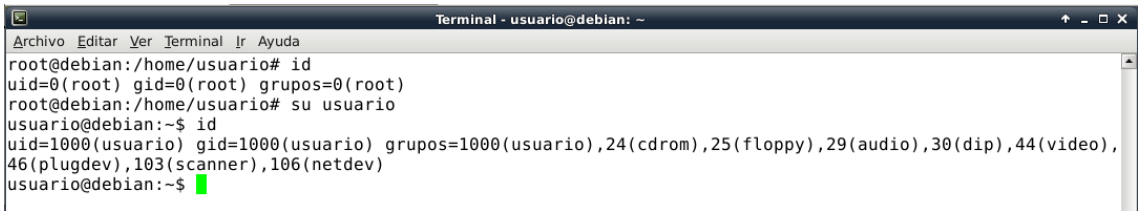


Figura 5.24 – Visualización de UID y los GID de una cuenta de usuario.

En el **Anexo A.V** se continúa con el aseguramiento de aspectos relacionados con los archivos *passwd, group y shadow* del sistema Linux-Debian.

5.3.3 El bit SUID, SGID y STICKY

Cumple con A.12.4.2 - Protección de los registros de información.
Cumple con A.12.4.3 - Registros de actividad del administrador y operador del sistema.
Cumple con A.9.4.2 - Procedimientos seguros de inicio de sesión. [119]

Los permisos de los archivos en Unix corresponden con un número en base octal que varía entre 000 y 777, sin embargo existen algunos permisos especiales que varían entre el número 0000 y 7777, estos *bits* se conocen como *bits* de permanencia (1000), SGID (2000) y SUID (4000).

Por ejemplo, el *bit* de SUID o *setuid* se activa sobre un archivo añadiéndole 4000 a la representación octal de los permisos del archivo y posteriormente añadiendo el permiso de ejecución al dueño del archivo, entonces el sistema automáticamente en lugar de colocar una letra x (ejecución) en la primera terna de los permisos, colocará una letra s o una S dependiendo de si se ha otorgado o no respectivamente, el permiso de ejecución correspondiente (ver Figura 5.25).

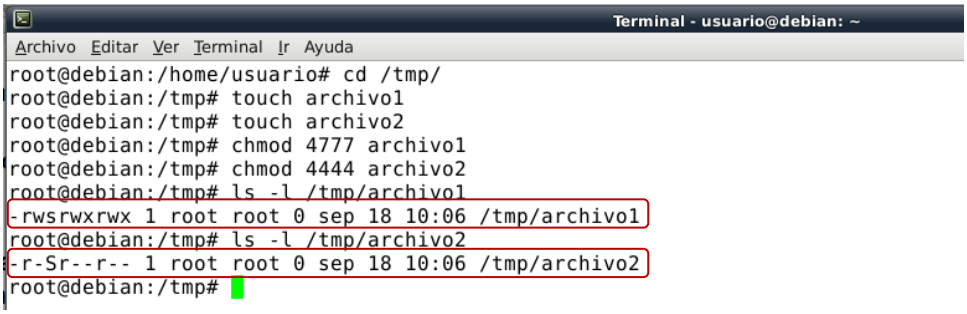


Figura 5.25 – El bit SUID.

Un archivo con el *bit* SUID activado le indica al sistema que todo usuario que ejecute el archivo va a tener durante la ejecución del mismo, los privilegios del usuario que creó dicho archivo, es decir, que si el administrador crea un archivo con el SUID activado, cualquier usuario que lo ejecute tendrá los mismos permisos que el administrador tenía sobre él.

119 El punto 5.3.3 se alinea con los controles A.12.4.2, A.12.4.3 y A.9.4.2 del ISO/IEC 27002:2013

Las características que se aplican al *bit* SUID también son aplicables al *bit* SGID la diferencia radica en que se aplican a un grupo y no a un usuario. Para activar el *bit* de SGID se suma 2000 a la representación octal del permiso del archivo y posteriormente se añade el permiso de ejecución a la terna correspondiente del grupo. Cabe destacar, que si el archivo es un directorio, el *bit* SGID afectará a los archivos y subdirectorios que se encuentren contenidos en él. [120]

Para el caso del *Sticky bit* o *bit* de permanencia se activa sumándole 1000 a la representación octal de los permisos de un archivo y otorgándole el permiso de ejecución; a diferencia de los *bits* de SUID y SGID se agregará una letra t o T (tercer terna de permisos) dependiendo si ha dotado al archivo de permisos de ejecución. En el **Anexo A.VI**, se describe a detalle el proceso de localización de archivos con bits especiales.

5.3.4 Usuarios con Shell válida e inválida

Cumple con A.9.4.4 - Uso de herramientas de administración de sistemas. [121]

En los sistemas Linux-Debian los usuarios que tiene un UID (*User ID*) menor a 1000 son usuarios de sistema, los cuales no tienen por qué utilizar una *Shell* (terminal intérprete de comandos) de manera interactiva, por lo tanto se les debe deshabilitar el uso de una *Shell* debido a que existe la posibilidad de que un intruso malicioso las utilice de trampolín para lanzar comandos perjudiciales al sistema.

Se recomienda asignar una *Shell* no válida a las cuentas de usuarios descritas en la Tabla 5.5 así como eliminar aquellas cuentas de usuario no útiles para las actividades del sistema *Linux-Debian* en cuestión. Para identificar las cuentas de usuario del sistema con Shell o sin Shell válida se debe revisar el archivo “/etc/passwd”.

Tabla 5.5 – Análisis de cuentas de usuarios con Shell válida

Cuenta de usuario	Propósito	¿Shell válido?	¿Remover?
daemon	Es el dueño de los procesos y se encarga de correrlos en el sistema.	No	No
bin	Dueño de los ejecutables.	No	No
sys	Usuario por default para montar los sistemas de archivos.	No	No
sync	Usuario de sincronización.	No	Sí
games	Usuario para juegos.	No	Sí
man	Usuario de la documentación de ayuda *man.*	No	No
lp	Usuario para impresión.	No	Sí
mail	Usuario del agente de transferencia de correo.	No	Sí (en caso de no usar el servicio

[120] Los bits SUID, SGID y sticky (2002). Obtenido el 22 de diciembre de 2013, de http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/SEGUNIX/unixsec-2.1-html/node56.html

[121] El punto 5.3.4 se alinea con el control A.9.4.4 del ISO/IEC 27002:2013

Cuenta de usuario	Propósito	¿Shell válido?	¿Remover?
			de correo)
news	Usuarios nuevos.	No	Sí
uucp	Protocolo para la transferencia de archivos.	No	Sí
proxy	Usuario por default para *proxy.*	No	Sí
www-data	Usuario del servidor Web.	No	No
backup	Usado para empaquetar archivos críticos utilizados como respaldos.	No	Sí (en caso de no usar el servicio de impresión)
list	Para usuarios de manejo de correo.	No	Sí
irc	Usuario de IRC.	No	Sí
gnats	Usuario de reporte de errores.	No	Sí
nobody	Usuario por default para apache.	No	Sí
libuuid	Usuario generador de id para librerías.	No	No
statd	Usuario para obtener información de bloqueos en sistemas NFS.	No	No
avahi	Permite que los programas publiquen y descubran los servicios de un equipo.	No	No
usbmux	Usuario del sistema que permite sincronizar un *Iphone.*	No	Sí
lightdm	Usado para que los usuarios gestionen las sesiones X *server.*	No	No
sshd	Usuario para dividir privilegios.	No	No

Para deshabilitar las *Shells* de cuentas de usuarios que no deban tener una por cuestiones de seguridad, se puede emplea el comando:

```
chsh –s /bin/false nombre_cuenta.
```

5.3.5 Caducidad de las cuentas de usuarios

Cumple con A.9.2.6 - Retirada o adaptación de los derechos de acceso.
Cumple con A.9.4.1 - Restricción del acceso a la información. [122]

Una buena práctica es recomendar a los usuarios el cambio periódico de su contraseña, sin embargo, por lo regular ni los usuarios ni administradores del sistema cambian la contraseña de acceso al sistema a menos que se vean obligados a cambiarla. Para visualizar la configuración de un usuario referente a la gestión de contraseñas se puede utilizar el comando `chage -l nombre_de_usuario`, como se muestra en la Figura 5.26 por defecto el sistema no implementa configuraciones para la gestión de seguridad de las contraseñas en el sistema.

122 El punto 5.3.5 se alinea con los controles A.9.2.6 y A.9.4.1 del ISO/IEC 27002:2013

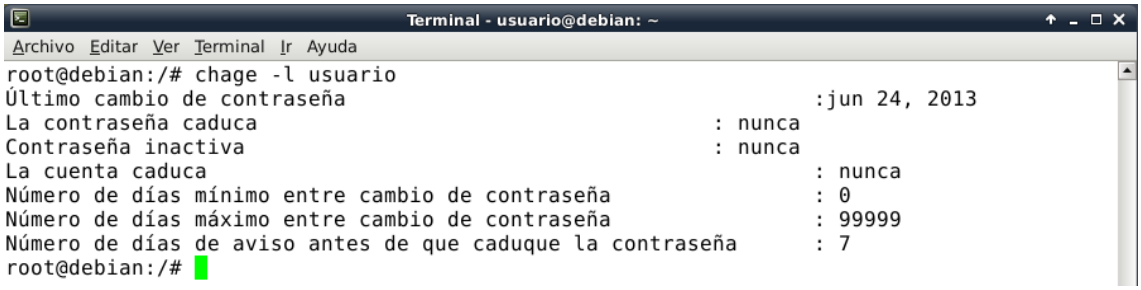


Figura 5.26 – Listado de configuraciones de políticas de contraseñas.

En el **Anexo A.VII**, se explica a detalle el procedimiento para la implementación de caducidad a las cuentas de los usuarios válidos del sistema.

5.3.6 Módulo para la gestión de contraseñas

Cumple con A.9.4.3 - Gestión de contraseñas de usuario. [123]

En el tema anterior, se describió cómo modificar las características de caducidad, validez y avisos de las cuentas, pero no basta con que las contraseñas sean renovadas periódicamente, también se debe asegurar que las contraseñas sean robustas, es decir, que sean de al menos 8 dígitos de longitud, que contengan letras mayúsculas, minúsculas, caracteres especiales (@#$&*%!) y que no estén relacionadas con palabras de diccionario o relativas a información personal del usuario, como números y letras de placas de un auto, fechas de nacimiento, nombres de mascotas, parientes, amigos, siglas del lugar de trabajo, lugares favoritos, etcétera.

Para tener un control sobre las contraseñas que los usuarios emplean al acceder al sistema, se utiliza el módulo PAM, el cual además de validar la complejidad de las contraseñas, verifica que el usuario al cambiar la contraseña no utilice alguna que previamente haya usado antes, es decir, guarda una memoria de las contraseñas utilizadas por cada usuario.

Otra de las opciones que se pueden configurar con el módulo de PAM, es el bloqueo de una cuenta de usuario después de un número determinado de intentos erróneos al introducir la contraseña. Esto permite mitigar el éxito de un posible ataque de diccionario o fuerza bruta.

El módulo *pam_cracklib* (*libpam-cracklib*) está diseñado específicamente para la gestión de contraseñas, se inserta en la fase *passwd* de PAM y antes de aceptar una nueva contraseña valida una serie de pruebas para determinar si es lo suficientemente robusta y no se parece a alguna previamente introducida:

- ¿Puede derivarse de una palabra del diccionario?
- ¿Es un palíndromo de la vieja contraseña?
- ¿Es sólo un intercambio de mayúsculas y minúsculas de la contraseña anterior?

123 El punto 5.3.6 se alinea con los controles A.9.4.3 del ISO/IEC 27002:2013

- ¿Es muy similar (caracteres repetidos) a la anterior?
- ¿Es muy simple (muy corta o no tiene suficientes caracteres distintos)?
- ¿Es una rotación de la vieja contraseña?
- ¿La contraseña ya fue usada en el pasado?

En el **Anexo A.VII** se describe a detalle la instalación y uso del módulo libpam-cracklib para la gestión de contraseñas en el sistema.

5.3.7 Sudo

Cumple con A.9.4.2 - Procedimientos seguros de inicio de sesión. [124]

Con la herramienta *sudo* (SUperuser DO) se puede tener un control estricto sobre las tareas que realizan los usuarios que requieren privilegios de *root*, las características que se pueden configurar son las siguientes:

- Definir los usuarios que pueden elevar privilegios.
- Definir las tareas que pueden realizar.
- Si los usuarios requieren ingresar un *password.*
- Restricciones horarias.
- Lugar de donde se está conectando el usuario.
- Registro de las acciones realizadas.

Cuando un usuario normal desea ejecutar un comando que le pertenece a *root*, entonces *sudo* verifica en su lista de control (/etc/sudoers), si el usuario que invoca a *sudo* está en dicha lista y a su vez se autentica correctamente, si las dos se cumplen, el sistema ejecutará el comando con los permisos de *root* definidos en dicha lista de control. Por tanto, la ventaja de esta herramienta radica en que no es necesario estar siempre como *root* en el sistema para realizar ciertas actividades que requieran la elevación de privilegios.

Por defecto, después de haberse autenticado el usuario tendrás cinco minutos para volver a usar el mismo comando u otros a los que tenga permisos, sin necesidad de ingresar nuevamente la contraseña. Para identificar las rutas absolutas de los comandos que se pueden utilizar con *sudo*, se usa la opción "-l" como se observa en la Figura 5.27.

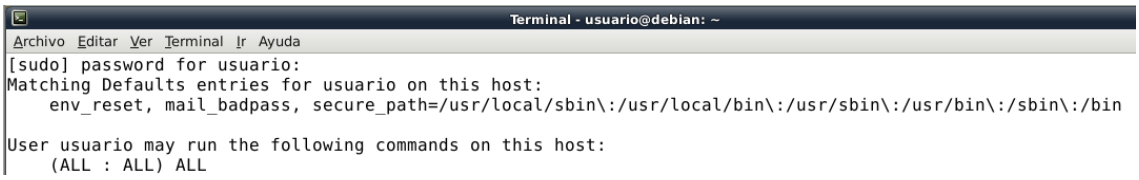


Figura 5.27 – Ejecución del comando "sudo –l".

También es posible ejecutar comandos de otros usuarios del sistema indicando la opción "-u", como se indica a continuación `sudo -u test /comando/de/test`.

124 El punto 5.3.7 se alinea con el control A.9.4.2 del ISO/IEC 27002:2013

5.4 Limitar el acceso directo a root

5.4.1 El grupo Wheel

Cumple con A.9.2.2 - Gestión de los derechos de acceso asignados a usuarios.
Cumple con A.9.2.3 - Gestión de los derechos de acceso con privilegios especiales.
[125]

El grupo *wheel* permite que los usuarios que pertenecen a él, sean los únicos que pueden escalar privilegios de *root* a partir del comando "su". En la gran mayoría de las distribuciones de Linux-Debian el grupo *wheel* no existe, por lo que en caso de no existir, es necesario crearlo y agregar al usuario *root* a este grupo, para ello se requiere editar el archivo "/etc/group" y añadir la línea "wheel:*:0:root,usuario" o ejecutar el siguiente comando "addgroup --system Wheel" y posteriormente agregar al usuario root al grupo *wheel*.

También se pueden limitar los usuarios que podrán ejecutar el comando "su*"* desde el módulo de PAM, para ello se requiere quitar el comentario de la línea que inicia con "*auth required pam_wheel.so deny…"* en el archivo "/etc/pam.d/su*"*. Su funcionamiento se basa en permitir ejecutar "su*"* sólo a los usuarios que pertenezcan al grupo *wheel*, pero si se desea especificar un grupo en particular se debe agregar a la variable *group* el nombre del grupo ver Figura 5.28.

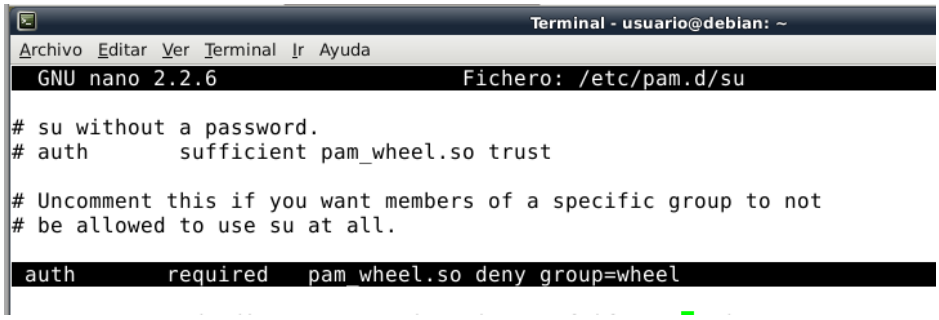


Figura 5.28 – Edición del módulo su de PAM.

Para agregar un usuario al grupo de wheel se emplea el siguiente comando:

```
usermod –G wheel usuario1
```

125 El punto 5.4.1 se alinea con los controles A.9.2.2 y A.9.2.3 del ISO/IEC 27002:2013

5.4.2 Tipos de consolas

Cumple con A.11.2.6 - Seguridad de los equipos y activos fuera de las instalaciones. [126]

Una de las características que diferencia al sistema operativo *UNIX* de otros sistemas, es su capacidad de proporcionar diferentes intérpretes de comandos o consola, es decir, las principales diferencias entre los diferentes tipos de *Shell* son básicamente el conjunto de instrucciones disponibles para el usuario y su estructura de programación.

La consola (*Shell*) también actúa como dispositivo de salida de texto para los mensajes de administración del sistema los cuales son emitidos por el núcleo, por el sistema de inicio y por el registro del sistema. Los sistemas GNU/Linux son sistemas operativos multiusuario, los cuales puede tener más de una terminal conectada a puertos serie del sistema llamados consolas físicas, sin embargo, también existen consolas virtuales o VC's las cuales permiten tener más de una sesión de trabajo abierta desde una consola, para alternar entre consolas virtuales se teclea Alt-F1 o AltF2, hasta Alt-F6, en algunas distribuciones de Linux se cuenta hasta con ocho consolas virtuales activas en el inicio, razón por la cual se debe restringir el acceso y disminuir la cantidad de consolas físicas y virtuales accesibles por el usuario ver **Anexo A.IX**. Algunas de las consolas virtuales más conocidas y utilizadas son:

a) Bourne Shell: escrito por *Steve Bourne* es el *Shell* original de *UNIX*, por lo tanto es más conocido y se encuentra en todas las distribuciones de UNIX, el comando que lo ejecuta es "/bin/sh", su símbolo de intérprete de comandos es "$".

b) Korn Shell: escrito por *David Korn* es compatible con el *Bourne Shell*, el comando que lo ejecuta es "/bin/ksh", su símbolo de intérprete de comandos es "$".

c) C Shell: escrito por *Bill Joy* fue diseñado para incorporar alias e historial de comandos, su sintaxis es parecida a la del lenguaje de programación "C", el comando que lo ejecuta es "/bin/csh", su símbolo de intérprete de comandos es el símbolo "%".

5.4.3 Configurar el acceso por la consola física

Cumple con A.11.2.6 - Seguridad de los equipos y activos fuera de las instalaciones.
Cumple con A.9.4.4 - Uso de herramientas de administración de sistemas. [127]

Para restringir el acceso a las consolas físicas (TTY) con el usuario *root*, se requiere editar el archivo "/etc/securetty", este archivo define la lista de consolas en las que *root* puede iniciar sesión. El archivo "/etc/securetty" es leído y validado por el programa "/bin/login", puesto que tiene un formato con la lista de nombres de dispositivos TTY permitidos, es posible restringir el acceso a cualquier TTY con tan solo agregar al principio de cada línea "#". Por lo general, se recomienda deshabilitar casi todas las

126 El punto 5.4.2 se alinea con el control A.11.2.6 del ISO/IEC 27002:2013

127 El punto 5.4.3 se alinea con los controles A.11.2.6 y A.9.4.4 del ISO/IEC 27002:2013

TTY exceptuando una, por ejemplo la TTY4. Para más detalle del aseguramiento de las consolas físicas de *Linux* revisar el **Anexo A.IX**.

5.4.4 Acceso y configuración de Secure Shell

Cumple con A.9.2.1 - Gestión de altas/bajas en el registro de usuarios.
Cumple con A.9.1.2 - Control de acceso a las redes y servicios asociados.
Cumple con A.9.4.2 - Procedimientos seguros de inicio de sesión. [128]

El servicio de SSH de tipo servidor no está instalado en los sistemas *Linux-Debian*, para instarlo se ejecuta el siguiente comando: `aptitude install openssh-server`. Para limitar que el usuario *root* inicie sesión de forma remota mediante SSH y forzar el uso de una cuenta de usuario normal para posteriormente elevar privilegios de root en el sistema con el comando "su", se requiere editar el archivo "/etc/ssh/sshd_config" y colocar la siguiente línea:

```
PermitRootLogin no
```

En la medida que las políticas de seguridad de la institución lo permitan se recomienda cambiar el puerto de escucha del servicio de SSH predefinido, puerto 22/TCP, por algún otro puerto del rango de los reservados, por ejemplo el 6616/TCP (agregar o modificar la línea con *port 6616*), esto confundirá a un potencial intruso al observar que el puerto común de SSH no es el 22. Cabe destacar que ésta es una técnica de seguridad por oscuridad (se basa en la ocultación de información con el fin de reducir la posibilidad de que un atacante entienda como funciona un sistema), el resto de las configuraciones de seguridad sobre el servicio de SSH se localizan en el **Anexo A.X**.

5.5 Control en las conexiones

5.5.1 Hosts.deny y Hosts.allow

Cumple con A.9.1.2 - Control de acceso a las redes y servicios asociados. [129]

TCP Wrappers es una herramienta que permite monitorear y controlar el tráfico que llega por la red, esta herramienta es útil para la protección de sistema y la detección de actividades indebidas hacia y desde el sistema.

El código original fue escrito por Wietse Venema de la Universidad Tecnológica de Eindhoven, Países Bajos, entre los años 1990 y 1995. Desde el 1 de junio de 2001, el programa es lanzado bajo su propia licencia tipo BSD. [130]]

TCP *Wrappers* permite controlar y proteger los servicios de red mediante la restricción de acceso, sus características de implementación más significativas son:

---

128 El punto 5.4.4 se alinea con los controles A.9.2.1, A.9.1.2 y A.9.4.2 del ISO/IEC 27002:2013

129 El punto 5.5.1 se alinea con el control A.9.1.2 del ISO/IEC 27002:2013

130 TCP Wrapper. Obtenida el 10 de diciembre de 2013, de http://es.cyclopaedia.net/wiki/TCP-Wrapper

- La seguridad lógica está concentrada en un solo programa.
- Los *wrappers* son sencillos.
- El programa protegido se mantiene como una entidad separada.
- Los w*rappers* llaman al programa protegido mediante la llamada al sistema estándar *exec*(), por tanto se usa un solo *wrapper* para controlar el acceso a diversos programas que se necesiten proteger.

El *wrapper* es un programa que controla el acceso a un segundo programa, es decir, cubre la identidad del segundo programa, obteniendo con esto un alto nivel de seguridad.

En el **Anexo A.XI** se indica cómo asegurar el sistema mediante el empleo de los TCP *Wrappers*.

5.5.2 Firewall

Cumple con A.12.4.1 - Registro y gestión de eventos de actividad.
Cumple con A.9.1.2 - Control de acceso a las redes y servicios asociados. [131]

Firewall es una herramienta que se encarga de filtrar el tráfico entre redes, éste puede ser un dispositivo físico (*appliance*) o un software sobre un sistema operativo. Ambas herramientas son efectivas, sin embargo, un dispositivo fabricado especialmente para ser un *firewall* permite mayor sencillez, simplicidad en la configuración y mantenimiento.

Un *firewall* cuenta con dos o más interfaces de red, en las que se establecen reglas de filtrado con las que se decide si una conexión determinada puede establecerse o no, es decir, un *firewall* es un dispositivo específico con un sistema operativo que se encarga de filtrar el tráfico (estático o dinámico) de red a través de diversos protocolos como: TCP, UDP, ICMP, etcétera, el cual implementa decisiones sobre si un paquete debe pasar, ser modificado o debe ser descartado. Un *firewall* colocado entre dos redes debe contar al menos con dos tarjetas de red, la topología clásica de un *firewall* se describe en la Figura 5.29.

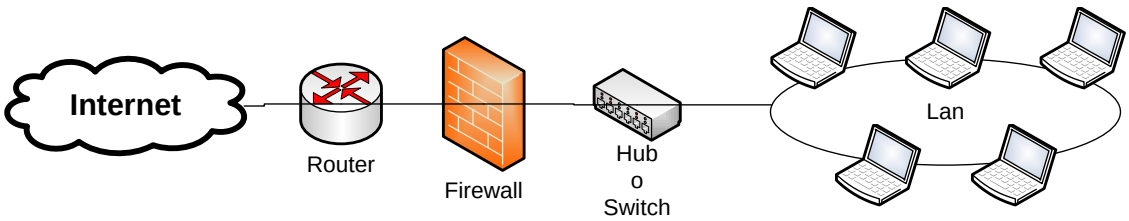


Figura 5.29 – Esquema de ubicación básica de un Firewall

131 El punto 5.5.2 se alinea con los controles A.12.4.1 y A.9.1.2 del ISO/IEC 27002:2013

La Figura 5.29 muestra el esquema típico de un *firewall* colocado para proteger una red local conectada a Internet a través de un *router*. El *firewall* debe colocarse entre el *router* (con un único cable) y la red local (conectado al *switch* o al hub de la LAN).

En el **Anexo A.XII** se detalla la forma de implementar un *firewall* en sistemas *Linux-Debian.*

5.5.3 Iptables

Cumple con A.9.1.2 - Control de acceso a las redes y servicios asociados. [132]

La herramienta *iptables* permite el filtrado y el monitoreo del trafico TCP/IP en sistemas GNU/Linux, las tablas más utilizadas de *iptables* son *filter* y *nat*, la primera ofrece una funcionalidad de filtrado de paquetes y la segunda realiza la traducción de direcciones, dentro de las cadenas de filtrado más usadas son INPUT, OUTPUT y FORWARD. En el caso de la cadena INPUT sólo aplica a paquetes que tienen como destino el *host* local, la cadena OUTPUT aplica sólo a paquetes generados por el *host* local hacia otro host o red y la cadena FORWARD se refiere a paquetes con dirección origen y destino diferentes a las del *host* local los cuales serán encaminados por el *firewall* hacia su destino, entonces el *firewall* tomará decisiones de aceptar el paquete (ACCEPT) o denegarlo (DROP).

La forma como trabajan las cadenas de filtrado es la siguiente: cada cadena tendrá asociada una lista de reglas que serán consultadas de forma secuencial por cada paquete, conforme los paquetes vayan pasando las reglas de las cadenas de filtrado que correspondan, éstos serán examinados para identificar coincidencias entre las reglas generadas y la información de las cabeceras de los paquetes, por último, con base en las coincidencias, la aplicación determina qué acción debe tomar. La Tabla 5.6 muestra un resumen de los comandos básicos de *iptables* para el tratamiento de cadenas.

Tabla 5.6 – Opciones de iptables para tratamientos de cadenas.

Opción	Descripción
-L	Ver las reglas introducidas iptables.
-F	Borrar todas las reglas de iptables.
-N	Creación de una nueva cadena.
-P	Cambia la política por defecto de una cadena.
-X	Elimina una cadena si está vacía, (excepto las cadenas básicas).

Aparte de las operaciones básicas para el tratamiento de cadenas, también se requiere un conjunto de operaciones para el manejo de reglas de una cadena (ver Tabla 5.7).

[132] El punto 5.5.3 se alinea con el control A.9.1.2 del ISO/IEC 27002:2013

Tabla 5.7 – Opciones de iptables para el manejo de reglas.

Opción	Descripción
-t	Indica en qué tabla ubicar la regla, si se omite este comando se agrega la regla a la tabla Filter Table.
-A	Agrega una regla al final de la lista de reglas de la cadena.
-I [pos]	Inserta una regla dentro de la cadena especificada en la posición [pos].
-D [pos]	Borra la regla en la posición especificada.
-R [pos]	Reemplaza una regla en la posición indicada por otra.

En la Tabla 5.8 se muestran las cadenas predefinidas de *iptables.*

Tabla 5.8 – Opciones predefinidas de iptables.

Opción	Descripción
INPUT	Los paquetes que llegan al equipo de cómputo.
OUTPUT	Los paquetes que salen del equipo de cómputo.
FORWARD	Los paquetes que pasan por el equipo de cómputo.

En la Tabla 5.9 se especifican las opciones para el manejo de paquetes.

Tabla 5.9 – Opciones para el manejo de paquetes.

Opción	Descripción
-s [-]	Especifica una dirección de origen*.
-d [-]	Especifica una dirección de destino*.
-p [-]	Especifica un protocolo*.
-i [-]	Especifica un interfaz de entrada*.
-o [-]	Especifica un interfaz de salida*.
-j	Especifica la acción a ejecutar sobre el paquete.
--sport	Puerto de origen*.
--dport [-]	Puerto de destino*.

En la Tabla 5.10 se muestran las acciones de *iptables* que puede implementar en el proceso de control de flujo de los paquetes.

Tabla 5.10 – Opciones para el manejo de paquetes.

Opción	Descripción
DROP	Elimina el paquete.
DENY	Deniega el paso al paquete.
REJECT	Rechaza el paquete.
ACCEPT	Acepta el paquete.
QUEUE	Encola el paquete.
RETURN	Termina la cadena actual y retorna a la inicial.

La sintaxis general de *iptables* es la siguiente, la implementación de políticas para el aseguramiento del sistema se puede observar en el **Anexo A.XIII**.

```
#iptables  <ubicación>  <tipo de cadena>  <especificación>  <acción>
```

5.6 **Mantenimiento**

5.6.1 Actualizaciones de seguridad

Cumple con A.11.2.4 - Mantenimiento de los equipos.
Cumple con A.12.5.1 Instalación del software en sistemas en producción.
Cumple con A.12.6.2 - Restricciones en la instalación de software.
Cumple con A.12.6.1 - Gestión de las vulnerabilidades técnicas. [133]

Hasta este punto, se han implementado acciones para limitar los servicios, las conexiones, los privilegios los usuarios, las respuestas del sistema, entre otros; sin embargo, también se debe mantener una posición proactiva que permita que el sistema siga operando de forma estable y segura. Para tal fin, es necesario tener en cuenta aspectos relacionados con las actualizaciones de seguridad, respaldos periódicos, posibles infecciones por código malicioso, violaciones a la integridad de directorios y archivos críticos, sistemas detectores de intrusos, etcétera. En otras palabras, el *hardening* garantiza que el sistema en "este momento" tiene un nivel de seguridad aceptable; sin embargo, para garantizar que siga teniendo un nivel de seguridad adecuado, es necesario implementar mecanismos para mantener su seguridad.

Es importante que el sistema *Linux-Debian* cuente con las últimas actualizaciones de seguridad, los sistemas *Debian* tienen varias herramientas para la gestión o actualización de paquetes (*apt-get, aptitude, dpkg y synaptic*) instalados en el sistema, a continuación se describen las características y funcionamiento de cada una.

1. APT (*Advanced Package Tool*): es el acrónimo del paquete de herramientas avanzado por sus siglas en inglés (no es un programa, es una biblioteca de funciones C++), el comando *apt* permite la instalación de paquetes a través de *Internet* (ftp o http), además es posible actualizar todos los paquetes del sistema de una forma simple y sencilla.

Por otro lado, la herramienta *aptitude* es una versión mejorada de *apt*, su principal ventaja es que no deja archivos huérfanos cuando se desinstala un paquete, mantiene un *log* de las acciones realizadas en "/var/log/aptitude" y se puede interactuar con la herramienta a través de un ambiente gráfico (ver Figura 5.30).

133 El punto 5.6.1 se alinea con los controles A.11.2.4, A.12.5.1, A.12.6.2 y A.12.6.1 del ISO/IEC 27002:2013

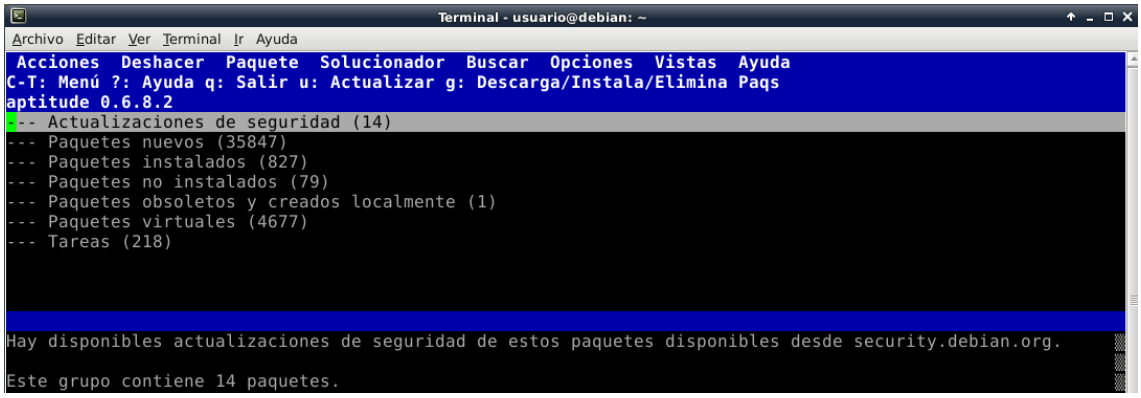


Figura 5.30 – Interfaz gráfica de la herramienta APT.

2. Dpkg: es una herramienta que puede utilizarse para instalar, desinstalar o consultar información sobre los paquetes instalados en el sistema. Se usa principalmente para instalar un paquete Debian ya disponible, puesto que esta herramienta no descarga paquetes desde Internet.

3. Synaptic: es una interfaz gráfica de usuario para APT (Gestor de paquetes), utiliza repositorios de Debian y gestiona los paquetes mediante un menú interactivo, además es capaz de reparar dependencias rotas de paquetes, permite deshacer y rehacer las selecciones de paquetes. *Synaptic* es uno de los gestores de paquetes más potente y flexible de los sistemas Linux (ver Figura 5.31).

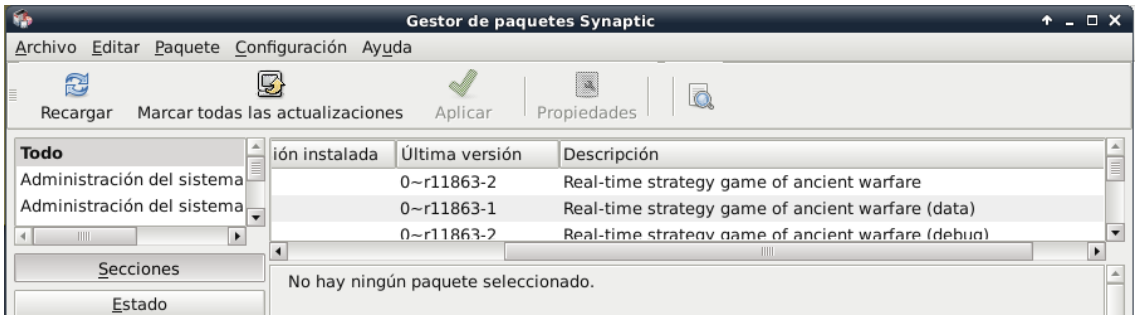


Figura 5.31 – Gestor de paquetes Synaptic.

En el caso de Debian-Wheezy, los repositorios se definen en el archivo "/etc/apt/source.list", el cual contiene las rutas de los servidores que contienen los paquetes disponibles para la versión del sistema, para el caso de Debian- Wheezy, los repositorios de los paquetes son los siguientes:

```
deb http://ftp.fr.debian.org/debian/ wheezy main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ wheezy main contrib non-free
deb http://security.debian.org/ wheezy/updates main
deb-src http://security.debian.org/ wheezy/updates main
```

Para actualizar la base de datos local de los paquetes disponibles en los repositorios, se puede utilizar el comando `aptitude update`, si lo que se desea es actualizar los paquetes que se tienen instalados en el sistema se emplea el comando: `aptitude upgrade` de esta forma no sólo se actualizarán los repositorios. El resto de las configuraciones para las actualizaciones de seguridad se pueden consultar en el **Anexo A.XIV**.

5.6.2 Respaldos

Cumple con A.12.4.1 - Registro y gestión de eventos de actividad.
Cumple con A.12.3.1 - Copias de seguridad de la información.
Cumple con A.12.4.4 - Sincronización de relojes. [134]

Dentro de las acciones de mantenimiento del sistema se incluyen los respaldos de los archivos críticos e importantes para los usuarios. La importancia de respaldar difiere de los escenarios y ambientes en los que se encuentre el equipo que resguarda la información, sin embargo, los equipos de trabajo de *CERTs* como cualquier otro equipo informático, se exponen más aun a diversas amenazas (*malware*, fallos de electricidad, errores de hardware y software, errores humanos, incendios, inundaciones, terremotos, sabotaje, entre otros) que podrían comprometer la seguridad del sistema y de la información que almacena, si bien no se pueden prevenir las amenazas, sí es posible evitar las consecuencias catastróficas sobre el sistema y su información, por lo tanto, para implementar respaldos se deben considerar los siguientes aspectos:

- La importancia y criticidad de los datos que se guardarán (esto es subjetivo y depende de cada organización).
- La periodicidad con la que se crearán los respaldos.
- La protección contra fallos en los medios de almacenamiento, que está relacionada con el número de dispositivos de almacenamiento usados.
- El almacenamiento alternativo, es decir, la posibilidad de guardar en una diferente ubicación (física) los respaldos realizados.
- Mecanismos de comprobación.
- Considerar el volumen a copiar.
- El tipo de respaldo que se implementará (total, incremental y diferencial).
- El costo beneficio de realizar el respaldo periódico.

En la Figura 5.32 se simplifican los aspectos más relevantes que se deben tener en cuenta en la realización de respaldos. [135]

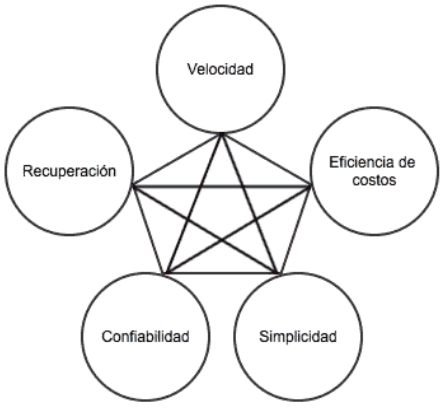


Figura 5.32 – Aspectos a considerar en los respaldos de información.

134 El punto 5.6.2 se alinea con los controles A.12.4.1, A.12.3.1 y A.12.4.4 del ISO/IEC 27002:2013

135 Consideraciones para comprar respaldo para un servidor. Obtenida el 14 de diciembre de 2013, de http://blog.iweb.com/es/2013/08/comprar-respaldo-para-servidor/2298.html

No existe un criterio definitivo que indique la periodicidad con la que deben realizarse las copias de seguridad, sin embargo, existen factores que pueden ser de utilidad para definir dicha periodicidad como: el tiempo invertido para la creación de los archivos, el costo, las consecuencias por su pérdida, etcétera. Además, se debe definir el periodo de retención de los respaldos, lo cual está relacionado con la naturaleza de la información y su utilización. Para un equipo de cómputo que pertenece a un CERT, la naturaleza de la información que se maneja puede estar relacionada con casos de investigación, estadísticas, información de inteligencia, manejo de incidentes, análisis de vulnerabilidades y artefactos, esta información exige un amplio periodo de retención pues no se sabe en qué momento la información pueda ser nuevamente requerida. En cuanto a los tipos de respaldos se tienen tres: respaldo total, incremental y diferencial:

a) Respaldo total: también denominado normal, crea una copia de todas las unidades, carpetas y archivos seleccionados, sin tener en cuenta si han sido modificados o creados desde la última copia de seguridad.
b) Respaldo Incremental: guarda únicamente los archivos que hayan sido modificados o creados desde la última copia incremental realizada. La primera copia de este tipo es idéntica, por lo tanto es una copia total. Este tipo de copia es la recomendada para respaldos más frecuentes.
c) Respaldo Diferencial: este tipo de respaldo también copia los archivos modificados o creados recientemente, pero en lugar de usar como referencia la última copia incremental, se usa la última copia completa.

El respaldo diferencial es más avanzado, debido a que sólo copia los archivos creados o modificados, es decir, no sólo compara la fecha de modificación, también compara el contenido de los archivos (integridad) es por ello que requiere menos espacio en disco para guardar los respaldos. La diferencia más significativa entre los respaldos diferenciales e incrementales es la siguiente:

- Para recuperar archivos desde un respaldo diferencial, se requiere el último respaldo completo y la versión del respaldo diferencial que se desea recuperar.
- Para recuperar archivos desde un respaldo incremental, se requiere el último respaldo completo y todas las versiones de los respaldos incrementales desde el primero hasta la versión deseada.

Por otro lado, para tener una correcta correlación de los eventos y respaldos del sistema así como interacción con otros sistemas, es importante establecer un esquema de sincronización de relojes, para realizar esta tarea se cuenta con la herramienta *ntpdate*, la instalación en el equipo Linux-Debian se muestra en la Figura 5.33.

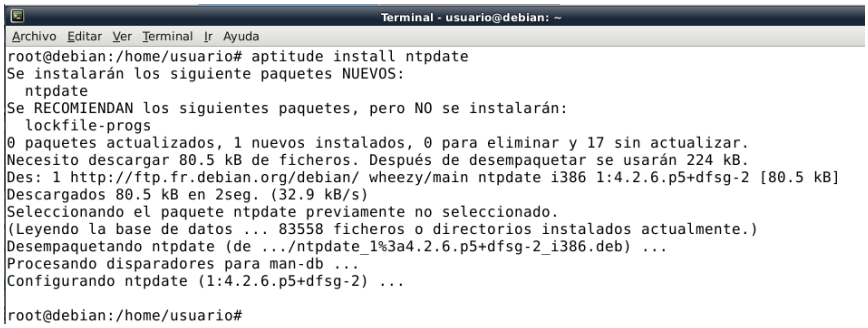


Figura 5.33 – Aspectos a considerar en los respaldos de información.

Una vez instalado, el siguiente paso es sincronizar el reloj del sistema con un servidor de NTP (*Network Time Protocol* – Protocolo de Red de Tiempo), para lo cual se utilizará la opción "-u" puesto que el sistema tiene configurado un *firewall* que bloquea el uso de puertos privilegiados, con "-u" se indica que use un puerto no privilegiado (> 1024). Para que la sincronización se realice de forma periódica con el servidor **ntp.crya.unam.mx**, se crea y edita el archivo "/etc/cron.hourly/ntp" como se muestra en la Figura 5.34, una vez que el archivo sea creado el sistema se sincronizará cada hora.

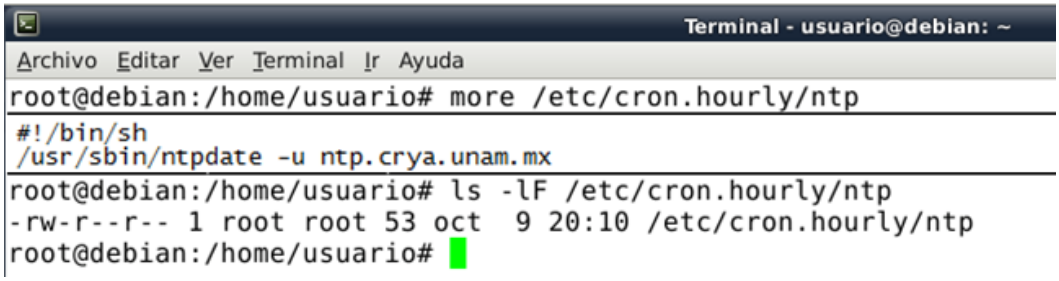


Figura 5.34 – Aspectos a considerar en los respaldos de información.

En el **Anexo A.XV** se continúa con la implementación de respaldos en el sistema *Linux-Debian,* en el cual se realiza un respaldo periódico y automatizado del archivo "/var/log/syslog" y posteriormente se envía mediante un mecanismo de seguridad a un servidor de respaldos externo.

5.6.3 Antivirus

Cumple con A.12.2.1 - Controles contra el código malicioso. [136]

Los programas antivirus también realizan un tipo de auditoría sobre el sistema, sólo que éstos se basan principalmente en comparar el *hash* de los archivos con listas de *hashes* relacionadas con códigos maliciosos. Existe una gran diversidad de software antivirus gratuitos para sistemas Linux, para este trabajo se empleará el antivirus ClamAV la forma sencilla de instalarlo es a través de `aptitude install clamav` (ver Figura 5.35)

136 El punto 5.6.3 se alinea con el control A.12.2.1 del ISO/IEC 27002:2013

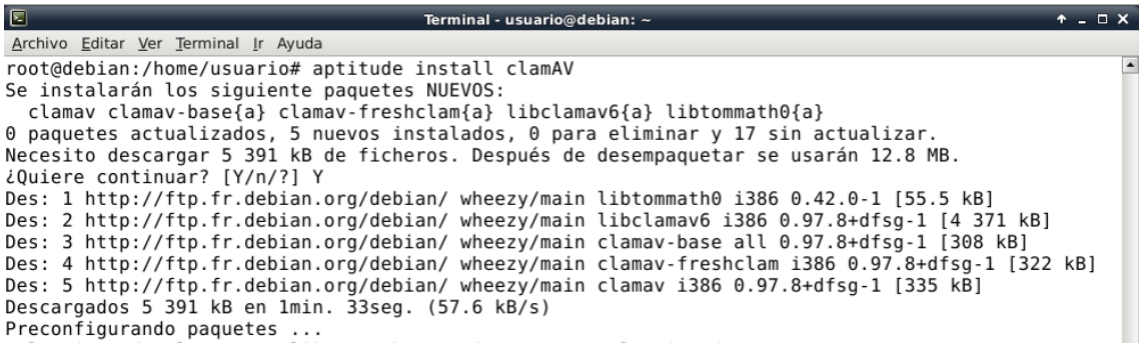


Figura 5.35 – Instalación de antivirus ClamAV.

En el **Anexo A.XVI** se continúa con la configuración del antivirus de forma gráfica.

5.6.4 Auditorías regulares

Cumple con A.12.2.1 - Controles contra el código malicioso.
Cumple con A.12.6.1 - Gestión de las vulnerabilidades técnicas.
Cumple con A.16.1.3 - Notificación de puntos débiles de la seguridad.
Cumple con A.12.7.1 Controles de auditoría de los sistemas de información. [137]

Los programas que auditan la seguridad en los sistemas operativos son herramientas indispensables para el mantenimiento de un sistema, ya que permiten detectar, problemas de seguridad para los que pudieran existir ataques conocidos. Los programas de auditoría pueden revisar diversos aspectos de seguridad, desde comprobar la pertenencia de archivos a usuarios y grupos del sistema, hasta pruebas sobre aplicaciones instaladas para verificar si éstas tienen vulnerabilidades conocidas. Existen diversas herramientas de auditoría de seguridad para Linux como: *Cops*, *Satan*, *Sara*, *Nessus*, *Tiger*, entre otros.

En este trabajo de tesis se optó por utilizar la herramienta de *Tiger*, esta aplicación permite revisar la seguridad del sistema mediante potentes verificaciones de archivos de configuración y el estado de varios elementos del sistema operativo de forma periódica. La herramienta *Tiger* analiza el sistema con el objeto de identificar diversas maneras de obtener privilegios de súper usuario. El diseño de esta herramienta parte de la hipótesis de que cualquier UID o GID puede ser obtenido por terceras personas no autorizadas, obtener acceso y posteriormente comprometer la seguridad del sistema. Las revisiones de la herramienta *Tiger* se enfocan en los siguientes aspectos:

- Variables de inetd.
- Variables del PATH.
- Permisos de archivos y directorios.
- Revisa existencia de parches de mantenimiento.
- *Paths* en archivos de configuración que representen un riesgo.

137 El punto 5.6.4 se alinea con los controles A.12.2.1, A.12.6.1, A.16.1.3 y A.12.7.1 del ISO/IEC 27002:2013

En el **Anexo A.XVII** se describe la forma de instalar la herramienta *Tiger* y cómo se ejecuta en el sistema para llevar a cabo la auditoría de archivos críticos del sistema.

5.6.5 Validación de integridad

Cumple con A.12.4.1 - Registro y gestión de eventos de actividad. [138]

La validación de integridad, permite identificar cambios en los archivos críticos del sistema o archivos necesarios para la operación, como primer paso, el análisis de integridad requiere generar un punto de partida "*base-line*", es decir, contar con un estado "ideal" el cual será tomado como referencia para identificar cuándo un archivo ha sufrido modificaciones en sus contenido. Existen varias herramientas para realizar la verificación de la integridad de los archivos existentes en el sistema de archivos:

- *Tripwire*: prácticamente la más antigua de todas (sólo de uso libre para Linux, y bajo ciertas condiciones de licenciamiento), alto nivel de configuración.
- *Integrit*: es un clon libre de tripwire.
- *Debsums*: verifica la integridad de los archivos con base en el *hash* contenido en el paquete *Debian* que lo provee, sin embargo, no todos los paquetes tienen los hashes necesarios.
- AIDE: actualmente una de las más usadas y recomendables.

En general, las herramientas anteriores proveen varias opciones de verificación, incluyendo *hashes* con uno o más algoritmos criptográficos, verificación de *MAC time* (Modificación, Acceso y Cambio), permisos, tamaño, etcétera. Los directorios que se sugiere sean supervisados por estas herramientas están en "/etc" el cual contiene todos los archivos de configuración, "/boot", donde se encuentran los archivos de inicio del sistema operativo. Por otro lado, los directorios que sufren constantes cambios son:

- /var/log tiene configuraciones específicas que seguramente causarán varios falsos positivos si la actividad en el equipo modifica los *logs* constantemente.
- /home sólo está contemplado como un ejemplo. Si el sistema posee muchos usuarios independientes, sólo interesa revisar que no cambie el dueño de los archivos.

En este trabajo se utilizará el software *Tripwire* para la verificación de integridad de los datos. *Tripwire* de manera automática y en intervalos regulares realiza la verificación de la información, en caso de detectar que uno de los archivos monitoreados ha sido cambiado, lo notifica al administrador del sistema. *Tripwire* puede identificar cuando los archivos han sido modificados, agregados o eliminados, facilitando la detección de intrusos y control de posibles daños en el sistema.

Tripwire requiere contar con un estado *base-line* del sistema, para iniciar la comparación de sistema, el base-line se compone de una base de datos de archivos, la cual se genera una vez que se tenga certeza de que el sistema tiene un grado de seguridad y funcionalidad adecuada para las necesidades del CERT. Después de

138 El punto 5.6.5 se alinea con el control A.12.4.1 del ISO/IEC 27002:2013

crear la base de datos de archivos “base”, *Tripwire* comparará la base de datos inicial con la base de datos de un segundo momento e informará de cualquier modificación, adición o eliminación. En la Figura 5.36 se muestra un diagrama de flujo el cual ilustra cómo funciona *Tripwire*.

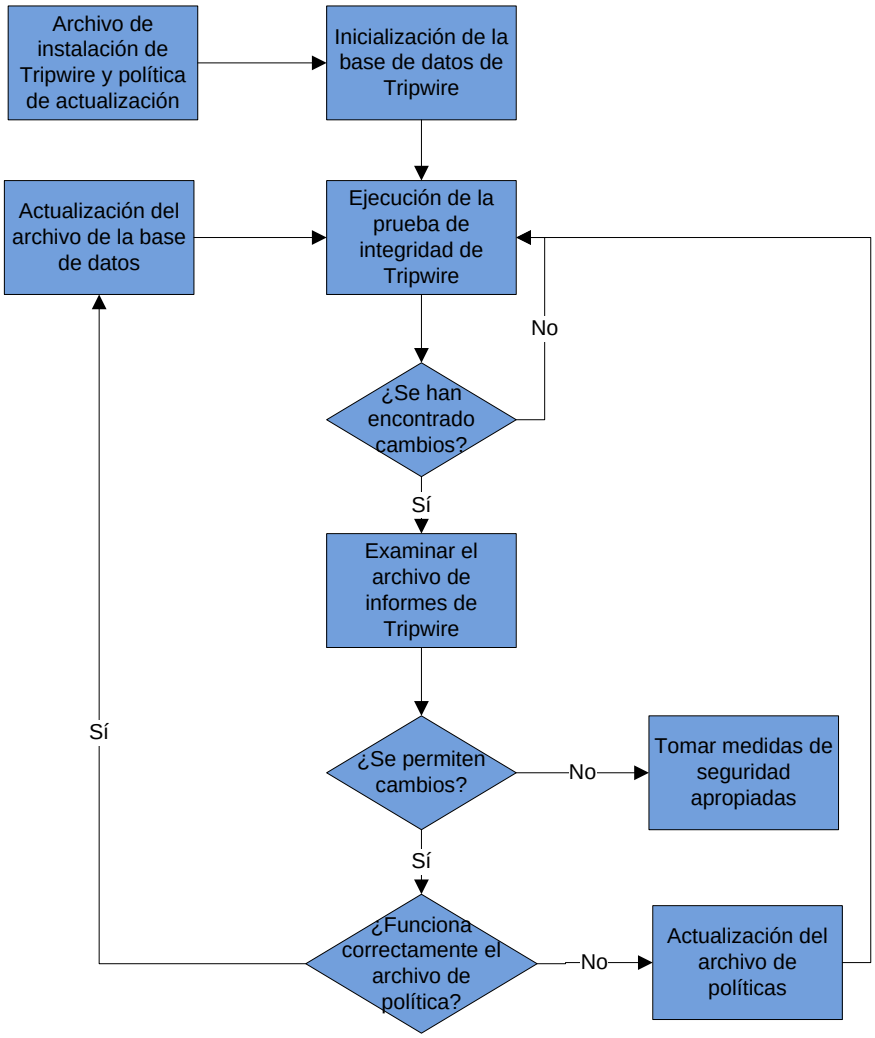


Figura 5.36 – Diagrama de funcionamiento de Tripwire.

La instalación y configuración de *Tripwire* se describe en el **Anexo A.XVIII** de este documento.

5.6.6 IDS/IPS de Host

Cumple con A.16.1.2 - Notificación de los eventos de seguridad de la información.
Cumple con A.12.4.1 - Registro y gestión de eventos de actividad. [139]

Un Sistema de Detección de Intrusos (*Intrusion Detection Systems*) es un mecanismo más dentro del modelo de seguridad. Este mecanismo consiste en detectar actividades no permitidas, incorrectas, anómalas y maliciosas que fluyen del exterior

139 El punto 5.6.6 se alinea con los controles A.12.4.1 y A.16.1.2 del ISO/IEC 27002:2013

hacia un sistema informático [140]. Hay tres tipos de Sistemas de Detección de Intrusos (IDS):

- Host IDS (IDS): los sensores se encuentran en cada equipo de cómputo, actúa como un demonio o servicio en el sistema el cual verifica señales de intrusión como puertas traseras, troyanos, ejecución de código malicioso, desbordamientos de *buffer overflow*, entre otros.
- *Network IDS* (NIDS): los sensores se encuentran en segmentos de red, por lo tanto revisan el tráfico que circula en dicha red.
- *Distributed IDS* (DIDS): en la práctica se trata de varios NIDS que se comunican con un sensor central que correlaciona la información que recibe.

Las principales fortalezas de los IDS son: suministrar información relevante sobre el tráfico malicioso en la red, detección del origen de los ataques, recolecta evidencia que permite identificar a intrusos, genera alertas de intentos de acceso y dificulta al intruso la posibilidad de eliminar el rastro de la intrusión.

Los IPS (*Intrusion Prevention System* – Sistemas de Prevención de Intrusiones) son mecanismos dedicados a la prevención de intrusiones a partir de la identificación de firmas, comportamiento en el sistema y posteriormente el bloqueo e informe de actividad potencialmente dañina para la seguridad del sistema.

Los IPS son especialmente útiles para comprobar los módulos cargados por el núcleo, monitorear la actividad del sistema de archivos, buscar *RootKits* en el sistema, además de las funcionalidades de los IDS, entre otros. No todos los IPS son iguales y debido a que trabajan a bajo nivel mediante la intercepción de llamadas al sistema es necesario tener en cuenta que el IPS debe ser confiable, no debe impactar en el rendimiento del sistema y no debe bloquear tráfico legítimo. Si un IPS no cumple con los requisitos antes señalados es preferible no tenerlo.

File2ban es un IDS eficiente (existen otras opciones como *DenyHost* y *BlockHosts*) para proteger al sistema ante ataques de diccionario o fuerza bruta remotos sobre el servicio SSH. Para ello, se debe instalar la herramienta con el comando: aptitude install *Fail2ban*, posteriormente en el archivo de configuración (/etc/fail2ban/fail2ban.conf) se deben agregar las siguientes líneas:

```
logtarget = /var/log/fail2ban.log # archivo donde se registran los eventos
background = true  # Indica que fail2ban se ejecute como un demonio
maxretry = 3  # El número máximo de intentos antes de ser bloqueado
bantime = 600  # El tiempo que será bloqueada la IP o cuenta 600/60=10 minutos
```

Posteriormente se debe reiniciar el servicio para que las configuraciones sean tomadas en cuenta: `/etc/init.d/fail2ban restart`. Con el propósito de verificar el bloqueo de la IP después de tres intentos desde otro se intentó conectarse mediante el servicio SSH, equivocando erróneamente las contraseñas, se observa en la Figura 5.37 después de 3 intentos fallidos.

[140] Detección de Intrusos en Tiempo Real. Obtenida el 14 de diciembre de 2013, de http://www.segu-info.com.ar/proteccion/deteccion.htm

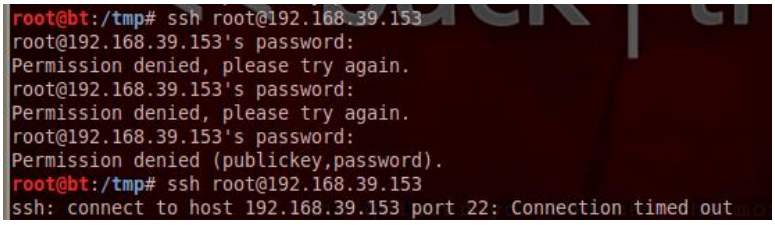


Figura 5.37 – Verificación de funcionamiento de IPS.

Para verificar lo anterior, se revisa el archivo de *logs* (ver Figura 5.38) de la herramienta *fail2ban* "/var/log/fail2ban.log" donde efectivamente se observa que el equipo con dirección IP 192.168.39.155 fue bloqueado.

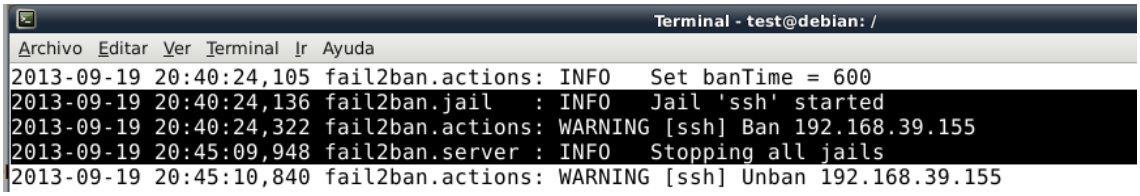


Figura 5.38 – Revisión de logs sobre bloqueo automático de direcciones IP.

5.6.7 Análisis de bitácoras

Cumple con A.12.4.1 - Registro y gestión de eventos de actividad.
Cumple con A.12.4.3 - Registros de actividad del administrador y operador del sistema.
Cumple con A.9.2.1 - Gestión de altas/bajas en el registro de usuarios. [141]

Los *logs* (registros o bitácoras) del sistema son muy valiosos al momento de realizar investigaciones relacionadas con intentos o accesos no autorizados que pudieran comprometer la seguridad de los sistemas. El monitoreo de los *logs* es una parte elemental de las políticas de seguridad de la información, ya que la detección oportuna de intentos de acceso para comprometer el sistema, podría evitar y prevenir impactos graves sobre la seguridad de la información.

En la mayoría de los sistemas *UNIX* y sistemas *GNU/Linux*, se utiliza el sistema *syslog* para el manejo de los *logs* del sistema. Este sistema cuenta con un demonio que recibe los diferentes mensajes de las aplicaciones y del propio sistema operativo, se configura con el archivo "/etc/syslog.conf", el cual tiene dos conceptos fundamentales: *facility* y *level*. El primero se refiere a la aplicación o componente del sistema operativo que genera *logs* y *level* hace referencia a la severidad del mensaje, por lo tanto para cada combinación de estos parámetros se realiza una acción específica en el registro de eventos en el sistema. Estos archivos, comúnmente residen en el directorio "/var/log" lo conveniente es que estos mensajes sean guardados en un repositorio externo para evitar que sean alterados en caso de una intrusión. Una herramienta que puede ser útil para la lectura de los *logs* almacenados, que suelen tener un tamaño importante, es *logwatch*.

141 El punto 5.6.7 se alinea con los controles A.12.4.1, A.12.4.3 y A.9.2.1 del ISO/IEC 27002:2013

Logwatch es un analizador de bitácoras (*logs*) desarrollado en lenguaje Perl, esta herramienta también puede reportar la actividad del sistema a través de correo electrónico, además revisa de forma periódica las bitácoras del sistema y clasifica los *logs* según diferentes niveles de alerta pre-configurados, al final genera un reporte para el administrador del sistema en un formato simple y concreto. La forma de instalación se realiza mediante *aptitude*.

Una vez instalado, el archivo que contiene la información de configuración se encuentra en "/usr/share/logwatch/dist.conf/logwatch.conf", en este archivo se indica a logwatch: la ruta en donde se alojan los *logs* de interés (*LogDir*), el nivel de detalle de la información (*Detail*) que se desea recibir de los registros y el rango de tiempo de búsqueda (Range) con las opciones *All, Today* o *Yesterday*, ver Figura 5.39.

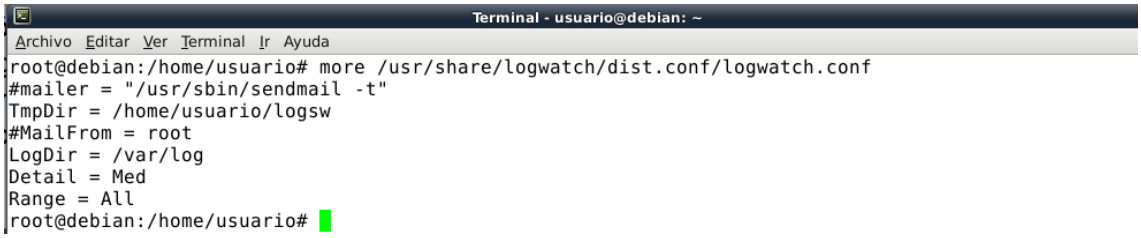


Figura 5.39 – Configuración de herramienta logwatch.

Una vez terminada la configuración, la forma de iniciar la revisión de bitácoras es ejecutando `logwatch` en una terminal (ver Figura 5.40). Posteriormente, *logwatch* generará un informe detallado, producto de la revisión de los registros.

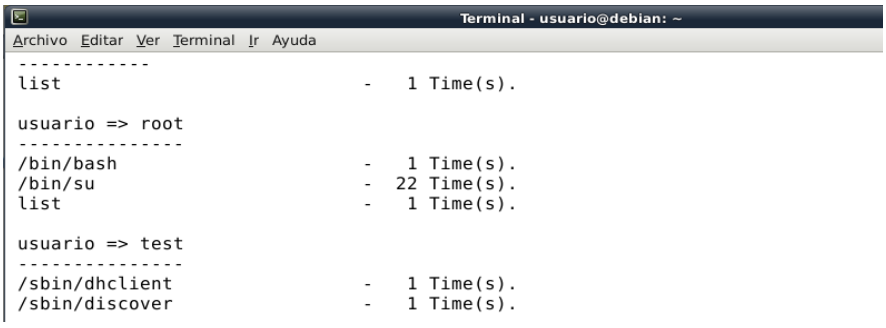


Figura 5.40 – Ejecución de logwatch en el sistema.

Para asegurar que la revisión se realice cada semana, se podría crear y editar un script que indique la ejecución de la herramienta y posteriormente se colocaría en "/etc/cron.weekly/", dicho *scritp* tendría el siguiente contenido:

```
#!/bin/sh
/usr/sbin/logwatch
```

5.7 Herramientas de seguridad en Debian

5.7.1 SELinux

Cumple con A.12.6.1 - Gestión de las vulnerabilidades técnicas. [142]

Existen herramientas que facilitan la implementación del *hardening* en un sistema operativo Linux, son útiles puesto que automatizan el procedimiento para fortalecer la seguridad, sin embargo, por ninguna razón se debe reemplazar el trabajo manual y la revisión de las configuraciones hechas por el administrador del sistema. En este trabajo se utiliza la herramienta SElinux y se habilitán las características del registro de actividades, pues ejecutar la herramienta en modo *enforcing*–forzado sobre el sistema ya *hardenizado,* podría alterar o modificar algunas configuraciones previamente realizadas en el sistema.

La herramienta *SELinux* (*Security-Enhanced Linux* – Seguridad Mejorada de Linux), es una colección de parches que se emplean en una arquitectura de seguridad que emplea los módulos de seguridad de *Linux Security Modules* (LSM), es un proyecto de la Agencia de Seguridad Nacional (NSA) de los Estados Unidos y de la comunidad *SELinux*.

SELinux proporciona un sistema flexible de control de acceso obligatorio (*Mandatory Access Control* – Control de Acceso Mandatorio) incorporado al *kernel* de Linux. Por otro lado para el sistema operativo Linux se emplea el Control de Acceso a Discreción - *Discretionary Access Control*, en el que un proceso o aplicación se ejecuta como un usuario (UID o SUID) y tiene los permisos que el usuario tiene sobre objetos, archivos y otros procesos.

Al ejecutar un *kernel* SElinux se protege al sistema de aplicaciones maliciosas o dañinas que pueden perjudicar o destruir el sistema. *SELinux* define el acceso y los derechos de cada usuario, aplicación, proceso y archivo en el sistema. Por lo tanto, *SELinux* es prácticamente invisible para la mayoría de los usuarios, salvo para el administrador del sistema, esto le da al *kernel SELinux* un control total y granular sobre el sistema.

Los requisitos previos para la instalación de *SElinux* son utilizar un sistema de archivos como: ext2, ext3, ext4, JFS y XFS; y utilizar un kernel de Linux. Los pasos siguientes describen cómo instalar y configurar *SELinux*, junto con la política específica que permite el uso de *SElinux* a las partes más importantes del sistema: `apt-get install selinux-basics selinux-policy-default`

Para activar *SElinux* (ver Figura 5.41) en el sistema se debe ejecutar: `selinux-activate` posteriormente reiniciar el sistema para que los cambios surtan efecto.

142 El punto 5.7.1 se alinea con los controles A.12.6.1 del ISO/IEC 27002:2013

```
Terminal - usuario@debian: ~
Archivo  Editar  Ver  Terminal  Ir  Ayuda
root@debian:~# selinux-activate
Activating SE Linux
Generating grub.cfg ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-3.2.15grsec2.2.0
Found initrd image: /boot/initrd.img-3.2.15grsec2.2.0
Found linux image: /boot/vmlinuz-3.2.0-4-686-pae
Found initrd image: /boot/initrd.img-3.2.0-4-686-pae
done
SE Linux is activated.  You may need to reboot now.
root@debian:~#
```

Figura 5.41 – Activación de SElinux en el sistema.

En el **Anexo A.XIX** se continúa con la configuración y ejecución de la herramienta *SElinux*.

5.8 Resultados obtenidos

Una vez que se concluyó con la implementación de las buenas prácticas de seguridad en el equipo con sistema operativo Linux-Debian, se evaluó la efectividad del *hardening*, para lo cual se realizaron escaneos de vulnerabilidades con la herramienta Nessus versión 5, con un enfoque externo de acuerdo a la Figura 5.42, comparando el estado inicial del equipo Linux (recién instalado) y el estado final (con *hardening*), para más detalle consultar el **Anexo C**.

Figura 5.42 – Esquema de escaneo de vulnerabilidades (Externo).

Posteriormene se realizó un análisis de vulnerabiliades con un enfoque interno con la herramienta Lynis versión 2.1, con el mismo escenario antes mencionado (estados inicial y final), como se observa en la Figura 5.43 para más detalle consultar el **Anexo C**.

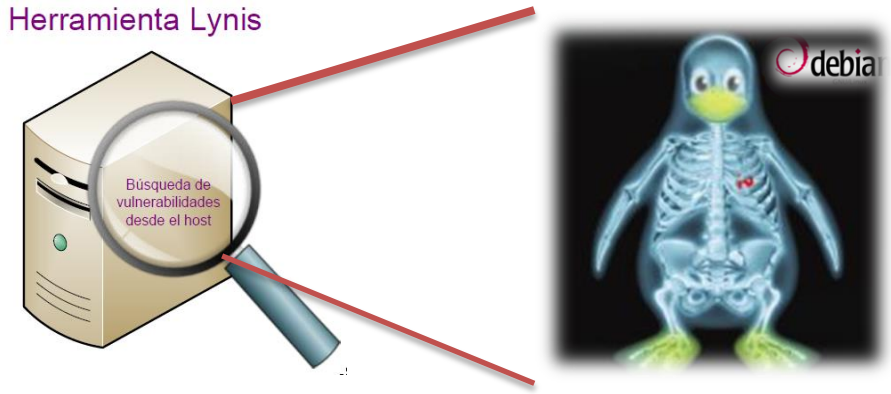


Figura 5.43 – Esquema de escaneo de vulnerabilidades (Interno).

En los dos tipos de análisis de vulnerabilidades realizados (enfoque externo e interno), se verificó que aumentó el nivel de seguridad y por consiguiente se redujó el número de vulnerabilidades del sistema Linux-Debian respecto del estado inicial y final.

Conclusiones

En este trabajo de tesis, el objetivo general fue elaborar una guía de buenas prácticas de seguridad informática que pudiera ser implementada en estaciones de trabajo Linux-Debian de un Equipo de Respuesta a Incidentes de Seguridad en Cómputo (CERT) y que además, fuera una guía alineada a los controles aplicables del ISO/IEC 27002:2013, y de esta forma facilitar la ardua labor de certificación del proceso de gestión de incidentes de un CERT, considerando que este proceso es su principal actividad y sobre la cual se detonan el resto de los servicios proactivos y reactivos que ofrecen a su comunidad objetivo. Lo anterior, con la finalidad de alcanzar un nivel de seguridad adecuado y óptimo en los equipos de cómputo, para así garantizar la confidencialidad, integridad y disponibilidad de la información en organizaciones que gestionan y atienden incidentes en cómputo.

Derivado del análisis e implementación de buenas prácticas de seguridad (alineadas al ISO/IEC 27002:2013) dirigidas al aseguramiento de sistemas Linux-Debian de un CERT, se concluye lo siguiente:

- Fue posible conseguir un punto de equilibrio en el cual el sistema sobre el cual se aplicaron las buenas prácticas de seguridad, es considerado con un nivel de seguridad adecuado acorde con el rol que podría desempeñar en un CERT y a su vez, es un sistema funcional, es decir, no presenta limitantes para realizar las tareas de análisis y procesamiento de datos relacionados con incidentes de seguridad de la información; adicionalmente se tiene un sistema alineado a los controles del ISO/IEC 27001:2013, lo cual facilitará el proceso de certificación del área de respuesta a incidentes.

- El proceso de hardening o aseguramiento de sistemas operativos no tiene como propósito forjar equipos de cómputo invulnerables, de acuerdo con el modelo de defensa en profundidad explicado en este trabajo, el *host* (equipo de cómputo), sólo es una capa de defensa, es decir, el *hardening* representa un factor más a considerar dentro del gran número de capas de protección que deberían ser contempladas para defender de forma integral un equipo de cómputo.

- El aseguramiento de sistemas de cómputo, es un proceso que NO debiera delegarse a ningún software, debido a que éste modifica de forma automática archivos de configuración del sistema con el objetivo de lograr un determinado nivel de seguridad, es decir, el usuario no tiene pleno control sobre las alteraciones realizadas por este tipo de software y se corre un alto riesgo de afectar la funcionalidad del sistema, por tal razón, la forma más adecuada de llevar a cabo este proceso, es realizarlo de manera paulatina, gradual y en la medida de lo posible hacer las modificaciones manualmente, lo cual permitirá ir verificando en cada paso la eficacia de las medidas de aseguramiento implementadas y cómo estas acciones de *hardening*, podrían afectar la funcionalidad del sistema Linux-Debian.

- Las buenas prácticas de seguridad propuestas en este trabajo de tesis, una vez que son implementadas en los sistemas Linux-Debian, les brindan un nivel

alto de seguridad sin afectar sustancialmente su funcionalidad, permitiendo que estos sistemas puedan ser empleados para las actividades que comúnmente se llevan a cabo en el proceso de respuesta a incidentes de seguridad que los CERTs implementan; sin embargo, por obvias razones dependiendo del tipo de actividad que el usuario tenga encomendada es posible que surja la necesidad de modificar el nivel de seguridad en ciertos aspectos, y es aquí donde se observa otro valor agregado de este trabajo de tesis, debido a que la mayoría de las actividades de aseguramiento se realizaron manualmente y se describen a detalle, el usuario por sí mismo podría incrementar o disminuir el nivel de seguridad de acuerdo con sus necesidades, sin que exista un riesgo alto de que el sistema deje de estar alineado al ISO/IEC 27001:2013, sólo en caso de que la organización ya tenga certificado el proceso de gestión de incidentes con este estándar.

- El proceso de seguridad informática es una estrategia de mejora continua, es decir, es un proceso que nunca termina ya que los riesgos nunca se eliminan, sólo se mitigan y están directamente relacionados con los constantes cambios en las Tecnologías de Información así como con las diversas amenazas cibernéticas que evolucionan, se perfeccionan y se especializan con el tiempo, como es el caso de las APTs (*Advanced Persistent Threats* – Amenazas Persistentes Avanzadas), es por esto, que probablemente las buenas prácticas de seguridad establecidas en este trabajo de tesis, en un futuro no muy lejano, ya no brinden a los sistemas donde fueron implementadas un nivel de aseguramiento adecuado y sea necesario actualizarlas para estar a la altura de las circunstancias. En este sentido, dado que las acciones de aseguramiento implementadas en este trabajo son flexibles y fáciles de calibrar debido a que se ocupó un sistema operativo Linux, no será difícil para el usuario recobrar un nivel de seguridad óptimo acorde con las amenazas del momento.

Finalmente se concluye que el proceso de aseguramiento (*hardening*) de sistemas de cómputo se torna complicado, cuando no se tienen claras las respuestas a las siguientes preguntas: ¿qué es lo que se quiere proteger?, ¿de quién se requiere proteger? y ¿con qué fin se desea proteger?, una vez que se tienen plenamente claras las respuestas a las preguntas anteriores, es más fácil realizar el aseguramiento de cualquier sistema operativo, sin afectar la funcionalidad del mismo y de estar forma evitar que el *hardening*; pase de ser una importante ayuda, a un dolor de cabeza para el usuario y la organización.

A. Anexo

I. (5.1.2) Instalación de LINUX-DEBIAN con particiones separadas

Para la instalación del sistema operativo *Debian* se empleó la imagen “debian-7.1.0-i386-xfce-CD.iso” la cual fue descargada desde el sitio Web oficial de *Debian*: http://www.debian.org/CD/http-ftp/. La instalación se realizó con una máquina virtual utilizando el software de virtualización VMWare versión 5.0 para sistemas MAC OS X, como se muestra en la Figura A. 1.

Figura A. 1 - Software de virtualización.

Antes de instalar el sistema operativo se creó una máquina virtual con 1024 MB de memoria RAM, un disco virtual de 13 GB capacidad y un procesador de tipo Intel i386. Cabe resaltar que se creó una máquina virtual con las capacidades antes mencionadas, porque se consideran capacidades suficientes para la instalación de aplicaciones y configuraciones producto del *hardening* implementado en este trabajo de tesis, en ambientes reales las configuraciones dependerán de las necesidades y recursos con los que se cuente. Una vez creada la máquina virtual se indica que inicie desde el CD/DVD el cual previamente se configuró con la ruta donde se localiza el archivo “debian-7.1.0-i386-xfce-CD.iso”, la interface inicial para la instalación se muestra en la Figura A. 2, el siguiente paso es elegir la opción de “Install”.

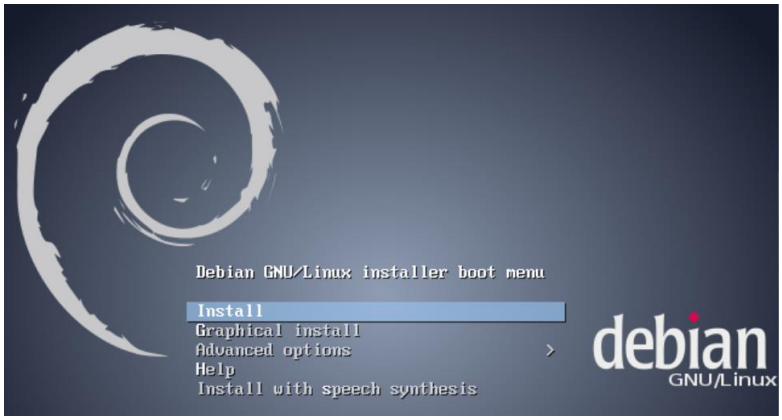


Figura A. 2 - Interface de instalación de Debian.

Debido a que el proceso de instalación es largo y sale del alcance de este trabajo, sólo se presentaran algunas fases de la instalación que se consideran las más relevantes para comprender el proceso de instalación de Debian con particiones separadas. Para más detalle del proceso de instalación consultar la página Web oficial de Debian. En la Figura XX se observa el establecimiento del nombre que llevara la máquina virtual, en este caso el nombre sugerido es “debian” (ver Figura A.3), es de notar que este nombre genérico no proporciona información del equipo, su entorno, propósito, etcétera. Es decir, nombres como “server-cert”, “cert”, “equipo-cert”, “juan.perez”, podrían proporcionar información de más que un intruso pudiera utilizar con fines maliciosos.

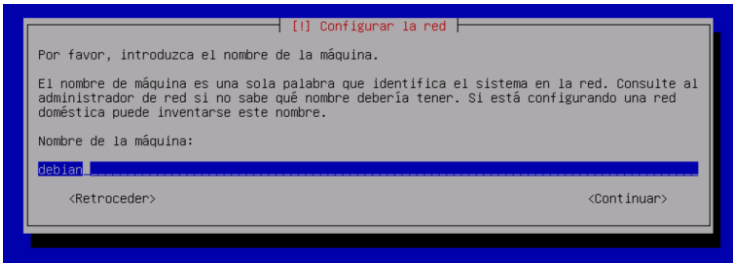


Figura A. 3 - Definición del nombre del sistema.

En la sección de “Particionado de discos” se debe elegir la primera opción que dice: utilizar todo el disco (Figura A. 4).

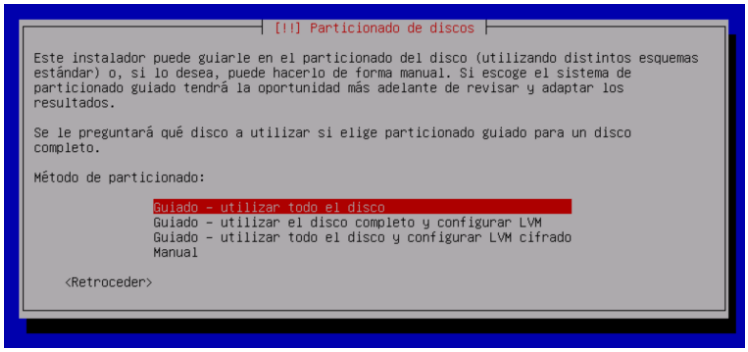


Figura A. 4 - Selección de todo el disco.

El siguiente paso es elegir el disco a particionar, en este caso se elige el único disco (SCSI3) que detecta el instalador el cual corresponde a un tamaño de 12.9 GB (VMWare) como se observa en la Figura A. 5.

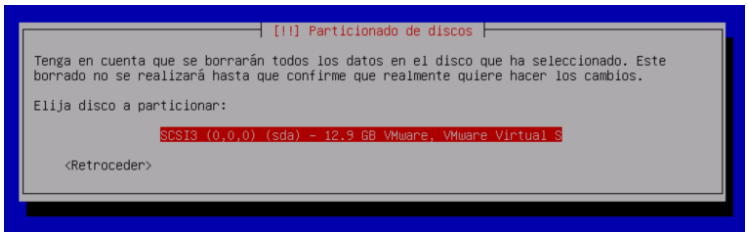


Figura A. 5 - Selección de disco a particionar.

A continuación se debe elegir la tercera opción la cual dice “Separar particiones /home, /usr, /var y /tmp”, como se muestra en la Figura A. 6.

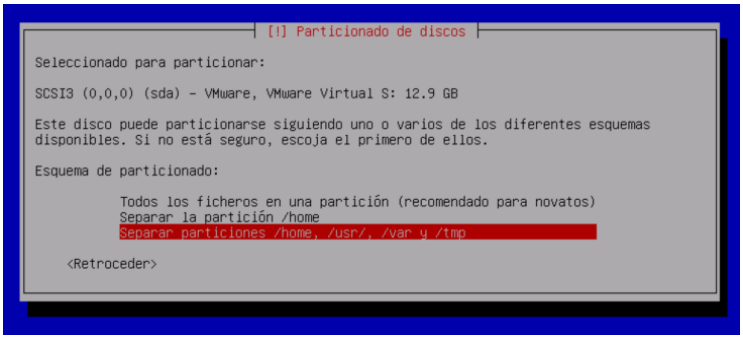


Figura A. 6 - Particiones separadas.

Debido a que el disco virtual es de 13 GB de capacidad, se definió un esquema de tamaño de particiones adecuado para la realización de este trabajo, el cual consiste en asignar la mayor cantidad de espacio a la partición *home* (4.7 GB), a la partición *usr* (4.5 GB) y *var* (2.2 GB). El resto de espacio se asigna a la partición raíz (*/*), *tmp* y *swap* (memoria de intercambio), al terminar la definición del tamaño de las particiones, se debe elegir “Finalizar el particionado y escribir los cambios en el disco” como se observa en la Figura A. 7.

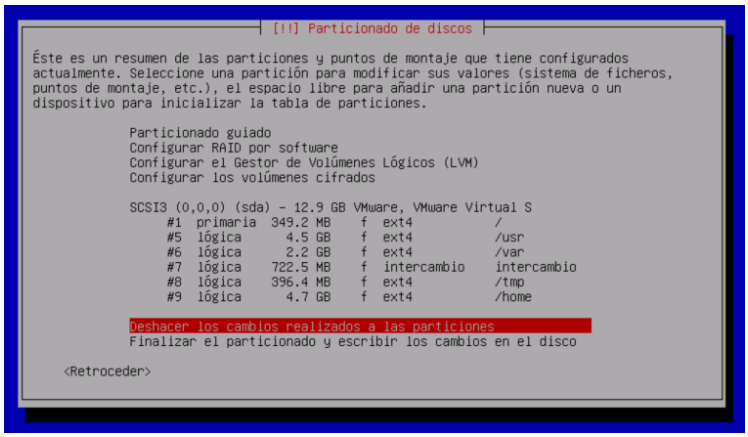


Figura A. 7 - Finalización de particionado de disco.

Los pasos adicionales que no se muestran en este trabajo se refieren a la selección de ubicación, configuración de teclado, establecimiento usuarios y contraseñas, configuración del gestor de paquetes y selección de programas (entorno gráfico, servidor Web, Servidor SSH, etcétera). Para este trabajo se eligió un sistema *Debian* con un entorno de escritorio XFCE, el cual es un entorno de escritorio ligero sin dejar de ser fácil de usar y amigable con el usuario, como se muestra en la Figura A. 8.

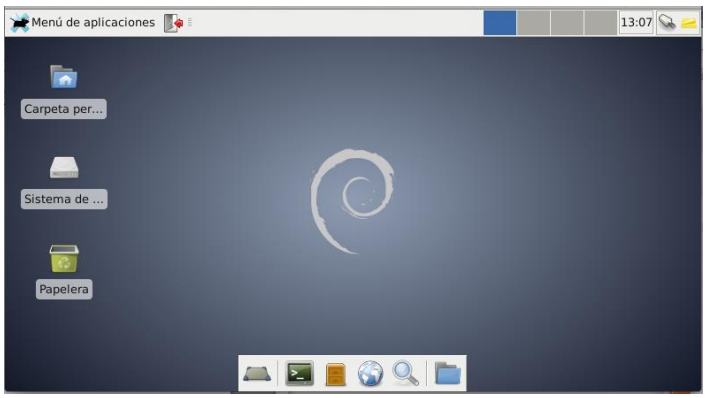


Figura A. 8 - Entorno virtual de XFCE.

Una vez iniciado el sistema con ayuda del comando `fdisk -l` se puede obtener información de las particiones del sistema, así como los bloques iniciales y finales de cada una de ellas (ver Figura A. 9).

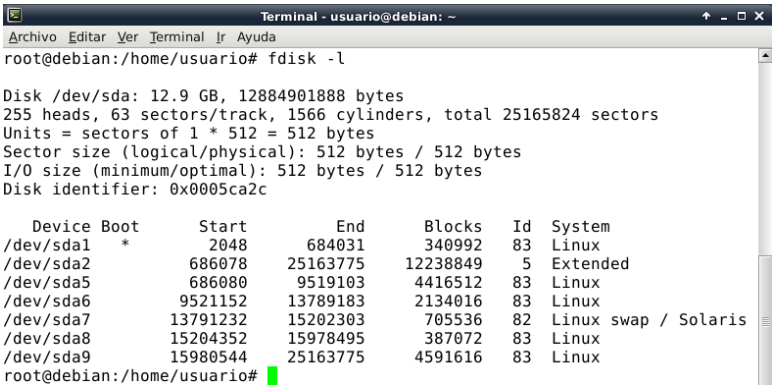


Figura A. 9 - Obtención de información del disco.

II. (5.1.5) Asegurar el dispositivo de almacenamiento

a) (5.1.5.1) Cifrado

Para que el directorio “/home/usuario” sea montado de forma automática cada vez que inicie sesión el usuario y que a su vez se realice la autenticación por medio de una “*passphrase”*, es necesario crear el directorio en “/mnt/usb” donde se montará automáticamente un dispositivo de tipo USB el cual contendrá un archivo con la *passphrase* (archivo_passwd.txt) que deberá validarse con el archivo “/root/.ecryptfsrc/sig-cache.txt” (ver Figura A. 10).

Figura A. 10 - Escenario para el “mount” automático con *passphrase*.

A continuación se debe crear el archivo “/root/.ecryptfsrc” (ver Figura A. 11).

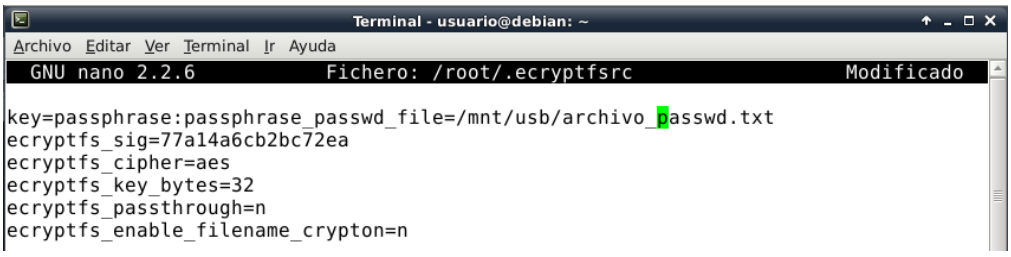


Figura A. 11 - Contenido del archivo “/root/.scryptfsrc”.

En el archivo “/mnt/usb/archivo_passwd.txt” se requiere colocar la *passphrase* en texto claro como se observa en la Figura A.12.

Figura A. 12 - Contenido del archivo "/mnt/usb/archivo_passwd.txt".

Posteriormente se debe editar el archivo "/etc/fstab" para indicarle al sistema que por defecto monte el dispositivo USB en "/mnt/usb" de forma automática y el directorio "/home/usuario" con el sistema de archivos *ecryptfs* en el propio directorio "/home/usuario" (ver Figura A. 13).

Figura A. 13 - Edición del archivo "/etc/fstab".

Es posible que después de reiniciar el sistema no se monte la partición cifrada, para evitar esta situación se debe editar el archivo "/etc/rc.local" y colocar antes de la línea que dice "exit 0" una línea con: "/bin/mount -a" para garantizar que el comando *mount* se ejecute siempre al inicio del sistema (ver Figura A. 14).

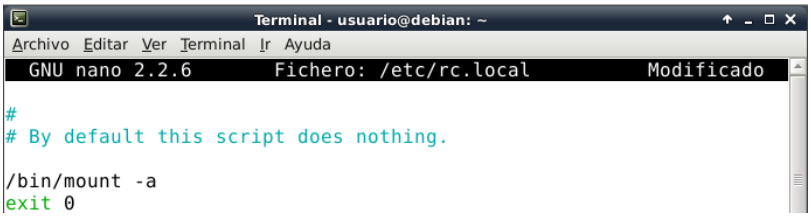


Figura A. 14 - Modificación de script de inicio "/etc/rc.local".

Para verificar que el montado se realice de forma automática se debe reiniciar el sistema y se revisar el registro de "/var/log/dmesg" el cual muestra que efectivamente se montó el dispositivo USB el cual a su vez permite el montado y descifrado del directorio "/home/usuario" y su contenido (ver Figura A. 15).

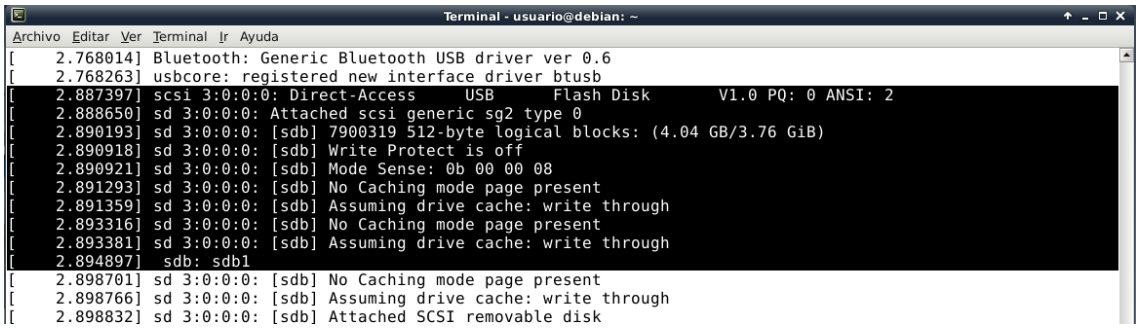


Figura A. 15 - Lectura del archivo "/var/log/dmesg".

Para finalizar, se muestra el contenido del archivo “/home/usuario/passwd” en texto claro (ver Figura A. 16).

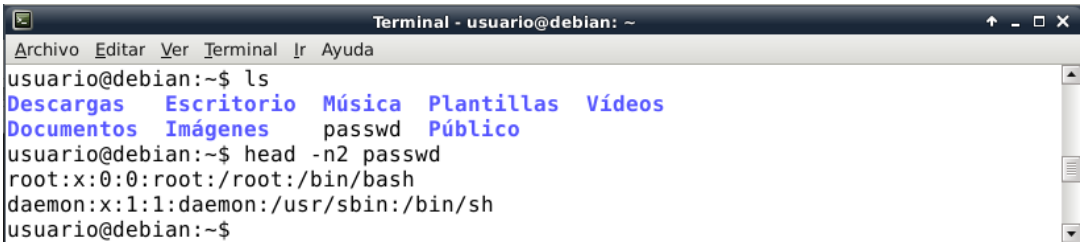


Figura A. 16 - Verificación de que el archivo passwd se localiza en el directorio.

Con la finalidad de demostrar que los archivos colocados en el directorio “/home/usuario” no pueden ser leídos a menos que se monten usando la *passphrase*, se utiliza un *Live-CD* de Backtrack versión 5, para montar la partición “/dev/sda9” referente al “/home” (ver Figura A. 17).

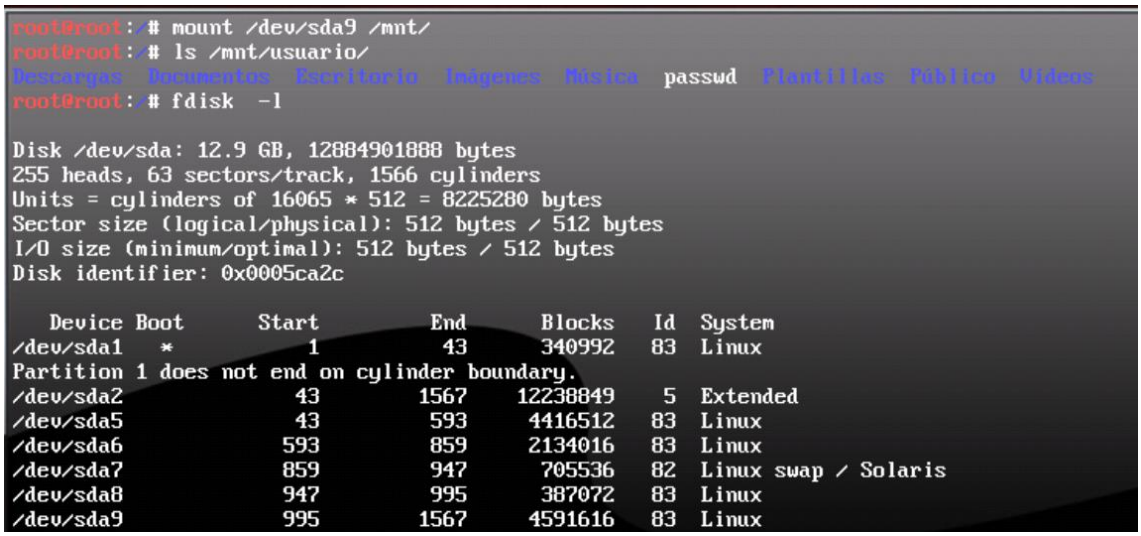


Figura A. 17 - Particiones reconocidas por Backtrack.

Al momento de intentar leer el contenido del archivo “/home/usuario/passwd” previamente montado en el directorio “/mnt” de *Backtrack* se puede observar que efectivamente aparece cifrado en el sistema, de esta forma se protege la confidencialidad del contenido del directorio “/home/usuario/” aun cuando el sistema sea montado con un Live-CD (ver Figura A. 18).

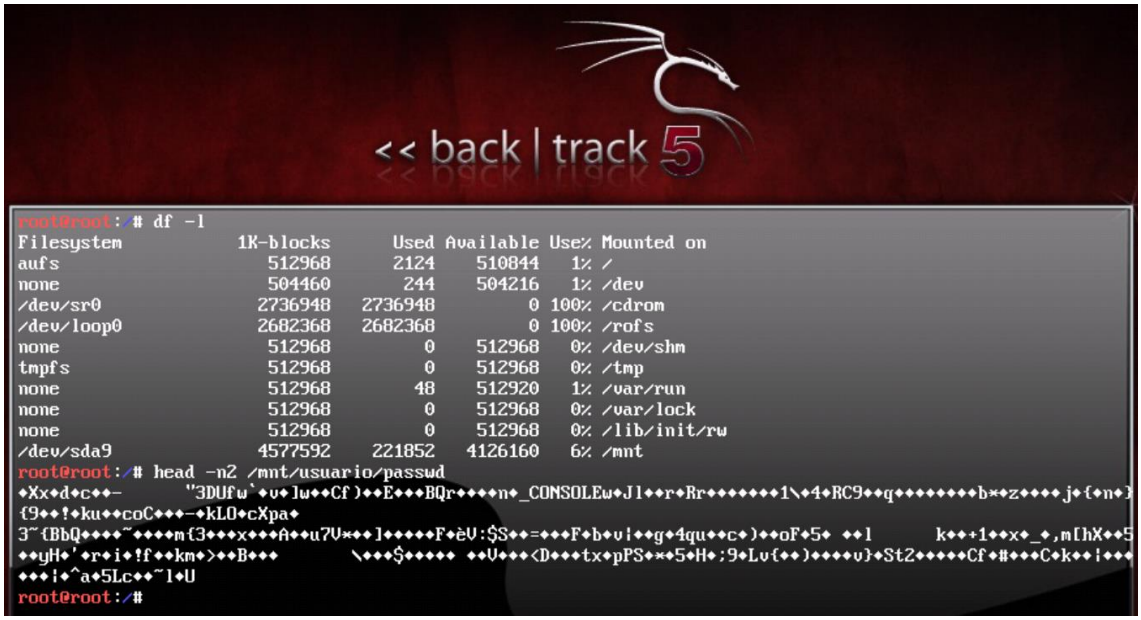


Figura A. 18 - Archivo "passwd" montado con Backtrack aparece cifrado.

b) (5.1.5.3) Uso de cuotas

Continuando con la implementación de cuotas en el sistema, se deben crear dos archivos en la partición que se deseen implementar las cuotas, estos dos archivos corresponden con las cuotas de usuario y grupo respectivamente (ver Figura A. 19).

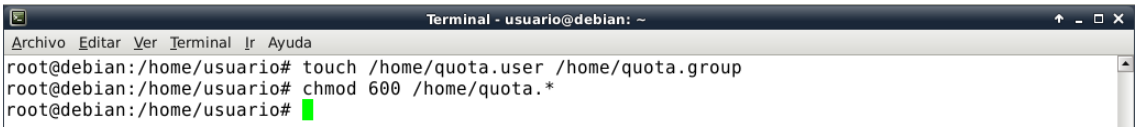


Figura A. 19 - Creación de archivos para cuotas de usuarios y grupos.

También se deben asignar permisos de lectura y escritura a los archivos creados anteriormente, posteriormente se debe remontar la partición de "home" para que se actualicen las opciones de cuotas en la partición /home, o bien reiniciar el sistema, la ejecución exitosa de los pasos anteriores se puede comprobar con el comando "mount", tal como se muestra en la Figura A. 20.

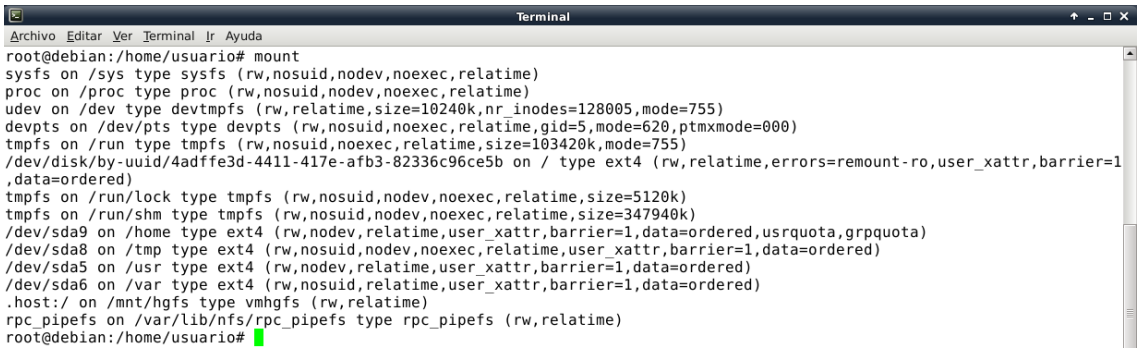


Figura A. 20 - Montado de "/home/" con cuotas de usuario y grupo.

Una vez que se verificó que las opciones de cuota se cargaron exitosamente para las particiones a las que se les configuro esa opción, se procede con la activación de las cuotas de disco recién configuradas, para ello se deben ejecutar el comando: `man quotacheck` para revisar los diversos formatos soportados para los archivos de cuotas, para este *hardening* se eligió el formato original de 16 bits (ver la Figura A. 21).

```
-F, --format=format-name
       Check  and fix quota files of specified format (ie. don't perform format auto-detec-
       tion). This is recommended as detection might  not  work  well  on  corrupted  quota
       files.   Possible  format names are: vfsold Original quota format with 16-bit UIDs /
       GIDs, vfsv0 Quota format with 32-bit UIDs / GIDs, 64-bit space usage,  32-bit  inode
       usage  and limits, vfsv1 Quota format with 64-bit quota limits and usage, rpc (quota
       over NFS), xfs (quota on XFS filesystem)
```

Figura A. 21 - Creación de archivos para cuotas de usuarios y grupos.

Posteriormente se debe ejecutar `quotacheck -F vfsv0 -avugm`, donde el parámetro “u” se refiere a la cuota del usuario, “g” a la cuota del grupo, “m” evita que se monte la partición como sólo lectura y “a” revisa que la partición soporte cuotas (ver Figura A. 22). Cabe destacar que la primera vez que se ejecuta el comando no identifica los archivos binarios “aquota.user” y “aquota.group” en los cuales se guarda la configuración de las cuotas.

```
root@debian:/home/usuario# quotacheck -F vfsv0 -avugm
quotacheck: Scanning /dev/sda9 [/home] done
quotacheck: Cannot stat old user quota file /home/aquota.user: No existe el fichero o el directorio.
Usage will not be substracted.
quotacheck: Cannot stat old group quota file /home/aquota.group: No existe el fichero o el directorio
. Usage will not be substracted.
quotacheck: Cannot stat old user quota file /home/aquota.user: No existe el fichero o el directorio.
Usage will not be substracted.
quotacheck: Cannot stat old group quota file /home/aquota.group: No existe el fichero o el directorio
. Usage will not be substracted.
quotacheck: Checked 78 directories and 88 files
quotacheck: Old file not found.
quotacheck: Old file not found.
root@debian:/home/usuario#
```

Figura A. 22 - Revisión de configuración de cuotas.

Al terminar la ejecución del comando anterior, en “/home” se generan dos archivos nuevos aquota.user y aquota.group como se observa en la Figura A. 23.

```
Terminal - usuario@debian: ~
Archivo  Editar  Ver  Terminal  Ir  Ayuda
root@debian:/home# ls -lcrt
total 36
drwx------  2 root    root    16384 jun 24 18:30 lost+found
drwxr-xr-x 18 usuario usuario  4096 jun 28 12:47 usuario
-rw-------  1 root    root        0 jul  9 20:23 quota.user
-rw-------  1 root    root        0 jul  9 20:23 quota.group
-rw-------  1 root    root     7168 jul  9 20:27 aquota.user
-rw-------  1 root    root     7168 jul  9 20:27 aquota.group
root@debian:/home#
```

Figura A. 23 - Creación de archivos para cuotas de usuarios y grupos.

A continuación se debe habilitar la cuota en el directorio “/home” como se muestra en la Figura A. 24.

```
root@debian:/# quotaon -avu
/dev/sda9 [/home]: user quotas turned on
root@debian:/#
```

Figura A. 24 - Comando para habilitar la cuota del usuario en "/home".

Una vez activadas las cuotas de usuario es necesario cambiar el nivel de ejecución para activar las cuotas duras y suaves, por lo general los sistemas Linux corren en el nivel de ejecución 2, para cambiar el nivel de ejecución a 3 se debe ejecutar el comando `init 3`. Para finalizar se deben asignar cuotas de disco a cualquier usuario o grupo con el comando `edquota -u usuario`, lo cual habilitará la consola mostrada en la Figura A. 25.

Figura A. 25 - Consola para la definición de cuotas para el disco.

Para este trabajo se especifica que la cuota sea de 512 MB (en bytes son 524288) para el limite duro y para el limite suave se definen 500000 bytes, como se muestra en la Figura A. 25. Para verificar que se asignó correctamente una cuota al usuario se ejecuta el comando mostrado en la Figura A. 26.

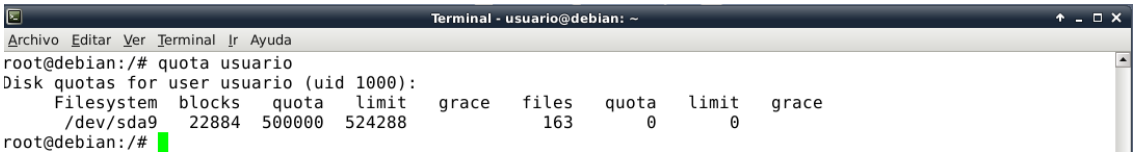


Figura A. 26 - Verificación de asignación de cuotas duras y suaves al usuario

Para configurar el tiempo de validez de la cuota, se usa el comando `edquota -t`, el cual habilita una consola que permite indicar el rango de tiempo máximo que los usuarios podrán seguir trabajado una vez que ha rebasado el límite de bloques o de ínodos definido acorde a la cuota establecida, este tiempo se puede especificar en días, horas e incluso minutos o segundos como se observa en la Figura A. 27.

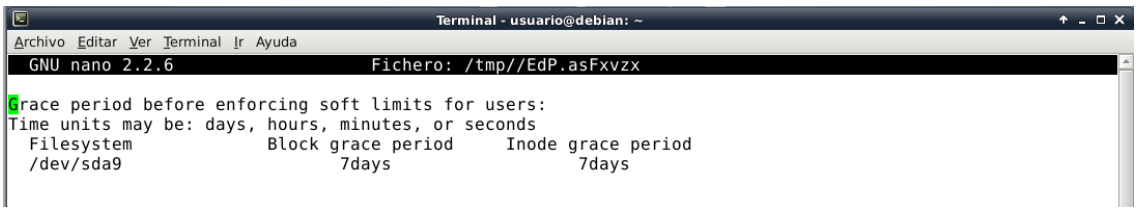


Figura A. 27 - Definición de parámetros de tiempo para las cuotas.

c) (5.1.5.4) Protección del kernel

Para prevenir que el sistema sea utilizado para reenviar paquetes de un tercero a nombre del equipo Debian, con la intención de realizar ataques de "hombre en el medio", se sebe desactivar el *forwarding* de los paquetes (ver Figura A. 28).

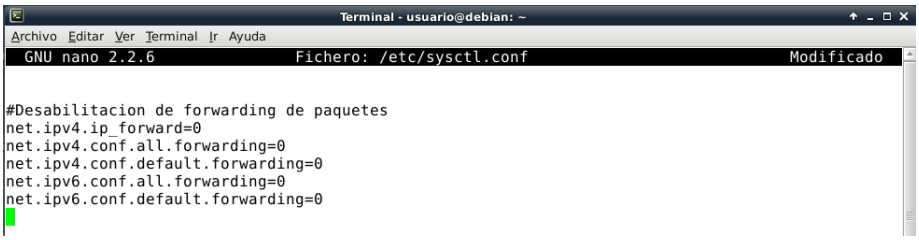


Figura A. 28 - Prevención de *forwarding*.

Para prevenir ataques de *IP Spoofing* y recepción de "paquetes marcianos (paquetes con la MAC del propio equipo)" se deben configurar los siguientes parámetros (ver Figura A. 29).

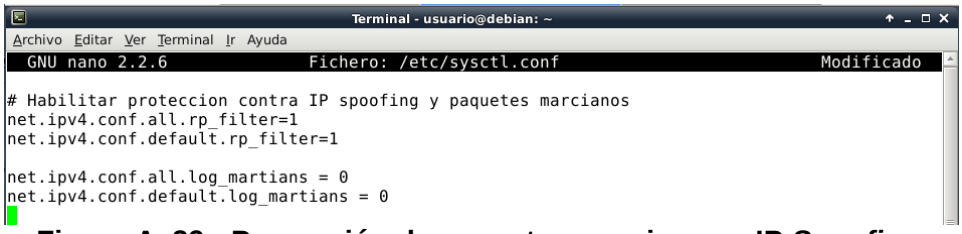


Figura A. 29 - Prevención de paquetes marcianos e IP *Spoofing*.

Para mitigar el impacto de ataques de Denegación de Servicio (DoS) es necesario definir el número máximo de *sockets* que el sistema puede sostener (ver Figura A. 30).

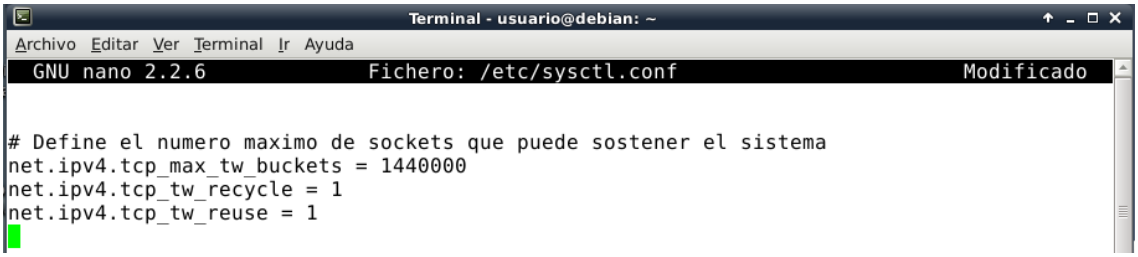


Figura A. 30 - Mitigación de impacto de ataques DoS.

Para mitigar el impacto de ataques de evasión de IDS y *Firewall* se define el mínimo (436 MB) y máximo (500 MB) de KB de memoria destinada para el re-ensamblado de paquetes, ver Figura A. 31.

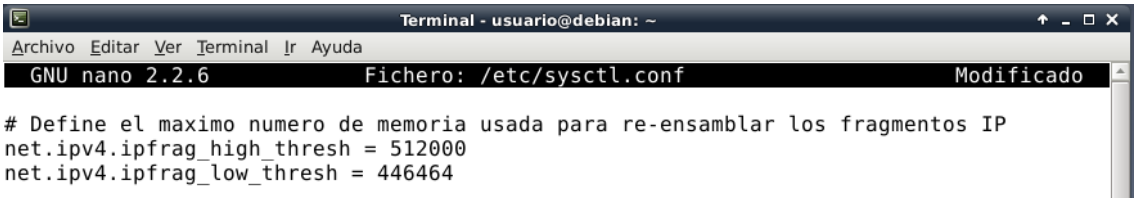


Figura A. 31 - Prevenir la evasión de IDS y *Firewalls*.

Una vez configurado el archivo sysctl.conf es necesario ejecutar `sysctl -p` para que el *kernel* cargue los nuevos parámetros (ver Figura A. 32).

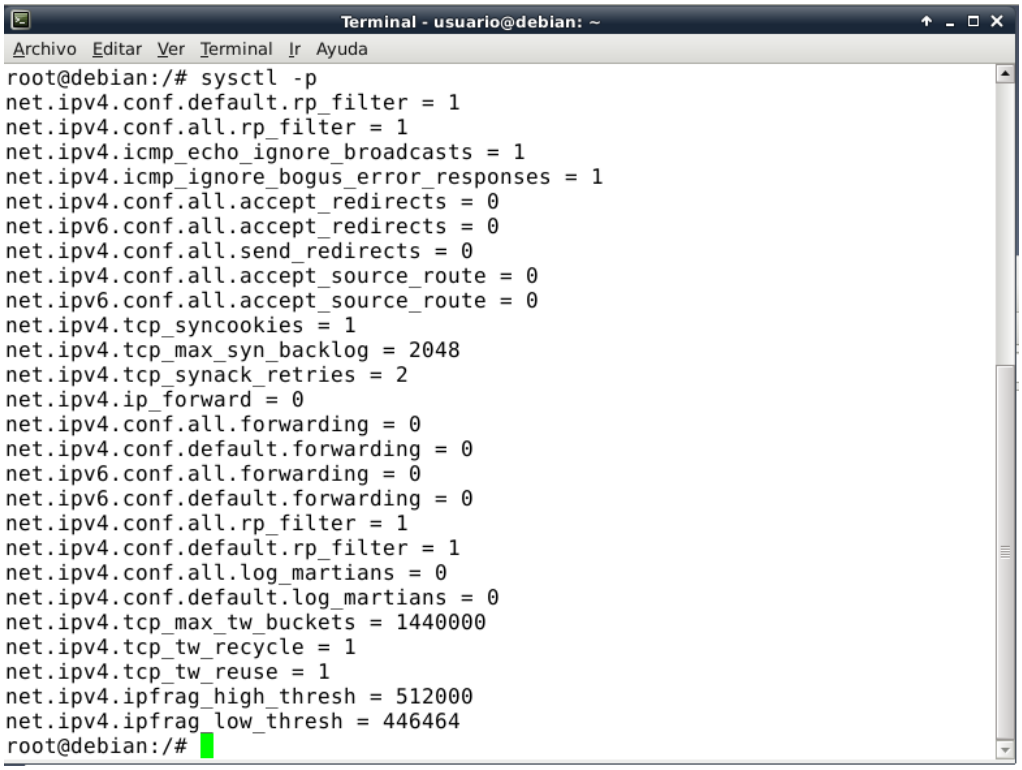


Figura A. 32 - Carga de parámetros de seguridad en el kernel.

d) (5.1.5.5) Parche de seguridad para el kernel

Una vez instalado el *kernel* de seguridad *Grsecurity* se debe proceder con la configuración para blindar al *kernel*, para lo cual es necesario reiniciar el sistema y en el menú de *GRUB* elegir el *kernel* de "Linux 3.2.15grsec2.2.0" para que arranque el sistema y cargue su configuración como se muestra en la Figura A. 33.

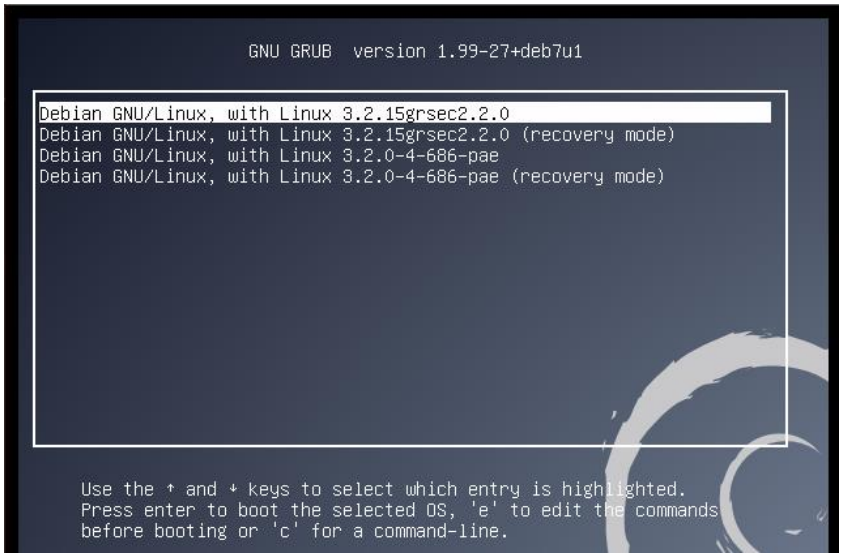


Figura A. 33 - Inicio del sistema con kernel *Grsecurity*.

Una vez que se ha instalado el parche de *Grsecurity* es posible instalar la herramienta *gradm* la cual permite al administrador del sistema habilitar políticas de seguridad y RBACs (*Role Based Access Control* – Controles de Acceso Basados en el Rol) para cada usuario, por defecto Linux implementa DAC (*Discretionary Access Control* –

Control de Acceso Discrecional) mediante la creación de un archivo qué define quien puede acceder a que archivo y quién no.

El primer paso es instalar las librerías necesarias: `apt-get install bison flex`, posteriormente dentro del directorio "/usr/src/" descargar con *wget* el paquete gradm-2.2.0-201011061849.tar.gz como se puede ver en la Figura A. 34.

```
root@debian:/usr/src# wget http://dev.gentoo.org/~blueness/hardened-sources/gradm/gradm-2.2.0-201011061849.tar.gz
--2013-09-09 18:54:53--  http://dev.gentoo.org/~blueness/hardened-sources/gradm/gradm-2.2.0-201011061849.tar.gz
Conectando con 10.241.208.243:8080... conectado.
Petición Proxy enviada, esperando respuesta... 200 OK
Longitud: 69493 (68K) [application/x-gzip]
Grabando a: "gradm-2.2.0-201011061849.tar.gz"

100%[======================================================================================>] 69 493      --.-K/s   en 0.003s

2013-09-09 18:54:53 (24.4 MB/s) - "gradm-2.2.0-201011061849.tar.gz" guardado [69493/69493]

root@debian:/usr/src#
```

Figura A. 34 - Instalación de librerías necesarias.

Posteriormente se debe instalar el paquete *gradm* con los siguiente comandos: `make` y `make install` (ver la Figura A. 35).

```
root@debian:/usr/src/gradm2# make install
Installing gradm...
Installing grlearn...
Installing gradm manpage...
root@debian:/usr/src/gradm2#
```

Figura A. 35 - Instalación de paquete gradm2.

El siguiente paso es definir una contraseña de administración para *gradm*, con el comando gradm –P (ver Figura A. 36).

```
root@debian:/usr/src# gradm -P
Setting up grsecurity RBAC password
Password:
Re-enter Password:
Password written to /etc/grsec/pw.
root@debian:/usr/src#
```

Figura A. 36 - Definición de contraseña para gradm.

Una de las características más potentes de esta herramienta (gradm) es que puede ser configurada en modo "auto aprendizaje" en el cual por medio del uso diario del sistema en relación a las actividades cotidianas que un usuario ejecuta a través del propio sistema, la herramienta *gradm* genera una propia política con los permisos necesarios para que el usuario desempeñe su actividades. Para lo cual se debe ejecutar el comando: `gradm -F -L /etc/grsec/learning.log`. Después que el usuario ha desempeñado las actividades habituales en el sistema en un tiempo suficientemente razonable para que la herramienta pueda "auto aprender" cuáles son los usos específicos y tareas para cada usuario, se debe proceder a generar las políticas que serán empleadas para tener una lista de control de acceso exitosa (ver Figura A. 37).

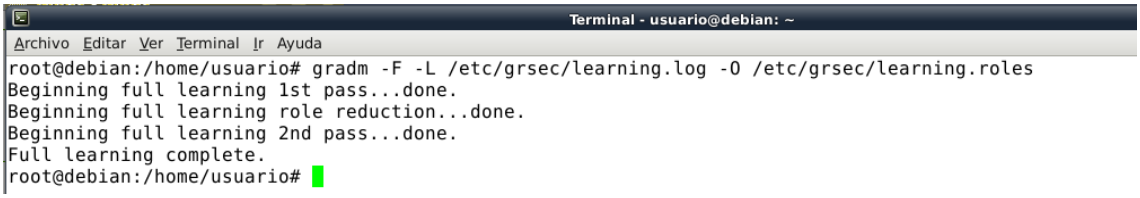


Figura A. 37 - Carga de políticas definidas para *Grsecurity*.

Una vez que se han generado las políticas y definidas en el archivo *learning.roles* es necesario intercambiarlas por las políticas establecidas por defecto en el sistema, para ello primero se debe realizar un respaldo de las mismas y después proceder a realizar la sobre escritura, previo a los pasos descritos es necesario deshabilitar *gradm* mediante el comando `gradm -D`, una vez terminada la ejecución de los comandos antes descritos se procede a habilitar de nuevo las *RBACs* con el comando `gradm -E`, como se muestra en la Figura A. 38.

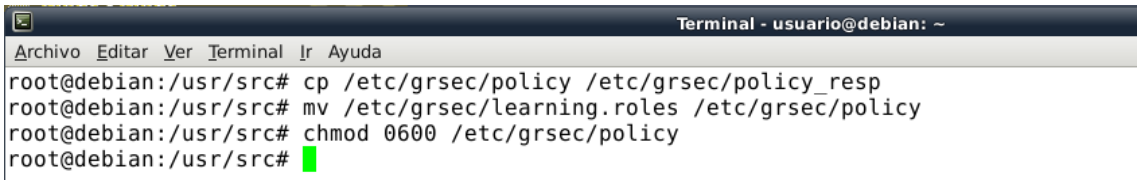


Figura A. 38 - Reemplazo de políticas por defecto para *Grsecurity*.

Una de las ventajas de parchar el *kernel* del Linux con *Grsecurity* es poder implementar protección y características de auditoría sobre el propio kernel. Para ello se requiere agregar las siguientes líneas al archivo "/etc/sysctl.conf" (ver Figura A. 39) y posteriormente se habilita el empleo de tales acciones mediante el comando sysctl como se puede ver en la Figura A. 40.

```
#### Hardening del comando chroot, estas caracteristicas harán que salir de entornos
#### chroot sea más complicado
kernel.grsecurity.chroot_deny_sysctl=1
kernel.grsecurity.chroot_caps=1
kernel.grsecurity.chroot_execlog=0
kernel.grsecurity.chroot_restrict_nice=1
kernel.grsecurity.chroot_deny_mknod=1
kernel.grsecurity.chroot_deny_chmod=1
kernel.grsecurity.chroot_enforce_chdir=1
kernel.grsecurity.chroot_deny_pivot=1
kernel.grsecurity.chroot_deny_chroot=1
kernel.grsecurity.chroot_deny_fchdir=1
kernel.grsecurity.chroot_deny_mount=1
kernel.grsecurity.chroot_deny_unix=1
kernel.grsecurity.chroot_deny_shmat=1

#### Caracteristicas de auditoria en el logging
kernel.grsecurity.audit_mount=1
kernel.grsecurity.forkfail_logging=1
kernel.grsecurity.signal_logging=1
kernel.grsecurity.timechange_logging=1

#### Prohibición de acceso al DMESG
kernel.grsecurity.dmesg=1

#### Limita las llamadas a la función execve, se mitiga el riesgo
#### de ataques de buffer overflow en aplicaciones del sistema.
kernel.grsecurity.execve_limiting=1

#### Protección contra Exploits, mediante la aplicación de aleatoriedad a partes especifícas
#### de programas obligando a la investigación de de direcciones de memoria, cualquier intento fallido provocará
#### que el kernel detenga las tareas e implemente protecciones al respecto
kernel.grsecurity.rand_pids=1
kernel.grsecurity.rand_tcp_src_ports=1

#### Protección contra Exploits que utilizan enlaces simbólicos o FIFOS que sean
#### propiedad de otros usuarios en directorios de lectura
kernel.grsecurity.fifo_restrictions=1
kernel.grsecurity.linking_restrictions=1
```

Figura A. 39 - Protección y características de auditoría al kernel.

```
root@debian:/etc# nano sysctl.conf
root@debian:/etc# sysctl -w kernel.grsecurity.exec_logging=1
kernel.grsecurity.exec_logging = 1
root@debian:/etc#
```

Figura A. 40 - Implementación de características del kernel.

Las características de protección y auditoría que se implementaron en el archivo "/etc/sysctl.conf" fueron las siguientes:

- *Hardening* del comando *chroot*, estas características harán que salir de entornos *chroot* sea más complicado, en esencia cualquier proceso con intenciones de ejecutarse fuera del entorno *chroot* será bloqueado.
- Características de auditoría en el *login,* cualquier ejecución de los comandos `mount` y `umount` será registrado, inicios de sesión fallidos, si la ejecución de un programa genera algún error este será registrado pues existe la posibilidad de que se trate de un *Exploit*.
- Prohibición de acceso al archivo *DMESG*
- Limita las llamadas a la función *execve*, se mitiga el riesgo de ataques de *buffer overflow* en aplicaciones del sistema.
- Protección contra *Exploits*, mediante la aplicación de aleatoriedad a partes específicas de programas obligando al intruso realizar una investigación exhaustiva de direcciones de memoria puntuales, cualquier intento fallido provocará que el *kernel* detenga las tareas e implemente protecciones al respecto.
- Protección contra *Exploits* que utilizan enlaces simbólicos que sean propiedad de otros usuarios en directorios de lectura "+t" a menos que el propietario del enlace simbólico sea el propietario del directorio, además los usuarios no podrán escribir en los FIFOS que no posean, sólo en directorios de tipo "+t" como "/tmp".

Para forzar que la aplicación de las características antes mencionadas sea hasta el reinicio del sistema, es necesario ejecutar el siguiente comando:

```
sysctl -w kernel.grsecurity.grsec_lock = 1
```

III. (5.2.1) Servicios y conexiones de red innecesarios

Antes de iniciar la desactivación de servicios innecesarios del sistema, es necesario analizar las características y funciones principales de cada uno, para ello se debe revisar el contenido del directorio "*/etc/init.d*" como se muestra en la Figura A. 41, la cual muestra los servicios en ejecución del sistema.

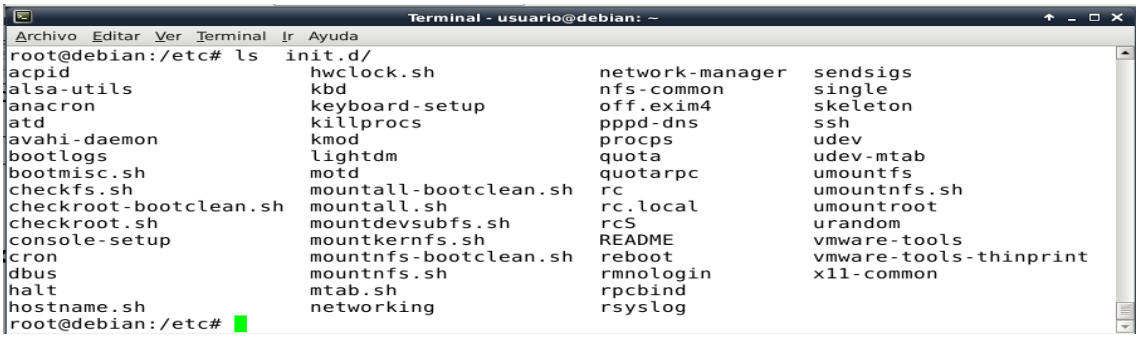


Figura A. 41 - Revisión de servicios del sistema

En la Tabla A. 1 se muestra la descripción de los servicios y a su vez cuales son candidatos para ser deshabilitados en el sistema.

Tabla A. 1 - Servicios del sistema candidatos a ser desactivados.

Servicio	Descripción del servicio
anacron	Es un programador de tareas periódico. A diferencia de *cron*, no asume que el sistema está encendido siempre. Por lo tanto, se puede usar para controlar la ejecución de tareas diarias, semanales o mensuales (o de cualquier período de "n" días), su ventaja es que aplica para sistemas que no están encendidos las 24 horas al día.
avahi	Permite detectar automáticamente los recursos de una red local y conectarse a ella, esto facilita el uso compartido de archivos, impresoras, etc.
atd	Es el demonio responsable de ejecutar las tareas programadas desde "at".
cups	Funciones de impresión.
Nfs-common	Servicios NFS.
portmap	Soporte de conexión RPC.

Para deshabilitar un demonio (o servicio) se pueden implementar algunos de los siguientes métodos:

- Eliminar las conexiones de /etc/rc${runlevel}.d/ o renombrar las conexiones (que no empiecen con 'S') sustituyendo "S" por "K".
- Mover el archivo de escritura (/etc/init.d/_nombre_del_servicio_) a otro nombre (por ejemplo /etc/init.d/OFF.nombre_nombre_del_servicio_)

Además de lo anterior, se puede inhabilitar un servicio de ejecución en todos los niveles del sistema, lo anterior se puede realizar con el comando: `update-rc.d`. En la Figura A. 42, se muestra la forma de desactivar el servicio "nfs-common" empleando el comando: `update-rc.d -f <servicio> remove`.

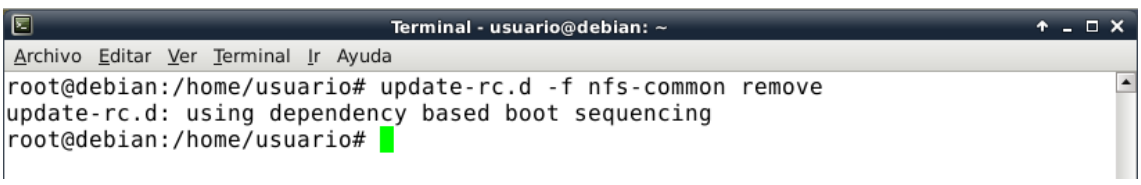


Figura A. 42 - Desactivación de servicios con update-rc.d.

Para eliminar los iconos en el “Escritorio” de Debian y disminuir la posibilidad de divulgación de información por dejar el equipo desatendido es necesario seguir la ruta: “Menú de aplicaciones” -> “Configuración” -> “Escritorio” y deshabilitar todos los iconos seleccionados (ver Figura A. 43) para que se remuevan de forma automática del “Escritorio”.

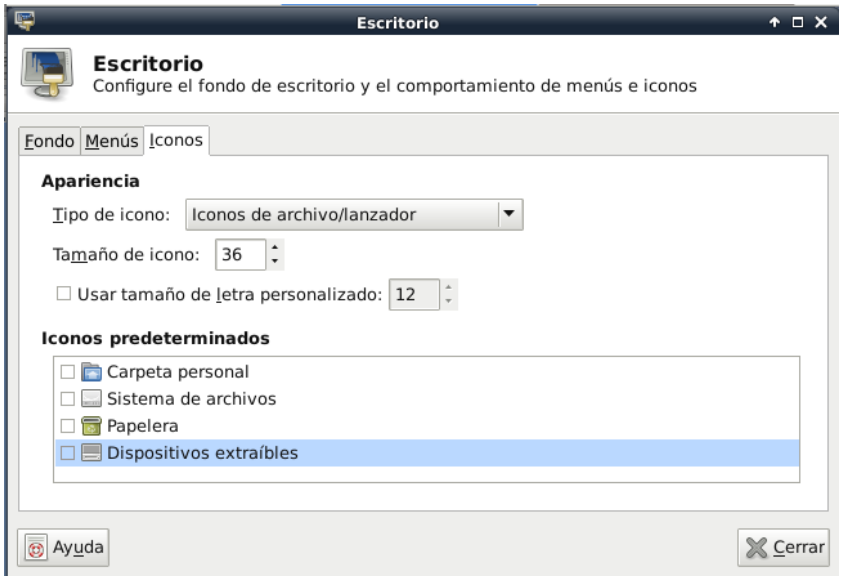


Figura A. 43 - Configuración de escritorio limpio.

Por último para impedir que los usuarios escriban sobre el “Escritorio” se deben cambiar los permisos de mismo, dejando únicamente permisos de lectura de forma recursiva. Esto se realiza dar clic derecho sobre el “Escritorio” -> Propiedades -> Permisos -> Acceso -> Sólo lectura-> Aplicar recursivamente (ver Figura A. 44).

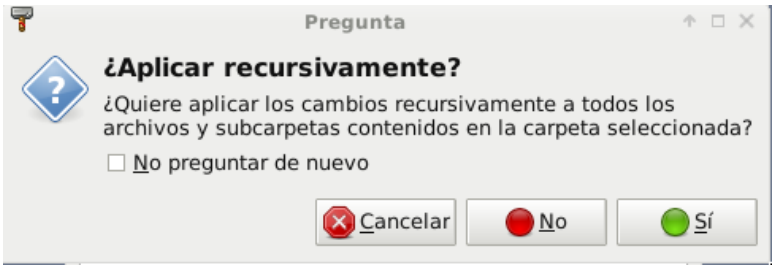


Figura A. 44 - Permisos recursivos de sólo lectura al escritorio.

Para la eliminación de medios e información críticos a “bajo nivel” es necesario instalar la herramienta *secure-delete* (ver Figura A. 45): `aptitude install secure-delete`. De esta forma si en algún momento se requiere eliminar archivos de forma segura del sistema y evitar el éxito de herramientas forenses el comando a ejecutar es: `srm archivo_a_eliminar`.

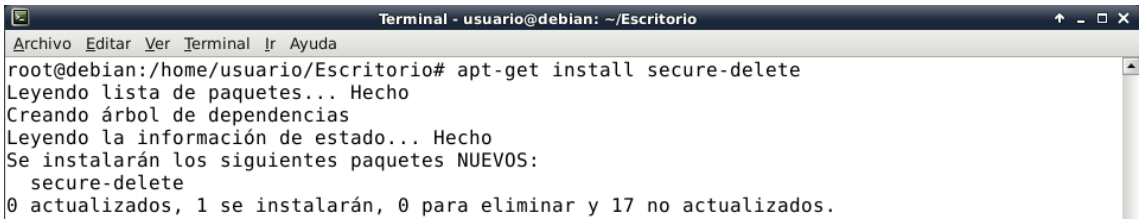


Figura A. 45 - Instalación de secure-delete.

IV. (5.2.3) Otros paquetes no utilizados

Para sistemas Debian que no requieran el uso de impresora se puede deshabilitar el servicio de *CUPS* (*Common Unix Printing System* - Sistema de Impresión Común de Unix), el servicio de *PORTMAP* es utilizado para la asignación dinámica de puertos de servicios *RPC* tales como *NIS y NFS*, y el *EXIM4* además se emplea para el envío de correos a través del sistema. Cabe señalar que dado que se instaló una distribución de Debian con un "ligero" *hardening*, el servicio de *PORTMAP* y *CUPS* no están instalados por defecto, no así el servicio *EXIM4* en caso de que versiones de sistema operativos más recientes a la utilizada en este trabajo de tesis, las cuales si incorporen estas aplicaciones desde la instalación, el proceso de inhabilitación es el mismo (ver Figura A. 46).

```
root@debian:/home/usuario# mv /etc/init.d/exim4 /etc/init.d/off.exim4
root@debian:/home/usuario# ls -l /etc/init.d/off*
-rwxr-xr-x 1 root root 6435 ene  2  2013 /etc/init.d/off.exim4
root@debian:/home/usuario#
```

Figura A. 46 - Deshabituación de servicios mediante el renombramiento.

Una alternativa gráfica para realizar la desactivación de servicios, es mediante la aplicación *sysv-rc-conf*, la cual despliega los niveles de ejecución y los servicios que inician en cada uno de ellos, además permite seleccionar (tecla +) y deseleccionar (tecla -) los servicios que se desean activar y desactivar respectivamente, como se observa en la Figura A. 47.

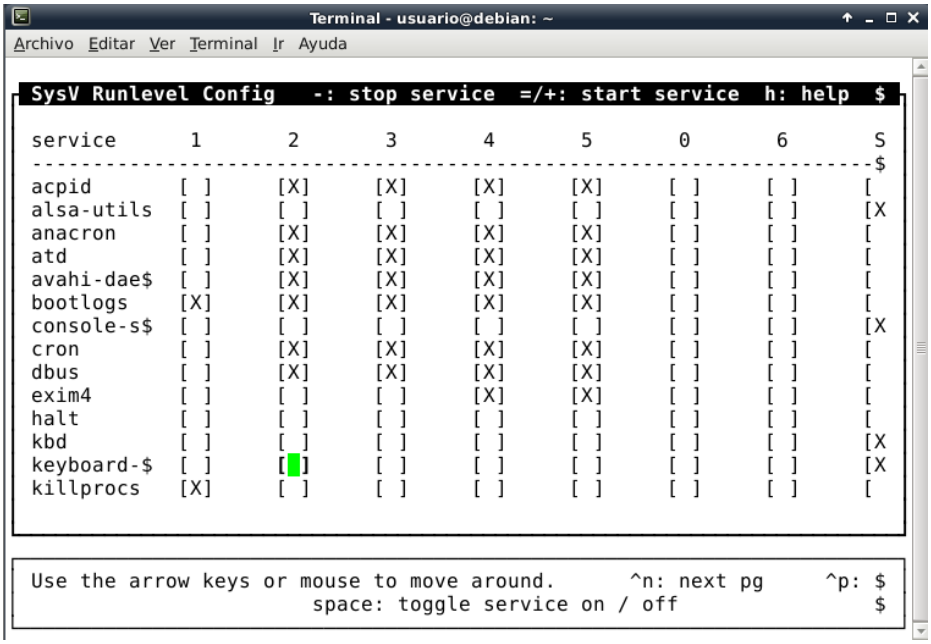


Figura A. 47 - Interface gráfica de la herramienta sysv-rc-conf.

Además para forzar el ingreso de la contraseña de *root* en cada cambio del nivel de ejecución se debe editar el archivo *inittab* agregando la línea mostrada en la Figura A. 48.

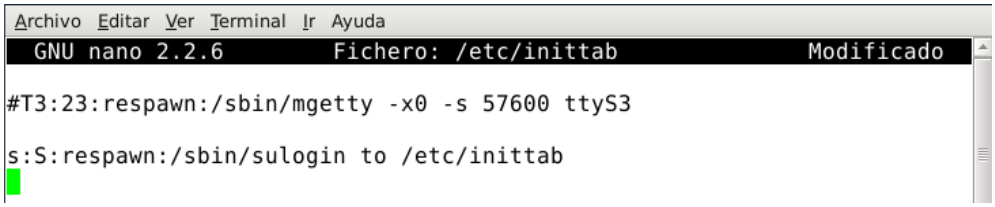


Figura A. 48 - Contraseña de root para acciones de cambio de nivel de ejecución.

Debido a que los puertos son utilizados por los servicios y otras aplicaciones para que dos sistemas puedan comunicarse a través de una red de datos. Los intrusos comúnmente suelen aprovechar estos puertos para ingresar de forma no autorizada a los sistemas o bien para recolectar información útil que les permita tener mayor probabilidad de éxito en las actividades de identificación y explotación de vulnerabilidades. Se deben dejar abiertos únicamente los puertos necesarios y cerrar el resto de los puertos, pues como previamente se explicó representan puertas en el sistema que un intruso pudiera aprovechar.

El primer paso es identificar los puertos abiertos del sistema, así como los servicios asociados a los mismos, también se deben identificar las conexiones de red activas, lo cual se puede realizar con el comando `netstat -tulp`, el cual muestra la relación entre los servicios y puertos de red del sistema, el parámetro "t" es para indicar los puertos que usan el protocolo *TCP*, el parámetro "u" para puertos que emplean *UDP*, "l" para mostrar los sockets que están escuchando y por último "p" para mostrar los PID (Identificadores de Proceso) ver Figura A. 49.

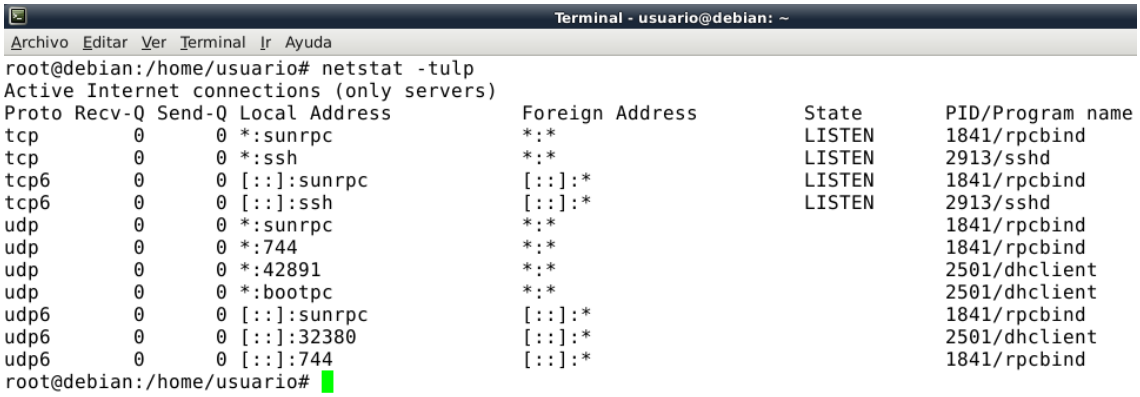


Figura A. 49 - Identificación de puertos TCP/UDP abiertos.

En la Figura A. 50, se puede identificar que los servicios *portmap, exim4* y *cups* no están presentes en el sistema producto de las acciones de *hardening* implementadas en el punto anterior. Además de la herramienta *deborphan* (previamente vista) para sistemas Debian, el comando: `dpkg -l` puede servir para visualizar los paquetes instalados en el sistema y ayudar a vislumbrar fácilmente cuales no están siendo utilizados, a continuación se describen algunas consideraciones para identificar este tipo de paquetes que no se están utilizando en el sistema.

- Si solo se emplea *Ethernet*, para la conexión a la red, se tendrían que eliminar los paquetes relacionados con *PPP3* (ppp, pppconfig, pppoe, pppoeconf), ya que estos paquetes sólo se emplean en conexiones de tipo *Wireless* (ver Figura A. 50).

- Los paquetes para desarrollo como los compiladores o interpretes (*perl, gcc*) y las bibliotecas con terminación en “-dev” por lo general son utilizadas para equipos de desarrollo de aplicaciones, dependiendo del rol que vaya a tener el equipo, estos paquetes se podrían dejar en el sistema o desinstalarlos.
- Los paquetes relacionados con dispositivos que no se tienen o no serán utilizados (*usbutils, eject, setserial, fdutils*) son buenos candidatos para su desinstalación.
- Programas inseguros para el traslado de información pueden ser desinstalados (por ejemplo; telnet y ftp).

Es importante evitar el borrado de paquetes de tipo *essential*. Estos paquetes como su nombre lo indica, son por lo general necesarios para el correcto funcionamiento del sistema operativo.

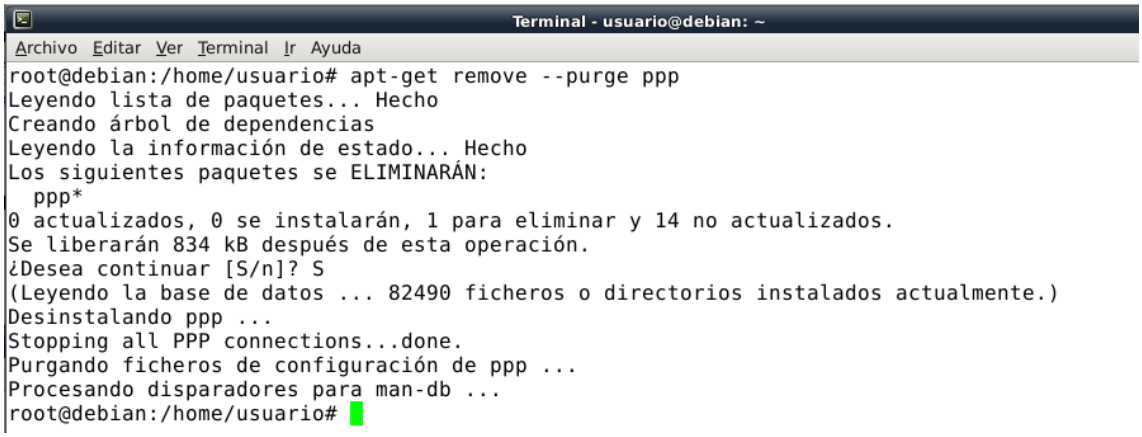


Figura A. 50 Herramienta para detectar y remover paquetes no utilizados.

Para automatizar la búsqueda de servicios candidatos a ser desinstalados del sistema se desarrolló un *script* que permitirá conocer los binarios que están en estado LISTEN (en espera de ser utilizados) que a su vez hagan referencia a que el binario está alojado en el directorio “/bin/” (sólo para usuarios normales) y por último que el paquete se encuentre en el directorio “/etc/init.d”. A continuación se muestra el código del script desarrollado.

```
#!/bin/sh
     for i in `lsof -i | grep LISTEN | cut -f1 -d" " | sort -u`; do
      paquete=`dpkg -S $i | grep bin | cut -f 1 -d ":" | uniq`
      echo "El servicio $i está instalado por: $paquete";
      init=`dpkg -L $paquete |grep init.d/`
      if [ ! -z "$init" ]; then
      echo "y está corriendo en el sistema: $init"
      fi
     done
```

La Figura A. 51 muestra el resultado de la ejecución del *script* anterior.

Figura A. 51 Ejecución de *script* para detección de paquetes no usados.

Se observa que el sistema se encuentra con un nivel adecuado de *hardening* puesto que los únicos servicios que podrían ser candidatos a desinstalarse son SSH(necesario para las actividades) y RPCBIND; sin embargo, ambos servicios son útiles y no representan huecos de seguridad en el sistemas, por lo que no se deshabilitarán.

Existen paquetes como FTP y TELNET que envían la información en claro (ver Figura A. 52), incluidas las credenciales de autenticación al servidor, es por ello que serán desinstalados del sistema empleando el siguiente comando: `apt-get remove --purge servicio` (ver Figura A. 53). De esta manera, se eliminarán tanto los paquetes como sus archivos de configuración, puesto que se emplea la opción "--purge".

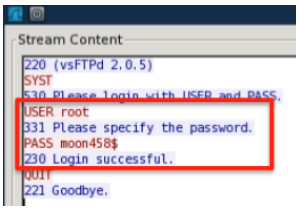


Figura A. 52 - Transferencia de datos en texto claro.

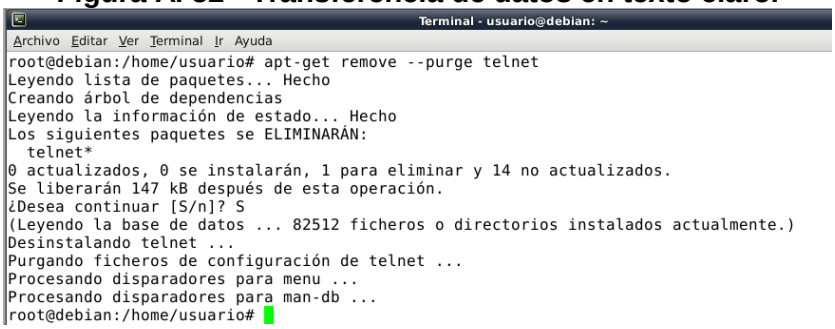


Figura A. 53 - Desinstalación de aplicaciones vulnerables.

V. (5.3.2) Archivo passwd, group y shadow

Si se crea un nuevo usuario en el sistema con el comando `adduser nombre_usuario` el sistema solicita información referente al usuario como se muestra en la Figura A. 54.

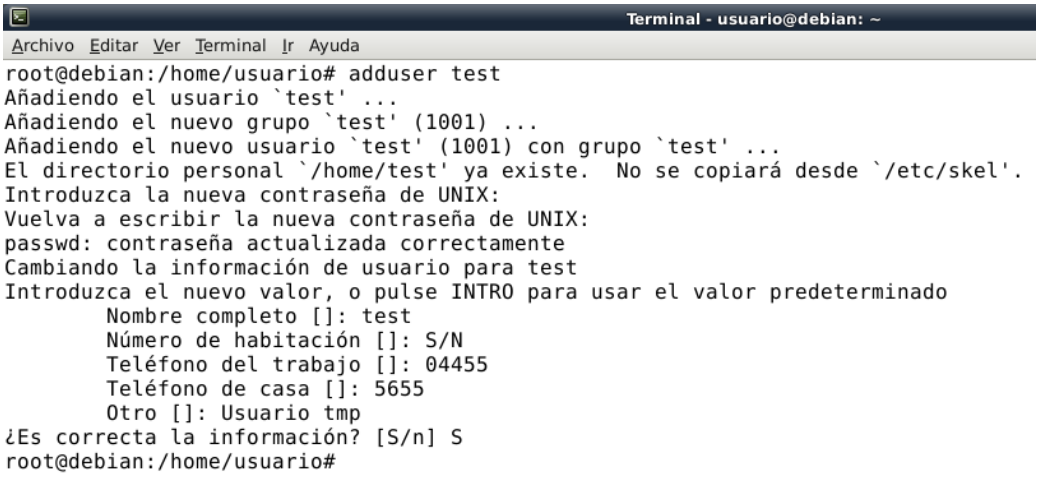


Figura A. 54 - Ejemplo de cómo se agrega un usuario y la información requerida.

El archivo “/etc/passwd” almacena información de los usuarios creados en el sistema y tiene el siguiente formato:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
…
test:x:1001:1001:test,S/N,04455,5655,Usuario tmp:/home/test:/bin/bash
```

En el archivo “passwd” (ver Figura A. 55) cada línea está separada en campos, los dos puntos “:” actúan como separador o limitador de los campos, el significado de cada campo es el siguiente:

- Login: el nombre del usuario, el cual no se puede repetir.
- Contraseña cifrada: se indica una referencia al propio archivo.
- User ID: el número de identificación del usuario.
- Group ID: el número de grupo al cual pertenece el usuario.
- Comentarios: campo reservado para introducir los comentarios que se deseen sobre el usuario.
- Directorio home: el directorio home (de trabajo) del usuario es donde éste podrá guardar sus archivos, generalmente se encuentran dentro del directorio /home/nombre_usuario y el nombre de cada directorio en este nivel es similar al del usuario proporcionado con el comando `passwd`.
- Intérprete de comandos: el intérprete de comandos (*Shell*) es un programa que se encarga de leer todo lo que se escribe en el teclado a través de una terminal y que se desea sea ejecutado por el propio sistema.

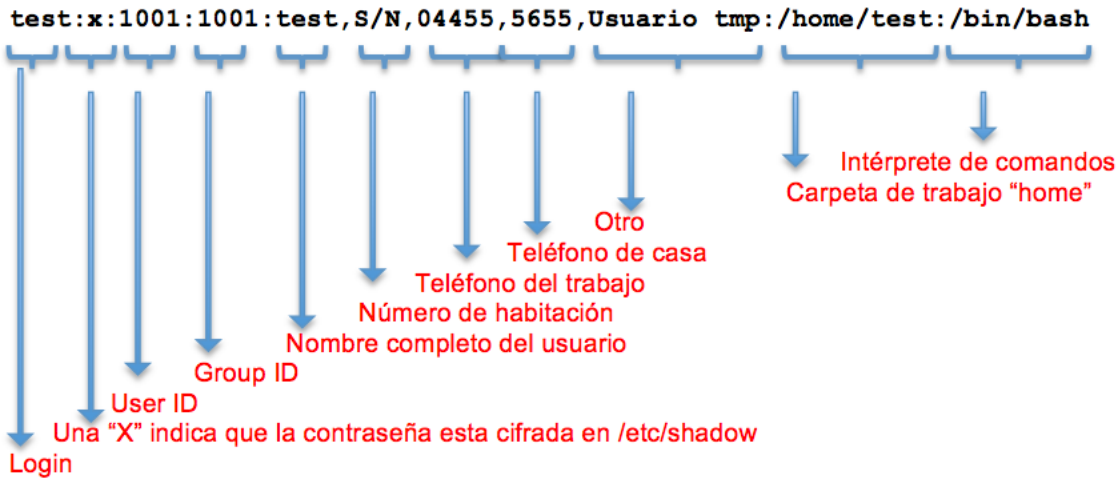


Figura A. 55 - Significado de los campos del archivo “/etc/passwd”.

El archivo “group” almacena la información de los grupos del sistema y tiene el siguiente formato:

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
…
scanner:x:103:usuario
```

```
messagebus:x:104:
colord:x:105:
netdev:x:106:usuario
utempter:x:107:
```

Al igual que el archivo "/etc/passwd" en el archivo "/etc/group" (ver Figura A. 56) cada línea está separada en campos, el separador de campos son los dos puntos ":" y cada campo representa lo siguiente:

- Nombre del grupo. Por defecto, con los comandos habituales se crea un grupo con el mismo nombre que el usuario que invoca dicha instrucción, aunque pueden existir otros grupos con nombres específicos.
- Contraseña cifrada: referencia a la contraseña.
- Group ID: número de identificación del grupo.
- Lista de usuarios: los nombres de los usuarios que pertenecen al grupo, separados por comas. Aunque todos los usuarios deben pertenecer a un determinado grupo, este campo se puede utilizar para que usuarios de otros grupos también dispongan de los mismos permisos que tiene el grupo al que se está referenciando.

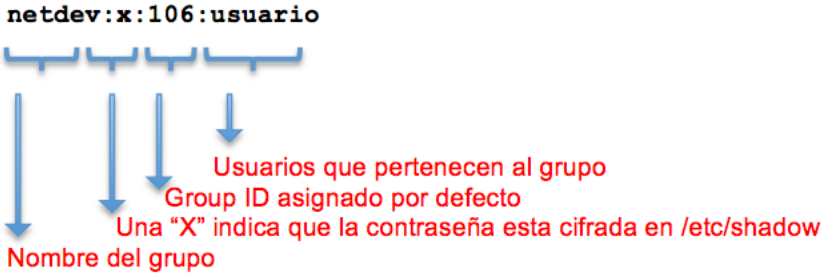


Figura A. 56 - Significado de los campos del archivo "/etc/group".

En el archivo "/etc/shadow" el sistema almacena las contraseñas cifradas (MD5-hash + semilla propia del sistema) del usuario las cuales tienen el siguiente formato:

```
root:$6$bRAtCZ2k$GsdQtbV2oiHpQb7GjTzG5qwKll95lLa5X.lOzOcQksHwLyCfcs/ks8gyAm1ql
PYNkCsRezF.Jc1Iy7fQIWSGg/:15880:0:99999:7:::
usuario:$6$rkOBl6xo$LEJtModsWe85zf7pxCua5ZP.nlHzYrAFTRV9RTBEniO1xrSJ1zH1YABd1j
wJKb8HzGx0jywLHyYy8NuScJEID0:15880:0:99999:7:::
```

Así como los archivos anteriores (*passwd* y *group*), en el archivo "/etc/shadow" (ver Figura A. 57) cada línea está separada en campos los cuales son delimitados por dos puntos ":" cada campo representa lo siguiente:

- Login: debe ser el mismo nombre que se utiliza en el archivo de *passwd*.
- Contraseña cifrada (MD5-hash + semilla).
- Indica los días que han pasado desde la última vez que la contraseña fue cambiada contados desde el día 1 de enero de 1970.
- Define los días que deben pasar como mínimo para que el usuario pueda cambiar la contraseña.
- Máximo de días durante los cuales la contraseña es válida, al terminar los días el usuario debe cambiar la contraseña.
- Días antes de caducar la contraseña en los cuales se avisará al usuario que debe cambiar la contraseña o la cuenta será bloqueada.
- Define el número de días que la cuenta será bloqueada, después de que la contraseña ha caducado.
- Días que será deshabilitada la cuenta contados desde el 1 de enero de 1970.

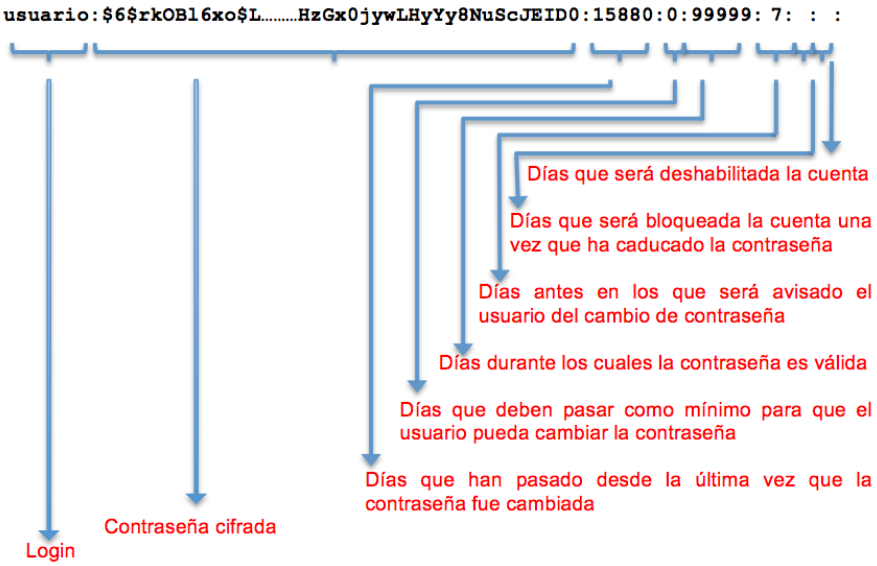


Figura A. 57 - Significado de los campos del archivo "/etc/shadow".

VI. (5.3.3) El bit SUID, SGID y STICKY

Los bits de *SUID, SGID* y *Sticky bit*, proporcionan a Unix una gran flexibilidad, pero al mismo tiempo constituyen un blanco importante para los ataques de elevación de privilegios en el sistema. Los sistemas Unix y similares cuentan con un determinado número de archivos ejecutables con los bits *SUID, SGID* y *Sticky bit* activados, debido a esto los intrusos tratan de crear más de estos archivos y ocuparlos para insertar funciones en su interior, las cuales no podrían ser ejecutadas con los permisos normales de un usuario, o bien una vez que han comprometido la seguridad del sistema hacen uso de los *bits* antes mencionados para cubrir sus huellas y pasar desapercibidos ante los ojos del administrador del sistema (ver Figura A. 58)

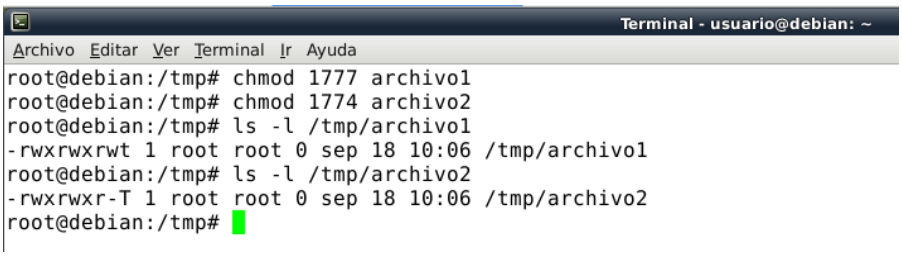


Figura A. 58 - Creación de archivos con permisos especiales.

Una de las vulnerabilidades asociadas a los archivos con el *bit* de permanencia activado, radica en que el sistema operativo interpreta que se trata de un archivo muy utilizado y en consecuencia garantiza que pase el mayor tiempo posible en la memoria principal, un intruso podría aprovecharse de esta característica y realizar ataques de *buffer overflow* para sobre-escribir localidades de memoria con código malicioso.

Para reducir al mínimo la posibilidad de que un intruso realice actividades maliciosas en el sistema mediante el uso de archivos especiales del sistema se deben realizar búsquedas específicas de estos archivos tomando como referencia los permisos que los caracterizan. Con el siguiente comando se pueden visualizar los archivos o

directorios del usuario “root” que a su vez cuenten con permisos especiales *SUID, SGID* y *Sticky bit* respectivamente.

```
find / -user root -perm -4000
find / -user root -perm -2000
find / -user root -perm -1000
```

Para buscar todos los archivos o directorios con permisos especiales sin importar a que usuario pertenecen, se deben ejecutar los siguientes comandos.

```
find / -perm -4000 -o -perm -2000
find / -perm -1000
```

Para la identificación de archivos o directorios con los tres tipos de permisos especiales se debe ejecutar el siguiente comando.

```
find / -perm -7000
```

Los comandos anteriores, fueron ejecutados en el sistema Linux-Debian que se está hardenizando, la ejecución de los comandos antes mencionados no mostró archivos sospechosos que hayan sido creados por algún usuario, sólo se localizaron algunos archivos propios del sistema (ver Figura A. 59) con los *bits* especiales. Debido a la función del equipo que se está hardenizando, a los archivos que se listan a continuación se les deben quitar los permisos especiales; “/usr/bin/Wall, /usr/bin/newgrp, /usr/bin/chsh, /bin/ping, /bin/ping6”; sin embargo si el sistema no es instalado desde “cero” es aconsejable ejecutar los comandos anteriores de forma periódica para identificar archivos potencialmente dañinos para el sistema. En caso de requerir quitarle a un archivo o directorio los permisos especiales (*SUID*) se debe ejecutar el comando `chmod ug-s`.

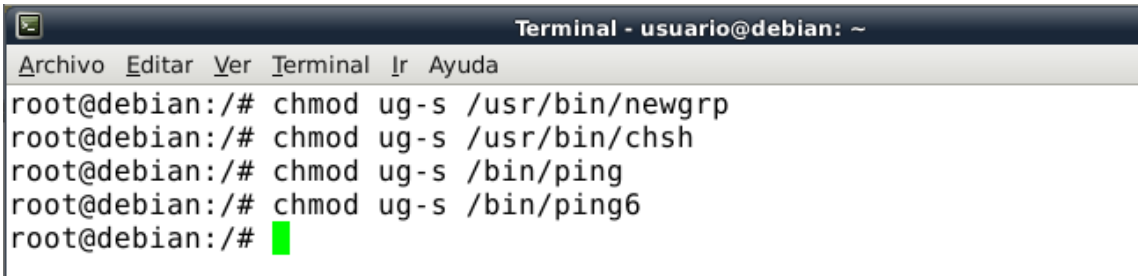


Figura A. 59 - Eliminación de permisos especiales con el comando chmod.

VII. (5.3.5) Caducidad de las cuentas de usuarios

Tanto el usuario *root* o algún otro con permisos de súper usuario pueden fijar la fecha de caducidad de contraseña para cualquier usuario. Para establecer la fecha de caducidad de la contraseña para la cuenta “usuario” se emplea el comando *chage* como se muestra en la Figura A. 60.

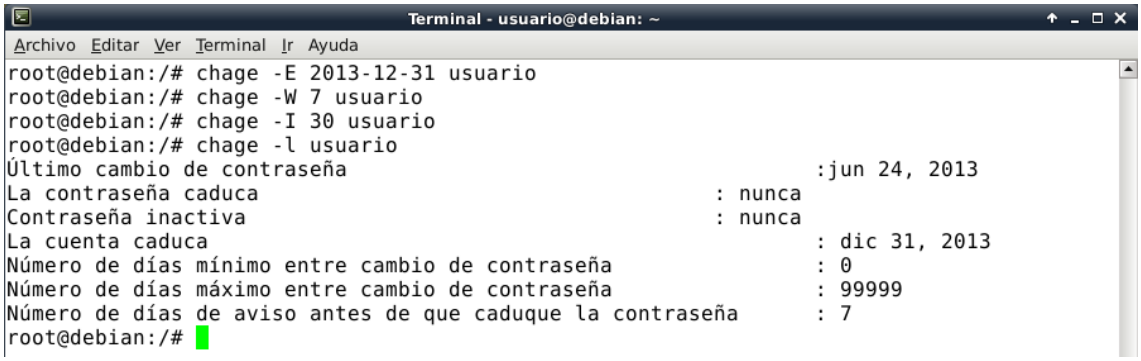


Figura A. 60 - Establecimiento de parámetros de caducidad a las contraseñas.

En la Figura A. 60 se puede observar que por medio del ejecutable `chage` se definió como fecha de expiración de la cuenta el último día del año 2013, se estableció que 7 días antes se avisara al usuario que debe cambiar su contraseña y si la cuenta permanece inactiva por más de 30 días será deshabilitada temporalmente.

La información anterior se puede corroborar con el contenido del archivo "/etc/shadow" el cual guarda las configuraciones referentes a las contraseñas.

```
usuario:$6$rkOBl6xo$LEJtModsWe85zf7pxCua5ZP.nlHzYrAFTRV9RTBEniO1xrSJ1zH1YABd1j
wJKb8HzGx0jywLHyYy8NuScJEID0:15880:0:99999:7:30:16070:
```

Por ejemplo si una política de seguridad indica que "las contraseñas deberán cambiarse máximo cada 180 días (6 meses), con un aviso previo de 10 días y 5 días de validez una vez vencida la contraseña", se debe ejecutar el siguiente comando:

```
chage –M 180  –W 7 –I 5  <usuario>
```

El calendario de sucesos y mensajes que se desplegarán en el sistema referente al comando anterior es:

- Día 0: Change your password.
- Día 173-179: *Warning: your password will expire in X days*.
- Día 180-184: *WARNING: Your password has expired. You must change your password now and login again!.*
- Día 185: *Your account has expired; please contact your system administrator.*

El resultado de la aplicación del comando anterior será que los usuarios que no hayan cambiado su contraseña en los diez días anteriores a la vigencia de seis meses, quedarán automáticamente bloqueados y tendrán cinco días para contactar al administrador del sistema para que les desbloque nuevamente su cuenta de usuario. Es importante señalar, que cada organización tiene sus propias políticas de seguridad de la información, por lo tanto las opciones con las que se use el ejecutable `chage` son variadas y particulares.

Al implementar políticas de contraseñas sobre usuarios ya existentes en el sistema es posible que la fecha del último cambio de contraseña no coincida con la fecha de implementación de las políticas en la organización; para evitar que existan estos conflictos se debe ejecutar el comando `chage` con la opción "-d" seguido de la fecha a partir de la cual será válida la política para el uso de las contraseñas.

chage -d 2013-09-1 <usuario>

Para que los parámetros de `chage` se apliquen automáticamente con cada usuario que se agregue al sistema se debe modificar el archivo “/etc/default/useradd”.

VIII. (5.3.6) Módulo para la gestión de contraseñas

La instalación de la librería para la gestión de contraseñas en el sistema (ver Figura A. 61), se debe realizar con el siguiente comando: `apt-get install libpam-cracklib.`

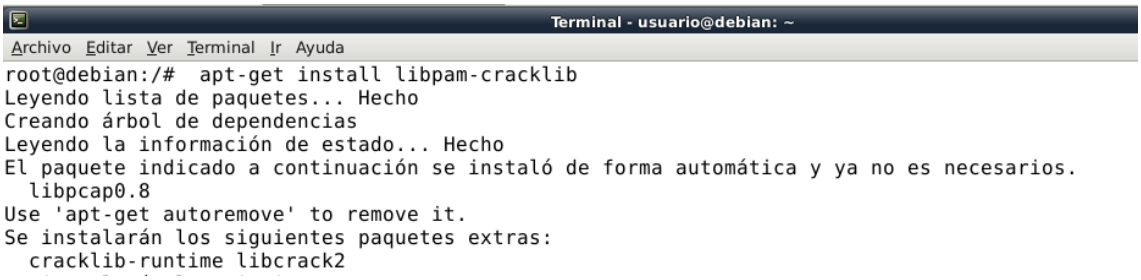


Figura A. 61 - Instalación del parche de seguridad *Grsecurity*..

Para implementar restricciones al uso de contraseñas se debe modificar el archivo “/etc/pam.d/common-password”, como se puede ver en la Figura A. 62.

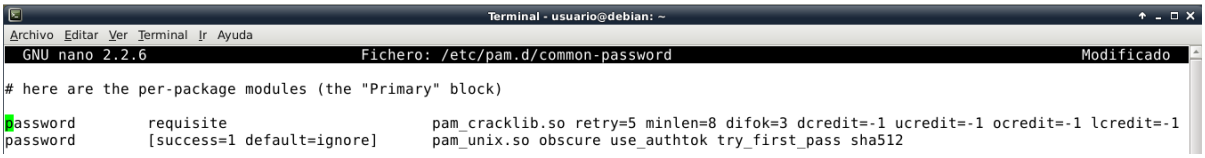


Figura A. 62 - Parámetros para la complejidad de las contraseñas.

Las opciones mostradas en la Figura A. 63 son; *difok=3 y dcredit=-1* (número), *ucredit=-1* (mayúsculas), lcredit=-1 (minúsculas) y ocredit=-1 (otro dígito), se emplean para definir la complejidad, el valor “-1” significa que se requiere al menos un carácter de este tipo. La siguiente vez que se desee definir una contraseña para un usuario, la cual no cumpla con las características ya definidas, el sistema mostrará un mensaje como el que se observa en la Figura A. 63.

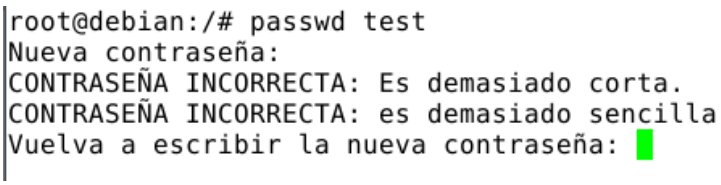


Figura A. 63 - Prueba de concepto para contraseñas débiles.

Para impedir que los usuarios reutilicen alguna contraseña previamente usada se debe usar el módulo *pam_unix*, que es el utilizado por defecto por *PAM*, el cual permite almacenar las contraseñas que se van utilizando en un archivo, de modo que *pam_cracklib* pueda verificar que no se vuelvan a utilizar, para lo cual se debe crear el archivo “opasswd” con los permisos adecuados como se muestra a continuación.

 # touch /etc/security/opasswd

```
# chmod 600 /etc/security/opasswd
```

Con los permisos de 600 dados al archivo “opasswd” se protege la confidencialidad de su información, debido a que este archivo guarda una copia del hash de las contraseñas de los usuarios las cuales también se localizan en “/etc/shadow”. Posteriormente se debe agregar al archivo “/etc/pam.d/common-password”, la siguiente línea para que en la autenticación de use el módulo de *PAM*.

```
password required pam_unix.so use_authtoknullok md5 remember=4
```

Para que la sesión se bloquee un determinado tiempo después de varios intentos de fallidos de inicio de sesión se debe utilizar el módulo *pam_tally*, que viene con la distribución estándar de PAM. Este módulo cuenta cada intento de inicio de sesión fallido ya sea por consola, comando `su`, o a través de un servicio vía red como SSH.

Los parámetros relevantes para el módulo de *PAM* son:

- onerr: indica qué hacer cuando algo falla (por ejemplo, por disco lleno), los valores son *succeed* (seguir como si nada) o *deny* (rechazar al usuario).
- deny: bloquear al usuario después de cierto número de intentos.
- unlock_time: si se incluye esta opción, la cuenta se desbloquea automáticamente después de la cantidad de segundos indicada.

Para bloquear a un usuario después de 3 intentos y desbloquearlo automáticamente 10 minutos después (unlock_time=600) se debe agregar al archivo “/etc/pam.d/common-auth” la siguiente línea:

```
auth required pam_tally.so onerr=succeed deny=3 unlock_time=600
```

Posteriormente agregar al archivo “/etc/pam.d/common-account” la siguiente línea:

```
account required pam_tally.so onerr=succeed
```

Con la configuración anterior se reduce el riesgo de vulnerabilidades o acceso al sistema mediante ataques de fuerza bruta, diccionario o híbridos.

IX. (5.4.3) Configurar el acceso por la consola física

En esta sección se continúa con él aseguramiento de las consolas físicas. Una vez comentadas las líneas deseadas descritas en la sección 5.4.3, al intentar la autenticación como usuario “root” en una *TTY* diferente de la *TTY4* aparecerá un mensaje como el mostrado en la Figura A. 64.

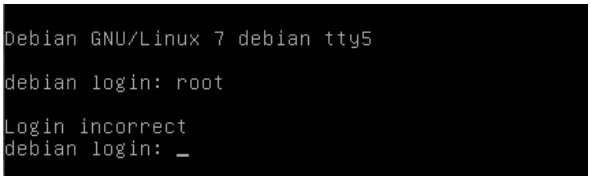


Figura A. 64 - Restricción de inicio de sesión en una TTY diferente a la TTY4.

Esto quiere decir, que un usuario no se podrá autenticar directamente como “root” en las consolas virtuales de la consola física, pero sí se podrá iniciar sesión como un usuario normal, para después elevar privilegios en él sistema, es decir, utilizar el comando `su`. También es necesario proteger el archivo *securetty* para asegurarse de que sea modificable sólo por el usuario “root”, esto se puede realizar modificando el dueño y permisos del archivo “/etc/securetty”.

```
# chown root:root /etc/securetty
# chmod 0600 /etc/securetty
```

Ya que se restringió el acceso del usuario “*root”* se debe proseguir con el aseguramiento de las consolas virtuales, ya que éstas también representan huecos de seguridad en el sistema. El principal riesgo es un equipo desatendido, el cual ocurre cuando se dejan sesiones abiertas sin antes haberlas bloqueado. Este tipo de situaciones se presentan comúnmente cuando se está trabajando en varias consolas virtuales y el equipo se deja de usar sin apagarlo, así pues el *hardening* consiste en impedir que estas sesiones queden abiertas.

El comando `vlock` permite bloquear la consola actual y también las consolas virtuales (es el equivalente a bloquear la sesión en un ambiente gráfico), de esta forma a menos que un usuario tenga la contraseña de la sesión activa, no podrá liberar la consola ni interferir con el proceso que esté en ejecución. La herramienta `vlock` viene instalada por defecto en algunas distribuciones de Linux, en el caso de la versión empleada en este trabajo no viene instalada por lo que será necesario instalarla con el siguiente comando.

```
# aptitude install vlock
```

Ahora bien, para bloquear sólo la consola que se esté utilizando, se debe emplear la opción “-c” tal como se muestra en la Figura A. 65.

```
# vlock –c
```

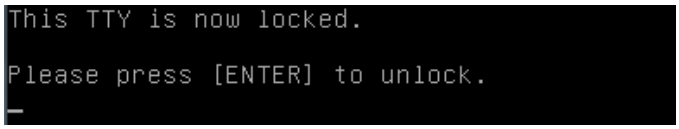


Figura A. 65 - Verificación de que la consola virtual está bloqueada.

Para bloquear todas las consolas en el equipo, se debe ejecutar el comando `vlock` con la opción “-a” que significa *ALL* (`vlock -a`), este comando bloqueará todas las consolas que tenga una sesión activa en el sistema.

X. (5.4.4) Acceso y configuración de Secure Shell

Continuando con el aseguramiento de Secure Shell (SSH), también se puede especificar qué usuarios serán los únicos que tendrán acceso a la conexión vía SSH, esto se puede hacer con la opción *AllowUsers (*archivo de configuración*)*, con los usuarios separados por un espacio, como se muestra en la siguiente línea.

```
AllowUsers usuario
```

Si se desea implementar una política prohibitiva se deben especificar los usuarios que no podrán conectarse a sistema vía SSH, para lo cual se debe agregar una línea en el mismo archivo de configuración con la opción *DenyUsers* y los usuarios separados por espacios.

```
DenyUsers test usuario2 usuario3
```

Para permitir conexiones al sistema únicamente desde *hosts* de los cuales se tiene la llave pública (relación de confianza) se debe agregar la siguiente línea:

```
HashKnownHosts yes
```

Para el cierre de sesión remota por inactividad después de 5 minutos, en la cual el servidor SSH enviará al cliente 5 mensajes como máximo y si no se recibe respuesta de parte del cliente la conexión será cerrada.

```
ClientAliveInterval 300
ClientAliveCountMax 5
```

Para limitar el tiempo que estará disponible la pantalla de *login* se realiza agregando LoginGraceTiem seguido del número de segundos que debe estar visible la solicitud del server.

```
LoginGraceTiem 20
```

Para forzar a que las configuraciones sean efectivas se debe reiniciar el demonio de SSH como se observa en la Figura A. 66.

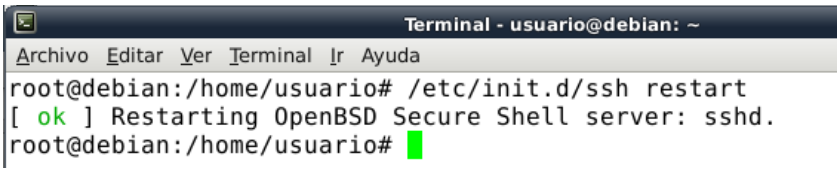


Figura A. 66 - Reinicio del demonio de SSH.

Si se requiere, por ejemplo que el servicio de SSH esté disponible sólo en un rango de tiempo laborar, por ejemplo de 09:00 horas a 20:00 horas se pueden utilizar las tareas programadas (*crontabs*) donde a las 09:00 horas se habilite el demonio de SSH y a las 20:00 horas se detenga, esta configuración sólo debe aplicarse del día lunes al día viernes, como se muestra en la Figura A. 167.

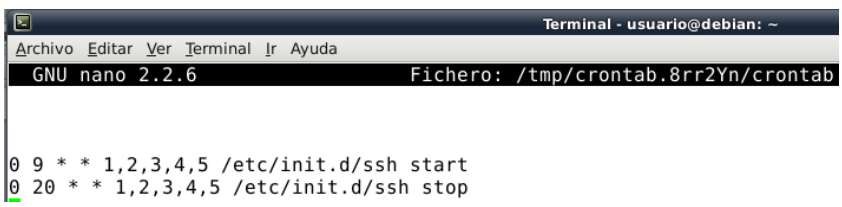


Figura A. 67 - Tarea programada para limitar el tiempo de conexión de SSH.

El mensaje de advertencia al acceder a Secure Shell es eficiente para disuadir a un posible intruso de seguir o no conectado a un sistema al cual no tiene acceso. De lo que se trata es que al iniciar sesión de forma remota en un sistema de información se muestre un mensaje (*banner*) que pone sobre advertencia a la persona que ha ingresado a un sistema sobre las repercusiones que podrían existir en caso de un acceso no autorizado o de que se realicen actividades no permitidas con el sistema en cuestión. En general, cada *banner* es muy particular de la institución o la finalidad del equipo, así como de las políticas de seguridad definidas, sin embargo para un equipo de trabajo de un CERT se propone que se contemplen los siguientes aspectos:

- Debe advertirse sobre el uso no autorizado del equipo y su información.
- Nunca se debe revelar el sistema operativo ni versión o dar más información de la necesaria. De esta manera los atacantes tendrán menos información, lo que provoca que sea más difícil para ellos penetrar en el sistema.
- Advertir que la actividad que se lleve a cabo en el equipo será registrada y que podría ser utilizada en la corte.
- Dejar claro que el usuario está ingresando a un equipo restringido y que en caso de haber iniciado sesión remota por error, debe desconectarse de forma inmediata y contactar al administrador.
- No usar palabras como “bienvenido”.
- Escribir el mensaje en el idioma local.
- El mensaje debe ser lo más corto posible y concreto.

Es necesario señalar que en este trabajo se configurarán dos tipos de *banners*, el primero se muestra cuando un usuario intenta iniciar sesión vía remota en el sistema y el segundo se muestra cuando el usuario ya ha iniciado sesión en él. Para el caso del primero es necesario modificar el contenido de los archivos “/etc/issue” y “/etc/issue.net” y para el segundo *banner* es necesario editar el archivo “/etc/motd”. El contenido de los archivos “issue*”* se muestra al usuario que intenta iniciar una sesión remota y el contenido del archivo “motd*”* cuando se ha ingresado correctamente de forma remota al equipo, local o a través de una terminal TTY.

Con base en los puntos descritos anteriormente se propone un mensaje de advertencia (ver Figura A. 68) y otro de bienvenida al usuario una vez que se ha autenticado correctamente (ver Figura A. 69).

Figura A. 68 - Mensaje de advertencia al intentar iniciar sesión con SSH.

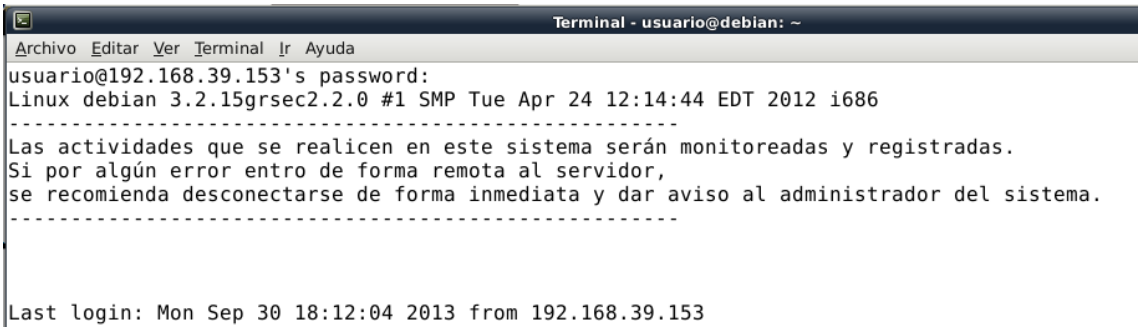


Figura A. 69 - Mensaje de advertencia al iniciar sesión con SSH.

XI. (5.5.1) Hosts.deny y Hosts.allow

Los *TCP Wrappers* nacieron de la necesidad de modificar el comportamiento del sistema operativo sin tener que modificar su funcionamiento. En sistemas Unix normalmente el demonio *inetd* se encarga de ejecutar los programas que atienden a cada tipo de cliente. El funcionamiento de los *Wrappers* se basa en hacer creer al demonio *inetd* que todos los protocolos son atendidos por el demonio de *TCP-Wrappers* llamado *tcpd* de tal forma que cada vez que llega un cliente *inetd* el sistema lanza el demonio *tcpd* sin importar de qué protocolo se trate como se muestra en la Figura A. 70.

Figura A. 70 - Engaño a demonio inetd de parte de TCP-Wrappers.

Cuando una conexión inicia, el demonio *tcpd* registra el nombre de la máquina remota, verifica las políticas de acceso y entonces corre el servicio original de red como se observa en la Figura A. 71.

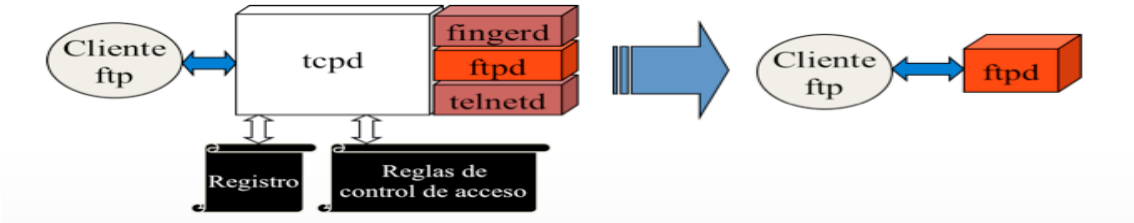


Figura A. 71 - TCP Wrappers en servicios de red.

TCP-Wrappers se componen de 5 programas:

- *Tcpd*: es el demonio del *TCP-Wrappers*.
- *tcpdmatch*: predice cómo el *tcpd* manejaría una petición en específico.
- *tcpdchk*: verifica las reglas de control de acceso contenidas en los archivos "/etc/hosts.allow" y "/etc/hosts.deny".
- *safe-finger*: versión de *finger* para implementar el finger reverso.
- *try-from*: Programa que permite probar si el sistema es capaz de reconocer qué máquina la está contactando.

Una de las características más potentes de los *TCP-Wrappers* con los archivos de control de acceso, es que el demonio de *tcpd* controla los accesos a los servicios

ofrecidos por el sistema con base en la configuración de los archivos “hosts.allow” y “hosts.deny”.

- /etc/hosts.allow. Contiene las reglas que especifican las máquinas y servicios que si están autorizados.
- /etc/hosts.deny. Contiene las reglas que especifican las máquinas y servicios que no están autorizados. [143]

La sintaxis de los archivos “hosts.allow” y “hosts.deny” es la siguiente:

```
<server_list>: <client_hosts_list>
```

- <server_list>
  - Lista de los demonios separados por coma o un espacio en blanco (sshd telnetd ...).
  - ALL, palabra clave que significa todos los servidores.
- <client_hosts_list>
  - Lista de los equipos separados por coma o un espacio en blanco.
  - Se pueden usar nombres de hosts y las direcciones IP.
  - Si utilizan nombres de equipos en lugar de direcciones IP, el DNS realizará la resolución de IPs correspondiente.

Los siguientes son puntos importantes a considerar cuando se usan *TCP-Wrappers* cuando se desea proteger los servicios de red:

a) Las reglas de acceso en “hosts.allow” se aplican primero (prioridad), toman precedencia sobre las reglas en “hosts.deny”. Por lo tanto, si el acceso a un servicio está permitido en “hosts.allow”, alguna regla que niegue el acceso a ese mismo servicio en “hosts.deny” será ignorada.
b) Las reglas en cada archivo se leen de arriba hacia abajo y la primera regla para un servicio determinado será la única que se aplique, en consecuencia el orden de las reglas es importante y debe ser definido con precaución.
c) Si no hay reglas para algún servicio en alguno de los archivos de *hosts* o si los archivos no existen, el acceso al servicio se concede.
d) Cualquier cambio en los archivos “hosts.allow” y “hosts.deny” será implementado por el sistema de forma inmediata sin necesidad de reiniciar los servicios de red.

En este trabajo se tomará la postura de una política prohibitiva la cual consiste en negar todo por defecto y posteriormente ir permitiendo los servicios que se desean (lo que está expresamente permitido). Esto se debe realizar editando el archivo “*/etc/host.deny”,* y colocando la línea: “ALL: ALL”. Posteriormente se debe editar el archivo de “*/etc/host.allow”* para indicar que sólo las direcciones IP de la red interna podrán acceder vía SSH al sistema (ver Figura A. 72).

143 Seguridad UNAM (n.d) Tutorial TCP-Wrappers. Obtenida el 18 de diciembre de 2013, de http://www.seguridad.unam.mx/descarga.dsc?arch=511

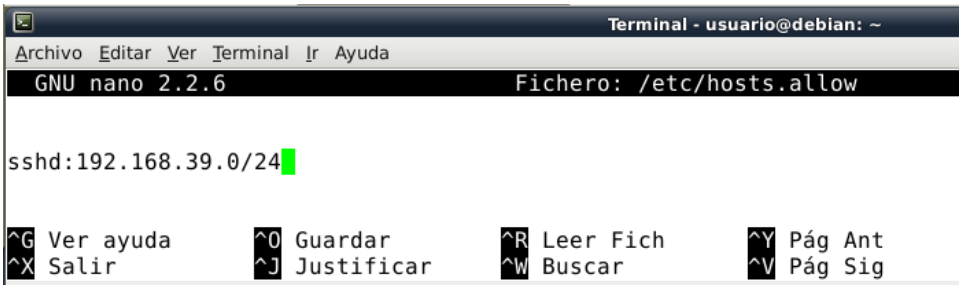


Figura A. 72 - Permitir conexiones con SSH sólo desde la red interna.

Con las reglas dispuestas en los archivos de *hosts* se puede prohibir el acceso a todos los servicios y direcciones IP, en este caso únicamente se permitió acceso a direcciones IP de la red interna 192.168.39.0/24 sobre el servicio de SSH, todos los demás servicios fueron denegados.

XII. (5.5.2) *Firewall*

Hay dos maneras de implementar un *firewall* las cuales deben ir acorde a las políticas de seguridad de la información dispuestas en la institución.

1) Política por defecto ACEPTAR: en principio todo lo que entra y sale por el *firewall* se acepta y sólo se denegará lo que se defina explícitamente.
2) Política por defecto DENEGAR: todo está denegado, y sólo se permitirá pasar por el *firewall* aquello que se permita explícitamente.

Si la política por defecto es DENEGAR, es muy difícil que un intruso pueda evadir el *firewall*, sin embargo esta política es difícil de aplicar si no se tiene pleno conocimiento de cómo es que funciona el sistema y sus servicios que en él operan, aunado a lo anterior esta configuración de *firewall* es ampliamente recomendada, aunque no es aconsejable usarla si no se domina el sistema.

A continuación se instalará un *firewall* para Linux (GuFw), con ambiente gráfico para agregar un mecanismo más de protección al sistema. La diferencia de usar este *firewall* y los iptables del sistema radica en que el *firewall* (GuFw) es más fácil de usar pero menos configurable, en cambio iptables es más complejo pero a la vez más flexible y potente en cuanto a configuración se refiere.

Para instalar el paquete se ejecuta: `aptitude install gufw`. Una vez instalado se ejecuta en la línea de comandos `gufw` y aparecerá una interface como la que se muestra en la Figura A. 73, en la cual se activa o desactiva el *firewall*.

Figura A. 73 - Interface gráfica de *firewall* (gufw).

En el menú editar se puede agregar una nueva regla (ver Figura A. 74), se observa que se puede tomar la plantilla de una regla pre-configurada, simple o avanzada, sin embargo debido a que en este trabajo la implementación del *firewall* se realizará con *iptables*, sólo se muestra de forma general el funcionamiento del *firewall*.

Figura A. 74 - Interface para configurar una regla en el *firewall* (gufw).

XIII. (5.5.3) Iptables

Continuando con la implementación de iptables, es preciso destacar que la política de restringir todo lo saliente y lo entrante hace que sea “imposible” para los usuarios comunicarse entre sí y por ende se puede afectar la operación y actividades diarias. Por tal razón, el realizar un *hardening* sobre un equipo con sistema operativo Debian que pertenece a un CERT, así como el conjunto de reglas que deberían ser aplicadas es una labor muy particular de cada organización, sin embargo en este trabajo se realizará la propuesta de las reglas que deberían ser incluidas o tomadas en cuenta para el *hardening* de este tipo de sistemas.

Para permitir el acceso al puerto 80 en el *firewall*, se debe añadir la siguiente regla:

```
iptables -A INPUT -p tcp -m tcp --sport 80 -j ACCEPT
iptables -A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Lo anterior permite la navegación normal desde los sitios Web que se comunican a través del puerto 80.

Para permitir el acceso a sitios Web seguros, se debe abrir el puerto 443.

```
iptables -A INPUT -p tcp -m tcp --sport 443 -j ACCEPT
iptables -A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

Para permitir el acceso a SSH, se deben utilizar las reglas siguientes:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p udp --sport 22 -j ACCEPT
```

Para permitir conexiones entrantes y salientes al puerto 22/TCP sólo a una red especifica (192.168.39.0/24).

```
iptables -A INPUT -p tcp -s 192.168.39.0/24 --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp -s 192.168.39.0/24 --dport 22 -j ACCEPT
```

Para verificar que las reglas han sido aplicadas correctamente se ejecuta el comando: "iptables –L" como se observa en la Figura A. 75.

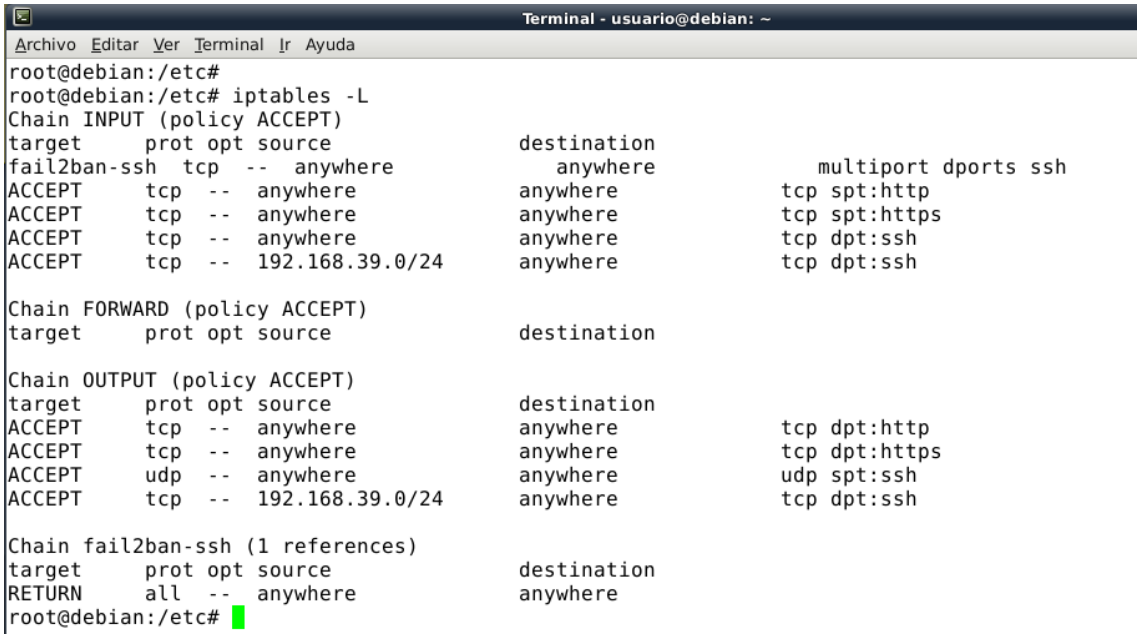


Figura A. 75 - Lista de reglas definidas con iptables.

Una forma automatizada de aplicar las reglas de *iptables* es a través de un *script*, el cual en primera instancia borra las reglas antes configuradas y pone a ceros los contadores de cadena de cada una de ellas, esto se debe realizar para que las nuevas reglas no causen conflicto ni comportamientos anómalos en el sistema, posteriormente se deben establecer las políticas por defecto (las cuales niegan todo lo entrante y saliente por defecto), el siguiente paso es permitir sólo lo que se desee que no sea filtrado por *iptables* (*firewall*), el *script* (programación en *Shell*) que realiza lo antes mencionado se muestra a continuación.

```
#!/bin/sh
## Política por defecto DROP

## FLUSH de reglas
iptables –F
iptables –X
```

```
iptables –Z
iptables -t nat -F

## Se establece política por defecto
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

## Reglas permisivas
iptables –A INPUT -p tcp -m tcp --sport 80 -j ACCEPT
iptables –A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT
iptables –A INPUT -p tcp -m tcp --sport 443 -j ACCEPT
iptables –A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT
iptables –A INPUT -p tcp --dport 22 -j ACCEPT
iptables –A OUTPUT -p udp --sport 22 -j ACCEPT
iptables –A INPUT -p tcp -s 192.168.39.0/24 --dport 22 -j ACCEPT
iptables –A OUTPUT -p tcp -s 192.168.39.0/24 --dport 22 -j ACCEPT
```

XIV. (5.6.1) Actualizaciones de seguridad.

Antes de continuar con los temas subsecuentes, como parte del mantenimiento del sistema se borrará la memoria cache, lo cual se puede realizar con el siguiente comando:

```
sync && echo 3 > /proc/sys/vm/drop_caches
```

En acciones relacionadas con las actualizaciones de seguridad, es importante tener certeza de que los repositorios que se agregan al archivo “sources.list” son repositorios seguros, puesto que abundan en Internet repositorios que no son confiables y representan un alto riesgo para la seguridad del sistema debido a que existe la posibilidad de descargar paquetes “troyanizados” desde fuentes no fiables.

Para prevenir lo anterior comentado, APT utiliza el archivo release.gpg asociado a cada paquete, el cual se emplea para revisar la integridad (no alteración) mediante el hash MD5 de los paquetes. Para hacer esta comprobación APT necesita conocer la llave pública del que firma el paquete, estas llaves se localizan en el directorio “/etc/apt/trsusted.gpg.d/”, como se muestra en la Figura A. 76.

Figura A. 76 - Llaves gpg de los repositorios del sistema.

Para ver las llaves públicas que vienen pre--configuradas con las llaves del archivo de Debian (ver Figura A. 77), se puede ejecutar el comando `apt-key list`.

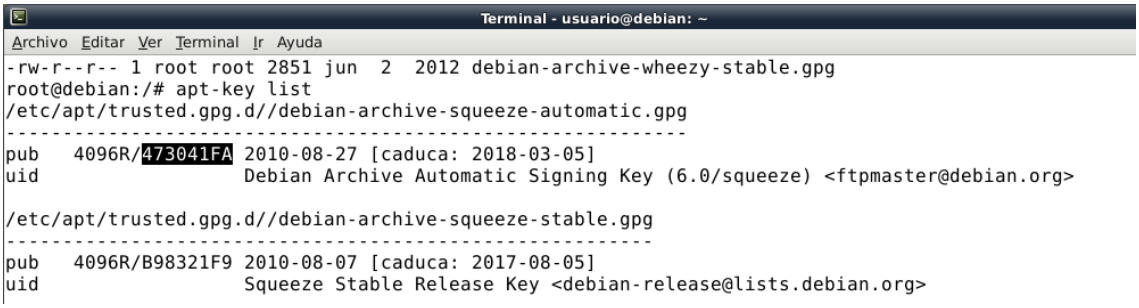


Figura A. 77 - Lista de llaves gpg y sus propiedades.

De la Figura A. 78, el valor “473041FA” se refiere a la identificación de la llave pública la cual en este caso es válida hasta el año 2018. Debian usa estas llaves como última línea de defensa con respecto a la instalación de paquetes. Si se añade un nuevo repositorio al archivo “/etc/apt/sources.list” se tiene que proporcionar a la aplicación APT la llave de dicho repositorio para que pueda certificar los paquetes descargados, para lo cual el proceso de añadir una llave pública de un repositorio se divide en 2 etapas:

a) Obtener la llave pública.
b) Exportar la llave y añadirla a APT.

Si se agrega un nuevo repositorio a Debian (no oficial) en el archivo “/etc/apt/sources.list”, una vez que se ejecute el comando “aptitude update”, el sistema mostrará un mensaje de error como el siguiente:

```
W: GPG error: http://www.deb-multimedia.org stable main Release: The following
signatures couldn't be verified because the public key is not available:
NO_PUBKEY 07DC563D1F41B907
```

El error anterior es mostrado por APT debido a que no tiene la llave pública para revisar la integridad y confidencialidad de dicho repositorio, para corregir el error se debe obtener dicha llave mediante el comando:

```
gpg --keyserver subkeys.pgp.net --recv-keys 07DC563D1F41B907
```

Una vez obtenida la llave, esta debe agregarse al llavero del sistema como se muestra en el siguiente comando:

```
gpg --export --armor 07DC563D1F41B907 | apt-key add -
```

Posteriormente se debe ejecutar el siguiente comando para que se realice la actualización de los repositorios:

```
aptitude update
```

Una vez hecho lo anterior, si se desea actualizar los paquetes instalados en el sistema de forma automática se puede utilizar el siguiente *script* (actualiza.pl) el cual sólo instala actualizaciones que no causen conflictos en el sistema (parámetro `safe-upgrade`), además almacena un registro de cada actualización en un archivo que lleva de nombre “upgrade_segura_año-mes-día.log” como se observa en la Figura A. 78.

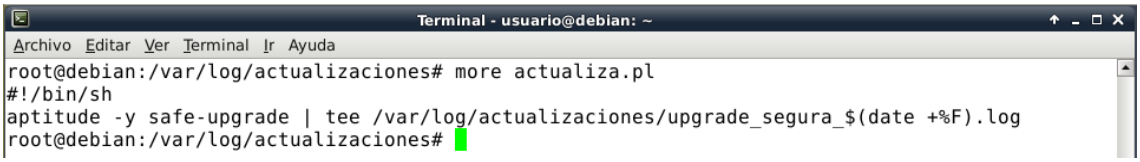


Figura A. 78 - Contenido del *script* actualiza.pl.

Siguiendo con la actualización automática del sistema se emplea la utilería del sistema "cron" sobre el script (actualiza.pl) para que este sea ejecutado todos los días lunes a las 15:00 horas (ver Figura A. 79).

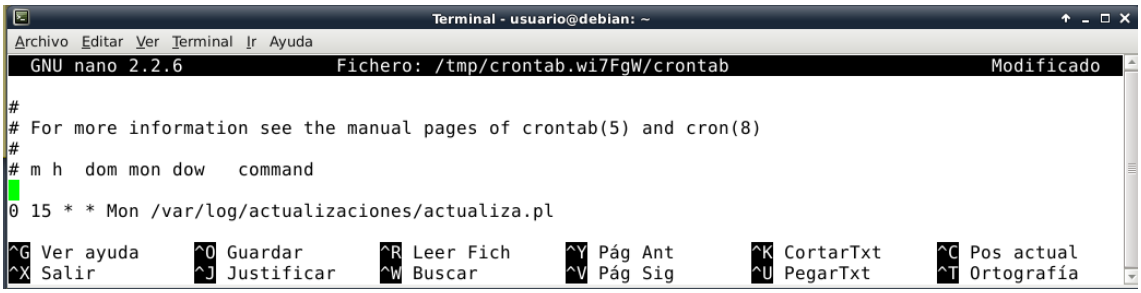


Figura A. 79 - Tarea programada para actualizaciones automáticas cada lunes.

XV. (5.6.2) Respaldos

A continuación se preparará el escenario para la ejecución de dos *scripts* usados para la obtención de respaldos totales automáticos y periódicos del archivo "/var/log/syslog", adicionalmente los respaldos serán colocados de forma remota en otro equipo de cómputo, cabe resaltar que la información será enviada de forma cifrada a través de la red con lo cual en caso de que exista un *sniffer* en la red, sólo permitirá que se observen datos sin ningún sentido, protegiendo así la confidencialidad de la información.

El primero paso es crear un directorio de "respaldos" en la máquina que tiene la información a respaldar, lo anterior debe hacerse con un usuario sin privilegios de administrador debido a que la conexión remota que extraerá esta información sólo se puede hacer a través de un usuario que no sea *root* debido al *hardening* previamente establecido para conexiones remotas (ver Figura A. 80).

Figura A. 80 - Creación de directorio de respaldos.

Posteriormente en la máquina que contiene la información a respaldar se debe crear un *script* (resp.sh) con permisos de 750, este *script* en esencia realiza el respaldo del archivo *syslog* y posteriormente le asigna como dueño a "usuario".

```
#!/bin/sh
fecha=`date '+%Y-%m-%d'`
tar zcvf /home/usuario/respaldos/syslog-$fecha.tar.gz /var/log/syslog
chown usuario:usuario /home/usuario/respaldos/syslog-$fecha.tar.gz
```

Para hacer que el *script* cree los respaldos automáticamente se utiliza la utilería *Cron*, de esta forma el *script* generará un respaldo todos días a una hora determinada, estas acciones se deben realizar desde una cuenta con privilegios puesto que se necesita acceder al archivo *syslog* el cual requiere de privilegios altos para su copia (ver Figura A. 81).

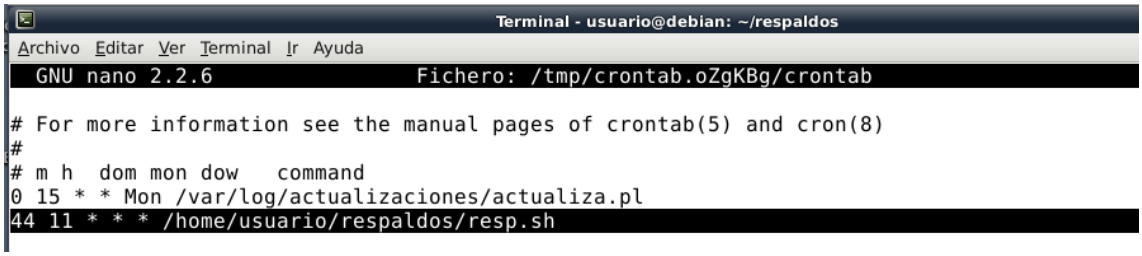


Figura A. 81 - Tarea programada para *script* de respaldos todos los días.

En este punto los respaldos son creados en la maquina local (Linux-Debian) como se muestra en la Figura A. 82, el siguiente paso es crear una relación de confianza con el servidor de respaldos, de esta forma los respaldos serán transferidos automáticamente de forma segura (cifrados y sólo con el sistema que contenga la llave privada).

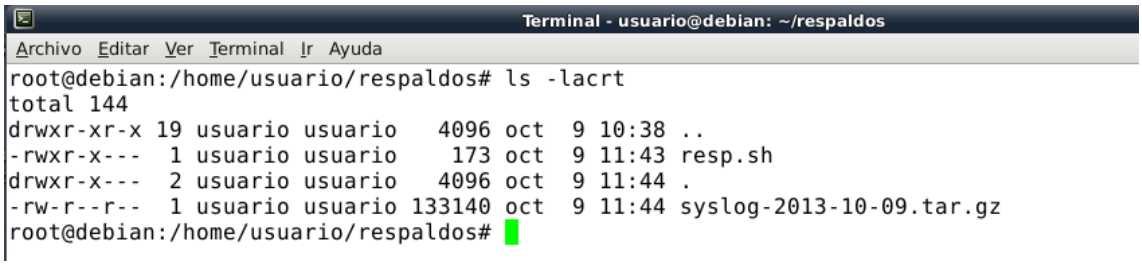


Figura A. 82 - Generación de respaldos periódicos del archivo syslog.

Para la creación de la relación de confianza se deben crear las llaves (pública y privada) en el equipo que fungirá como el servidor de respaldos, utilizando el comando `ssh-keygen -d` y el *passphrase* vacío, como se muestra en Figura A. 83.

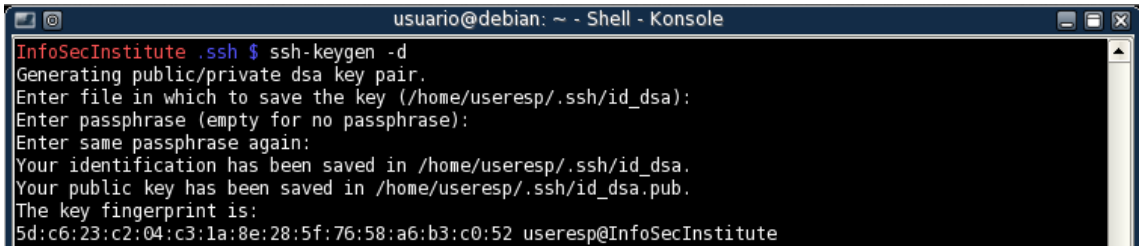


Figura A. 83 - Creación de llaves para SSH y relación de confianza.

El siguiente paso es exportar la llave pública (/home/useresp/.ssh/id_dsa.pub) al equipo Linux-debian (ver Figura A. 84) con dirección IP 192.168.39.153 y agregar la llave pública al archivo “authorized_keys2” con el siguiente comando:

```
cat /home/usuario/id_dsa.pub.cliente >> .ssh/authorized_keys2
```

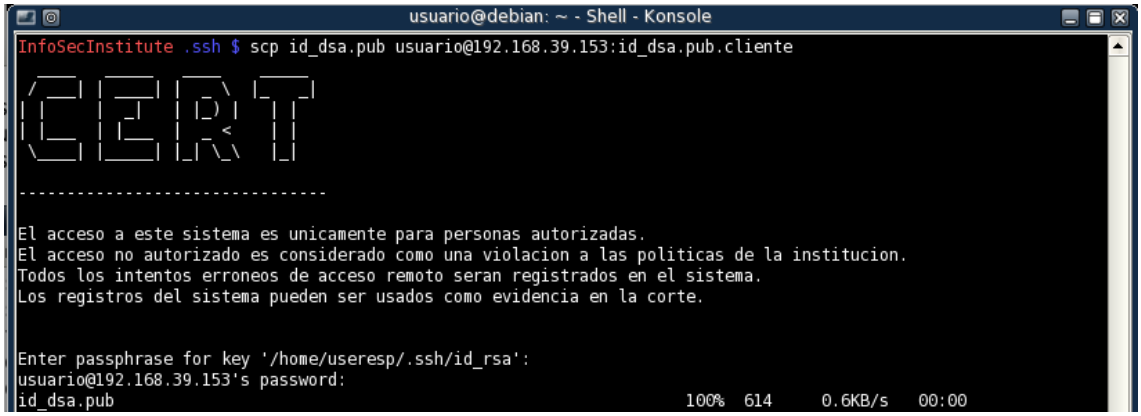


Figura A. 84 - Traslado de llave pública al servidor de respaldos.

En el equipo Linux-Debian se debe tener la llave del servidor de respaldos como se muestra en la Figura A. 85, de esta forma cada vez que el servidor de respaldos se conecte por medio de SSH (con el usuario **useresp**) al equipo Linux-Debian a través de la cuenta “usuario” para realizar la copia del respaldo diario (ubicado en “/home/usuario/respaldos”) el sistema no pedirá que se autentique, pues ya cuenta con la llave pública del servidor de respaldos, permitiendo así que el respaldo sea automatizado y los datos transiten en la red de forma cifrada.

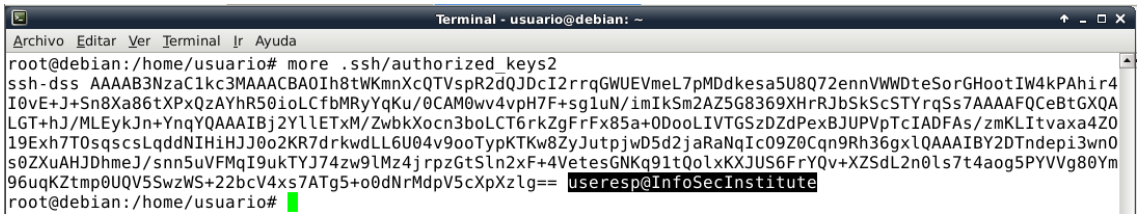


Figura A. 85 - Linux-Debian con la llave pública del servidor de respaldos.

El penúltimo paso consiste en crear un segundo *script* (**traeResp.sh**) en el servidor de respaldos el cual tendrá la tarea de conectarse al equipo Linux-Debian y copiar el respaldo correspondiente a la fecha del día en cuestión (ver Figura A. 86).

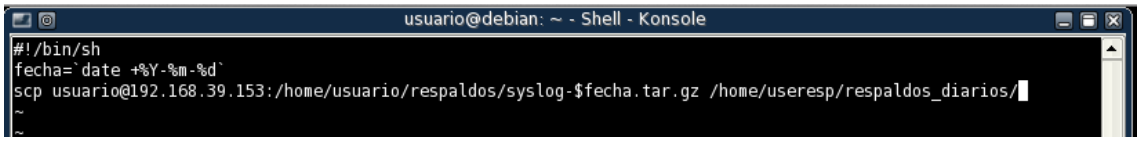


Figura A. 86 - *Script* para el traslado automático de los respaldos diarios.

El paso final es verificar que una vez que se ejecuta el anterior *script* se realice la conexión al equipo Linux-Debian y automáticamente se copie el respaldo (el sistema no solicita el *password*) para posteriormente ubicarlo en el directorio “/home/useresp/respaldos_diarios/” como se muestra en la Figura A. 87.

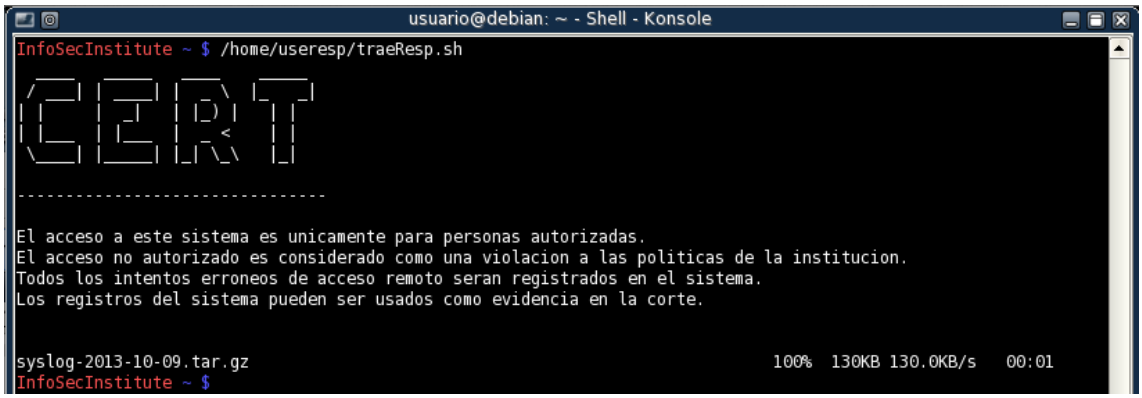


Figura A. 87 - Traslado del respaldo sin solicitud de contraseña.

Para automatizar el proceso antes descrito, también se debe crear en el servidor de respaldos una tarea programada con la utilería *Cron* (ver Figura A. 88), el cual debe ejecutarse tiempo después de que el respaldo es generado (11:44 horas) en el equipo Linux-debian (14:36 horas).

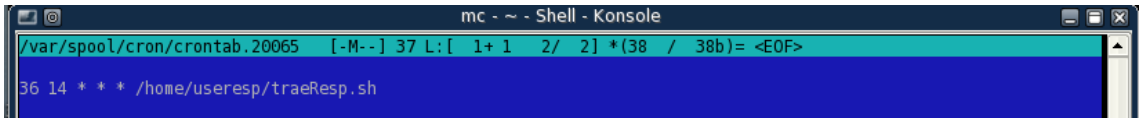


Figura A. 88 - Tarea programada para el traslado automático del respaldo.

Adicionalmente a lo ya comentado para la realización de respaldos, existe la herramienta gráfica (*FwBackups*) que permite realizar respaldos incrementales y periódicos, el código fuente puede ser descargado desde el sitio Web: http://www.diffingo.com/downloads/fwbackups/, la instalación y configuración se deja a criterio de las necesidades de la organización, la ventaja principal de la herramienta es que tiene una interface gráfica simple y a la vez con mucho potencial para la especificación de las características de los respaldos (ver la Figura A. 89).

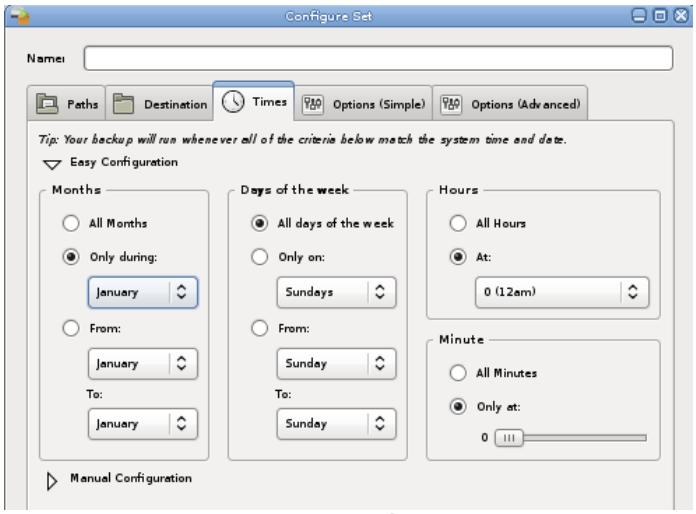


Figura A. 89 - Interface gráfica de *FwBackups*.

XVI. (5.6.3) Antivirus

En caso de que desee instalar el antivirus con interface gráfica se deben instalar los paquetes *clamav-daemon* y *clamtk*. Posteriormente, para acceder a la interface gráfica la ruta es “Menú de aplicaciones -> Accesorios -> ClamTk”. El siguiente paso es configurar el Proxy como se muestra en la Figura A. 90. En este caso el equipo Linux-Debian se encuentra detrás de un Proxy con dirección IP 10.241.208.243.

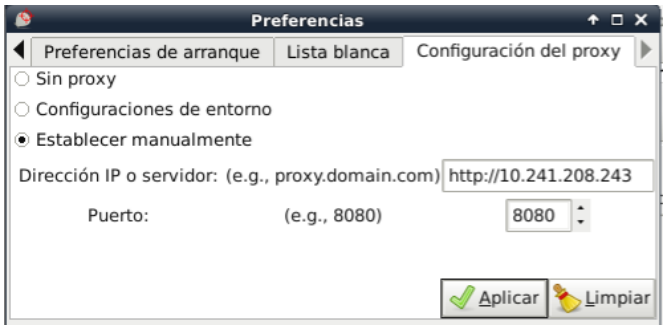


Figura A. 90 - Preferencias del antivirus ClamAV.

Se prosigue con la actualización de la base de datos del antivirus, para lo cual se debe ejecutar en la terminal el comando `freshclam`. Una de las principales ventajas de este software antivirus es que permite agregar una lista de los archivos (bien conocidos) que deberían ser omitidos durante el análisis (lista blanca) y planificar la hora en que el antivirus deberá realizar el análisis (ver la Figura A. 91).

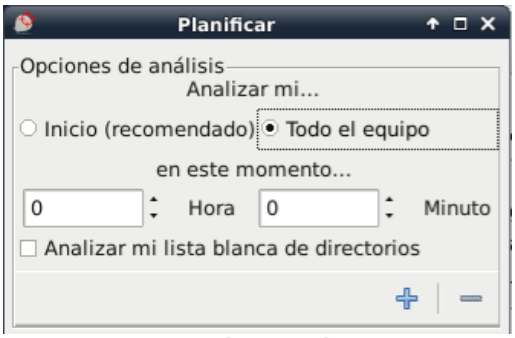


Figura A. 91 - Revisión periódica con ClamAV.

El paso final es configurar el análisis del sistema, para lo cual en la pestaña “Analizar” se debe elegir el directorio que se requiere revisar, seguido de esto el análisis comienza de forma automática como se muestra en la Figura A. 92.

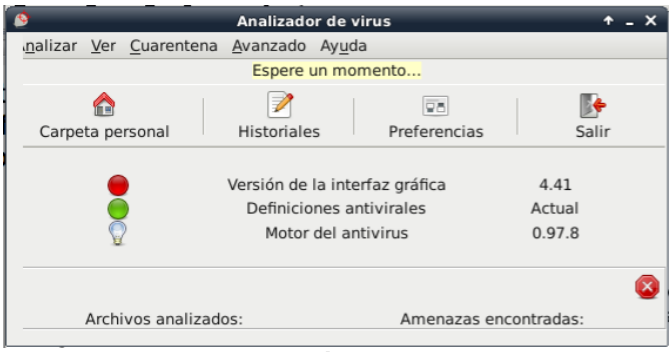


Figura A. 92 - Ejecución de antivirus ClamAV.

Al finalizar el análisis, el antivirus muestra que no se localizaron amenazas en el sistema lo cual refleja que el sistema puede utilizarse como un sistema base para verificaciones futuras de la integridad en el sistema.

XVII. (5.6.4) Auditorías regulares

La forma de instalar *Tiger* (herramienta para auditorías regulares) es mediante el comando `aptitude install tiger`. Para iniciar una revisión de la seguridad del sistema basta con ejecutar `tiger` en una terminal como se muestra en la Figura A. 93, al terminar la ejecución genera un reporte en formato de texto, el cual es ubicado en el directorio "/var/log/tiger/" este reporte incluye las vulnerabilidades encontradas del sistema (ver Figura A. 94).

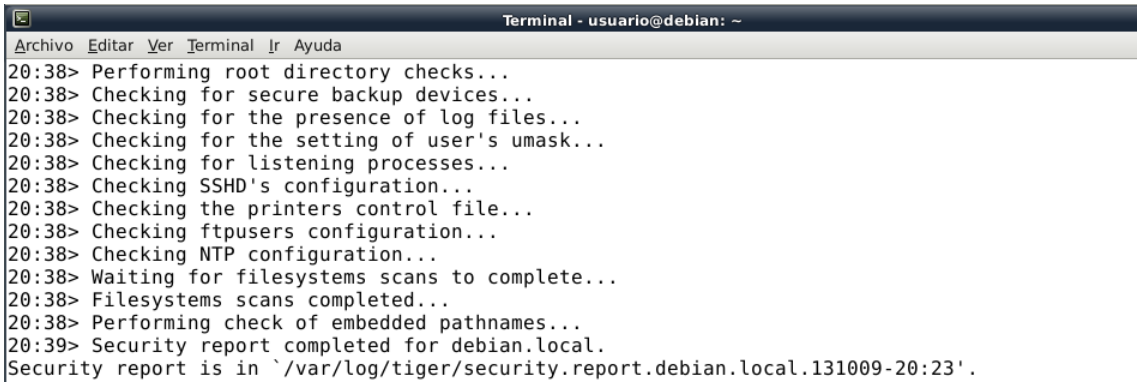


Figura A. 93 - Ejecución de la herramienta *tiger* en el sistema.

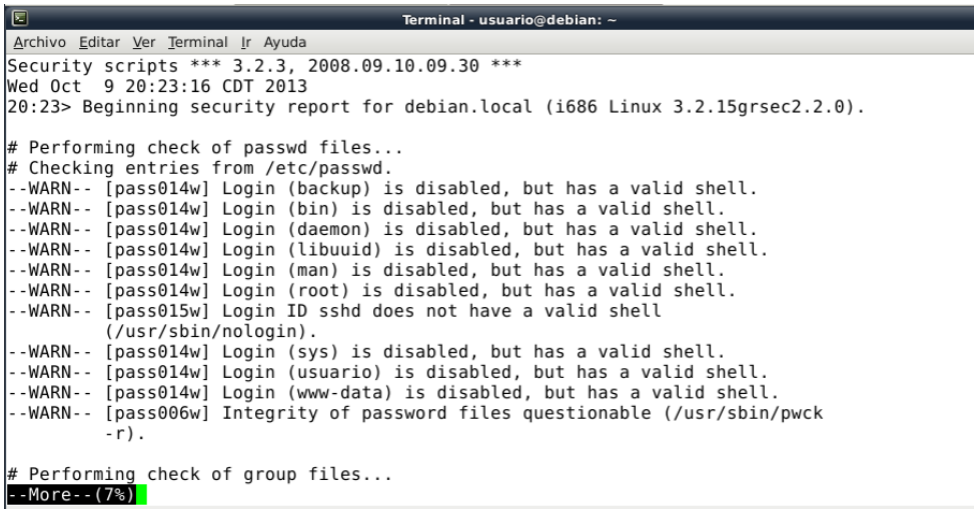


Figura A. 94 - Reporte de vulnerabilidades desarrollado por *tiger*.

Producto de la ejecución de la herramienta de seguridad *Tiger* se generó un reporte, el cual fue revisado y con base en lo desplegado se deshabilitaron los *shells* válidos para cuentas del sistema, que si bien ya están deshabilitadas no deberían tener asociado un *Shell*. También se realizaron cambios de permisos de SUID en los binarios *su*, *at* y *wall*.

Una vez corregidas las vulnerabilidades mostradas por *Tiger,* se ejecutó de nuevo la herramienta y se pudo observar en el informe de *Tiger* que el nivel de seguridad del sistema es alto, sin embargo el verdadero reto del *hardening* es mantener el nivel de seguridad y funcionalidad, es por ello que a continuación se abordará el aspecto de cómo mantener la integridad del sistema.

XVIII. (5.6.5) Validación de integridad

Para realizar la verificación de integridad de los archivos críticos del sistema, se requiere instalar la herramienta de *Tripwire* configurarse con el objetivo de poder realizar revisiones periódicas de la integridad. Para instalar la herramienta basta con ejecutar el comando: `aptitude install tripwire`, posteriormente mediante una interface gráfica el sistema de instalación le preguntará al usuario si desea definir una contraseña para firmar archivos que son compartidos con otros sistemas (ver Figura A. 95), y también solicita otra contraseña para firmar los archivos locales de *Tripwire.*

Figura A. 95 - Solicitud de contraseña para firma de archivos.

En la Figura A. 96 se muestra el proceso para definir la contraseña o clave de sitio.

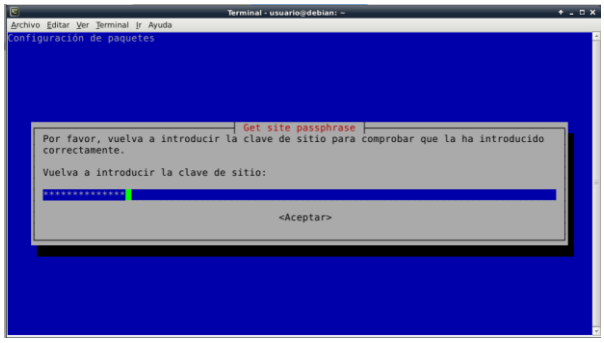


Figura A. 96 - Solicitud de clave de sitio.

Posteriormente el sistema solicita la clave local como se muestra en la Figura A. 97.

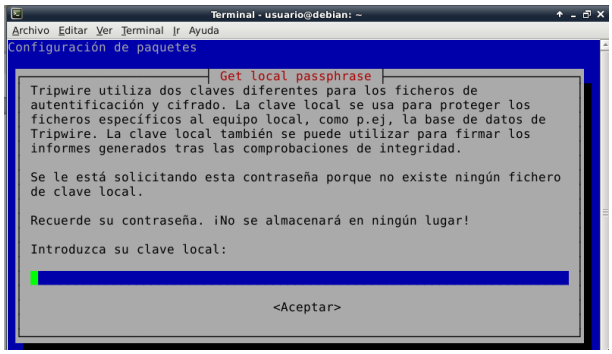


Figura A. 97 - Solicitud de clave local.

Finalmente se indica que ha terminado la instalación de *Tripwire* de forma exitosa (ver Figura A. 98)

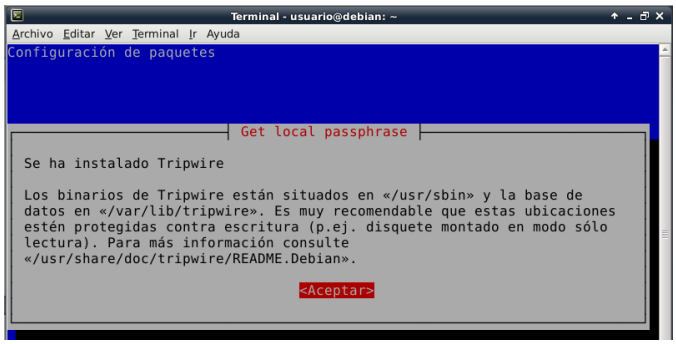


Figura A. 98 - Fin de la instalación de *Tripwire*.

Una vez terminada la instalación de *Tripwire* es necesario hacer *hardening* a los archivos de configuración debido a que se encuentran en claro en el sistema. Los permisos de los archivos deben tener únicamente permisos de lectura y escritura para el usuario "root", las instrucciones necesarias se hacen con el comando `chmod` como se muestra en la Figura A. 99.

Figura A. 99 - Permisos de archivos de configuración de *Tripwire*.

Para modificar las políticas predeterminadas de *tripwire*, es necesario editar el archivo "/etc/tripwire/twpol.txt", en el cual se definen los archivos y carpetas que no se desea que *Tripwire* revise. Cuando el archivo de políticas contiene todo lo que se pretende monitorear además del nivel de criticidad, el siguiente paso es cargar las políticas a *Tripwire* para lo cual se emplea una versión compilada y cifrada de este archivo localizada en el archivo "/etc/tripwire/tw.pol". Para generar y regenerar el archivo binario con las políticas cuantas veces sea necesario se debe ejecutar el siguiente comando e introducir la clave (contraseña) de sitio que se introdujo en la fase de instalación de *Tripwire*.

```
/usr/sbin/twadmin -m P /etc/tripwire/twpol.txt
```

El siguiente paso es respaldar los archivos /etc/tripwire/twcfg.txt y /etc/tripwire/twpol.txt, quitar los permisos de lectura (*old.txt) para cualquier usuario y finalmente eliminar los archivos originales, debido a que si un intruso logra acceder a ellos, podría saber en qué directorio esconder archivos maliciosos para que no sean detectados por *Tripwire*.

```
cp /etc/tripwire/twcfg.txt /etc/tripwire/twcfg_old.txt
cp /etc/tripwire/twpol.txt /etc/tripwire/twpol_old.txt
chmod 000 /etc/tripwire/*old.txt
rm /etc/tripwire/twpol.txt /etc/tripwire/twcfg.txt
```

Si en un futuro se requiere generar de nuevo el archivo de políticas y configuración en texto claro los comandos que se deberían ejecutar son:

```
/usr/sbin/twadmin -m p > /etc/tripwire/twpol.txt
/usr/sbin/twadmin -m f > /etc/tripwire/twcfg.txt
```

Por otro lado, para construir la base de datos de *Tripwire* se necesita recolectar la información actual de los archivos que se desean monitorear. Dicha información se almacena en una base de datos especial, la cual es generada mediante el comando `tripwire –m i`, sin embargo, en ocasiones existen algunos errores relacionados con archivos o rutas no existentes, por lo tanto para depurar la base de datos inicial se sugiere mandar los errores a un archivo (2> /tmp/mensajes) y repetir el procedimiento hasta que no se registren más errores, este paso se realiza mediante el siguiente comando:

```
tripwire –m i 2> /tmp/mensajes
```

En el comando anterior se ha redirigido la salida de errores al archivo "/tmp/mensajes". Es probable que haya archivos especificados en las políticas (twpol.txt) que no existen o están incorrectamente escritos, esto quedará registrado en "/tmp/mensajes", los errores deberán corregirse en twpol.txt, para proceder a reconstruir la base de datos de *Tripwire*. Este procedimiento se repetirá mientras subsistan errores en el archivo de políticas.

> Nota: Se debe borrar el archivo "/tmp/mensajes" cuando se tenga un estado adecuado para la base de datos.

Si en un futuro se desea dejar de monitorear ciertos archivos o iniciar el monitoreo de otros, se debe configurar el archivo de políticas "/etc/tripwire/twpol.txt como se vio en pasos anteriores, posteriormente regenerar dicho archivo de políticas que permita generar de nuevo la base de datos que *Tripwire,* la cual se tomará como base para la comparación de archivos sospechosos, de haber sido comprometidos.

Ahora que *Tripwire* está correctamente configurado con su base de datos, es el momento de verificar la integridad del "*filesystem*". Esto se consigue con el comando:

```
tripwire –m c
```

El comando anterior se usará cada vez que se requiera saber el estado de integridad del sistema (ver Figura A. 100), si por alguna razón algunos de los archivos monitoreados son modificados (por ejemplo, por una actualización en el software) entonces se debe reconstruir nuevamente la base de datos a fin de que no aparezcan discrepancias con el estado actual del sistema en verificaciones futuras.

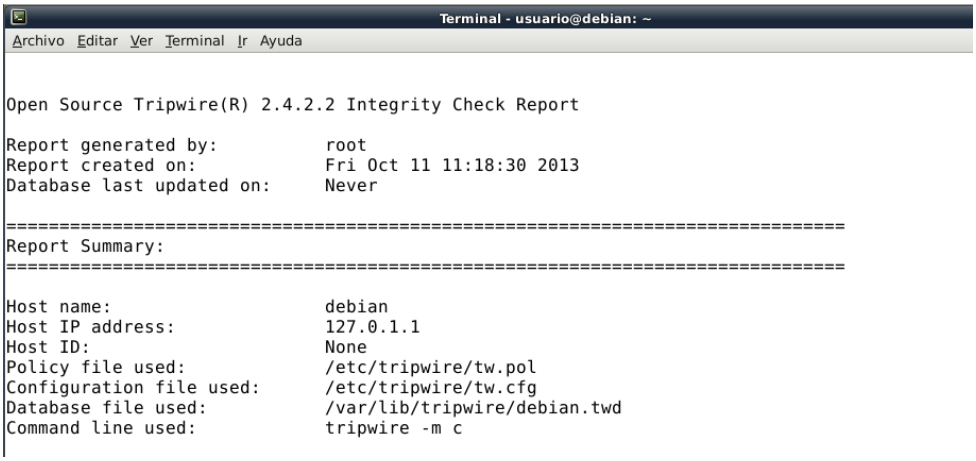


Figura A. 100 - Ejecución de *Tripwire* para verificar la integridad del sistema.

Ahora que se ha probado la correcta ejecución de *Tripwire*, es necesario programar la ejecución automática, para lo cual dependiendo del uso que se le dé al equipo y la interacción de los usuarios con él, se definirá el periodo de ejecución de *Tripwire*. Por ejemplo, para este trabajo de tesis se propone que la ejecución sea diaria, por lo tanto se debe agregar al directorio de cron "/etc/cron.daily/" un archivo "/etc/cron.daily/tripwire_p "con el siguiente contenido:

```
#!/bin/sh
/usr/sbin/tripwire -m c > /root/informeIntegridad.txt
```

Para garantizar que la utilidad del sistema (*Cron)* funcione correctamente se deben asignar los siguientes permisos.

```
# chmod 750 /etc/cron.daily/tripwire
```

El *script* anterior enviará el resultado de la ejecución de *Tripwire* a un archivo ubicado en el directorio "/root, el cual deberá ser revisado por el administrador en busca de cambios en archivos críticos del sistema.

XIX. (5.7.1) SElinux

Continuando con la configuración de la herramienta de seguridad en Linux (SElinux), el siguiente paso es editar el archivo de configuración "/etc/selinux/config" poniendo en la variable SELINUX el valor de *permissive* de esta forma sólo advierte y registra las acciones que considere maliciosas y no causa conflictos con alguna otra política configurada previamente en el sistema, con la variable SELINUXTYPE en *targeted* (ver Figura A. 101) se protegen por defecto los demonios: *httpd, named, dhcpd, mysqld* entre otros.

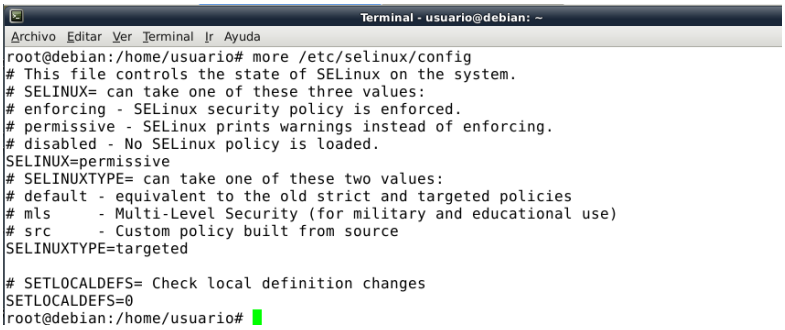


Figura A. 101 - Archivo de configuración de SElinux.

Ahora es necesario ejecutar el comando `sestatus` para comprobar que todo ha sido configurado correctamente como se observa en la Figura A. 102. De esta forma el sistema está listo para proteger servicios comunes como los descritos en el párrafo anterior y además registrará toda aquella actividad maliciosa relacionada con intentos de acceso y errores de autenticación en el sistema.

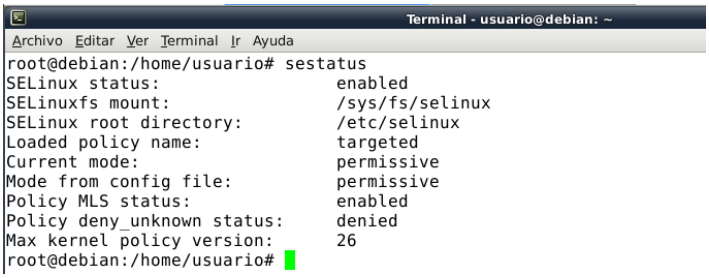


Figura A. 102 - Verificación de configuraciones de SElinux.

Para saber las políticas definidas por defecto y el valor de cada una de ellas se debe emplear el comando `getsebool -a` como se observa en la Figura A. 103.

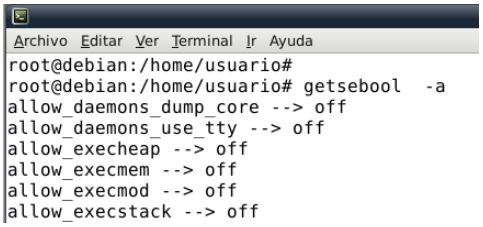


Figura A. 103 - Políticas definidas por SElinux.

Por defecto las políticas que tienen un valor de "*on"* serán aplicadas al sistema, cabe destacar que existen políticas que es mejor dejar con un valor de off puesto que así es más seguro el sistema como las mostradas en la Figura A. 103. Hasta este momento el sistema tiene un nivel de seguridad óptimo para realzar las funciones propias de un equipo de trabajo de un CERT, sin embargo en caso de que existan políticas de seguridad de la información más restrictivas al interior de la organización, la forma de cambiar los valores de las políticas definidas por SElinux será de la siguiente forma:

```
setsebool nombre-politica valor(on|off)
```

B. Anexo

El presente anexo muestra la relación de equivalencia del ISO/IEC 27001:2005 y la actual ISO/IEC 27001:2013. El número entre paréntesis significa el número de controles en cada grupo que hacen sentido con el control definido en la versión 2005. [144]

Tabla B. 1 - Equivalencia del ISO/IEC 27001:2005 y 2013

Mapeo de controles ISO/IEC 27001 de versión 2005 a versión 2013													
Equivalencia		**ISO/IEC 27001:2005**											
ISO/IEC 27001:2013		Políticas de seguridad	Organización de la seguridad de la información	Gestión de activos	Seguridad de recursos humanos	Seguridad física y ambiental	Gestión de las comunicaciones y operaciones	Control de acceso	Adquisición, desarrollo y mantenimiento de los sistemas	Gestión de incidentes de seguridad de la información	Gestión de continuidad del negocio	Complimiento	
		A.5	**A.6**	**A.7**	**A.8**	**A.9**	**A.10**	**A.11**	**A.12**	**A.13**	**A.14**	**A.15**	
A.5.1	Dirección de gestión de la seguridad de la información (2)	X											
A.6.1	Organización interna (5)		X		X		X						
A.6.2	Los dispositivos móviles y telecomunicaciones (2)							X					
A.7.1	Antes de la implementación (2)				X								
A.7.2	Durante la implementación (3)				X								
A.7.3	Terminación y cambio de la implementación (1)				X								
A.8.1	Responsable de activos (4)			X	X								
A.8.2	Clasificación de información (3)			X			X						
A.8.3	Manejo de medios (3)						X						
A.9.1	Requerimientos de la institución para el control de acceso (2)							X					
A.9.2	Administración de accesos de usuarios (6)				X			X					
A.9.3	usuarios responsables (1)							X					
A.9.4	Sistema y aplicación de control de accesos (5)							X	X				
A.10.1	Controles de criptografía (2)								X				
A.11.1	Áreas de seguridad (6)					X							
A.11.2	Equipamiento (9)					X		X					
A.12.1	Procedimientos operacionales y responsables (4)						X						
A.12.2	Protección para Malware (1)						X						
A.12.3	Respaldo (1)						X						
A.12.4	Monitoreo y registro (4)						X						
A.12.5	Operación de control de software (1)								X				
A.12.6	Administración de técnicas de seguridad (2)								X				
A.12.7	Sistemas de auditoría de información considerados (1)											X	
A.13.1	Administración de la seguridad de la red (3)						X	X					
A.13.2	Transferencia de información (4)		X				X						

[144] BSI (2013). Moving from ISO/IEC 27001:2005 to ISO/IEC 27001:2013, Mapas y tablas, consultado el 20 de diciembre de 2013, disponible en http://www.bsigroup.com/LocalFiles/en-GB/iso-iec-27001/resources/BSI-ISO27001-transition-guide-UK-EN-pdf.pdf

Mapeo de controles ISO/IEC 27001 de versión 2005 a versión 2013													
E q u i v a l e n c i a		**ISO/IEC 27001:2005**											
ISO/IEC 27001:2013		Políticas de seguridad	Organización de la seguridad de la información	Gestión de activos	Seguridad de recursos humanos	Seguridad física y ambiental	Gestión de las comunicaciones y operaciones	Control de acceso	Adquisición, desarrollo y mantenimiento de los sistemas	Gestión de incidentes de seguridad de la información	Gestión de continuidad del negocio	Complimiento
		A.5	**A.6**	**A.7**	**A.8**	**A.9**	**A.10**	**A.11**	**A.12**	**A.13**	**A.14**	**A.15**
A.14.1	Requerimientos de sistemas de información de seguridad (3)						X		X			
A.14.2	Desarrollo en seguridad y soporte en procesos (9)						X		X			
A.14.3	Datos de prueba (1)								X			
A.15.1	Relación del proveedor en seguridad de la información (3)		X									
A.15.2	Servicio de administración de entrega a proveedores (2)						X					
A.16.1	Administración de los incidentes de seguridad de la información y mejoras (7)									X		
A.17.1	Continuidad de la seguridad de la información (3)										X	
A.17.2	Redundancias (1)											
A.18.1	Conformidad legal y requerimientos contractuales (5)											X
A.18.2	Observaciones de la seguridad de la información (3)		X								X	

C. Anexo

El presente anexo muestra las acciones implementadas para verificar la eficacia de las buenas prácticas de seguridad realizadas en este trabajo de tesis en un equipo de cómputo con sistema operativo Linux-Debian versión 7.1. Los resultados muestran que el nivel de seguridad del sistema se incrementó significativamente respecto de las condiciones iniciales del sistema operativo recién instalado.

Con el objetivo de verificar el nivel de seguridad del sistema operativo como resultado del *hardening* implementado, se realizó una comparativa entre el antes y después, es decir, se ejecutaron herramientas de análisis de vulnerabilidades primero sobre el sistema operativo Linux-Debian recién instalado y posteriormente sobre el sistema al cual se le aplicaron las buenas prácticas de seguridad. Además, los análisis de vulnerabilidades se realizaron con un enfoque externo (herramienta Nessus) y con un enfoque interno (Lynis) del sistema Linux-Debian, de esta forma se logró tener una perspectiva integral (externa e interna) sobre el nivel de seguridad.

I. Análisis de vulnerabilidades (enfoque interno)

Para realizar el análisis de seguridad desde el interior del propio sistema, se utilizó la herramienta “Lynis” versión 2.1.0, la cual analiza el software instalado, información general del sistema, paquetes, archivos de configuración, usuarios, grupos, procesos, *firewall*s, kernel, servicios, puertos, etc. Esta herramienta puede ser descargada desde el sitio Web “https://cisofy.com/download/lynis/”, para ejecutarse no es necesario instalarla en el sistema.

Como se puede ver en la Figura C.1, la herramienta fue ejecutada sobre el equipo con sistema operativo Linux-Debian 7.1, versión de kernel 3.2.0.

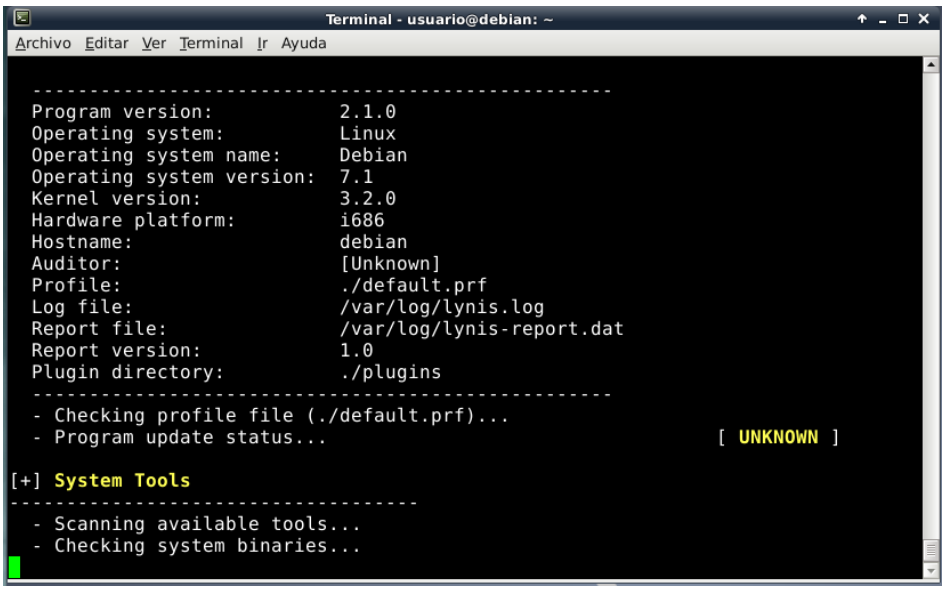


Figura C.1. Ejecución de herramienta Lynis antes del *hardening*.

Una vez concluida la ejecución de la herramienta de auditoría de seguridad “Lynis”, en el equipo con el sistema operativo Linux-Debian (recién instalado) se observa que la

herramienta asigna un porcentaje de fortaleza (*hardening*) del **52%.** Como se puede observar en la Figura C.2.

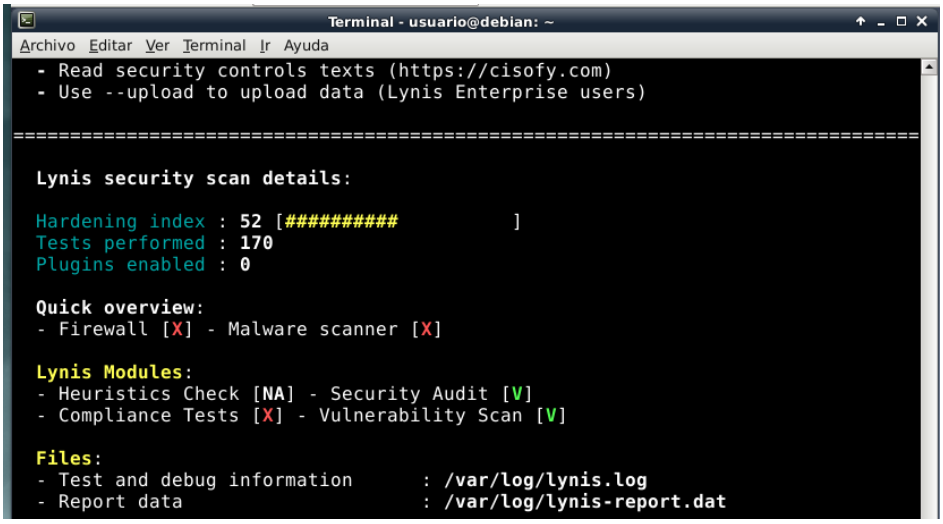


Figura C.2. Resultados de ejecución de herramienta Lynis antes del *hardening*.

Posteriormente, la misma herramienta se ejecutó sobre el sistema que representa el producto final de este trabajo de tesis, dando como resultado un porcentaje de fortaleza (hardening) del **69%,** lo cual significa que las acciones implementadas en este trabajo de tesis cumplieron con el objetivo de incrementar el nivel de fortaleza del sistema operativo en un **17%** como se observa en la Figura C.3.

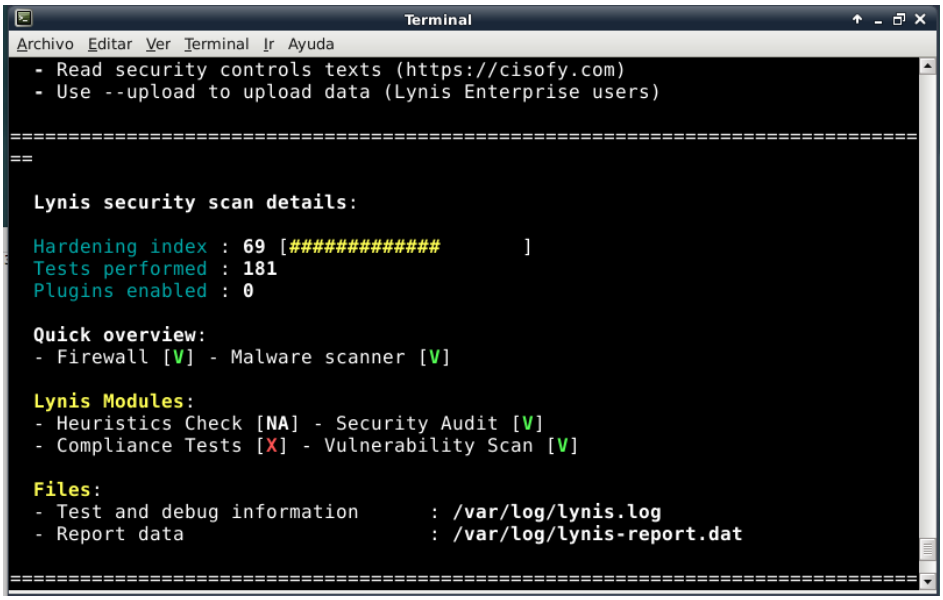


Figura C.3. Ejecución de herramienta Lynis después del *hardening*.

II. Análisis de vulnerabilidades (enfoque externo)

El análisis de seguridad desde el exterior del sistema, se realizó mediante el escáner de vulnerabilidades “Nessus” versión 5 con los *plugins* actualizados al 2015.

El primer escaneo de vulnerabilidades se realizó sobre el sistema operativo Linux-Debian recién instalado, dando como resultado 1 vulnerabilidad catalogada como media y 15 catalogadas como bajas como se muestra en la Figura C.4.

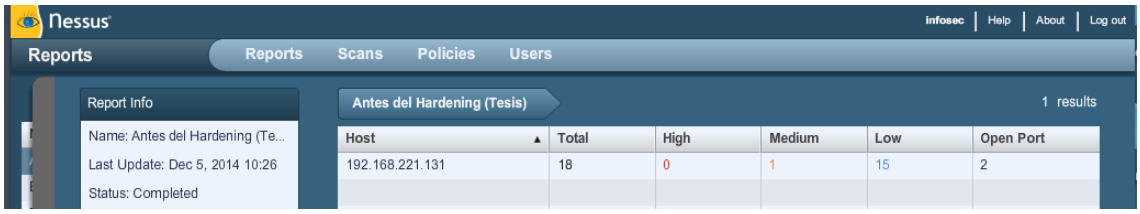


Figura C.4. Ejecución de la herramienta Nessus antes del *hardening*.

Posteriormente se realizó el mismo procedimiento en el sistema operativo Linux-Debian con las configuraciones de buenas prácticas de seguridad, dando como resultado 10 vulnerabilidades catalogadas como bajas, ver Figura C.5.

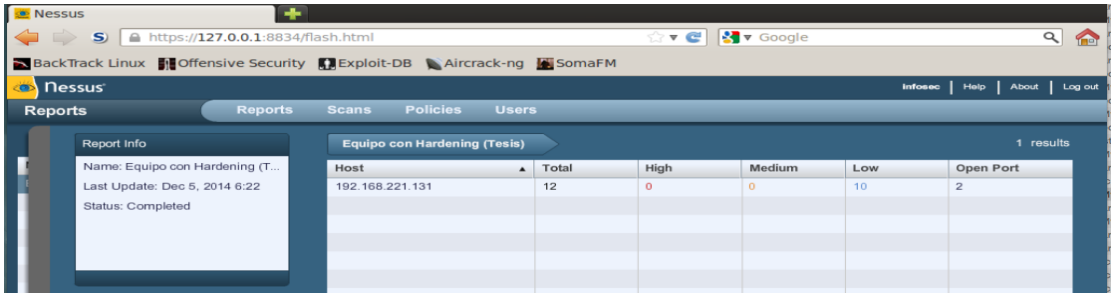


Figura C.5. Ejecución de la herramienta Nessus después del *hardening*.

Índice de Glosario

Glosario de términos y acrónimos

AAA Autenticación, Autorización y Registro; Conjunto de herramientas, procedimientos y protocolos que garantizan un tratamiento coherente de las tareas de autenticación, autorización y registro de la actividad de las entidades que tienen acceso a un sistema de información.

ACL Lista de Control de Acceso. Una ACL es una lista que especifica los permisos de los usuarios sobre un archivo, carpeta o recurso del sistema.

ACLs Listas de Control de Acceso, plural de ACL.

AESC ANSI se estableció originalmente como el Comité de Normas de Ingeniería estadounidense (AESC-*American Engineering Standards Committee*), supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos.

AMENAZAS Todo elemento o acción capaz de poner en riesgo la seguridad de la información. Las amenazas surgen a partir de la existencia de vulnerabilidades, es decir que una amenaza sólo puede existir si existe una vulnerabilidad que pueda ser aprovechada, e independientemente de que se comprometa o no la seguridad de un sistema de información.

APCERT Equipos de Respuesta a Incidentes del área Asia Pacífico.

BACKDOORS Es un código malicioso que se introduce en un sistema de cómputo de manera encubierta aparentando ser inofensivo. Una vez que se ejecuta en el sistema víctima, se establece una "puerta trasera", la cual puede ser empleada por un intruso para realizar acciones que comprometan aún más la seguridad del sistema.

BCP	Plan de Continuidad del Negocio, el cual define las acciones que una organización debe implementar para recuperar y restaurar sus funciones críticas de forma parcial o totalmente interrumpidas por algún desastre o acción intencional o no intencional.
BIOS	Acrónimo de (*Binary Input Output System*), es el responsable de permitir el arranque del sistema operativo. Para lo cual analiza los dispositivos de entrada y salida, verifica el estado de la memoria RAM y los configura para que el sistema operativo los pueda usar.
BITÁCORAS	Son registros del sistema que almacenan información detallada de determinados eventos.
BOOTLOADER	Es un programa que actúa como un gestor de inicio o arranque el cual tiene la responsabilidad de iniciar la ejecución del sistema operativo. Algunos gestores de inicio son LILO, GNU GRUB y NTLDR.
BOTS	Es un código malicioso que le permite a un intruso tomar el control de un sistema de cómputo. Por lo general, los bots son parte de redes de máquinas infectadas, conocidas como "botnets", las cuales pueden recibir y ejecutar determinadas instrucciones al mismo tiempo y de forma coordinada.
BSI	Es una empresa de normas empresariales que ayuda a las organizaciones a lograr mejores resultados, mediante la transformación de normas de mejores prácticas en hábitos de excelencia.
BUFFER OVERFLOW	El desbordamiento de memoria, ocurre cuando un programa informático excede el uso de cantidad de memoria asignado por el sistema operativo, y comienza a escribir código en los bloques de memoria contiguos.
CC	*Common Criteria* (Criterios Comunes), norma para certificar los procesos de implementación, especificación, desarrollo y evaluación en productos de seguridad informática, actualmente CC versión 2.2 están contemplados en la norma ISO-15408:2005.

CERT	El término CERT es un acrónimo de las palabras en inglés *Computer Emergency Response Team* (Equipo de Respuesta a Incidentes Informáticos) que define a una organización responsable del proceso de Respuesta a Incidentes de Seguridad Informática, el cual comprende las fases de: preparación, identificación, contención, erradicación y recuperación de todos aquellos sistemas informáticos pertenecientes a su comunidad objetivo.
CISCO	(Cisco Systems, Inc.) Compañía global con sede en San José, California (EE.UU.). Principalmente dedicada a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones
COADYUVAR	Contribuir o ayudar en la realización de una actividad enfocada a cumplir con objetivo en común.
COBIT	Es modelo para auditar la gestión y control de los sistemas de información y tecnología, orientado a todos los sectores de una organización, es decir, administradores IT, usuarios y auditores involucrados en el proceso de negocio.
CONTITUENCY	Comunidad perteneciente a un CERT/CSIRT, de la cual es responsable y ofrece sus servicios de seguridad proactivos y reactivos.
CRON	Es una utilería de sistemas UNIX/LINUX que sirve para definir tareas de ejecución automática en un momento del tiempo definido por el usuario. Es usado normalmente para la ejecución programada de tareas administrativas tales como la realización de respaldos de información, sin embargo, los crons pueden ser usados para ejecutar prácticamente cualquier comando o *script* en el sistema.
CUPS	*Common Unix Printing System*: es un sistema de impresión modular para sistemas operativos de tipo UNIX/LINUX que permite que un equipo actúe como servidor de impresión. Un equipo que ejecuta CUPS actúa como un servidor que puede aceptar tareas de impresión desde otros equipos clientes, los procesa y los envía al servidor de impresión apropiado.
CYMRU	Equipo de Respuesta a Incidentes enfocado a la investigación de temas relacionados con la seguridad informática en Internet, esta labor la realiza sin fines de lucro con el único fin

	de hacer del Internet un lugar más seguro. Team Cymru ayuda a las organizaciones a identificar y erradicar los problemas de seguridad informática en sus redes de datos.
DAÑO COLATERAL	Se produce como consecuencia de una acción encaminada a un objetivo concreto en el cual se ven afectadas terceras partes.
DCS	Un Sistema de Control Distribuido o SCD, más conocido por sus siglas en inglés DCS (*Distributed Control System*), es un sistema de control aplicado a procesos industriales complejos en las grandes industrias como petroquímicas, papeleras, metalúrgicas, generación de energía, plantas de tratamiento de aguas, industria farmacéutica, etc.
DENIAL OF SERVICE	Es un ataque de denegación de servicio que afecta primordialmente la disponibilidad de la información, un ejemplo de este tipo de ataques, son aquellas actividades malintencionadas en la red que tengan como propósito saturar los recursos de un servidor Web impidiéndole responder de manera adecuada a las peticiones que los clientes Web realizan hacia el servidor.
DFSG	El *Debian Free Software Guidelines*, fue diseñada inicialmente como un conjunto de criterios para definir lo que es software libre, la cual fue adoptada posteriormente por la comunidad de software libre como base para la definición del *Open Source*.
DMZ	Se refiere a una Zona Desmilitarizada o red perimetral, la cual es una red local que se ubica entre la red interna de una organización y una red externa.
DNS	El Sistema de Nombres de Dominio, un sistema para asignar nombres a equipos y servicios de red que se organizan en una jerarquía de dominios. La asignación de nombres DNS se utiliza en las redes TCP/IP, para localizar equipos y servicios con nombres descriptivos.
DOS	DOS es un sistema operativo modular que consiste de múltiples componentes con funciones especiales cada uno. Existen múltiples versiones de DOS, el más conocido es el MS-DOS de Microsoft, sin embargo también existe el PC-DOS (de IBM), DR-DOS, el FreeDOS y el QDOS.

DRP	El Plan de Recuperación de Desastres es la estrategia que se seguirá para restablecer los servicios de TI (Hardware y Software) después de haber sufrido una afectación por una catástrofe natural, epidemiológica, daño premeditado o ataque de cualquier tipo el cual atente contra la continuidad del negocio.
EAL	Evaluación de Nivel de Aseguramiento de un producto o sistema, sirve para asignar una calificación numérica después de la finalización de la evaluación de criterios comunes.
EGC	Grupo de CERTs gubernamentales en Europa, se encarga del desarrollo de medidas para el tratamiento de incidentes de seguridad de la red, así como facilitar el intercambio de información y tecnología relacionada con incidentes de seguridad y de vulnerabilidades y amenazas de código malicioso, así como fomentar la formación de CSIRTs gubernamentales entre los países europeos.
ENGINEERING TECHNOLOGIES	Categoría de tecnologías de ingeniería de ISO
ESTEGANOGRAFÍA	Es la aplicación de técnicas que permiten ocultar mensajes u objetos, dentro de otros, llamados portadores, de modo que no se perciba su existencia. Es decir, procura ocultar mensajes dentro de otros objetos y de esta forma establecer un canal encubierto de comunicación, de modo que el propio acto de la comunicación pase inadvertido para observadores que tienen acceso a al canal.
ETHERNET	También conocido como estándar IEEE 802.3) es un estándar de transmisión de datos para redes de área local.
EXIM4	Servicio presente en sistemas Linux usado para enviar correos, ya sea desde consola o bien utilizando una interface gráfica.
EXPLOITS	Es un programa informático malicioso que trata de forzar alguna deficiencia o vulnerabilidad en los sistemas.

FIFOS FIFO es un acrónimo de *First In, First Out* (primero en entrar primero en salir) que es una abstracción en relación con las formas de organización y manipulación de los datos en relación con el tiempo y las prioridades.

FINGERPRINTING Es una técnica que consiste en analizar las huellas que deja un sistema operativo en sus conexiones de red. Está basada en los tiempos de respuesta al enviar y recibir ciertos paquetes de datos a través del protocolo TCP/IP.

FIRST Es el Foro Internacional de Equipos de Respuesta a Incidentes de Seguridad, su principal función es coordinar los esfuerzos de cada uno de los equipos de respuesta a incidentes en pro de hacer del espacio de Internet un lugar más seguro.

FTP El Protocolo de Transferencia de Archivos, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (*Transmission Control Protocol*), basado en la arquitectura cliente-servidor.

GID Es un identificador de tipo entero, presente en sistemas UNIX/LINUX, que permite organizar a los usuarios por grupos, de esta forma la asignación de permisos para cada usuario se simplifica.

GPL La licencia GPL o General Public License, está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

GRSECURITY Es una serie de parches de seguridad para el kernel de Linux, los cuales le brindan al sistema operativo la capacidad de defenderse en contra de diversas amenazas de seguridad.

GRUB Es un programa que instala un gestor de arranque en el registro MBR lo cual permite insertar instrucciones específicas en el MBR que carga un entorno de comandos o menú de GRUB para así poder iniciar el sistema operativo que el usuario elija.

HACKEAR	Se refiere a la acción de comprometer la seguridad de un sistema informático mediante la implementación de las fases de: reconocimiento o escaneo del sistema, explotación de vulnerabilidades, mantenimiento del acceso y cubrir las huellas.
HARDENING	Es el proceso de asegurar un sistema mediante la reducción de vulnerabilidades en el mismo, es decir se trata de hacer más difícil el trabajo de los hackers.
HASH	Es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud.
ICMP	El Protocolo de Mensajes de Control y Error de Internet, se usa para manejar mensajes de error y de control necesarios para los sistemas de la red, informando con ellos a la fuente original para que evite o corrija el problema detectado.
IDS	El Sistema de Detección de Intrusiones, hace referencia a un mecanismo que sigilosamente escucha el tráfico en la red para detectar actividades anormales o sospechosas y de este modo, reducir el riesgo de intrusión. Existen IDS a nivel de host y a nivel de red.
IEC	La Comisión Electrotécnica Internacional, es una organización de normalización que se especializa en los campos: eléctrico, electrónico y tecnologías relacionadas.
IEEE	El Instituto de Ingeniería Eléctrica y Electrónica; es una asociación mundial de técnicos e ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas
IPS	Dispositivos dedicados a la prevención de intrusiones a partir de la identificación y bloqueo de patrones específicos de ataque con base en una amplia gama de firmas y algunas detecciones basadas en métodos relacionados con el comportamiento.
IRC	El *Internet Relay Chat*, es un Protocolo que sirve para mantener conversaciones en tiempo real con otros usuarios

	utilizando un programa llamado cliente, para conectarse con un servidor IRC, que a su vez, se vincula con otros servidores IRC.
ISAC	*Information Sharing and Analysis Centre* - Centro de Intercambio y Análisis de Información.
ISO	Es una Organización Internacional para la Estandarización (ISO), es decir se trata de una federación de alcance mundial integrada por cuerpos de estandarización nacionales de 153 países, uno por cada país. Su objetivo es promover el desarrollo de la estandarización para facilitar el intercambio de servicios, bienes y para promover la cooperación en aspectos de carácter: intelectual, científico, tecnológico y económico.
ISO/IEC	Es un conjunto de estándares desarrollados en conjunto por ISO (*International Organization for Standardization*) e IEC (*International Electrotechnical Commission*).
ITIL	La *Information Technology Infrastructure Library*, es una Biblioteca de Infraestructura de Tecnologías de Información, la cual es un marco de trabajo de las mejores prácticas destinadas a facilitar la entrega de servicios de tecnologías de la información (TI) de alta calidad y eficientar las operaciones de TI.
ITSEC	La *Information Technology Security Evaluation Criteria*, es un esfuerzo conjunto entre los miembros de la Unión Europea (UE) para desarrollar un criterio de evaluación de la seguridad estandarizado para la Unión Europea. La ITSEC concede a los productos que se evalúan satisfactoriamente niveles de seguridad de E1 (el más bajo) al E6 (el más alto).
ITU	La *International Telecommunications Union*, es una Unión internacional de telecomunicaciones, encargada de regular las telecomunicaciones entre las distintas administraciones y empresas operadoras.
JUNIPER NETWORKS	Es una Compañía multinacional de la industria de las telecomunicaciones dedicada a ofrecer soluciones redes y seguridad.

JURE	Situación que está reconocida formalmente.
KERNEL	Es un software que constituye una parte fundamental del sistema operativo, se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo). El kernel o núcleo de Linux se puede definir como el corazón de este sistema operativo, es el encargado de que el software y el hardware de una computadora puedan trabajar juntos.
KEY	Se refiere a la clave o frase de la contraseña para el cifrado.
LAN	LAN son las siglas de *Local Area Network*, Red de área local. Una LAN es una red de datos que conecta a los equipos de cómputo en un área relativamente pequeña y predeterminada como una habitación o un edificio.
LIVE-CD	Es un sistema operativo almacenado en un medio extraíble, tradicionalmente un CD o un DVD (de ahí su nombre), que puede ejecutarse desde éste sin necesidad de ser instalado en el disco duro de la computadora.
LOGIN	Es el proceso mediante el cual se controla el acceso de un usuario a un sistema informático.
LOGS	Son registros del sistema que almacenan información detallada de determinados eventos.
MAC	La Media Access Control, es una dirección física con un identificador de 48 bits, que corresponde de forma única a una tarjeta o dispositivo de red. La MAC permite las transmisiones de datos entre equipos de cómputo en la red, debido a que cada equipo de cómputo es reconocido mediante esa dirección MAC de forma inequívoca.
MAIN	Se trata de la función de entrada, y debe existir siempre, será la que tome el control cuando se ejecute un programa en C.
MALWARE	El Malware es la abreviatura de “Malicious software”, término que engloba a todo tipo de programa o código informático malicioso, que tiene la función de dañar un sistema o causar un mal funcionamiento.

MD5 Es un algoritmo que tiene como finalidad calcular la huella digital de un archivo y de esta forma comprobar la integridad de los archivos. El algoritmo de MD5 genera un número único de 128 bits a partir del análisis y suma de los caracteres de una cadena binaria (función no reversible), de esta forma si un sólo bit del archivo cambia, su suma MD5 cambiará totalmente.

NFS El Network File System, es un protocolo que permite acceso remoto a un sistema de archivos a través de la red. De esta forma posibilita que distintos sistemas conectados a una misma red accedan a archivos de forma remota como si éstos estuvieran almacenados de forma local.

NIDS Sistema de detección de intrusos en una Red. Detecta anomalías que indiquen un riesgo potencial, tales como ataques de denegación de servicio, escaneos de puertos o intentos de entrar a un equipo de cómputo, esto se realiza analizando el tráfico que circula en la red en tiempo real.

NIS *El Servicio de Información de Red*, es el nombre de un protocolo de servicios de directorios cliente-servidor desarrollado por Sun Microsystems para el envío de datos de configuración en sistemas distribuidos tales como nombres de usuarios y hosts entre computadoras sobre una red.

NIST El Instituto Nacional de Normas y Tecnología, es una agencia de la administración de tecnología del departamento de comercio de los Estados Unidos. La misión de este instituto es promover la innovación y la competencia industrial en Estados Unidos mediante avances en metrología, normas y tecnología de forma que mejoren la estabilidad económica y la calidad de vida.

NO-BREAKS Es un aparato que regula el voltaje, además ante la falta del suministro eléctrico, provee un periodo de algunos minutos con alimentación eléctrica a todos los aparatos contactados a él. En el caso de las computadoras les permite a los usuarios tomar precauciones con el fin de respaldar y guardar toda aquella información considerada como importante.

NSA La Agencia de Seguridad Nacional, es una agencia de inteligencia del Gobierno de los EE.UU que se encarga de todos los aspectos relacionados con la seguridad de la

	información de su país.
NSP-SECURITY FORUM	Lista de correo de respuesta a incidentes de voluntarios, que coordina la interacción entre los ISP y los NSP en tiempo casi real
NTP	El Network Time Protocol, es usado para sincronizar los relojes de hosts y routers en Internet con una única referencia horaria.
OPTICAL NETWORKING	Es un medio de comunicación que utiliza señales codificadas en luz para transmitir información entre los diferentes nodos de una red de telecomunicaciones.
PAM	El PAM (Pluggable Authentication Module) es un mecanismo que proporciona una interfaz entre las aplicaciones de usuario y diferentes métodos de autenticación, lo cual le permite al administrador del sistema establecer una política de autenticación sin la necesidad de recompilar los programas de autenticación existentes en el sistema.
PC	Personal Computer - Computadora personal o equipos de cómputo.
PEN-TESTING	Es el método de evaluar la seguridad de los equipos y las redes de comunicación, de esta forma se tendrá un mejor panorama de la eficiencia de los mecanismos de protección a perimetral y a nivel de host.
PGP	El sistema Pretty Good Privacy, es un sistema de criptografía híbrido que usa una combinación de funciones tomadas de la criptografía de clave pública y de la criptografía simétrica. PGP ofrece las siguientes disponibilidades: firmas digitales y verificación de la integridad de los mensajes, cifrado de archivos, generación de claves públicas o privadas, administración de claves, certificación de claves, etc.
PHISHING	Consiste en la suplantación de identidad de sitios Web, con la intención de engañar y estafar a usuarios legítimos de una empresa u organización para que proporcionen datos confidenciales como usuarios, contraseñas, números de tarjetas de crédito, etc.

PORTMAP	Es un demonio de asignación de puertos dinámico para servicios RPC, tales como NIS y NFS. Es importante limitar qué redes o hosts tengan acceso al servicio portmap puesto que éste no posee autenticación incorporada.
PP	Conjunto de requerimientos de seguridad válidos para una categoría de TOE, independiente de su implementación, que satisface necesidades específicas de usuarios" (el TOE – Target Of Evaluation es el objeto de evaluación, es decir, el sistema evaluado).
RACE CONDITIONS	Una condición de carrera se da principalmente cuando varios procesos acceden al mismo tiempo y cambian el estado de un recurso compartido (por ejemplo una variable), obteniendo de esta forma un valor no esperado de ese recurso, lo que puede ser aprovechada por exploits locales para vulnerar los sistemas.
RAM	La Random Access Memory (Memoria de Acceso Aleatorio). Este tipo de memoria es de tipo volátil, es decir, que pierde sus datos cuando deja de recibir energía, otra característica de este tipo de memoria es que se puede acceder aleatoriamente; es decir, se puede acceder a cualquier byte de memoria sin acceder a los bytes precedentes por esta razón también se le conoce como memoria de acceso directo.
ROOT	Es el nombre convencional de la cuenta de usuarios en sistemas UNIX/LINUX que posee permisos totales para realizar cualquier modificación en el sistema de archivos del sistema. El root es también llamado superusuario y por lo regular esta cuenta es administrada por el administrador del sistema.
ROOTKITs	Es un código malicioso comúnmente empleado por los intrusos para ocultar su actividad maliciosa en el sistema vulnerado ante la vista del administrador y de esta forma pasar desapercibido el mayor tiempo posible. Existen rootkits a nivel de sistema operativo y a nivel de kernel, siendo éstos últimos los más complicados de detectar y erradicar.
ROUTER	Se trata de un producto de hardware que permite interconectar computadoras que funcionan en el marco de una red.

RPC	La Llamada a Procedimiento Remoto, es un protocolo que le permite a un programa ejecutarse de forma remota sin tener que preocuparse por las comunicaciones entre ambas partes.
SCADA	SCADA es el acrónimo de Supervisory Control And Data Acquisition. Los sistemas SCADA utilizan las computadoras y tecnologías de comunicación para automatizar el monitoreo y control de procesos industriales. Estos sistemas son partes integrales de la mayoría de los ambientes industriales complejos y geográficamente dispersos ya que pueden recoger la información de una gran cantidad de fuentes y presentarla a un operador de forma muy entendible lo cual permitirá facilitar la toma de decisiones.
SCANERS	Son equipos de cómputo que se encuentran realizando escaneos de vulnerabilidades a otros equipos en una subred.
SCRIPT	Son programas usualmente pequeños o simples, para realizar tareas muy específicas, este tipo de programas no requieren ser compilados para generar un archivo ejecutable.
SGSI	El Sistema de Gestión de Seguridad de la Información, para garantizar que la seguridad de la información es gestionada correctamente, se debe hacer uso de un proceso sistemático, documentado y conocido por toda la organización, desde un enfoque de riesgo del negocio, este proceso es el que constituye un SGSI y es el concepto central sobre el que se construye el ISO/IEC 27001.
SHA-1	El Secure Hash Algorithm (Algoritmo seguro de Hash), es un algoritmo criptográfico de Hash, que genera un valor Hash de 160 bits a partir de una secuencia de longitud arbitraria. SHA-1 es considerado lo suficientemente seguro para la mayoría de las aplicaciones prácticas, sin embargo si el usuario lo desea puede utilizar versiones más robustas como: SHA-256, SHA-384 y SHA-512 que generan valores Hash de 256, 384 y 512 bits respectivamente.
SHA-2	Algoritmo hash criptográfico desarrollados por el NIST (Instituto Nacional de Normalización y Tecnología de los Estados Unidos) que sustituyó a los antiguos algoritmos hash SHA-1.

SHELL	Se emplea para referirse a aquellos programas que proveen una interfaz de usuario para acceder a los servicios del sistema operativo. Éstos pueden ser gráficos o de texto simple, dependiendo del tipo de interfaz que empleen. Los Shells están diseñados para facilitar la forma en que se invocan o ejecutan los distintos programas disponibles en la computadora.
SHELLs	Plural de Shell.
SMART CARD	Tarjeta Inteligente que posee un circuito integrado el cual tiene la capacidad de almacenar información y procesarla.
SNIFFEO DEL TRÁFICO	Es la acción de emplear un programa sniffer para capturar el tráfico de red en una subred específica.
SNIFFER	Un sniffer es un programa que trabaja en conjunto con la tarjeta de interfaz de red, para capturar indiscriminadamente todo el tráfico que esté dentro del umbral de audición del sistema de escucha. Y no sólo el tráfico que vaya dirigido a una tarjeta de red, sino a la dirección de difusión de la red, es decir a cualquier dirección IP.
SPOOF	Es una técnica de suplantación de identidad en la red generalmente con usos maliciosos o de investigación. El funcionamiento es el siguiente: un intruso simula la identidad de otra máquina de la red para conseguir acceso a recursos de un tercer sistema que ha establecido algún tipo de confianza basada en el nombre o la dirección IP de un host en particular.
SSH	El Secure Shell, es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente-servidor que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, el protocolo SSH sí cifra la sesión de conexión.
STACK	Una pila es un método de estructuración de datos usando la forma LIFO (último en entrar, primero en salir), que permite almacenar y recuperar datos.

SUID Set User ID - Establecer ID de usuario, permisos de acceso que pueden asignarse a archivos o directorios en un sistema operativo basado en Unix.

TCP El Protocolo de Control de Transmisión, es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. Su principal característica es que TCP es un protocolo orientado a conexión, es decir, que permite que dos máquinas que están comunicadas controlen el estado de la transmisión.

TCSEC Estándar que establece los requisitos básicos para evaluar la eficacia de la seguridad informática y de los controles integrados en un sistema informático. El TCSEC se utilizó para evaluar, clasificar y seleccionar los sistemas informáticos que realizan actividades de procesamiento, almacenamiento y recuperación de la información sensible o información clasificada.

TELNET El protocolo Telnet es un protocolo de Internet estándar que permite conectar terminales y aplicaciones en Internet. El protocolo proporciona reglas básicas (no ofrece cifrado de los datos transmitidos) que permiten vincular a un cliente con un intérprete de comandos del lado del servidor. El protocolo Telnet se aplica en una conexión TCP para enviar datos en formato ASCII codificados en 8 bits.

TF-CSIRT Es un foro de Equipos de Respuesta a Incidentes de seguridad, sólo que en este a diferencia de FIRST, únicamente es para países del continente Europeo.

TOE El Target of Evaluation, producto o sistema que es sujeto de una evaluación de seguridad de la información.

TRIAGE El triage es un término en inglés ampliamente utilizado en los Equipos de Respuesta a Incidentes de Seguridad (CERTs) para referirse a las acciones de categorización, correlación, priorización y asignación de incidentes de seguridad.

TROYANOS Este tipo de malware se hace pasar por un programa inofensivo para vulnerar la seguridad de los sistemas, posteriormente se activa de manera discreta y sin consentimiento del usuario, cumpliendo así su propósito

	nocivo para el que fue creado.
TTY	La tty es una terminal que permite en sistemas GNU/Linux acceder al sistema operativo fuera de su entorno gráfico (X-Windows). En general se disponen de hasta seis terminales de este tipo.
UID	Es un identificador de tipo entero, único para cada usuario presente en sistemas UNIX/LINUX.
USB	Universal Serial Bus, es un puerto que sirve para conectar periféricos a una computadora, una característica importante es que permite a los dispositivos trabajar a velocidades mayores, en promedio a unos 12 Mbps, lo cual significa que es de 3 a 5 veces más rápido que un dispositivo de puerto paralelo y de 20 a 40 veces más rápido que un dispositivo de puerto serial.
VMWARE	Es un software que sirve para virtualizar un sistema operativo dentro de otro. La palabra "virtualizar" quiere decir, ejecutar en un ambiente simulado otro sistema operativo.
VULNERABILIDAD	Hace referencia a una debilidad en un sistema permitiendo a un atacante violar la confidencialidad, integridad, disponibilidad, control de acceso y consistencia del sistema o de sus datos y aplicaciones
WANK	Gusano WANK (Worms Against Nuclear Killers), infectó equipos, obtuvo privilegios adecuados para sustituir el desplegado del sistema por un anuncio que llenaba la pantalla con la palabra WANK, atacó una red compartida entre la NASA y el departamento de Energía de Estados Unidos.
WIFI	También conocido como Wi-Fi, es una marca comercial de Wi-Fi Alliance una organización que adopta y certifica los equipos que cumplen con los estándares 802.11 de las redes inalámbricas de área local.
WRAPPERS	Un Wrapper es un programa que controla el acceso a un segundo programa. El Wrapper literalmente cubre la identidad de este segundo programa, obteniendo con esto un nivel más alto de seguridad.

Referencias

Artículos y libros.

[1] Álvarez Torres M. G. (2006). Manual para Elaborar Manuales de Políticas y Procedimientos. Primera edición. México. Editorial Panorama.

[2] Caralli R. A., Allen J. H., Curtis P.D, White D. W., Young·L. R. (2010). CERT Resilience Management Model versión 1.0. Estados Unidos de America. Carnegie Mellon University.

[3] Carnegie Mellon University. (2008). Creating and Managing Computer Security Incident Handling Teams (CSIRTs) Organizational Models. Estados Unidos de America.

[4] Cichonski P., Millar T., Grance T., Scarfone K. (2012). Computer Security Incident Handling Guide. Estados Unidos de America. NIST.

[5] Donald C. Latham. (1985). Department of Defense Trusted Computer System Evaluation Criteria. Department of Defense Standard.

[6] ISO/IEC 17799:2005. (2005) Information technology – Security techniques – Code of practice for information security management. (2a ed.).

[7] ISO/IEC 27002:2013. (2013). Information technology – Security techniques – Code of practice forinformation security controls. (2a ed.).

[8] ITSEC. (1991). Information Technology Security Evaluation Criteria, Provisional Harmonised Criteria, Document COM (90) 314. Alemania. Luxemburgo.

[9] Jason Andress. (2011). The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice. Elsevier.

[10] Kimberly Graves. (2007) CEH Official Certified Ethical Hacker Review Guide. EC-Council. SYBEX.

[11] López Barrientos M. J., Quezada Reyes C. (2006). Fundamentos de seguridad informática. Primera Edición. Facultad de Ingeniería UNAM.

[12] National Institute of Standards and Technology. (1995). An Introduction to Computer Security: The NIST Handbook, Special Publication 800-12. USA.

[13] Proyecto Amparo. (2011). Manual básico de Gestión de Incidentes de Seguridad Informática. México. LACNIC.

[14] West-Brown M. J., Stikvoort D., Kossakowski K. P., Killcrece G., Ruefle R., Zajicek M. (2003). Handbook for Computer Security Incident Response Teams (CSIRTs). (2a ed.). Estados Unidos de America. Carnegie Mellon University.

Páginas de internet.

[15] Aguilar Roselló V. J., Domínguez Jover R., Sánchez Medina I. (2002). Implementación de redes privadas virtuales (VPN) utilizando el protocolo IPSEC, consultado el 04-02-2013, disponible en http://www.redes-linux.com/manuales/vpn/trabajo.pdf.

[16] Alvy (2009). Una breve historia de Linux, consultado el 06 de septiembre de 2012, disponible en http://noticias.lainformacion.com/ciencia-y-

tecnologia/ciencias-informaticas/una-breve-historia-de-linux_Zj2ge6h8QOZgWu0h8eyqM6/.

[17] ANSI. (n.d). Historical Overview, consultado el 20-04-2014 , disponible en http://www.ansi.org/about_ansi/introduction/history.aspx.

[18] Benedict Torvalds L. (1992). Linux'sHistory, consultado el 07 de septiembre de 2012, disponible en http://www.cs.cmu.edu/~awb/linux.history.html.

[19] Borbón Sanabria J. S. (2011). ¿Qué son las buenas prácticas? Revista de seguridad de la UNAM, consultado el 21 de enero de 2013, disponible en http://revista.seguridad.unam.mx/numero-11/buenas-pr%C3%A1cticas-est%C3%A1ndares-y-normas.

[20] Borbón Sanabria J. S. (2011). Buenas prácticas, estándares y normas, consultado el 22-01-2013, disponible en http://revista.seguridad.unam.mx/numero-11/buenas-prácticas-estándares-y-normas.

[21] BSI. (n.d.). Beneficios del ISO/IEC 27001, consultado el 21-04-2014, disponible en http://www.bsigroup.es/certificacion-y-auditoria/Sistemas-de-gestion/estandares-esquemas/Seguridad-de-la-Informacion-ISOIEC27001/.

[22] CERT RED IRIS. (n.d.). Autenticación, consultado el 15-09-2013, disponible en http://www.rediris.es/CERT/doc/unixsec/node14.html.

[23] Cristian Borghello. (2009). Detección de Intrusos en Tiempo Real, consultado el 14-12-2013, disponible en http://www.segu-info.com.ar/proteccion/deteccion.htm.

[24] Debian Policy Manual. (2012). Archive areas, consultado el 25-09-2012, disponible en http://www.debian.org/doc/debian-policy/ch-archive.html.

[25] Debian. (2012). Desarrolladores a nivel mundial, consultado el 27-09-2012, disponible en http://www.debian.org/devel/developers.loc.

[26] Distribuciones de Linux (2012), consultado el 26-09-2012, disponible en http://www.argentinawarez.com/linux-y-gnu/2407598-distribuciones-linux.html.

[27] El Proceso de Arranque en Linux (2010), consultado el 14-12-2013, disponible en http://www.linuxyyo.es/el-proceso-de-arranque-en-linux.

[28] Equipo de Documentación de Debian (2004). Una breve historia de Debian, consultado el 24 de septiembre de 2012, disponible en http://www.debian.org/doc/manuals/project-history/ch-intro.es.html.

[29] Fenzi K. y Wreski D. (2004). Linux Security HOWTO, What Are You Trying to Protect?, consultado el 03 de septiembre de 2012, disponible en http://www.linuxdoc.org/HOWTO/Security-HOWTO/x82.html#AEN85.

[30] Fernández J. y Peña S. (2005). Manual de Seguridad de Debian, Capitulo 2. consultado el 12 de octubre de 2012, disponible en http://www.linux-cd.com.ar/manuales/debian-seguridad/ch2.es.html.

[31] FIRST. (2015). Información de CERT-MX, consultado el 20-04-2015, disponible en http://www.first.org/members/teams/cert-mx.

[32] FIRST. (2015). Members around the world, consultado el 30-01-2015, disponible en http://www.first.org/members/map.

[33] FISRT History (2015), consultado el 30-01-2015, disponible en http://www.first.org/about/history.

[34] Fogel K., Martilotti R., Ayuso A., Puerta Peña J. M., Bonilla Polo P. A., Vadillo Batista A., Urbano García F., López Espínola C., Casbas Jiménez E., Colina H. (2007). Producir Software de Código Abierto – Libre vs Abierto,

consultado el 23-09-2012, disponible en http://producingoss.com/es/producingoss.pdf.

[35] Garbee B. (2004). Publicaciones de Debian, consultado el 24 de septiembre de 2012, disponible en http://www.debian.org/doc/manuals/project-history/project-history.es.txt.

[36] Glosario y abreviaturas (2011), consultado el 11-04-2011, disponible en https://www.ccn-cert.cni.es/publico/seriesCCN-STIC/series/400-Guias_Generales/401-Glosario_y_abreviaturas/401/index.html.

[37] GNU (2012). La definición de software libre, consultado el 21 de septiembre de 2012, disponible en http://www.gnu.org/philosophy/free-sw.es.html.

[38] GNU, Proyecto GNU (2008). Logo de GNU, consultado el 21 de septiembre de 2012, disponible en http://lacasadetux.wordpress.com/2008/09/06/historia-del-proyecto-gnu/.

[39] Hertzog R. y Roland M. (2013). The Debian Administrator's Handbook – Lifecycle of a Release, consultado el 25-09-2013, disponible el http://debian-handbook.info/download/stable/debian-handbook.pdf.

[40] Historia del proyecto Debian (2003), consultado el 23-09-2012, disponible en http://debianitas.net/doc/minicomos/Historia%20y%20FAQ%20sobre%20Debian/pdf/historia-julio2003.pdf.

[41] IEC (n.d), consultado el 19-04-2014, disponible en http://www.unit.org.uy/miembros/iec.php.

[42] INTENCO (CERT). (n.d.). Modelo PDCA, consultado el 18-02-2012, disponible. en http://cert.inteco.es/Formacion/SGSI/Conceptos_Basicos/Modelos_PDCA_SGSI/.

[43] ISO (n.d.), consultado el 21-04-2012, disponible en http://www.unit.org.uy/miembros/iso.php.

[44] ISO 27000 (n.d), consultado el 19-04-2012, disponible en http://www.iso27000.es/download/doc_iso27000_all.pdf.

[45] ISO in figures (2011), consultado el 18-04-2012, disponible en http://www.iso.org/iso/iso-in-figures_2011.pdf.

[46] ISO members (2012), consultado el 18-04-2012, disponible en http://www.iso.org/iso/about/iso_members.htm.

[47] ISO/IEC 17799 (n.d), IT news, consultado el 22-04-2014, disponible en http://itnews.ec/marco/000130.aspx.

[48] ISO/IEC JTC1 (n.d.), consultado el 10-04-2014, disponible en http://www.iso27000.es/otros.html#section4a.

[49] Jeimy J. Cano, Ph.D. (2011). La Gerencia de la Seguridad de la Información, consultado el 04-02-2013, disponible en http://www.isaca.org/Journal/Past-Issues/2011/Volume-5/Pages/JOnline-La-Gerencia-de-la-Seguridad-de-la-Informacion-Evolucion-y-Retos-Emergentes.aspx.

[50] Las 14 mejores distribuciones de Linux (2010), consultado el 06 de septiembre de 2012, de http://mediterrahosting.com/wp-content/uploads/2012/07/linux-logo-full.jpg.

[51] Línea de tiempo de Debian (n.d.), consultado el 26 de septiembre de 2014, disponible el http://linuxgnublog.org/imgblog/relevante.png.

[52] Línea de tiempo de Debian GNU/Linux (n.d.), consultado el 21 de septiembre de 2012, disponible en http://linuxgnublog.org/imgblog/lintem.png.

[53] Línea de tiempo de las distribuciones de Linux (n.d.), consultado el 18 de septiembre de 2012, disponible en http://robertbriones.files.wordpress.com/2007/07/linea-tiempo-linux.png.

[54] Logotipo de Debian (2012), consultado el 24 de septiembre de 2012, disponible en http://www.debian.org/Pics/openlogo-50.png.

[55] Logotipo de ISO (n.d.), consultado el 11-11-2014, disponible en http://www.bitcompany.biz/wordpress/wp-content/uploads/2011/07/iso-iec-logos.jpg.

[56] Martínez E. (1999). Estándares de comunicaciones, consultada el 20-04-2011, disponible en http://www.eveliux.com/mx/estandares-de-telecomunicaciones.php.

[57] Microsoft TechNet. (2008). Comprender la defensa en seguridad consultado el 27-10-2012, disponible en http://www.microsoft.com/latam/technet/articulos/articulos_seguridad/2008/junio/sm0608.mspx.

[58] NOM Certification. (2012), consultado el 21-04-2014, disponible en http://www.e-switch.com/Portals/0/NOM.png.

[59] Normalización en México (n.d), consultado el 21-04-2012, disponible en http://www.mitecnologico.com/Main/Normalizaci%F3nEnM%E9xico.

[60] Normas y Estándares (n.d), consultado el 21-04-2012, disponible en http://es.wikitel.info/wiki/Normas_y_estandares.

[61] Open SourceIniative (n.d.), consultado el 23 de septiembre de 2012, disponible en http://opensource.org/files/osi_symbol.png.

[62] Opentia (2007). Estudio sobre Estándares informáticos tipos y caracterizaciones, consultado el 21-04-2012, disponible en http://people.ffii.org/~abarrio/estandares/OPENTIA-estudioTiposDeEstandares-20070129.pdf.

[63] PDCA Cycle (n.d), consultado el 22-04-2014, disponible en http://www.hci.com.au/hcisite3/toolkit/pdcacycl.htm.

[64] Políticas de Seguridad (n.d), consultado el 03-04-2012, disponible en http://auditoriasistemas.com/auditoria-informatica/politicas-de-seguridad/.

[65] Raymond E.S. (2003). The Art of Unix Programming, Origins and History of Unix, 1969-1995, consultada el 05-09-2014, disponible en http://www.faqs.org/docs/artu/ch02s01.html.

[66] Real Academia Española. (n.d.). Estándar, consultado el 21 de abril de 2012, disponible en http://buscon.rae.es/.

[67] Roberto. (2012). Círculo de Deming PDCA, consultado el 22-04-2014, disponible en http://www.empresasandalucia.com/circulo-de-deming-pdca-plan-do-check-act-planificar-hacer-verificar-actuar/.

[68] Seguridad UNAM (n.d.). Tutorial TCP-Wrappers, consultado el 18 de diciembre de 2013, disponible en http://www.seguridad.unam.mx/descarga.dsc?arch=511.

[69] Southwest Florida GNU/Linux Users Group. (2010). Consultado el 06 de septiembre de 2012, disponible en http://files.meetup.com/95734/Unix_history-simple.svg.

[70] Spam (2011), consultado el 06-04-2012, disponible en http://milcris.multiply.com/journal/item/218.

[71] Stallman R. (2012). La primera comunidad que comparte software, consultado el 21 de septiembre de 2012, disponible en http://www.gnu.org/gnu/thegnuproject.es.html.

[72] Turner S. (2013). Consideraciones para comprar respaldo para un servidor, consultado el 14-12-2013, disponible en http://blog.iweb.com/es/2013/08/comprar-respaldo-para-servidor/2298.html.

[73] UNIX Tutorial Home page (n.d.), consultado el 05 de septiembre de 2012, disponible en http://people.rit.edu/~dqn8613/409/Unix/index.xhtml.

[74] Villalón Huerta A. (2002). Los bits SUID, SGID y sticky, consultado el 22 de diciembre de 2013, disponible en http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/SEGUNIX/unixsec-2.1-html/node56.html.

[75] Wikipedia (2014). Normatividad Mexicana, consultado el 21-04-2014, disponible en http://es.wikipedia.org/wiki/Normatividad_Mexicana.

[76] Wikipedia. (n.d.). TCP Wrapper, consultado el 10 de diciembre de 2013, disponible en http://es.cyclopaedia.net/wiki/TCP-Wrapper.

[77] BSI (2013). Moving from ISO/IEC 27001:2005 to ISO/IEC 27001:2013, Mapas y tablas, consultado el 20 de diciembre de 2013, disponible en http://www.bsigroup.com/LocalFiles/en-GB/iso-iec-27001/resources/BSI-ISO27001-transition-guide-UK-EN-pdf.pdf.