Universidad de las Ciencias Informáticas

Facultad 1

“Framework de seguridad para Nova Servidores”

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Dairon Vera Rodriguez

Tutor: Mtr. Yoandy Pérez Villazón

Ing. Gustavo Quezada Arévalo

La Habana, junio 2016

“Año 58 de la Revolución”

Declaración de Autoría

Declaro ser el autor del presente Trabajo de Diploma y se reconoce a la Universidad de las Ciencias Informáticas, los derechos patrimoniales del mismo con carácter exclusivo. Para que así conste firmo la presente declaración jurada de autoría en La Habana a los días _____ del mes de _____ del año _____.

Dairon Vera Rodriguez

_____

Firma del autor

Mtr. Yoandy Pérez Villazón

_____

Firma del tutor

Ing. Gustavo Quezada Arévalo

_____

Firma del tutor

"Crezcan como buenos revolucionarios. Estudien mucho para poder dominar la técnica que permite dominar la naturaleza. Acuérdense que la Revolución es lo importante y que cada uno de nosotros, solo, no vale nada. Sobre todo, sean siempre capaces de sentir en lo más hondo cualquier injusticia cometida contra cualquiera en cualquier parte del mundo. Es la cualidad más linda de un revolucionario."

Ernesto Che Guevara

Dedicatoria

Este trabajo va dedicado especialmente a mi hermano menor, aunque no esta físicamente entre nosotros, está en el corazón de cada uno de quien lo conoció, principalmente en el mío.

A mi abuelo Paquito, por ser tan bueno y enseñarme cosas buenas en la vida.

A mi mamá, mis dos papá, mi abuela, mi tía Marcia y a mis primos David y Daniar, por darme apoyo en todo momento en mi vida.

A mis hermanos por ser parte de mí.

Agradecimientos

Agradezco a la Revolución por permitir de estudiar en esta maravillosa escuela.

A mis padres por darme lo mejor que se le puede dar a un hijo, amor, cariño y felicidad.

A mis amigos de la vieja escuela como les digo, Orlay, Marisol, Roberto y Yorgenis.

A mis amigos de esta nueva estapa de mi vida, Alexander Chávez, Lázaro Placencia y José.

A los que una vez compartieron conmigo y no se encuentran presente en estos momentos y a los que están.

A mi Hormiguita Retozona, como le digo, por ser una persona que me ayudó en momentos difíciles en la carrera.

A mis familiares en general, porque nunca me dieron la espalda y siempre tenia un pedazo de su casa disponible para mi.

Mi madrina y padrino por ser mis padres aquí en la Habana y que me cuidaron como siempre lo han hecho.

A todos en general. Son muchas personas a la cual agradezco en el corazón y todos ellos saben que sí lo tengo presente.

A todos muchas gracias por compartir parte de mi vida.

Resumen

Un *framework* de seguridad es un mecanismo fundamental para todos los sistemas operativos, sin su utilización son más vulnerables y pueden ser víctimas de ataques, ya sea interno o externo. En la actualidad los *framework* de seguridad son necesarios para controlar las acciones que una aplicación o servicio pueda realizar en el sistema. Estos mecanismos persiguen un objetivo, elevar el nivel de seguridad de los sistemas informáticos. La distribución cubana de GNU/Linux Nova hereda el *framework* de seguridad *AppArmor* que se encuentra en la distribución GNU/Linux Ubuntu. La no realización de cambios y adición de nuevas configuraciones al *framework*, constituye un factor de riesgo en la seguridad que ofrece esta distribución a los usuarios. La mayor preocupación está dada en el proceso de su creación al realizarse variaciones con respecto al *software* original y la incorporación de aplicaciones propias implementadas por el equipo de desarrollo.

La presente investigación persigue como objetivo definir perfiles asociados a las nuevas aplicaciones que son incorporadas al sistema operativo y la actualización de los que ya existen con determinadas políticas que garanticen elevar el nivel de seguridad. Para lograr las metas trazadas se compararon un conjunto de *frameworks*, permitiendo determinar *AppArmor* como el más adecuado para la distribución cubana. Se realizó experimento para validar la propuesta realizada. Se aplicó la técnica de IADOV a un grupo de especialistas de Administración de Servicios Telemáticos, como mecanismo para medir el nivel de satisfacción con las mejoras realizadas, el índice de satisfacción grupal fue de 0.72.

Palabras clave: distribución, *framework*, Nova, seguridad.

Índice

Índice de Figuras

Índice de tablas

Introducción

La constante evolución de las tecnologías integradas a la interconexión, ha convertido a las Tecnologías de la Información y la Comunicación (TIC) en un elemento estratégico para el crecimiento y transformación de las organizaciones. Las TICs proporcionan un perfeccionamiento de los productos de *software* que hacen posible la informatización de los procesos en las diferentes esferas de la sociedad.

La seguridad es uno de los mecanismos que necesita una mejora continua. Constituye uno de los factores más vulnerables que tienen los sistemas informáticos, que es cada vez más difícil de controlar. Dichos mecanismos se clasifican en diferentes tipos: preventivos, consisten en prevenir la ocurrencia de un ataque informático; detectores, tienen como objetivo detectar todo aquello que pueda ser una amenaza para los bienes; correctivos, se encargan de reparar los errores o daños causados una vez que se haya cometido un ataque, modifican el estado del sistema de modo que vuelva a su estado original y adecuado y los disuasivos, se encargan de desalentar a los perpetradores de que cometan su ataque para minimizar los daños que puedan tener los bienes (1).

La seguridad constituye uno de los temas principales en la rama de la informática a nivel mundial. Existen cientos de sitios en Internet que ofrecen información, herramientas y métodos para vulnerar sistemas informáticos. Antes los problemas de seguridad estaban dado por los virus, actualmente son otros los tipos de ataques que preocupan a los usuarios como son: los *phishing*[1], *spamming*[2], *pharming*[3], *hacker*[4], *cracker*[5], *adware*[6], *spyware*[7] (2). Hay que tener en cuenta que la seguridad informática es un proceso

1 ***Phishing**:* es utilizado para referirse a uno de los métodos más utilizados por delincuentes cibernéticos para estafar y obtener información confidencial de forma fraudulenta.

2 ***Spamming**:* es el hecho de enviar mensajes electrónicos (**spam**) no solicitados y en cantidades masivas.

3 ***Pharming**:* es la explotación de una vulnerabilidad en el *software* de los servidores DNS (Domain Name System).

4 ***Hacker**:*Persona con grandes conocimientos de informática que se dedica a acceder ilegalmente a sistemas informáticos ajenos y a manipularlos.

5 ***Cracker**:* El término **cracker** fue acuñado por primera vez hacia 1985 por hackers que se defendían de la utilización inapropiada por periodistas del término hacker.

6 **Adware** es el nombre que se da a los programas diseñados para mostrar publicidad en su computadora, redirigir sus solicitudes de búsqueda a sitios web de publicidad.

7 **Spyware** es *software* o *hardware* instalado en una computadora, generalmente sin el conocimiento del usuario, que recoge información de dicho usuario para más tarde enviarla por Internet a un servidor.

dinámico, suele siempre estar encaminado a la actualización permanente de mecanismos, métodos, técnicas y procedimientos que ayudan a contrarrestar los ataques o amenazas informáticas que cada día aparecen en *internet* (3).

Cuba, en aras de ganar en soberanía tecnológica y seguridad, así como garantizar la informatización de todas las esferas de la sociedad, inició su incursión en el año 2002 en el desarrollo del Proyecto Futuro, nombre dado por el Comandante en Jefe Fidel Castro Ruz a la Universidad de las Ciencias Informáticas (UCI). En la actualidad la Universidad cuenta con un total de 14 centros de desarrollo de *software,* dentro de ellos se encuentra el Centro de Software Libre (CESOL), cuyo objetivo es el desarrollo de la distribución cubana de GNU/Linux Nova. Luego del recién finalizado Primer Taller Nacional de Informatización y Ciberseguridad, la distribución cubana de GNU/Linux Nova fue elegida para ser desplegada en los Órganos y Organismos de la Administración Central del Estado (OACE). La selección de Nova para la migración del país responde a las necesidades de la informatización segura de la sociedad cubana.

El *framework* de seguridad se encarga de controlar las acciones que el *software* instalado puede realizar en el sistema (4). Aunque estos mecanismos persiguen un objetivo común, proteger al sistema operativo, divergen en cómo lograrlo, haciendo evidentes diferencias en aspectos como el entorno al que están dirigidas, el factor usabilidad y la curva de aprendizaje.

Actualmente la distribución cubana de GNU/Linux Nova, hereda el *framework* de seguridad *AppArmor* que se encuentra disponible en la distribución GNU/Linux Ubuntu, al cual no se le realizan cambios, ni se le añaden nuevas configuraciones. Este último elemento es un factor de riesgo en la seguridad que ofrece Nova a los usuarios, puesto que en el proceso de creación de la distribución se realizan variaciones con respecto al *software* original disponible en la distribución GNU/Linux Ubuntu y se añaden aplicaciones propias desarrolladas por el equipo de la distribución nacional.

Además de los elementos enunciados anteriormente, no se tiene certeza de que el *framework* de seguridad heredado de la distribución GNU/Linux Ubuntu, sea el más adecuado para la distribución cubana de GNU/Linux Nova, surgiendo la necesidad de realizar un conjunto de mejoras que agrupen los

aspectos referentes a las acciones que puede realizar el *software* en el sistema operativo y que a su vez, tribute a la obtención de un mecanismo de seguridad más adecuado a la distribución cubana de GNU/Linux Nova, contribuyendo así a la soberanía tecnológica del país.

Tomando como punto de partida la problemática anteriormente descrita, se plantea como **problema de la investigación:** ¿Cómo aumentar los niveles de seguridad en la distribución cubana de GNU/Linux Nova?

Se plantea como **objeto de estudio:** los mecanismos de seguridad en los sistemas de código abierto.

Enmarcado en el **campo de acción:** los mecanismos de seguridad utilizados para controlar las acciones de las aplicaciones instaladas en la distribución cubana de GNU/Linux Nova.

La presente investigación tiene como **objetivo general**: implementar mejoras para el *framework* de seguridad de la distribución cubana de GNU/Linux Nova que permita aumentar los niveles de seguridad en los servicios que se despliegan en los OACE.

Para darle cumplimiento al objetivo general se definen los siguientes **objetivos específicos:**

- Identificar los conceptos asociados al dominio del problema.
- Analizar los *frameworks* de seguridad disponibles para sistemas de código abierto.
- Definir los perfiles de configuración para el *framework* de seguridad en la distribución cubana de GNU/Linux Nova.
- Evaluar el nivel de seguridad en la distribución cubana de GNU/Linux Nova a partir de la incorporación de los nuevos perfiles.

Para dar solución al problema planteado y dar respuesta a las preguntas científicas formuladas se proponen las siguientes **tareas de la investigación:**

1. Análisis de los conceptos asociados a los *frameworks* de seguridad.
2. Caracterización de los diferentes *frameworks* de seguridad disponibles para sistemas código abierto.
3. Selección del *framework* de seguridad más adecuado para distribución cubana de GNU/Linux

Nova.

4. Creación de perfiles de configuración para el *framework* de seguridad seleccionado a implantar en la distribución cubana de GNU/Linux Nova.
5. Evaluación del nivel de seguridad proporcionado a partir de la introducción de los nuevos perfiles de *AppArmor* en la distribución cubana de GNU/Linux Nova mediante un experimento práctico y de la aplicación de la técnica de IADOV.

Como **idea a defender** se tiene que la implementación de mejoras para el *framework* de seguridad de la distribución cubana de GNU/Linux Nova permitirá aumentar los niveles de seguridad en los servicios que se despliegan en los OACE.

Para el desarrollo de la investigación se utilizan los siguientes **métodos científicos:**

Métodos teóricos

Histórico-Lógico: la utilización de este método permite conocer la evolución que han tenido los procesos de los mecanismos de seguridad en la distribución cubana de GNU/Linux Nova, así como las causas que dieron paso a su surgimiento, las características comunes que se han mantenido con el transcurso del tiempo y las nuevas que han surgido.

Analítico-Sintético: se utiliza para dividir el problema en subproblemas, que faciliten el estudio del proceso de desarrollo de la distribución cubana de GNU/Linux Nova y luego sintetizarlo en una solución general acorde al problema propuesto.

Análisis bibliográfico: empleado para la revisión de la literatura necesaria durante el proceso de investigación sobre los procesos de mecanismos de seguridad.

Métodos empíricos:

Observación: utilizada para hacer un análisis y evaluación del comportamiento de los mecanismos de seguridad, con el objetivo de identificar elementos que aporten a la investigación.

Entrevistas: utilizada para obtener los servicios más utilizados por la OACE y para aplicar el método IADOV para conocer el grado de satisfacción de los expertos.

Justificación de la investigación

La distribución cubana no dispone de un mecanismo propio, previamente evaluado y analizado que permita conocer las acciones que puedan realizar las aplicaciones sobre el sistema operativo. El despliegue generalizado previsto para este sistema requiere incorporar mecanismos que posibiliten aumentar el nivel de seguridad en el sistema operativo. Estudios previos a esta investigación definen que el framework de seguridad es un factor clave para este propósito.

El aporte práctico de esta investigación es la definición del framework de seguridad más adecuado para GNU/Linux y de un conjunto de nuevos perfiles que permiten aumentar el nivel de seguridad de la distribución cubana de GNU/Linux Nova.

El presente documento está estructurado de la siguiente manera:

Capítulo 1: Fundamentación Teórica. Este capítulo abarca el estudio de los diferentes *frameworks* de seguridad, realizando un análisis de cada uno, en cuanto a características, funciones, ventajas y desventajas. Se establece una comparación entre dos de los *frameworks* estudiados, para seleccionar el más adecuado en la distribución cubana GNU/Linux Nova. Se analizan las herramientas y tecnologías a utilizar para el desarrollo del trabajo de investigación.

Capítulo 2: Perfiles para el *framework* de seguridad. En este capítulo *s*e analizan un conjunto de servicios que son los más utilizados por los distintos OACE. Se proponen además los nuevos perfiles a incluir al *framework* de seguridad, describiéndose también las funcionalidades de cada uno.

Capítulo 3: Pruebas de los perfiles de *AppArmor*. La propuesta realizada en el capítulo anterior propone un conjunto de perfiles para incrementar la seguridad en Nova servidores. A continuación se realiza pruebas a las propuestas, para ellos se realiza una comparación teniendo en cuenta la propuesta actual con respecto a lo presente antes de esta investigación. Se aplica además la técnica de IADOV para evaluar el nivel de satisfacción de especialistas en administración de servicios telemáticos.

La investigación cuenta también con recomendaciones, referencias bibliográficas, anexos y glosario de términos.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

Este capítulo abarca el estudio de los diferentes *frameworks* de seguridad, realizando un análisis de cada uno, en cuanto a características, funciones, ventajas y desventajas. Se establece una comparación entre dos de los *frameworks* estudiados para seleccionar el más adecuado para la distribución cubana GNU/Linux Nova. Se analizan las herramientas y tecnologías a utilizar para el desarrollo del trabajo en cuestión.

1.2 Conceptos generales asociados al dominio del problema

1.2.1 *Framework* (infraestructura, marco)

Un *framework* es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de *software* concretos con base a la cual otro proyecto de *software* puede ser organizado y desarrollado. Estos pueden incluir soporte de programas, bibliotecas, lenguajes interpretados, entre otras aplicaciones para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de *software* que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo que extiende o utiliza las aplicaciones del dominio (4).

1.2.2 Seguridad

La Organización Internacional de Normalización (*ISO por sus siglas en inglés*) hace referencia a que la seguridad consiste minimizar la vulnerabilidad de bienes y recursos. La seguridad en un sistema se basa en los mecanismos de protección que ese sistema proporciona. Estos mecanismos deben permitir controlar qué usuarios tienen acceso a los recursos del sistema y qué tipo de operaciones pueden realizar sobre esos recursos (6). Para controlar el acceso de los dominios a los recursos se utilizan las Listas de Control de Acceso (*ACL por sus siglas en inglés*) por cada recurso. La ACL especifica qué dominios tienen acceso a los recursos y qué operaciones asociadas al recurso pueden utilizar. El problema que plantea la ACL es su tamaño variable, ya que depende del número de dominios que tengan acceso al recurso y de las operaciones que pueda realizar cada uno de ellos (6).

1.2.3 Seguridad Informática

Se denomina Seguridad Informática al conjunto de métodos y herramientas destinados a proteger los bienes informáticos de una institución. La seguridad en la información tiene el objetivo de garantizar (7):

- **Confidencialidad:** la información o los activos informáticos son accedidos sólo por las personas autorizadas para hacerlo. La información confidencial debe estar disponible únicamente para un grupo de individuos pre-establecido. La transmisión y uso de información no autorizada debe restringirse. Por ejemplo, la confidencialidad de información garantiza que la información personal o financiera no esté al alcance de individuos no autorizados con propósitos malintencionados tales como: robo de identidad o fraude de crédito.
- **Integridad:** los activos o la información sólo pueden ser modificados por las personas autorizadas y de la forma autorizada. La información no se debe alterar de forma tal que la reproduzcan incompleta o incorrecta. Se debe restringir a los usuarios no autorizados de la capacidad de modificar o destruir información confidencial.
- **Disponibilidad:** los activos informáticos son accedidos por las personas autorizadas en el momento requerido. La información debe estar accesible a usuarios autorizados, en cualquier momento, es decir, cuando se necesite. Es decir, cuando se necesite. La disponibilidad garantiza que la información pueda obtenerse con una frecuencia y puntualidad acordadas. Suele medirse en términos de porcentajes y se acepta de manera formal en los Acuerdos de Nivel de Servicio (SLA por sus siglas en inglés) usados por los proveedores de servicios de red y los clientes corporativos.

1.2.4 Técnicas de protección al sistema operativo Linux

Los perjuicios económicos provocados por los ataques mediante todo tipo de técnicas, ya sean virus, *exploits*[8], usuarios malintencionados, entre otros, son tales que la seguridad en cualquier instalación

8 ***Exploits****:* es un fragmento de *software*, fragmento de datos o secuencia de comandos y/o acciones, utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo.

informática se ha convertido en un tema prioritario. Para cualquier administrador de un sistema informático la seguridad es crítica. Se puede decir que un sistema informático es seguro si está 'fuera de peligro', pero es prácticamente imposible conseguir llegar a esta situación. No existe un sistema completamente seguro. Todos los sistemas tienen sus vulnerabilidades, bien del propio sistema operativo o de las aplicaciones ejecutadas sobre él. Todo lo que se puede hacer es aumentar la dificultad para que el sistema quede comprometido. Existen algunas técnicas de protección al sistema Linux entre ellas (6):

- **Control de acceso al sistema:** permite que el *Basic Input/Output System* (BIOS) pueda determinar la configuración de la máquina y proporcionar un sistema básico de control sobre los dispositivos. Este *software* se suele almacenar en Memoria de Solo Lectura *(*ROM por sus siglas en inglés) o FLASH[9] que permite la actualización de la BIOS sin cambiar el chip.
- **Protección de los archivos:** la protección forma parte del contenido del nodo-i del archivo, que es la estructura que almacena información relativa a dicho archivo, como son el propietario, grupo, fechas de creación-modificación-acceso, tamaño, entre otros.
- **Contraseñas y encriptación:** las contraseñas de Linux utilizan el algoritmo de cifrado de IBM Estándar de Encriptación de Datos (DES por sus siglas en inglés). Estas contraseñas son de longitud fija de ochos caracteres.
- **Seguridad de la cuenta de administración:** esta técnica tiene su propia cuenta de usuario y se conecta siempre como usuario normal. Cuando tenga que realizar tareas de administración puede pasar a modo de superusuario. Es importante que la contraseña del administrador del sistema no viaje por la red en texto plano. Debe hacerlo de forma cifrada. Evitar el uso de órdenes -r- (*rlogin, rcp, rsh, entre otros*) y utilizar las órdenes *ssh* o *scp*, en las que las contraseñas viajan encriptados.

1.3 Estudios y análisis de *frameworks* de seguridad existentes

Los *framework*s de seguridad utilizan Control de Acceso Discrecional (DAC por sus siglas en inglés),

9 ***Memoria Flash**: es una manera desarrollada de la memoria EEPROM que permite que múltiples posiciones de memoria sean escritas o borradas en una misma operación de programación mediante impulsos eléctricos, frente a las anteriores que sólo permite escribir o borrar una única celda cada vez EPROM.*

Control de Acceso Obligatorio (MAC por sus siglas en inglés) y ACL. El control de acceso constituye una poderosa herramienta para proteger la entrada a una web completa o sólo a ciertos directorios concretos e incluso a ficheros o programas individuales (10). Linux por defecto implementa DAC, permitiendo que cada sujeto (usuario, grupo, roles) pueda determinar quién puede leer y escribir sus ficheros y otros objetos (aplicaciones y procesos) que le pertenecen. Los procesos heredan todos los privilegios del usuario que los ejecuta garantizando el acceso, sin restricciones, a cualquier recurso al que pueda acceder el usuario (11). DAC es una forma de acceso a recursos, basada en los propietarios y grupos a los que pertenece un objeto, es discrecional, ya que un sujeto puede transmitir sus permisos a otro sujeto (12). MAC se basa en políticas mediante la cual el administrador establece una política en la que se especifica la forma en la que los sujetos pueden acceder a los recursos del sistema operativo. Existen un conjunto de reglas de autorización (políticas), las cuales determinan si una operación sobre un objeto realizada por un sujeto está o no permitida basándose en los atributos de ambos (12). La ACL es una forma de determinar los permisos de accesos, es usado para el filtrado de tráfico. La misma cuenta de dos tipos: estándar donde se tiene que especificar una dirección de origen y la extendida donde aparece el protocolo y una dirección de origen y de destino (13).

A continuación se hace una descripción de algunos *framework*s de seguridad utilizados en Linux, dejando plasmadas algunas de las características que tienen en común, rasgos que la diferencian y principales funcionalidades.

1.3.1 *OpenBSD Cryptographic Framework* (OCF)

La estructura criptográfica de *OpenBSD* es una capa de servicios de virtualización de máquina asíncrona en el interior del núcleo, que proporciona un acceso uniforme a *hardware* de tarjetas aceleradoras criptográficas. El *OpenBSD* implementa dos *Application Programming Interface* (*API* por sus siglas en inglés) para el uso de otros subsistemas del *kernel*, uno para los consumidores (otros subsistemas del *kernel*) y otro para los productores (controladores de dispositivo criptotarjeta). El *OpenBSD* admite clases de algoritmos simétricos, entre ellos: *Data Encryption Standard* (DES por sus siglas en inglés), *Advanced Encryption Standard (*AES por sus siglas en inglés), dentro de los asimétricos se encuentra el algoritmo

Rivest, Shamir and Adleman (RSA por sus siglas en inglés).

En los algoritmos simétricos las operaciones se basan en el concepto de la sesión, ya que este tipo de algoritmo se utiliza normalmente para el procesamiento de datos y aprovechar el almacenamiento en caché de sesión y características disponibles en muchos aceleradores. Los algoritmos asimétricos se implementan como operaciones individuales, ya que no se realiza ninguna sesión de almacenamiento en caché (19).

1.3.2 SYSINIT

SYSINIT es utilizado para una llamada de clasificación y expedición de mecanismo genérico. Permite subsistemas del *kernel* de *FreeBSD* para ser reordenadas y añadir, eliminar y reemplazarlas en tiempo de enlace del núcleo o uno de sus módulos que se carga sin tener que editar un enrutamiento de inicialización estática ordenada y recopilar el *kernel*. Este sistema también permite a los módulos del núcleo, actualmente llamados *Kernel Linker Dynamic* (KLD por sus siglas en inglés), establecida a tal fin por separado, vinculado entre sí, e inicializan en el arranque y carga incluso más tarde, mientras que el sistema ya está en marcha. Esto se logra utilizando el "enlazador *kernel*" y "conjuntos de enlazador" (20).

SYSINIT se basa en la capacidad del enlazador para tomar datos estáticos declarados en múltiples localizaciones a través de la fuente y el grupo trata de un programa junto como un solo trozo contiguo de datos. Esta técnica enlazador se llama "conjunto de enlazador". SYSINIT enlazador utiliza dos conjuntos de mantener conjuntos de datos que contienen el fin de llamada, la función de cada consumidor y un puntero a los datos que se pasan a la función.

1.3.3 TrustedBSD MAC

El *framework* MAC *TrustedBSD* permite que los módulos del núcleo pueda extender la política de seguridad del sistema operativo, así como proporcionar la funcionalidad de la infraestructura requerida por muchos módulos de control de acceso. Si hay varias políticas se cargan al mismo tiempo, en la estructura MAC será útil, por alguna definición de utilidad componer los resultados de las políticas (21).

La estructura MAC contiene una serie de elementos del núcleo:

- *Framework* de las interfaces de administración.
- Concurrencia y sincronización primitivas.
- Política de la inscripción.
- Etiqueta de seguridad extensible para los objetos del núcleo.
- Operadores de composición de política a punto de entrada.
- Primitivas de gestión de etiquetas.
- Punto de entrada de la API invocada por los servicios del núcleo.
- Punto de entrada API para los módulos de políticas.
- Puntos de entrada de implementaciones.
- Llamada al sistema múltiples *mac_syscall().*
- Diversas políticas de seguridad implementadas como módulos de políticas MAC.

El *framework* MAC TrustedBSD puede gestionarse directamente a través de *sysctl*, carga sintonizable y las llamadas al sistema.

1.3.4 *Systrace*

La herramienta *Systrace* ayuda a analizar el rendimiento de la aplicación mediante la captura y visualización de los tiempos de ejecución de los procesos de aplicaciones y otros procesos del sistema *Android*. La herramienta combina datos desde el núcleo de *Android* como el planificador de la CPU, la actividad del disco y hebras de la aplicación para generar un informe HTML que muestra una visión global de los procesos del sistema de un dispositivo *Android* durante un período de tiempo determinado (22). Si una aplicación ejecuta código intensivo de la CPU en más de un tema por "*tiempo suficiente*", el núcleo desplazará al hilo por otro núcleo diferente (23).

1.3.5 *Smack*

Smack es un módulo de seguridad del *kernel* Linux que protege la interacción de datos y el proceso de la manipulación maliciosa utilizando un conjunto de control de acceso personalizado obligatorio, como su principal objetivo el diseño. Se ha fusionado oficialmente desde el lanzamiento de Linux 2.6.25, y fue el principal mecanismo de control de acceso para el sistema operativo móvil *MeeGo*. También se utiliza para aplicaciones HTML5 web *sandbox* en la arquitectura *Tizen*, en las soluciones comerciales de *Wind River* Linux para el desarrollo de dispositivos embebidos, y en productos de la TV *Philips* digital. Smack consta de tres componentes:

- Un módulo del *kernel* que se implementa como un módulo de seguridad de Linux. Funciona mejor con los sistemas de archivos que soportan atributos extendidos.
- Una secuencia de comandos de inicio que se asegura de que los archivos de dispositivos tienen la Smack correcta atributos y carga la configuración Smack.
- Un conjunto de parches para el paquete Core *Utilities* GNU para que sea consciente de los atributos de archivo Smack extendidas. También se crea un conjunto de parches similares a *Busybox*. Smack no requiere soporte de espacio de usuario.

1.3.6 *SELinux*

SELinux es una arquitectura de seguridad integrada en el *kernel* usando los módulos de seguridad Linux. Este es un proyecto de la Agencia de Seguridad Nacional (NSA por sus siglas en inglés) de los Estados Unidos (E.E.U.U.) y de la comunidad *SELinux*. La integración de *SELinux* en *Red Hat Enterprise* Linux fue un esfuerzo conjunto entre al NSA y *Red Hat.* Proporciona un sistema flexible incorporado en el *kernel.* Bajo el Linux estándar se utiliza el DAC, en el que un proceso o aplicación ejecutándose como un usuario tiene los permisos y de los objetos, archivos, *sockets* y otros procesos. Al ejecutar un *kernel SELinux* MAC se protege al sistema de aplicaciones maliciosas o dañadas que pueden perjudicar o destruir el sistema. *SELinux* define el acceso y los derechos de transición de cada usuario, aplicación, proceso y archivo en el sistema. *SELinux* implementa la interacción de estos sujetos y objetos usando una política de seguridad

que especifica cuán estricta o indulgente una instalación de GNU/Linux dada debería de ser (25).

Es un módulo de seguridad para el *kernel* de Linux que proporciona un mecanismo de políticas para el control de acceso, logrando un mayor nivel de abstracción para los usuarios. Trabaja a través de contextos de seguridad, controles de acceso impositivos u obligatorios y en base a roles; ofreciendo un control más granular del acceso a los recursos del sistema por parte de los objetos y los sujetos. Este no reemplaza el modelo tradicional de seguridad de los sistemas tipo Unix, por el contrario, sirve de complemento de este en los puntos que la seguridad tradicional no es suficiente. La seguridad está dividida por niveles de usuarios, grupos, permisos, ACL y atributos extendidos de acceso; donde un usuario puede ejecutar un conjunto de aplicaciones a las que tienen acceso y estas son ejecutadas con los niveles de acceso que posee el usuario. Por tal motivo uno de los mecanismos de control de acceso utilizados por este *framework* es el DAC (10).

SELinux introduce además un sistema de control tipo MAC basado en contextos, donde se indica cuándo un objeto o sujeto puede acceder a otro objeto. El administrador debe definir los permisos de cada usuario que acceda a algunas aplicaciones o a cualquier objeto del sistema. Para evitar que esta operación sea tediosa se definen Controles de Acceso por Roles (*RBAC por sus siglas en inglés*) (11).

La idea central de RBAC es que los permisos están asociados a los roles y los usuarios se asignan a roles apropiados. Es por esto que este mecanismo de control de acceso es compatible con tres principios de seguridad fundamentales: privilegio, separación de funciones y abstracción de datos (12). Los roles se crean para las diversas funciones de trabajo en una organización y los usuarios se asignan roles en función de sus responsabilidades. Los usuarios pueden fácilmente ser reasignados de un papel a otro, pueden conceder nuevos permisos a las nuevas aplicaciones y sistemas que se incorporan, además de que estos pueden ser revocados de los roles según sea necesario (13).

Cuando un usuario o una aplicación intenta acceder a un archivo, el servidor de aplicaciones de políticas verifica la Caché de Vector de Acceso (*AVC por su siglas en inglés*), donde se registran todos los privilegios de acceso. Si no se puede tomar una decisión basada en los datos del AVC, la petición continúa al servidor de seguridad, el cual busca el contexto de seguridad de la aplicación y del archivo, luego los

permisos otorgados o negados son notificados a través de un mensaje en los registros del sistema (12). Ver Figura 2.

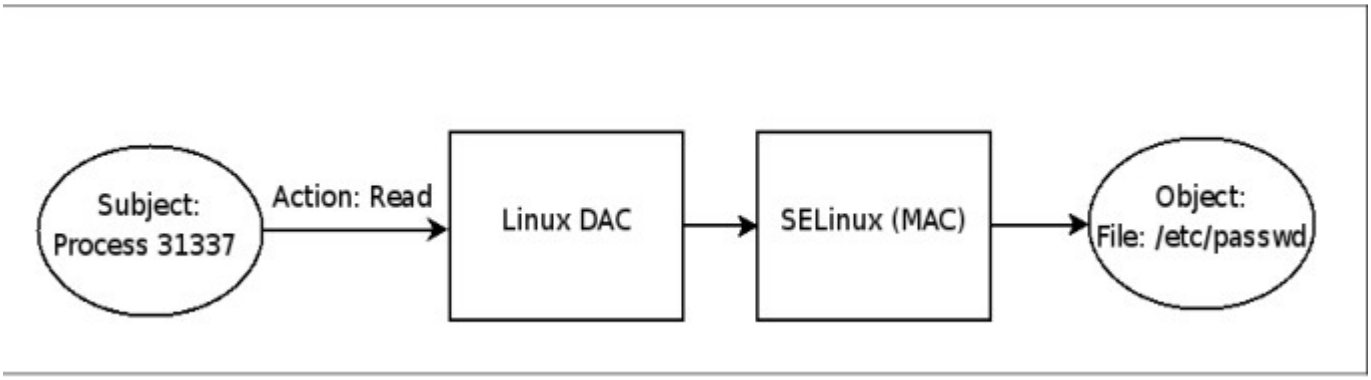

Figura 1: Funcionamiento de *SELinux*.
Fuente: (Sanchez, 2008)

En *SELinux* existen dos formas de configurar bajo *Red Hat Enterprise* Linux, usando la herramienta de configuración de nivel de seguridad (*system-config-securitylevel*), o manualmente editando el archivo de configuración *(/etc/sysconfig/selinux)*. El archivo (*/etc/sysconfig/selinux)* es el archivo de configuración principal para habilitar o inhabilitar *SELinux*, así como también para configurar cuál política debe imponer en el sistema y cómo hacerlo. Este archivo contiene un enlace simbólico al archivo de configuración real, (*/etc/selinux/config)* (25).

1.3.7 *AppArmor*

AppArmor es una implementación relativamente simple de MAC sobre Linux. Está incluida en la distribución SUSE, pero también puede utilizarse en otras distribuciones. Su función es la de aplicar restricciones de acceso sobre las aplicaciones que, a diferencia del DAC, no se basa en privilegios de los usuarios sino en políticas del sistema, implementadas como perfiles. En estos perfiles, se especifican explícitamente las acciones que la aplicación puede realizar dentro del sistema. Dado que las restricciones sobre las aplicaciones se basan en las políticas del sistema y no en los permisos de los usuarios, la

aplicación se encontrará restringida independientemente del usuario que la esté ejecutando, incluso si fuera el superusuario "root". Por otro lado, el cumplimiento de las políticas no es responsabilidad de una aplicación o servicio, sino del propio *kernel* de Linux, (como en el caso del Firewall "iptables") lo cual dificulta las posibilidades de saltear los mecanismos de control de acceso (16).

Está diseñado para ofrecer servicios de seguridad de fácil manejo a las aplicaciones de los servidores y las estaciones de trabajo. Es un sistema de control de acceso que permite especificar qué archivos puede leer, escribir y ejecutar en cada programa. Protege las aplicaciones mediante la imposición de buenas prácticas de comportamiento, de forma que se puedan prevenir estos ataques, incluso si intentan explotar vulnerabilidades anteriormente desconocidas (16).

AppArmor está incluido en el *kernel* de Linux, diseñado para proveer protección al sistema operativo y que su utilización no fuera compleja. Previene de los efectos nocivos de los ataques internos o externos, aplicaciones maliciosas y virus (17). Complementa el modelo tradicional de DAC de Unix proporcionando el MAC, centrado en las aplicaciones construidas con Módulos de Seguridad de Linux (LSM por sus siglas en inglés).

Esta herramienta permite aplicar políticas de seguridad asignando un perfil a cada aplicación para marcarle límites. El sistema monitoriza la forma en que los procesos acceden a los ficheros, distinguiendo entre accesos de lectura y escritura, así como el uso del privilegio de administración. Establece una capa intermedia entre una aplicación (web, cliente-servidor o *script*) y los recursos del sistema operativo que la aplicación demanda durante su ejecución (*sockets*[10], acceso a ficheros y librerías).

Para cada aplicación que se desee proteger, se debe generar un perfil en el que se establezca el rango de acción de la aplicación en dos aspectos:

***Capabilities* POSIX:** es un conjunto de capacidades que se le pueden otorgar a un proceso para interactuar con el sistema operativo. Estas capacidades incluyen acciones, entre ellas: cambiar la hora del sistema, bloquear memoria, modificar la tabla de ruteo, abrir un *socket* en un puerto menor a 1024, cargar y descargar módulos del sistema, reiniciar el equipo. Al listar explícitamente las *Capabilities* POSIX que se

10 *Socket*: se utiliza en los sistemas operativos POSIX(**P**ortable **O**perating **S**ystem **I**nterface Unix) para comunicación entre procesos.

le otorgan a una aplicación, se restringe el rango de acción de la misma ante eventuales ataques que pudieran realizarse sobre la misma.

Control de acceso al sistema de archivos: se especifica una lista exhaustiva de todos los archivos y directorios a los que la aplicación o proceso tendrán acceso, indicando el tipo de acceso a cada uno de ellos. Esta restricción es independiente de los permisos que dichos archivos y directorios tengan en el sistema de archivos. De esta manera, si en el perfil de *Apparmor* para la utilidad "*syslog*[11]" se especifica que dicho binario tenga acceso de sólo lectura al archivo "*/etc/syslog.conf*", dicha utilidad no podrá escribir en el archivo aunque los permisos en el sistema de archivo se lo permitan.

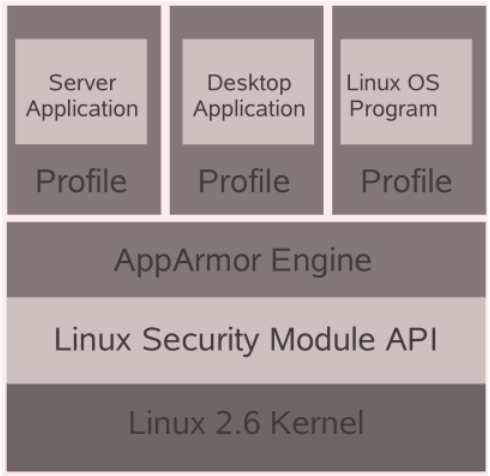

Figura 2: Funcionamiento de AppArmor.
Fuente: (Novell, 2007)

1.3.8 *Grsecurity*

Grsecurity es una amplia mejora de seguridad para el núcleo de Linux, que lo defiende contra una amplia

11 **syslog** es un estándar de facto para el envío de mensajes de registro en una red informática IP. Por **syslog** se conoce tanto al protocolo de red como a la aplicación o biblioteca que envía los mensajes de registro

gama de amenazas a la seguridad a través del control de acceso inteligente. Se ha desarrollado activamente y mantenido durante los últimos quince años (18). Ofrece protección contra ataques y otras amenazas avanzadas que compran los administradores en tiempo valioso, mientras corrige la vulnerabilidad hacen su camino a la distribución y producción de pruebas, esto es posible gracias al enfoque en la eliminación de clases enteras de errores y explotación de vectores, en lugar de la eliminación del *status-quo* de las vulnerabilidades individuales (18). *Grsecurity* limita sus cambios en el *kernel* de Linux en sí, por lo que es posible utilizarlo con cualquier distribución o dispositivo: integrado, servidor o escritorio. Entre sus controles de seguridad implementados en el sistema se destacan:

- Protección a nivel de funcionamiento del *kernel*.
- Prevención de la ejecución del código arbitrario.
- Control de ejecución de las tareas en el *stack.*
- Control de las actividades de los usuarios.
- Permisos de ejecución en determinadas áreas del sistema.
- Implementación de controles adicionales a la seguridad impuesta por *chroot*.
- Alarmas e intervenciones de seguridad que contienen el IP del que causa la alarma.
- Implementación de un control de acceso basado en roles.

1.4 Comparación de *frameworks* de seguridad analizados

En este epígrafe se realiza una comparación para determinar los *framework* más semejantes para ser utilizado en la distribución cubana de GNU/Linux Nova. Atendiendo a los siguientes criterios:

- **Sistema Operativo:** es parte de la base en que se desarrolla la investigación en cuestión, ya que se requiere Linux.
- **Función:** tipo de función que realiza en el *kernel* del sistema operativo al que está orientado los

distintos *frameworks.*

- **DAC:** se analiza si el *framework* posee o no este tipo de mecanismo.
- **Seguridad al sistema operativo Linux:** indica si el *framework* brinda servicios de protección al sistema operativo Linux.
- **MAC:** se evalúa si el *framework* analizado posee este tipo de mecanismo.
- **Ayuda disponible:** indica si el *framework* tiene sitios oficiales para una mejor búsqueda de información.

Tabla 1: Comparación de los *frameworks* de seguridad analizado.

Framework	**Sistema operativo**	**Función**	**DAC**	**Seguridad al sistema operativo Linux**	**MAC**	**Ayuda disponible**
OpenBSD Cryptografic	OpenBSD	Capa de servicios	no	no	si	si
SYSINIT *Framework*	FreeBSD	clasificación y expedición de mecanismo genérico	no	no	si	si
TrustedBSD MAC *Framework*	TrustedBSD	Extiende política de seguridad	no	no	si	no
Systrace	Android	Seguridad Android	no	no	no	no
Smack	Linux	Módulo de seguridad	si	si	si	no
SELinux	Linux	Módulo de seguridad	si	si	si	si
AppArmor	Linux	Servicio de seguridad	si	si	si	no
Grsecurity	Linux	Mejora de seguridad	si	si	si	si

Dada la descripción de los *frameworks* anteriormente analizados y de acuerdo a la comparación realizada se puede decir que *SELinux y AppArmor* son los *frameworks* adecuados para cumplir el objetivo general. No se analizan los restantes porque no brindan un nivel de seguridad apropiado para el objetivo de esta

investigación. Los *frameworks Smack* y *Grsecurity* poseen características similares a los seleccionados, pero debido a la falta de documentación y ayuda de éstos, no se tuvieron en cuenta.

1.5 Selección del *framework* a utilizar

Dada la comparación que se realizó en el epígrafe anterior se selecciona *AppArmor* y *SELinux* como *frameworks* de seguridad con las características adecuadas para ser utilizado en la distribución cubana de GNU/Linux Nova.

SELinux cuenta con mecanismos de seguridad implementados en el sistema que proporcionan soporte flexible para una amplia gama de políticas de seguridad. Además posee una menor vulnerabilidad a los ataques de escala de privilegios, contando con un poderoso mecanismo de control de acceso. Estos mecanismos garantizan la confidencialidad y la integridad de los datos, proporcionando la protección de los procesos. Además cuenta con una herramienta de la Línea de Comando (CLI por sus siglas en inglés) para escribir y/o modificar normas establecidas.

AppArmor protege la infraestructura de los atacantes que intentan encontrar y explotar las fallas de las aplicaciones. Define las políticas que marcan el alcance del acceso de una aplicación a los recursos del sistema, que ofrece las herramientas para proteger su infraestructura crítica, sin grandes inversiones en tiempo, recursos o formación, ofrece además prevención de intrusos en *host*, protegiendo al sistema operativo y las aplicaciones de los efectos de los ataques internos o externos y aplicaciones maliciosas. Se encuentra implementado en el *kernel* de Nova servidores 5.0, debido a que éste lo hereda de la distribución GNU/Linux Ubuntu.

Teniendo en cuenta que estos *frameworks* de seguridad cumplen con las características para ser utilizados en la distribución cubana de GNU/Linux Nova, en este epígrafe se hará una comparación para determinar cuál será utilizado en esta distribución. Para la selección del *framework* se utilizó el modelo de madurez *Qualification and Selection of Open Source* (QSOS), que es un método diseñado para calificar, seleccionar y comparar *software* libre/ código abierto de manera objetiva.

Licencia de uso/distribución: permite conocer si el producto es realmente de *software* libre o no,

para verificar se recomienda emplear el sitio de la Iniciativa para el Código Abierto[12], donde se enuncian todas aquellas licencias que son compatibles con el *software* libre.

Propietario: permite conocer si el producto es desarrollado por una persona o una institución.

Antigüedad: los productos más antiguos poseen generalmente mayor madurez.

Documentación: el grado de disponibilidad de la documentación debe ser tenido en cuenta. La existencia de foros de discusión, listas de correo, manuales de uso y desarrollo son elementos a considerar.

Soporte: existen varios niveles de soporte, los más populares son: soporte comunitario y soporte profesional. Se debe tener en cuenta que exista en adecuado soporte comunitario.

Facilidad de uso: permite la facilidad de utilizar un mecanismo. Se debe tener en cuenta que el más fácil de utilizar.

Curva de aprendizaje: permite saber el nivel de entendimiento del producto. Se recomienda que se menor.

En el grupo de características las funcionales serán evaluadas con una puntuación del 1-5, siendo las funcionalidades de 5 puntos las mejores. En la actualidad los *frameworks* de seguridad deben de cumplir con la siguiente actividad:

- Soporte flexible para una amplia gama de seguridad.
- Vulnerabilidades a escala de privilegios.
- Mecanismo de control de acceso.
- Implementación de MAC.
- Implementación de DAC.
- Implementación de RBAC.
- Herramienta de líneas de comandos.

12 https://opensource.org/licenses

- Prevención de intrusos en *host*.

Tabla 2: Comparación SELinux y AppArmor.

Frameworks	**SELinux**	**AppArmor**
Genéricos		
Licencia	GPL	GPL
Propietario	Agencia de Seguridad Nacional(E.E.U.U.)	Canonical
Antigüedad	16 años	14 años
Documentación	Si	Si
Soporte	Profesional	Comunitario
Facilidad de uso	Difícil	Fácil
Curva de aprendizaje	Alta	Baja
Funcionales		
Soporte flexible	5	3
Vulnerabilidades	3	4
MAC	5	3
DAC	5	5
RBAC	5	1
líneas de comandos	5	3
Prevención de intrusos	3	5
Recubrimiento funcional	31	24

Desde el punto de vista funcional el *framework* de seguridad *SELinux* es superior a *AppArmor atendiendo* a las funcionalidades definidas. Por otra parte, atendiendo los aspectos genéricos se puede determinar que *SELinux* es creado por la Agencia de Seguridad Nacional de los E.E.U.U. de América y que AppArmor es fácil de utilizar, posee una menor curva de aprendizaje, e implementa una correcta política de seguridad deseada, se propone como resultado del estudio del estado del arte, mantener *AppArmor* como *framework* a utilizar en la distribución cubana de GNU/Linux Nova.

1.6 Tecnologías y herramientas utilizadas

Las tecnologías y herramientas definidas para la mejora de los perfiles fue producto de un estudio realizado por el equipo de especialistas en migración de servicios telemáticos, pertenecientes al departamento Servicios Integrales de Migración de Asesoría y Soporte (SIMAYS) de CESOL. Teniendo en cuenta las características del entorno donde se utilizará la herramienta, se definió que las tecnologías a utilizar deben ser libres y multiplataforma.

1.6.1 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (*IDE por sus siglas en inglés*) es un programa compuesto por un conjunto de herramientas para que el programador las utilice. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, puede utilizarse para varios. El entorno de desarrollo es imprescindible en la producción de un *software*. *Geany* es un pequeño y ligero entorno de desarrollo integrado. Fue desarrollado para proporcionar un IDE, que tiene sólo unas pocas dependencias de otros paquetes. Otro objetivo era ser lo más independiente posible de un entorno especial de escritorio como KDE o GNOME. *Geany* sólo requiere las bibliotecas de tiempo de ejecución GTK2[13].

A continuación se exponen algunas de las características básicas de *Geany* (14):

- Resaltado de sintaxis.
- Plegado de código.
- Autocompletado.
- Cierre automático de etiquetas XML y HTML.
- Muestra de consejos.
- Archivos soportados de múltiples tipos de lenguajes tales como *C, Java, PHP, Python, Perl, Pascal*.
- Listas de símbolos.
- Código de navegación.

13 GTK (GIMP Tool Kit) es una biblioteca que contiene los objetos y funciones básicos para crear interfaces gráficas de usuario.

- Construir un sistema (conjunto de ejecuciones) para compilar y ejecutar el código.
- Fácil gestión de proyectos.
- Soporte para *plugins*.

Esta herramienta es utilizada para la implementación de los perfiles a los diferentes servicios a proteger.

1.6.2 Sistema de control de versiones

Un sistema de control de versiones es una combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los ficheros del proyecto, en particular en el código fuente, en la documentación y en las páginas web. Todo este proceso se realiza manteniendo una correcta gestión sobre las versiones de la información almacenada (15).

RapidSVN: es usado como cliente SVN para le gestión del código fuente de la aplicación en el sistema de control de versiones (39). Está escrito en C++ y distribuido bajo licencia GPL. Facilita el versionado de ficheros, desde una interfaz sencilla e intuitiva y se encuentra disponible para plataformas *Windows, Linux, MAC OS X* y *Solaris*. Es una herramienta rápida y eficiente. Es utilizada por el centro de desarrollo de CESOL.

1.7 Consideraciones finales

El estudio de los conceptos asociados al dominio del problema permitió una mejor comprensión del tema. La comparación entre los *frameworks* identificados a partir de un análisis de sus características condujo a seleccionar a *AppArmor* como *framework* de seguridad para la distribución cubana de GNU/Linux Nova elementos decisivos fueron su sencillez de uso, poseer una menor curva de aprendizaje e implementar una correcta política de seguridad, además es de origen comunitario a diferencia de SELinux que aunque es de código abierto es propiedad de la NSA. Como herramientas y tecnologías a utilizar, el *Geany* para implementar las políticas de seguridad y el *RapidSVN* para controlar las versiones de las mismas.

Capítulo 2: Perfiles para el *framework* de seguridad

2.1 Introducción

En este capítulo *s*e analizan un conjunto de servicios que son los más utilizados por los distintos OACE. Se proponen además los nuevos perfiles a incluir al *framework* de seguridad, describiéndose también las funcionalidades de cada uno.

2.2 Criterios de selección de servicios

Para la selección de los servicios telemáticos se aplicó una entrevista a los especialistas de CESOL, que se encargan de realizar el proceso de migración de los servicios telemáticos en los diferentes OACE, con el objetivo de determinar cuáles son los servicios telemáticos empleados comúnmente en las diferentes instituciones del país. A partir de dicha entrevista se decide crear nuevos perfiles para incrementar el nivel de seguridad a los servicios Apache, Squid y DHCP. Además actualizar NTP, Samaba4, SSH, Correo electrónico y DNS que provienen de la distribución GNU/Linux Ubuntu, que a consideración del autor de esta investigación su seguridad puede ser fortalecidos.

2.3 Servicios para la creación de perfiles

2.3.1 NTPD

Network Time Protocol (NTP) es un protocolo de red para sincronizar el reloj de un computador con la hora de una fuente de referencia, logrando una precisión de orden de milisegundos con respecto a la Hora Universal Coordinada (UTC por sus siglas en inglés). La hora UTC, que ha sido adoptada como la escala de tiempo estándar por la mayoría de las naciones del mundo. Básicamente, un cliente solicita la hora actual a un servidor, y usa la respuesta para poner en hora su propio reloj. El reloj sincronizado con NTP está siempre a la hora oficial y no es necesario ajustarlo cada cierto tiempo. Los problemas asociados a un reloj desincronizado son múltiples. Por ejemplo, el sello de tiempo cuando se crea o modifica un archivo puede quedar con la hora y fecha equivocada. El correo electrónico que se envía desde el computador podría llevar un sello de tiempo equivocado (26).

A continuación se presenta el perfil de *AppArmor* para el servicio NTP.

```
/usr/sbin/ntpd {
 #include <abstractions/base>
 #include <abstractions/nameservice>
 #include <abstractions/user-tmp>
 capability ipc_lock,
 capability net_bind_service,
 capability setgid,
 capability setuid,
 capability sys_chroot,
 capability sys_resource,
 capability sys_time,
 capability sys_nice,
 network inet6 dgram,
 network inet stream,
 network inet6 stream,
 @{PROC}/net/if_inet6 r,
 @{PROC}/*/net/if_inet6 r,
 @{NTPD_DEVICE} rw,
 /{,s}bin/ r,
 /usr/{,s}bin/ r,
 /usr/sbin/ntpd rmix,
 /etc/ntp.conf r,
 /etc/ntp.conf.dhcp r,
```

```
 /etc/ntpd.conf r,
 /etc/ntpd.conf.tmp r,
 /var/lib/ntp/ntp.conf.dhcp r,
 /etc/ntp.keys r,
 /etc/ntp/** r,
 /etc/ntp.drift rwl,
 /etc/ntp.drift.TEMP rwl,
 /etc/ntp/drift* rwl,
 /var/lib/ntp/*drift rw,
 /var/lib/ntp/*drift.TEMP rw,
 /var/log/ntp w,
 /var/log/ntp.log w,
 /var/log/ntpd w,
 /var/log/ntpstats/clockstats* rwl,
 /var/log/ntpstats/loopstats* rwl,
 /var/log/ntpstats/peerstats* rwl,
 /var/log/ntpstats/protostats* rwl,
 /var/log/ntpstats/rawstats* rwl,
 /var/log/ntpstats/sysstats* rwl,
/etc/init.d/ntp start r,
 /{,var/}run/ntpd.pid w,
}
```

Este perfil está compuesto por un conjunto de abstracciones, interfaces de red, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza NTPD son *base*, *nameservice* y *user_tmp*.

Estas abstracciones son un conjunto de ficheros y aplicaciones que sirven para definir de manera global los permisos de accesos a librerías y archivos básicos del sistema operativo.

La abstracción base, como su nombre lo indica es la base de todos los perfiles. Está compuesto por ficheros que permiten la unión de puertos efímeros[14], utilización de ficheros como ld.so.cache e ld que se utilizan para cargar las bibliotecas compartidas que mejor disponibilidad tengan en todas partes. Las abstracciones nameservice son utilizadas para realizar operaciones de nombre de servicios similares búsqueda de usuario por ID o nombre, dirección IP, al utilizar libnss-extrausers y sssd los ficheros de contraseñas y grupos se almacenan en una ruta alternativa, evitando pérdida de información. La abstracción usr_tmp contiene directorios temporales para usuarios y globales.

Las interfaces de red que utiliza este servicio son de versión 4 y versión 6 para *User Datagram Protocol* (UDP) y TCP respectivamente.

Las capacidades que se le otorgan a este servicio, permiten bloquear segmentos de memoria compartida, además permite el uso de *chroot*, el aumento de prioridades y establecimiento de prioridades en otros procesos, así como establecer la afinidad de CPU en otros procesos. Anula los límites establecidos de recursos, permite la manipulación del reloj del sistema de tiempo real, maneja los id de usuario y grupo del proceso, permite abrir un *Socket* en un puerto de 0-1024.

Los ficheros relacionados a este servicio contienen permisos independientes al que ya tienen por defecto en el sistema. Se le permite al servicio NTPD permiso de escritura a los archivos /var/log/ntp, escrituras, lecturas y manejo de *links* simbólicos a todos los archivos del directorio /tmp cuyo nombre comience con ntp.

2.3.2 DHCPD

El Protocolo de Configuración Dinámica de Host (DHCP por sus siglas en inglés), proporciona automáticamente los datos necesarios para configurar una dirección IP. Esta tecnología permite conectar un dispositivo a Internet de una manera más rápida y simple. Permite configurar la red de una manera más

14 **Puertos Efímeros**: Los diferentes sistemas operativos, con objeto de mantener los servicios de red hacen uso de un determinado rango de puertos TCP y UDP para atender estas conexiones.

directa, cambia la ubicación del dispositivo, tan solo es necesario volver a solicitar la configuración al servidor, la configuración tiene una duración que puede ser renovada fácilmente (29). Uno de los principales problemas de este protocolo es cuando un cliente intenta obtener o verificar una dirección IP, es posible que se registren problemas en syslog o en la salida del modo de depuración del servidor. Algunas de las causas de estos errores es debido a que un cliente solicita una dirección IP, específica o intenta ampliar un permiso para su dirección IP actual. El servidor DHCP no puede encontrar la tabla de red DHCP para esa dirección y otra es que la dirección IP que se iba a ofrecer a un cliente DHCP ya se está utilizando. Este problema puede surgir si hay más de un servidor DHCP propietario de la dirección. También puede ocurrir si se ha configurado manualmente una dirección para un cliente de red no DHCP (30).

```
#include <tunables/global>
/usr/sbin/dhcpd {
#include <abstractions/base>
#include <abstractions/nameservice>
#include <abstractions/console>
capability dac_override,
capability setgid,
capability setuid,
capability net_bin_service,
capability net_raw,
capability sys_chroot,
capability fsetid,
capability sys_tty_config
network inet stream,
network inet dgram,
```

network inet6 stream,

network inet6 dgram,

/db/dhcpd.leases* lrw,

/etc/dhcpd.conf r,

/etc/named.d/* r,

/etc/hosts.allow r,

/etc/hosts.deny r,

@{PROC}/net/dev r,

/usr/sbin/dhcpd rmix,

/var/lib/dhcp/ {db/,}dhcpd.leases* rwl,

/var/lib/dhcp/etc/dhcpd.conf r,

/{,var/}run/dhcpd.pid wl,

}

Este perfil está compuesto por un conjunto de abstracciones, interfaces de red, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza DHCPD son *base*, *nameservice* y *console*.

La abstracción consola permite tres formas comunes para referirse a este tipo de consola /dev/console rw, /dev/tty rw, estas entradas es un poco desafortunado; /dev/tty siempre estará asociado con la terminal de control por el núcleo, pero si un programa utiliza el directorio /dev/pts/interface, lo que realmente tiene acceso a -todos- xterm, sshd y algunas terminales en los sistemas.

Las interfaces de red que utiliza este servicio son de versión 4 y versión 6 para UDP y TCP respectivamente.

Las capacidades que se le otorgan a este servicio permiten anular los accesos del DAC incluyendo ACL de ejecución, manejar los id de usuario y grupo del proceso, además permite abrir un *Socket* en un puerto de 0-1024, permite el uso de los conectores directos de paquetes y la unión de cualquier tipo de proxy de

transporte, permite el uso de *chroot*, anula restricciones que el identificador de usuario efectivo debe coincidir con el id propietario del archivo y permite la configuración de dispositivos tty.

Los ficheros relacionados a este servicio contienen permisos independientes al que ya tienen por defecto en el sistema. Se le permite al servicio DHCPD permiso de lectura a los archivos /etc/, escrituras, lecturas y manejo de *links* simbólicos a todos los archivos.

2.3.3 Named

Sistema de Nombres de Dominio (DNS por sus siglas en inglés) es un servidor que proporciona resolución de nombres para redes basadas en TCP/IP. Hace posible que los usuarios de equipos clientes utilicen nombres en lugar de direcciones IP numéricas para identificar *hosts* remotos. Un equipo cliente envía el nombre de un *host* remoto a un servidor DNS, que responde con la dirección IP correspondiente. El equipo cliente puede entonces enviar mensajes directamente a la dirección IP del *host* remoto. Si el servidor DNS no tiene ninguna entrada en su base de datos para el *host* remoto, puede responder al cliente con la dirección de un servidor DNS que pueda tener información acerca de ese *host* remoto, o bien puede consultar al otro servidor DNS. Este proceso puede tener lugar de forma recursiva hasta que el equipo cliente reciba las direcciones IP o hasta que se establezca que el nombre consultado no pertenece a ningún *host* del espacio de nombres DNS especificado (31). Uno de los problemas presentados es cuando los usuarios generalmente no se comunican directamente con el servidor, la resolución de nombres se hace de forma transparente por las aplicaciones del cliente (navegadores, clientes de correo y otras aplicaciones que usan Internet). Al realizar una petición que requiere una búsqueda de DNS, la petición se envía al servidor DNS local del sistema operativo (32).

```
#include <tunables/global>

/usr/sbin/named {

#include <abstractions/base>

#include <abstractions/nameservice>

capability sys_resource,
```

```
capability net_bind_service,
capability setgid,
capability setuid,
capability sys_chroot,
/usr/local/lib/libsodium.so* mr,
/bin/false r,
/dev/null rw,
/dev/urandom r,
/etc/ld.so.chache r,
/etc/localtime r,
/etc/nsswitch.conf r,
/etc/passwd r,
/lib/*-linux-gnu*/libc-*.so mr,
/lib/*-linux-gnu*/libm-*.so mr,
/lib/*-linux-gnu*/libnsl-*.so mr,
/lib/*-linux-gnu*/libnss_compat-*.so mr,
/lib/*-linux-gnu*/libnss_files-*.so mr,
/lib/*-linux-gnu*/libnss_nis-*.so mr,
/usr/lib/libsodium.so* mr,
/usr/local/lib/libsodium.so* mr,
/usr/lib/libdns.so* mr,
/usr/sbin/named rmix,
}
```

Este perfil está compuesto por un conjunto de abstracciones, interfaces de red, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza Named son *base* y *nameservice*.

Las capacidades que se le otorgan a este servicio permiten anular los límites de recursos establecidos, manejar los id de usuario y grupo del proceso, permite abrir un *Socket* en un puerto de 0-1024 y permite el uso de *chroot*.

Los ficheros relacionados a este servicio contienen permisos independientes al que ya tienen por defecto en el sistema. Se le permite al servicio Named permiso de lectura a los archivos /etc/ y /dev/ exceptuando al fichero /dev/null, lecturas y asignación de ejecución a los restantes ficheros.

2.3.4 Correo (Postfix, Dovecot)

Postfix es un servidor de correo de *software* libre código abierto, utilizado para el envío de correo electrónico y creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura al utilizar Sendmail. Postfix es el agente de transporte por omisión en diversas distribuciones de Linux, proporciona una alternativa a *sendmail* convirtiéndose en un *software* de diseño más simple, más modular y más fácil de configurar y administrar (33). Dovecot es un servidor de IMAP y POP3 de código abierto para GNU/Linux, puede trabajar con el estándar mbox y sus propios formatos nativos dbox de alto rendimiento y es completamente compatible con implementaciones de servidores UW IMAP y Courier IMAP, así como con clientes que accedan directamente a los buzones de correo (33).

```
#include <tunables/global>
/usr/sbin/dovecot flags=(complain) {
 #include <abstractions/authentication>
 #include <abstractions/base>
 #include <abstractions/mysql>
 #include <abstractions/nameservice>
 #include <abstractions/ssl_certs>
 #include <abstractions/ssl_keys>
```

```
capability chown,
capability dac_override,
capability fsetid,
capability kill,
capability net_bind_service,
capability setgid,
capability setuid,
capability sys_chroot,
/etc/dovecot/** r,
/etc/mtab r,
/etc/lsb-release r,
/etc/SuSE-release r,
@{PROC}/@{pid}/mounts r,
@{PROC}/filesystems r,
/usr/bin/doveconf rix,
/usr/lib/dovecot/anvil Px,
/usr/lib/dovecot/auth Px,
/usr/lib/dovecot/config Px,
/usr/lib/dovecot/dict Px,
/usr/lib/dovecot/dovecot-auth Pxmr,
/usr/lib/dovecot/imap Pxmr,
/usr/lib/dovecot/imap-login Pxmr,
/usr/lib/dovecot/lmtp Px,
```

```
/usr/lib/dovecot/log Px,
/usr/lib/dovecot/managesieve Px,
/usr/lib/dovecot/managesieve-login Pxmr,
/usr/lib/dovecot/pop3 Px,
/usr/lib/dovecot/pop3-login Pxmr,
/usr/lib/dovecot/ssl-build-param rix,
/usr/lib/dovecot/ssl-params Px,
/usr/sbin/dovecot mrix,
/var/lib/dovecot/ w,
/var/lib/dovecot/* rwkl,
/var/spool/postfix/private/auth w,
/var/spool/postfix/private/dovecot-lmtp w,
/{,var/}run/dovecot/ rw,
/{,var/}run/dovecot/** rw,
}
```

Este perfil está compuesto por un conjunto de abstracciones, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza Correo (Dovecot) son *base*, *nameservice,* authentication, mysql, ssl_certs y ssl_keys.

La abstracción autenticación en algunos servicios lo necesitan para realizar la autenticación de los usuarios, es casi seguro que necesita tener acceso a las bases de datos de usuarios locales que contienen contraseñas, archivos de configuración. MySQL tiene acceso a archivos en varios lugares en el sistema de archivos, en una instalación estándar de MySQL en Ubuntu, los datos va en /var/lib/mysql. Pone los archivos de configuración en /etc/, los registros y los archivos binarios en varios lugares, e incluso se necesita para acceder a algunos archivos del sistema operativo, como /etc/hosts.allow. La

abstracción ssl_certs es un parche que crea una nueva abstracción que puede ser heredado por todos los perfiles que lo utilizan.

Las capacidades que se le otorgan a este servicio permiten anular la restricción de cambios de propiedad de los archivos y la propiedad de grupo, anula los accesos incluyendo la ACL de ejecución, anula restricciones que el identificador de usuario efectivo debe coincidir con el id propietario del archivo, anula la restricción que el id de usuario real o efectivo de un proceso que envía una señal debe coincidir con el id de usuario real o efectivo del proceso de recepción de la señal, maneja los id de usuario y grupo del proceso, permite abrir un *Socket* en un puerto de 0-1024 y el uso de *chroot*.

Los ficheros relacionados a este servicio contienen permisos independientes al que ya tienen por defecto en el sistema. Se le permite al servicio de Correo (Dovecot) permiso de lectura a los archivos /etc/, ejecución discreta de perfiles se encuentran algunos ficheros de /usr/lib/.

```
#include <tunables/global>
/usr/sbin/postfix {
#include <abstractions/base>
#include <abstractions/nameservice>
#include <abstractions/consoles>
#include <abstractions/usr-tmp>
/etc/mtab r,
/etc/postfix r,
/etc/postfix/aliases r,
/etc/postfix/aliases.db rw,
/etc/postfix/main.cf r,
/etc/postfix/postfix-script Px,
@{PROC}/net/if_inet6 r,
```

/usr/lib/postfix r,

/usr/lib/postfix/master Px,

/usr/lib/postfix/showq Px,

/usr/sbin/postalias Px,

/usr/sbin/postdrop Px,

/usr/sbin/postqueue Px,

/usr/sbin/postfix rmix,

/var/spool/postfix/ r,

/var/spool/postfix/active r,

/var/spool/postfix/bounce r,

/var/spool/postfix/corrupt r,

/var/spool/postfix/defer r,

/var/spool/postfix/deferred r,

/var/spool/postfix/incoming r,

/var/spool/postfix/maildrop/ r,

/var/spool/postfix/maildrop/* lrw,

/var/spool/postfix/pid r,

/var/spool/postfix/private r,

/var/spool/postfix/public r,

/var/spool/postfix/public/pickup w,

/var/spool/postfix/public/showq w,

/var/spool/postfix/public/qmgr w,

/var/spool/postfix/saved r,

}

Este perfil está compuesto por un conjunto de abstracciones, interfaces de red, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza Correo (Postfix) son *base*, *nameservice, console,* y *user_tmp*.

Los ficheros relacionados a este servicio contienen permisos independientes al que ya tienen por defecto en el sistema. Se le permite al servicio de Correo (Postfix) permiso de lectura de algunos de los archivos /var/spool/, en modo de ejecución discreta de perfiles se encuentran algunos ficheros de /usr/lib/.

2.3.5 Squid

Squid es un servidor proxy para web con caché. Es una de las aplicaciones más populares y de referencia para esta función, *software* libre publicado bajo Licencia Publica General (GPL). Entre sus utilidades está la de mejorar el rendimiento de las conexiones de empresas y particulares a Internet guardando en caché peticiones recurrentes a servidores web y DNS, acelerar el acceso a un servidor web determinado o añadir seguridad realizando filtrados de tráfico. Proporciona un servicio de proxy que soporta peticiones HTTP, HTTPS y FTP a equipos que necesitan acceder a Internet y a su vez provee la funcionalidad de caché especializado en el cual almacena de forma local las páginas consultadas recientemente por los usuarios. De esta forma, incrementa la rapidez de acceso a los servidores de información Web y FTP que se encuentran fuera de la red interna (34). Uno de los problemas que presenta es que al intentar acceder a ciertas páginas, aparecen con fechas desactualizadas (35).

```
#include <tunables/global>

/usr/sbin/squid {

#include <abstractions/base>

#include <abstractions/nameservice>

#include <abstractions/consoles>

capability setgid,

capability setuid,
```

```
capability net_bind_service
/usr/lib/squid/* rmix,
/usr/sbin/squid rmix,
/usr/sbin/unlinkd rmix,
/var/cache/squid/** lrw,
/dev/tty rw,
/etc/mtab r,
/etc/squid/* r,
@{PROC}/[0-9]*/mounts r,
@{PROC}/mounts r,
/usr/share/squid/** r,
/var/log/squid/access.log w,
/var/log/squid/cache.log rw,
/var/log/squid/store.log w,
/{,var/}run/squid.pid lrw,
/usr/sbin/digest_pw_auth rmix,
/usr/sbin/diskd rmix,
/usr/sbin/getpwname rmix,
/usr/sbin/ip_user_check rmix,
/usr/sbin/msnt_auth rmix,
/usr/sbin/ncsa_auth rmix,
/usr/sbin/no_check.pl rmix,
/usr/sbin/ntlm_auth rmix,
```

/usr/sbin/pam_auth rmix,

/usr/sbin/rcsquid rmix,

/usr/sbin/smb_auth rmix,

/usr/sbin/smb_auth.pl rmix,

/usr/sbin/smb_auth.sh rmix,

/usr/sbin/squid rmix,

/usr/sbin/squid_ldap_auth rmix,

/usr/sbin/squid_ldap_group rmix,

/usr/sbin/squid_ldapauth rmix,

/usr/sbin/squid_unix_group rmix,

/usr/sbin/squidclient rmix,

/usr/sbin/unlinkd rmix,

/usr/sbin/wbinfo_group.pl rmix,

/usr/sbin/yp_auth rmix,

}

Este perfil está compuesto por un conjunto de abstracciones, interfaces de red, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza Squid son *base*, *nameservice* y *console*. Estas abstracciones no son más que un conjunto de ficheros y aplicaciones que sirven para definir de manera global los permisos de accesos a librerías y archivos básicos del sistema operativo.

Las capacidades que le se otorgan a este servicio, permiten manejar los id de usuario y grupo del proceso y permite abrir un *Socket* en un puerto de 0-1024.

Los ficheros relacionados a este servicio contienen permisos independientes al que ya tienen por defecto en el sistema. Se le permite al servicio de Squid permiso de modo de lectura, asignación de ejecutables y modo de ejecución heredado a los archivos /usr/sbin/, en modo de lectura, escritura y manejo de *links* a

/var/cache/squid/**.

2.3.6 SSHD

Secure SHell (SSH) es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un *host* remotamente. A diferencia de otros protocolos de comunicación remota tales como *File Transfer Protocol* (FTP) o Telnet. SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. Está diseñado para reemplazar los métodos más viejos y menos seguros para registrarse remotamente en otro sistema a través de la *shell* de comando, tales como *telnet* o rsh (36). Los problemas más comunes están dados por las conexiones.

#include <tunables/global>

/usr/sbin/sshd {

#include <abstractions/base>

#include <abstractions/nameservice>

#include <abstractions/consoles>

#include <abstractions/wutmp>

#include <abstractions/authentication>

capability chown,

capability dac_override,

capability fowner,

capability fsetid,

capability kill,

capability net_bin_service,

capability setgid,

capability setuid,

```
capability sys_chroot,
capability sys_resource,
capability sys_tty_config,
/bin/ash rUx,
/bin/bash rUx,
/bin/bash2 rUx,
/bin/bsh rUx,
/bin/csh rUx,
/bin/ksh rUx,
/bin/sh rUx,
/bin/tcsh rUx,
/bin/zsh rUx,
/dev/ptmx rw,
/dev/pts/[0-9]* rw,
/dev/urandom r,
/etc/** r,
/proc/*/oom_adj rw,
/proc/*/oom_score_adj rw,
/sbin/nologin rUx,
/tmp/ssh-*/agent.[0-9]* rwl,
/tmp/ssh-*[0-9]*/ w,
/usr/sbin/sshd rmix,
/var/log/* rw,
```

```
/{,var/}run w,
/{,var/}run/sshd{,.init}.pid wl,
@{HOME}/.ssh/authorized_keys{,2} r,
@{PROC}/[0-9]*/fd/ r,
@{PORC}/[0-9]*/loginuid w,
@{PROC}/[0-9]*/mounts r,
^AUTHENTICATED {
#include <abstractions/authentication>
#include <abstractions/console>
#include <abstractions/nameservice>
#include <abstractions/wutmp>
capability setuid,
capability setgid,
capability sys_tty_config,
/dev/log w,
/dev/ptmx rw,
/etc/default/passwd r,
/etc/localtime r,
/etc/login.defs r,
/etc/motd r,
/tmp/ssh-*/agent.[0-9]* rwl,
/tmp/ssh-*[0-9]*/ w,
 }
```

```
^EXEC {
#include <abstractions/base>

/bin/ash Ux,
/bin/bash Ux,
/bin/bash2 Ux,
/bin/bsh Ux,
/bin/csh Ux,
/bin/ksh Ux,
/bin/sh Ux,
/bin/tcsh Ux,
/bin/zsh Ux,
/sbin/nologin Ux,
 }
^PRIVSEP {
#include <abstractions/base>
#include <abstractions/nameservice>
capability setgid,
capability setuid,
capability sys_chroot,
 }
^PRIVSEP_MONITOR {
#include <abstractions/base>
```

```
#include <abstractions/nameservice>
#include <abstractions/authentication>
#include <abstractions/wutmp>

capability chown,
capability setgid,
capability setuid,

/dev/ptmx rw,
/dev/pts/[0-9]* rw,
/dev/urandom r,
/etc/hosts.allow r,
/etc/hosts.deny r,
/etc/ssh/modu;i r,
@{HOME}/.ssh/authorized_keys{,2} r,
@{PROC}/[0-9]*/mounts r,
 }
}
```

Este perfil está compuesto por un conjunto de abstracciones, interfaces de red, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza SSHD son *base*, *nameservice,* wutmp, authentication y *consol*e.

Las capacidades que le se otorgan a este servicio, permiten anular los accesos incluyendo la ACL de ejecución, permiten manejar los id de usuario y grupo del proceso, permite abrir un *Socket* en un puerto de 0-1024, sustituye todas las restricciones sobre operaciones permitidas en los archivos, anula los límites de

recursos establecidos, permite la configuración de dispositivos tty, anula la restricción de cambio de propiedad de los archivos y la propiedad de grupo, anula restricciones que el identificador de usuario efectivo debe coincidir con el id propietario del archivo, anula la restricción que el id de usuario real o efectivo de un proceso que envía una señal debe coincidir con el id de usuario real o efectivo del proceso de recepción de la señal y permite el uso de *chroot*.

2.3.7 SMBD

Samba es un *software* de re-implementación libre del protocolo de red SMB / CIFS. Proporciona servicios de archivo e impresión para diversos clientes y se puede integrar con un dominio de servidor, ya sea como un controlador de dominio o como un miembro del dominio. Samba es estándar en casi todas las distribuciones de Linux y es comúnmente incluido como un servicio del sistema básico sobre otros sistemas operativos basados en Unix también. Samba es liberado bajo los términos de la GPL de GNU. El nombre proviene de Samba SMB (*Server Message Block*), el nombre del protocolo estándar utilizado por el sistema de archivos de red (37).

```
#include <tunables/global>
/usr/sbin/smbd flags=(complain) {
 #include <abstractions/authentication>
 #include <abstractions/base>
 #include <abstractions/consoles>
 #include <abstractions/cups-client>
 #include <abstractions/nameservice>
 #include <abstractions/samba>
 #include <abstractions/user-tmp>
 #include <abstractions/wutmp>
 capability dac_override,
 capability dac_read_search,
```

```
capability fowner,
capability lease,
capability net_bind_service,
capability setgid,
capability setuid,
capability sys_resource,
capability sys_tty_config,
/etc/mtab r,
/etc/netgroup r,
/etc/printcap r,
/etc/samba/* rwk,
@{PROC}/@{pid}/mounts r,
@{PROC}/sys/kernel/core_pattern r,
/usr/lib*/samba/vfs/*.so mr,
/usr/lib*/samba/charset/*.so mr,
/usr/lib*/samba/auth/script.so mr,
/usr/lib*/samba/pdb/*.so mr,
/usr/lib*/samba/{lowcase,upcase,valid}.dat r,
/usr/sbin/smbd mr,
/usr/sbin/smbldap-useradd Px,
/var/cache/samba/** rwk,
/var/cache/samba/printing/printers.tdb mrw,
/var/lib/samba/** rwk,
```

```
 /var/lib/sss/pubconf/kdcinfo.* r,
 /{,var/}run/cups/cups.sock rw,
 /{,var/}run/dbus/system_bus_socket rw,
 /{,var/}run/samba/** rk,
 /{,var/}run/samba/ncalrpc/ rw,
 /{,var/}run/samba/ncalrpc/** rw,
 /{,var/}run/samba/smbd.pid rw,
 /var/spool/samba/** rw,
 @{HOMEDIRS}/** lrwk,
}
```

Este perfil está compuesto por un conjunto de abstracciones, interfaces de red, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza SMBD son *base*, *nameservice, console,* authentication, cups-client, samba, wutmp y *user_tmp*.

Las capacidades que se le otorgan a este servicio, permiten anular los accesos incluyendo la ACL de ejecución, permite manejar los id de usuario y grupo del proceso, permite abrir un *Socket* en un puerto de 0-1024, sustituye todas las restricciones sobre las operaciones permitidas en los archivos, en los que el propietario del archivo id debe ser igual a la de id de usuario, salvo a que *Fsetid* es aplicable, anula también todas las restricciones del DAC en cuanto a leer y buscar en archivos y directorios, incluyendo las restricciones de ACL, excluyendo el acceso DAC cubierto por Linux Inmutable, anula los límites de recursos establecidos, permite toma de arrendamiento en los ficheros y permite la configuración de dispositivos tty.

Los ficheros relacionados a este servicio contienen permisos independientes al que ya tienen por defecto en el sistema. Se le permite al servicio de SMBD permiso de lectura de algunos de los archivos /etc/, en modo de asignación de ejecución y lectura en algunos archivos /usr/lib/.

2.3.8 Apache2

Apache es el servidor web más utilizado en los sistemas Linux. Los servidores Web se utilizan para servir páginas web solicitadas por los equipos cliente. Los clientes normalmente solicitan y ven páginas web usando las aplicaciones del navegador web como Firefox, Opera, Chrome, o Internet Explorer (27). Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar de manera remota, o explotar por los usuarios locales en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. Una de la ventaja de este servicio es que es modular. Consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web (28).

```
#include <tunables/global>
/usr/sbin/apache2 {
#include <abstractions/base>
#include <abstractions/nameservice>
 capability dac_override,
 capability kill,
 capability net_bind_service,
 capability setgid,
 capability setuid,
 capability sys_tty_config,
 / rw,
 /** mrwlkix,
 ^DEFAULT_URI flags=(complain) {
 #include <abstractions/base>
```

```
#include <abstractions/nameservice>
/ rw,
/** mrwlkix,
}
^HANDLING_UNTRUSTED_INPUT flags=(complain) {
#include <abstractions/nameservice>
/ rw,
/** mrwlkix,
}
}
```

Este perfil está compuesto por un conjunto de abstracciones, interfaces de red, capacidades y ficheros relacionados a este servicio. Las abstracciones que utiliza Apache2 son *base*, *nameservice*.

Las capacidades que le se otorgan a este servicio, permiten anular los accesos, incluyendo la ACL de ejecución, anula la restricción que el id de usuario real o efectivo de un proceso que envía una señal debe coincidir con el id de usuario real o efectivo del proceso de recepción de la señal, permite manejar los id de usuario y grupo del proceso, permite abrir un *Socket* en un puerto de 0-1024 y permite la configuración de dispositivos tty.

Este perfil contiene dos perfiles ^DEFAULT_URI flags=(complain), ^HANDLING_UNTRUSTED_INPUT flags=(complain), los cueles están en modo de quejas. Uno de los perfiles incluido contiene los archivos *apparmor* específicos para dicho paquete de aplicaciones web y el otro los sitios específicos permitidos y anulados.

2.4 Consideraciones finales

La entrevista realizada a los especialistas del centro de CESOL, permitió identificar los servicios telemáticos comúnmente empleados en los diferentes OACE (NTP, Correo Electrónico, Samba4, Apache2, SSH, DHCP, Squid y DNS). A partir de que algunos perfiles ya implementados no cumplían con el principio

mínimo de privilegio fueron restructurados contribuyendo a aumentar la seguridad del sistema, en este caso fueron cambiadas las políticas de: DNS, Correo electrónico, Samba4, SSH y NTP. La implementación de nuevos perfiles para Apache2, DHCP y Squid proporcionó un grado mayor de adaptación a las necesidades de la distribución cubana de GNU/Linux Nova. Todos los perfiles creados y modificados fueron documentados para elevar el nivel de entendimiento de la funcionalidad de los mismos.

Capítulo 3: Pruebas de los perfiles de *AppArmor*

3.1 Introducción

La propuesta realizada en el capítulo anterior propone un conjunto de perfiles para incrementar la seguridad en Nova Servidores. A continuación se realizan pruebas a las propuestas, para ellos se realiza una comparación teniendo en cuenta la propuesta actual con respecto a lo presente antes de esta investigación. Se aplica además la técnica IADOV a un grupo de especialistas en administración de servicios telemáticos, para evaluar el nivel de satisfacción.

3.2 Método experimental

La aplicación del método experimental se basa en la identificación de un elemento de estudio para controlar el comportamiento del mismo. Tiene como objetivo afectar un escenario y observar el efecto que causa a partir de los cambios realizados (40). Para la aplicación del método se determinaron dos escenarios. El escenario 1 consta de un sistema operativo Nova 5.0 sin sufrir cambios en las reglas heredadas de la distribución GNU/Linux Ubuntu. El escenario 2 consiste en la realización de cambios a los perfiles con el objetivo de adaptarlos a la versión de Nova Servidores.

3.3 Aplicación del método

Para la aplicación del método se tienen en cuenta las siguientes condiciones:

El escenario de pruebas se describe a continuación:

- Sistema operativo: Nova servidores 5.0.
- Recursos de *hardware*: Procesador Corei3, 2GB RAM, 2.4GHz.
- *Framework AppArmor* en su versión 2.10.95.

Los pasos a seguir para la realización de las pruebas fueron:

1. Instalar dos computadoras con Nova servidor 5.0.
2. Instalar los diferentes servicios.

3. Incorporar los perfiles que no se encontraban en el *framework* de seguridad.
4. Realizar cambios a los perfiles que se encontraban en el *framework* de seguridad.
5. Analizar el comportamiento de los cambios realizados. Para ellos se verificó en la salida de los logs que se encuentra en /var/log/syslog.

3.3.1 Servicio DNS

Para el caso del servicio de DNS se restringieron los ficheros siguientes:

Tabla 3: Mejoras de política de Named

Nova 5 (antes)	Nova Servidores (después)
lib/*-linux-gnu*/libc-*.so mr, /lib/*-linux-gnu*/libm-*.so mr, /lib/*-linux-gnu*/libnsl-*.so mr, /lib/*-linux-gnu*/libnss_compat-*.so mr, /lib/*-linux-gnu*/libnss_files-*.so mr, /lib/*-linux-gnu*/libnss_nis-*.so mr,	lib/*-linux-gnu*/libc-*.so r, /lib/*-linux-gnu*/libm-*.so r, /lib/*-linux-gnu*/libnsl-*.so r, /lib/*-linux-gnu*/libnss_compat-*.so r, /lib/*-linux-gnu*/libnss_files-*.so r, /lib/*-linux-gnu*/libnss_nis-*.so r, /var/lib/samba/private/** rw, deny /etc/passwd rw, deny /etc/sudoers rw,

Se restringen las librerías en modo lectura, puesto que contiene las bibliotecas necesarias para que se ejecuten los programas que tiene en /bin/ y /sbin/ únicamente. El contenido del fichero /etc/passwd determina quién puede acceder al sistema de manera legítima y que se puede hacer una vez dentro del sistema. Este fichero es la primera línea de defensa del sistema contra accesos no deseados. En el tenemos registrados las cuentas de usuarios, así como las claves de accesos y privilegios. El contenido del fichero /etc/sudoers permite o no la ejecución al usuario que lo invocó sobre un determinado programa

propiedad de otro usuario, generalmente del administrador del sistema 'root'. Además implementa un control de acceso altamente granulado de que usuarios ejecutan que comandos. Por lo anteriormente descrito se deniega los permisos de escritura y lectura de estos dos ficheros. Para la instalación de Samba4, solución usada en Nova para la migración del directorio activo en las instituciones cubana, como controlador de dominio es necesario permitir la convivencia entre la herramienta de nombre de dominio Bind9 y dicha aplicación. Con el objetivo de utilizar el controlador implementado por el equipo de desarrollo Samba4 para Bind9 se establecen los permisos necesarios en los perfiles de *AppArmor* asociados a Bind9.

3.3.2 Servicio Correo electrónico

Para el caso del servicio de correo electrónico se les restringen los siguientes ficheros:

Tabla 4: Mejoras de política de Correo electrónico

Nova 5 (antes)	Nova Servidores (después)
/usr/lib/dovecot/dovecot-auth Pxmr, /usr/lib/dovecot/imap Pxmr, /usr/lib/dovecot/imap-login Pxmr, /usr/lib/dovecot/managesieve-login Pxmr, /usr/lib/dovecot/pop3-login Pxmr,	/usr/lib/dovecot/dovecot-auth Px, /usr/lib/dovecot/imap Px, /usr/lib/dovecot/imap-login Px, /usr/lib/dovecot/managesieve-login Px, /usr/lib/dovecot/pop3-login Px, deny /etc/passwd rw, deny /etc/sudoers rw,

Se restringen las bibliotecas de los programas a modo de ejecución discreta de perfiles, puesto que no es recomendable tener modo lectura y modo de asignación de ejecutables a estos directorios.

3.3.3 Servicio SSH

Para el caso SSH se restringieron los ficheros siguientes:

Tabla 5: Mejoras de política de SSHD

Nova 5 (antes)	Nova Servidores (después)
/bin/ash rUx,	/bin/ash rix,
/bin/bash rUx,	/bin/bash rix,
/bin/bash2 rUx,	/bin/bash2 rix,
/bin/bsh rUx,	/bin/bsh rix,
/bin/csh rUx,	/bin/csh rix,
/bin/ksh rUx,	/bin/ksh rix,
/bin/sh rUx,	/bin/sh rix,
/bin/tcsh rUx,	/bin/tcsh rix,
/bin/zsh rUx,	/bin/zsh rix,
bin/ash Ux,	bin/ash ix,
/bin/bash Ux,	/bin/bash ix,
/bin/bash2 Ux,	/bin/bash2 ix,
/bin/bsh Ux,	/bin/bsh ix,
/bin/csh Ux,	/bin/csh ix,
/bin/ksh Ux,	/bin/ksh ix,
/bin/sh Ux,	/bin/sh ix,
/bin/tcsh Ux,	/bin/tcsh ix,
/bin/zsh Ux,	/bin/zsh ix,
/sbin/nologin Ux,	/sbin/nologin ix,

	deny /etc/passwd rw, deny /etc/sudoers rw,

El uso de los permisos modo de ejecución sin restricción (Ux) no es recomendable debido a que le permiten a un servicio ejecutarse sin restricciones. Por lo que se utiliza el modo de ejecución heredado que hereda del perfil padre.

3.3.4 Servicio SMB

Para el caso de SMB se restringieron los ficheros siguientes:

Tabla 6: Mejoras de política de SMBD

Nova 5 (antes)	Nova Servidores (después)
/usr/lib*/samba/vfs/*.so mr, /usr/lib*/samba/charset/*.so mr, /usr/lib*/samba/auth/script.so mr, /usr/lib*/samba/pdb/*.so mr,	/usr/lib*/samba/vfs/*.so r, /usr/lib*/samba/charset/*.so r, /usr/lib*/samba/auth/script.so r, /usr/lib*/samba/pdb/*.so r, deny /etc/passwd rw, deny /etc/sudoers rw,

Se restringen las bibliotecas de los programas a modo lectura, puesto que para estos directorios no son recomendables asignarle modo de asignación de ejecutables.

3.3.5 Servicio NTP

Para el caso de NTP se restringieron los ficheros siguientes:

Tabla 7: Mejoras de política de NTP

Nova 5 (antes)	Nova Servidores (después)
/etc/group rw,	/etc/group r,

	deny /etc/passwd rw, deny /etc/sudoers rw,

Se restringen el fichero /ect/group a modo lectura debido a que este fichero es crítico en el sistema.

En el directorio /var/log/syslog se guardan las violaciones de los permisos. Cuando un servicio quiere acceder a un directorio especifico y no tiene los permisos necesarios, al mismo tiempo se manda un mensaje de denegación. El propio *framework* es el encargado de que se cumplan estas restricciones. A continuación se describe lo que contiene el mensaje de denegación de un servicio.

May 9 17:32:19 cesol-93 kernel: [4571.121794] type=1400 audit(1462829539.608:66): apparmor="DENIED" operation="open" profile="/usr/sbin/named" name="/etc/bind/named.conf" pid=5782 comm="named" requested_mask="r" denied_mask="r" fsuid=119 ouid=0.

apparmor="DENIED": *AppArmor* deniega la acción que se quiere realizar.

operation="open": define qué tipo de acción fue denegada, en este caso intentaba abrir un archivo.

profile="/usr/sbin/named": el perfil que hace denegar esta acción.

name="/etc/bind/named.conf": el nombre del archivo que intentaba abrir.

Pid=5782: el PID[15] del proceso que intenta abrirlo.

comm="named": el nombre o comando que intentó abrir.

requested_mask="r": named lo quería hacer con el fichero de modo lectura.

denied_mask="r": *AppArmor* detiene la acción.

3.4 Prueba de validación

NTP es un servicio de sincronización de hora en la red. Exactamente la funcionalidad ntpdate se encarga de actualizar la hora local del sistema con un servidor de hora en la red.

Para esta prueba fue descargado y modificado el código fuente de esta aplicación, añadiéndole

15 **PID:** es un sistema al que le entra un error calculado a partir de la salida deseada menos la salida obtenida y su salida es utilizada como entrada en el sistema que se quiere controlar. El controlador intenta minimizar el error ajustando la entrada del sistema.

instrucciones maliciosas que permiten modificar el fichero /etc/group del sistema operativo.

Las operaciones realizadas fueron las siguientes:

1) Descargar el código fuente de ntpd desde el repositorio de fuentes de ubuntu (apt-get source ntpd)

2) Analizado el código fuente y modificado el fichero ntpdate.c, fue añadida la instrucción system ("echo "error" >> /etc/group"), luego de un profundo estudio fue verificado que el binario generado por este fichero se invoca para actualizar la hora del sistema, tarea común que ejecutan todos los sistemas GNU/Linux.

3) Fue compilado y empaquetado el código fuente de ntpd con los cambios realizados.

4) Fue instalado el paquete modificado (dpkg -i *.deb).

5) Se procedió a verificar los resultados, en ambos escenarios.

Escenario 1: NTP tiene permisos de escritura sobre el fichero /etc/group ó no hay política habilitada en *AppArmor*.

Al habilitar el perfil usr.sbin.ntpdate con permisos de escritura o no contar con política alguna implementada y solicitar una actualización del tiempo del sistema se pudo observar que el fichero /etc/group se modificaba, cuestión crítica que puede ser aprovechada para vulnerar la seguridad de Nova.

Los efectos pudieron ser vistos en tiempo real a través de la utilidad *tail* disponible en GNU/Linux. Se ejecutó tail -f /etc/group a la vez que se invocó a ntpdate 10.0.0.4, el fichero fue modificado, quedando como se muestra a continuación:

pulse-access:x:124:
rtkit:x:125:
saned:x:126:
gdm:x:127:
usser:x:1000:
sambashare:x:128:usser
vboxusers:x:129:
bind:x:130:

ntp:x:131:

error

error

error

error

En este caso se añadió al finalizar el fichero la línea “error”, pudiera verse eliminado el fichero, eliminado grupos o eliminados usuarios de grupos específicos. Esta actividad maligna pudo verse realizado sobre otros ficheros claves del sistema.

Escenario 2: NTP no tiene permisos de escritura sobre el fichero /etc/group

Luego se procede a habilitar el perfil que evitará esta falla de seguridad aa-enforce /etc/apparmor.d/usr.sbin.ntpdate. Y de ahí se procede a ejecutar el comando ntpdate 10.0.0.4 para verificar el funcionamiento del perfil obteniendo, que este a su vez se observa en el siguiente registro de syslog a través de tail -f /var/log/syslog, el cual muestra el siguiente mensaje:

May 25 17:13:59 cesol-135 kernel: [156023.961912] audit: type=1400 audit (1464210839.070:2867): apparmor="DENIED" operation="exec" profile="/usr/sbin/ntpdate" name="/bin/dash" pid=9233 comm="ntpdate" requested_mask="x" denied_mask="x" fsuid=0 ouid=0.

Este mensaje se repite cuatros veces, que significa los cuatros intentos de escritura al fichero /etc/group. Verificando el fichero /etc/group con el comando tail -f se obtiene la siguiente salida:

usser@cesol-135:~$ tail -f /etc/group

pulse-access:x:124:

rtkit:x:125:

saned:x:126:

gdm:x:127:

usser:x:1000:

sambashare:x:128:usser

vboxusers:x:129:

bind:x:130:

ntp:x:131:

Lo que significa que mientras se realizó la operación de ntpdate 10.0.0.4 el fichero group no sufrió cambios e informó del error en el registro del sistema.

La aplicación de este procedimiento evidencia un incremento en la seguridad del sistema, la creación de los nuevos perfiles, permitió limitar el acceso de los servicios telemáticos más importantes usados en los OACE a algunos de los ficheros claves del sistema.

3.5 Valoración de la satisfacción de los expertos

La técnica de IADOV, en su versión original fue creada por su autor V. A. IADOV, para el estudio de la satisfacción por la profesión en carreras pedagógicas (38). En la presente investigación, la técnica de IADOV constituye una vía para el estudio de la satisfacción de los especialista de administración de servicios telemáticos, ya que los criterios que se utilizan, se fundamentan en la relaciones que se establecen entre tres preguntas cerradas (3, 8, 10) que se intercalan dentro del cuestionario que aparece en el **Anexo 1**. Dichas preguntas se relacionan a través de lo que se denomina “Cuadro Lógico de IADOV” y cuya relación el encuestado desconoce (**Ver Anexo 2**). El cuestionario contempla además siete preguntas complementarias de carácter abierto.

A continuación se muestra los criterios tenidos en cuenta para la selección de los especialistas que fueron parte del estudio realizado:

- Experiencia como mínimo de tres años en administración de servicios telemáticos.
- Experiencia en la ejecución de al menos un proceso de migración de servicios telemáticos hacia GNU/Linux.
- Experiencia en actividades de gestión de seguridad informática en sistemas operativos.
- Conocimientos avanzados del sistema operativo Linux.
- No fueran parte del centro CESOL.

El número resultante de la interrelación de las preguntas cerradas, indica la posición de cada especialista en la siguiente escala de satisfacción:

1: Clara satisfacción.

2: Más satisfecho que insatisfecho.

3: No definidas.

4: Más insatisfecho que satisfecho.

5: Clara insatisfacción.

6: Contradictoria.

A través de la siguiente fórmula, se obtiene el índice de satisfacción grupal (**ISG**):

$$\text{ISG} = \frac{A(1)+B(0.5)+C(0)+D(-0.5)+E(-1)}{N}$$

Donde ***A*** representa el número de sujetos con índice individual **1**; ***B*** la cantidad con índice **2**; ***C*** la cantidad con índice **3** o la cantidad con índice **6**; ***D*** la cantidad con índice **4**; ***E*** la cantidad con índice **5** y ***N*** representa el número total de sujetos del grupo.

El ISG se expresa en una escala numérica que oscila entre 1 y -1 de la siguiente forma:

Tabla 8: Escala numérica del índice de satisfacción.

Índice de satisfacción		
Insatisfechos	**No definidos o contradictorios**	**Satisfechos**
-1< = ISG <= -0.5	-0.49 <= ISG <= 0.49	0.5<= ISG <= 1

Luego de aplicar el cuadro lógico de IADOV a cada uno de los encuestados, se obtuvieron los siguientes resultados:

Tabla 9: Resultado de la aplicación del método de IADOV.

Escala de satisfacción	**Total 11**

		Cantidad	%
1	Clara satisfacción	**6**	**54,54**
0.5	Más satisfecho que insatisfecho	**4**	**36,36**
0	No definido o contradictoria	**1**	**9,10**
-0.5	Más insatisfecho que satisfecho	0	0
-1	Clara insatisfacción	0	0

Como el análisis de los resultados obtenidos en la técnica de IADOV se obtiene el siguiente **ISG**:

$$ISG = \frac{6(1)+4(0.5)+1(0)+0(-0.5)+0(-1)}{11} = 0.72$$

Del **ISG** obtenido y en relación con la tabla 7, se observa que existe **satisfacción** de los especialistas encuestados.

3.6 Consideraciones finales

La actualización de los perfiles, permitió incrementar el nivel de seguridad en los servicios telemáticos que ofrece la distribución cubana de GNU/Linux Nova. La realización del experimento práctico con la simulación de la introducción de una vulnerabilidad en el servicio NTP permitió verificar el correcto funcionamiento del framework de seguridad seleccionado y del perfil modificado. Los especialistas se mostraron satisfecho con la propuesta.

Conclusiones generales

Luego de realizada la presente investigación se arriba a las siguientes conclusiones:

- A partir de los *frameworks* analizados, se considera *AppArmor* como el más apropiado para la distribución cubana de GNU/Linux Nova, cumpliendo con las condiciones, facilidad de uso, bajo curva de aprendizaje e implementar una correcta política de seguridad.
- Las restricciones definidas en los perfiles para los servicios telemáticos comúnmente utilizados en los diferentes OACE permitió aumentar el nivel de seguridad de la distribución cubana de GNU/Linux Nova.
- La ejecución de un caso simulación de ataque al sistema permitió verificar la correcta funcionalidad del *framework AppArmor* para este escenario.
- Los especialistas de administración de servicios telemáticos mostraron una clara satisfacción con la propuesta realizada.

Recomendaciones

Los objetivos trazados al inicio de este trabajo de manera general han sido logrados, pero al mismo tiempo, a lo largo del proceso de investigación, ha quedado claro que es la primera fase de una investigación que puede ser mucho más ambiciosa. Por tanto se declaran las siguientes recomendaciones:

1. Realizar un estudio en el que se identifiquen las aplicaciones utilizadas en los OACE para añadir perfiles para las mismas, garantizando una mayor seguridad en el sistema.
2. Desarrollar un módulo para el sistema HMAST desarrollado en el centro CESOL que permita la gestión y configuración de políticas de forma centralizada para el *framework AppArmor*.

Referencias Bibliográficas

(1) redyseguridad.fi-p.unam.mx [Online] [Citado noviembre 2015, 15] http://redyseguridad.fi-p.unam.mx/proyectos/buenaspracticas/mecanismos%20de%20seguridad.html

(2) asera.com [Online] [Citado octubre 2015, 8] http://www.asersa.com/asersa/Articulos/Articulo49.pdf

(3) elheraldo.co [Online] [Citado octubre 2015, 8] http://www.elheraldo.co/tecnologia/como-va-el-mundo-en-seguridad-informatica-182638

(4) theserverlabs.com [Online] [Citado noviembre 2015, 10] http://www.theserverlabs.com/es/soluciones/framework-de-seguridad.html

(5) González Rodríguez, Leover Armando. Alternativas para el desarrollo de Aplicaciones Web. Junio 2011.

(6) recursostic.educacion.es [Online] [Citado noviembre 2015, 10] http://recursostic.educacion.es/observatorio/web/en/software/software-general/562-elvira-misfud-

(7) Pfleeger, Charles P. Security in computing. 2006. ISBN: 978-0-13-239077-4.

(8) blog.hectorbenitez.com [Online] [Citado nobiembre 2015, 11] http://blog.hectorbenitez.com/2009/08/frameworks-de-php-un-acercamiento

(9) jordisan.net [Online] [Citado noviembre 2015, 11] http://jordisan.net/blog/2006/que-es-un-framework/

(10) seguridaddensistemascomputacionales.zonalibre.org [Online] [Citado noviembre 2015, 15] http://seguridadensistemascomputacionales.zonalibre.org/Control%20De%20Acceso%20En%20Los%20Sistemas%20Computacionales.pdf

(11) ditec.um.es [Online] [Citado noviembre 2015, 18] http://ditec.um.es/deiso/apuntes/tema7.pdf

(12) securityartwork.es [Online] [Citado noviembre 2015, 18] http://www.securityartwork.es/2010/03/12/sistemas-de-control-de-acceso-mac-y-dac/

(13) atc2.aut.uah.es [Online] [Citado diciembre 2015, 3] http://atc2.aut.uah.es/~rosa/LabRC/Prac_5/Listas%20de%20Control%20de%20acceso.pdf

(10*) National Security Agency.NSA&CSS.[Online] [Citado noviembre 2015, 24]. https://www.nsa.gov/public_info/press_room/2001/se-linux.shtml

(11) PETER LOSCOCCO, N. S. A. Integrating flexible support for security policies into the Linux operating system. En *Proceedings of the FREENIX Track:... USENIX Annual Technical Conference*. The Association, 2001. p. 29.

(12) Dr. Saeed Rajput, Roled Based Access Control Models

(13) SANDHU, Ravi S., et al. Role-based access control models. *Computer*, 1996, no 2, p. 38-47.

(14) Sitio oficial de Red/Hat. Sitio oficial de Red-Hat.[Online] [Citado Octubre 2015, 24]. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security-Enhanced_Linux/chap-Security-Enhanced_Linux-Introduction.html

(15) National Security Agency.NSA&CSS. [Online] [Citado noviembre 2015, 24]. https://www.nsa.gov/research/selinux/

(16) CAMPBELL, leona; JAEGER, Jana: Guia de administración de Novell AppArmor 2.0. 2006

(17) SUSE Linux Enterprise.SUSE Linux Enterprise. [Online] [Citado octubre 2015, 15] https://www.suse.com/support/security/apparmor/

(18) grsecurity.net [Online] [Citado Diciembre 2015, 4] https://grsecurity.net/

(19) openbsd.org [Online] [Citado febrero 2016, 7] http://www.openbsd.org/papers/ocf.pdf

(20) freebsd.org [Online] [Citado febrero 2016, 7] https://www.freebsd.org/doc/en/books/arch-handbook/sysinit.html

(21) freebsd.org [Online] [Citado febrero 2016, 8] https://www.freebsd.org/doc/en/books/arch-handbook/mac-framework-kernel-arch.html

(22) developer.android.com [Online] [Citado febrero 2016, 8] http://developer.android.com/intl/es/tools/help/systrace.html

(23) bigflake.com [Online] [Citado febrero 2016, 9] http://bigflake.com/systrace/

(24) alcancelibre.org [Online] [Citado febrero 2016, 9] http://www.alcancelibre.org/staticpages/index.php/introduccion-selinux-centos-fedora

(25) web.mit.edu [Online] [Citado diciembre 2015, 5] http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-selinux.html

(26) man-es.debianchile.org [Online] [Citado mayo 2016, 3] http://man-es.debianchile.org/ntp.html

(27) help.ubuntu.com [Online] [Citado mayo 2016, 3] https://help.ubuntu.com/lts/serverguide/httpd.html

(28) httpd.apache.org [Online] [Citado mayo 2016, 3] https://httpd.apache.org/docs/2.0/es/logs.html

(29) pamarke.com [Online] [Citado mayo 2016, 3] http://pamarke.com/blog/2012/11/03/ventajas-del-protocolo-dhcp/

(30) docs.oracle.com [Online] [Citado mayo 2016, 4] https://docs.oracle.com/cd/E19957-01/820-2981/dhcp-trouble-35/index.html

(31) technet.microsoft.com [Online] [Citado mayo 2016, 4] https://technet.microsoft.com/es-es/library/cc753635%28v=ws.10%29.aspx

(32) taringa.net [Online] [Citado mayo 2016, 4] http://www.taringa.net/post/info/15315280/Servidores-DNS-de-sencillo-a-complicado-de-solucionar.html

(33) slideshare.net [Online] [Citado mayo 2016, 4] http://es.slideshare.net/andreslds/servidor-de-correo-postfix-presentation

(34) recursostic.educacion.es [Online] [Citado mayo 2016, 4] http://recursostic.educacion.es/observatorio/web/es/software/servidores/589-elvira-mifsud

(35) ecualug.org [Online] [Citado mayo 2016, 5] http://www.ecualug.org/?q=2007/aug/06/forums/problemas_cache_squid

(36) web.mit.edu [Online] [Citado mayo 2016, 5] http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html

(37) linux/magazine.com [Online] [Citado mayo 2016, 5] http://www.linux-magazine.com/Online/Features/What-s-New-in-Samba-4

(38) efdeportes.com [Online] [Citado mayo 2016, 9] http://www.efdeportes.com/efd47/iadov.htm

(39) GNU Free Documentation License 1.2. rapidsvn. In: rapidsvn [online]. [Citado marzo 2016, 8]. [Accessed 6 February 2013]. Available from: http://www.rapidsvn.org/.

(40) MURILLO, Javier. Métodos de investigación de enfoque experimental. *USO DE LOS RECURSOS DIDÁCTICOS POR PARTE DE LOS MAESTROS Y MAESTRAS DE PRIMERO A CUARTO AÑO BÁSICO DE LAS ESCUELAS DE LA PARROQUIA*, 2011, vol. 5.

Bibliografía Consultada

ANCINCOVÁ, Barbora, et al. Fedora 22 SELinux User's and Administrator's Guide. 2014.

BADILLO BERNAL, David Hernán. Estudio comparativo de las distribuciones Linux orientado a la seguridad de redes de comunicación. 2015.

BRATUSD, Sergey, et al. Beyond SELinux: the Case for behavior-based policy and trust languages. computer science Technical report TR2011–701, Dartmouth College, 2011.

CHEN, Hong; LI, Ninghui; MAO, Ziqing. Analyzing and Comparing the Protection Quality of Security Enhanced Operating Systems. En NDSS. 2009. p. 11-16.

ESTEVE, Josep Jorba. Administración de seguridad.

HANSON, Chad. Selinux and mls: Putting the pieces together. En Proceedings of the 2nd Annual SELinux Symposium. 2006.

JAEGER, Trent; SAILER, Reiner; ZHANG, Xiaolan. Analyzing integrity protection in the SELinux example policy. En Proceedings of the 12th conference on USENIX Security Symposium-Volume 12. USENIX Association, 2003. p. 5-5.

LEITNER, Achim. Counterpoint: AppArmor vs SELinux. Linux Magazine (69), 2006, p. 40-42.

MURILLO, Javier. Métodos de investigación de enfoque experimental. *USO DE LOS RECURSOS DIDÁCTICOS POR PARTE DE LOS MAESTROS Y MAESTRAS DE PRIMERO A CUARTO AÑO BÁSICO DE LAS ESCUELAS DE LA PARROQUIA*, 2011, vol. 5.

Anexo 1: Encuesta de satisfacción

Estimado especialista:
Lea cuidadosamente cada pregunta antes de responder. Este cuestionario se realiza de manera anónima. Le agradecemos su participación y franqueza al decir honestamente lo que piensa sobre lo que le preguntamos.

1 - ¿Ha empleado Ud en su actividad como administrador/especialista de migración de servicios telemáticos los mecanismos de protección disponibles en Linux a través del framework de seguridad AppArmor?
____ Sí ____ No _____ No sé
2 - ¿Cuál de los siguiente servicios telemáticos a administrado/migrado Ud?
_____ NTP
_____ DNS
_____ DHCP
_____ Correo electrónico
_____ SSH
3 – ¿Considera Ud factible la administración y gestión de servicios telemáticos en GNU/Linux sin la existencia de un framework de seguridad para proteger los servicios y las aplicaciones?
____ Sí _____ No _____ No sé
4 - ¿De los mecanismos de protección de un sistema operativo selecciones cuales usted considera de mayor importancia?
_____ Antivirus
_____ Firewall
_____ Fraamework de seguridad
_____ Sistema de detección de intrusos
5 – ¿Considera útil el hecho de que Nova pueda contar un framework de seguridad con políticas diferentes a las que hereda de Ubuntu ? Argumente

6 – ¿Considera Ud. al framework de seguridad de un sistema GNU/Linux como un elemento clave para incrementar su nivel de seguridad?
____ Sí _____ No _____ No sé
7 – ¿Considera Ud importante algún servicio a aplicación informática de las existentes en Nova GNU/Linux a la que se le deba incrementar su nivel de seguridad? Argumente.

8 – ¿Si Ud. fuera a ejecutar un proceso de migración con Nova servidores lo realizaría con los nuevos perfiles de seguridad añadidos o empleando los mismos que se heredan desde Ubuntu ?
____ Sí _____ No _____ No sé
9 - ¿Qué considera sobre la seguridad de Nova, tiene algún elemento que desee señalar ? Argumente
10 – ¿Considera Ud positivo el incremento de la seguridad de Nova con los nuevos perfiles añadidos como resultado de esta investigación ?
____ Me gusta mucho.
____ Me gusta más de lo que me disgusta.
____ Me da lo mismo.
____ Me disgusta más de lo que me gusta.
____ No me gusta nada
____ No sé que decir

Anexo 2 Cuadro lógico de IADOV

¿Le gusta las acciones declaradas en las etapas de la estrategia metodológica para desarrollar el pensamiento lógico de los estudiantes en la asignatura Matemática III?	¿Consideras que la estrategia, fomenta tu preparación metodológica para contribuir al desarrollo del pensamiento lógico de los estudiantes en la asignatura Matemática III, mediante la formación de los procedimientos lógicos asociados a las diferentes formas lógicas del pensamiento?								
	Sí			No sé			No		
	¿Utilizarías las orientaciones metodológicas plasmadas en la estrategia metodológica y el sistema de actividades para desarrollar el pensamiento lógico de los estudiantes en la asignatura?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta mucho	2	2	3	2	3	3	6	3	3
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé que decir	2	3	6	3	3	3	6	3	4

Glosario de términos

ACL: Listas de Control de Acceso, es una lista que especifica los permisos de los usuarios sobre un archivo, carpeta u otro objeto.

AES: Advanced Encryption Standard, es uno de los algoritmos criptográfico más seguros y utilizados hoy en día y está disponible para uso público.

API: Application Programming Interface, es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-*software*.

AVC: Caché de Vector de Acceso,

AppArmor: Application Armor, es un programa de seguridad para Linux, lanzado bajo la licencia GPL.

BIOS: Basic Input/Output System, es un *software* que reside en un chip instalado en la *motherboard* de la PC y que realiza tarea apenas que se presiona el botón de encendido del equipo.

CESOL: Centro de Software Libre.

DAC: Control de Acceso Discrecional.

DES: Data Encryption Standard.

EEUU: Estados Unidos.

IDE: entorno de desarrollo integrado.

ISO: Organización Internacional de Normalización.

KLD: Kernel Linker Dynamic.

MAC: Control de Acceso Obligatorio.

MVC: Modelo Vista Controlador.

NSA: Agencia de Seguridad Nacional.

OACE: Organismos de la Administración Central del Estado.

OCF: OpenBSD Cryptographic *Framework*.