Universidad de las Ciencias Informáticas

Facultad 2

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Protección contra taques de red para el software Segurmática Antivirus para Linux.

Autor: Dennys Lázaro Alvarez Medina

Tutores: Ms. C. Diannet Sospedra López

Ing. Javier Ricardo Ponce Pérez

La Habana, junio 2018

"Año 60 de la Revolución"

"El aspecto fundamental en el cual la juventud debe señalar el camino es precisamente en el aspecto de ser vanguardia en cada uno de los trabajos que le compete."

Che

DECLARACIÓN DE AUTORÍA

Se declara ser el autor único del trabajo de diploma Protección contra taques de red para el software Segurmática Antivirus para Linux. Se reconoce a la Universidad de las Ciencias Informáticas los derechos de hacer uso del mismo en su beneficio.

Para que así conste se firma la presente a los ____ días del mes de julio del año 2018

Dennys Lázaro Álvarez Medina Firma del autor	Ms. C. Diannet Sospedra López Firma del tutor	Ing. Javier R. Ponce Pérez Firma del tutor

Datos de Contacto

Dennys Lázaro Alvarez Medina.

Estudiante de 5to año de la carrera Ingeniería en Ciencias Informàtica. Correo electrónico: dlalvarez@estudiantes.uci,cu

MSc. Diannet Sospedra López.

Graduada de Ingeniería en Ciencias Informáticas (2016) en la Universidad de las Ciencias Informática en el 2013. Jefa del Departamento de Aplicaciones del Centro de TLM de la Facultad 2. Correo electrónico: dsospedra@uci.cu

Ing. Javier Ricardo Ponce Pérez.

Graduado de Ingeniería en Ciencias Informáticas (2016) en la Universidad de las Ciencias Informática en el 2016. Perteneciente al departamento de Aplicaciones del Centro de TLM de la Facultad 2. Correo electrónico: jrponce@uci.cu

Dedicatoria

Le dedico esta tesis a dos personas que, aunque no están físicamente a mi lado, sé que si estuvieran presentes estarían contentos por este triunfo: mis abuelos Paula y Leoncio. Que sus bendiciones siempre me alcancen.

A mi mamá y a mi papá por ser las personas que han estado a mi lado brindándome su apoyo y amor en todo momento, los amo.

A mi familia en general por confiar en mí en todo momento. A todas muchas gracias. Los quiero.

AGRADECIMIENTO

Agradezco este trabajo a mis padres, por el orgullo de ser su hijo, por darme tanto amor, por ser la luz de mis días y el motivo de seguir adelante, por ser mi ejemplo a seguir mí meta para ser mejor cada día. Por cada consejo y cada regaño. Por apoyarme en los momentos más difíciles de mi vida cuando mi hijo nació y estaba en los exámenes del primer semestre de 4to año. Gracias los amo con mi vida mil gracias.

A mi hermano y mi cuñada que, aunque no están aquí a mi lado sé que han estado pendiente en todo momento, ya la vida se encargará de unirnos nuevamente y poder compartir juntos nuevamente.

A mi abuela por sus bendiciones en todo momento, por estar pendiente de mi de lo que hacía, de mis resultados, gracias abuela.

A mi pequeño gigante que llegó en el transcurso de esta carrera. La verdad fue lo mejor que me ha pasado en mi vida. Gracias por existir, por darme más motivos y razón de esforzarme para graduarme. Para que puedas ver en mí el ejemplo a seguir como lo tuve yo con mis padres.

A la madre de mi hijo y esposa, a su familia, por estar pendiente de mí en todo momento, por estar ahí en el transcurso de esta trayectoria, por darme apoyo gracias por todo.

A mi familia y los que han sido como si lo fueran, en espacial a mis primos, mis tíos, mis ahijados y ahijadas Mayelin, Barbón, Maydelin, Mileydi gracias por todo, los quiero mucho de verdad.

A los profesores que han formado parte de mi formación en especial a la profesora Arlety, al profesor Roberto de Base de Datos, a la profe Eliana Bárbara, por todas esas discusiones que tuvimos en los turnos de MIC, gracias por todo.

Agradecerles esta tesis a mis dos tutores, por todo los consejos que me dieron, por su apoyo incondicional, sin ustedes no hubiese podido llevar a cabo este trabajo.

A todos mis compañeros, en especial a los de mi grupo 2503.

A todo aquel que de una forma u otra formó parte de mi formación como profesional.

Muchas gracias

Resumen

Debido al desarrollo de la informática, los sistemas de comunicación actual, y la generación de grandes volúmenes de información por parte de las entidades de cualquier sector, surge la llamada sociedad de la información, la cual sufre constante y progresiva transformación que la impulsa cada vez más a los delitos informáticos. El aseguramiento de la integridad y seguridad debería ser aplicado a los sistemas de computación y datos. Los usuarios maliciosos buscan objetivos vulnerables tales como sistemas no actualizados, sistemas infectados con troyanos y redes ejecutando servicios inseguros. Las alarmas son necesarias para notificar a los administradores y a los miembros del equipo de seguridad que ha ocurrido una entrada ilegal para que así estos puedan responder en tiempo real a la amenaza, se han diseñado los sistemas de detección y prevención de intrusos como tales sistemas de notificación. La empresa Segurmática firmó con el Centro de Telemática (TLM) de la Universidad de las Ciencias Informáticas un convenio de unidad docente. En este centro, se está desarrollando actualmente la interfaz visual y por consola para el antivirus SAVUnix para Linux. Este software presenta funcionalidades, como son la protección permanente, cuarentena, exclusiones, estadísticas, servidor corporativo y búsquedas. Pero una de las debilidades que presenta SAVUnix es que no garantiza la seguridad ante ataques de red.

El objetivo de la presente investigación, fue desarrollar el módulo de protección contra ataques de red para el software Segurmática Antivirus para Linux que permita bloquear posibles ataques de red basado en reglas previamente configuradas por el usuario. Para el desarrollo de la investigación se utilizaron las herramientas QTcreator como IDE de desarrollo, Visual Paradigm para el modelado de los diagramas y SQLite como gestor de base de datos. Se obtuvo como resultado, una herramienta que permite bloquear ataques de red por reglas configuradas por el usuario.

Palabras Clave

Antivirus, ataques de red, software, protección, Segurmática Antivirus

Índice

Introducción 7
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN 11
1.1 Conceptos asociados al dominio del problema. 11
1.1.1 Ataques informáticos 11
1.1.2 Seguridad informática. 11
1.1.4 Sistema de detección de intrusiones (IDS) 11
1.2 Descripción general del objeto de estudio. 12
1.2 Funcionamiento de los IDS e IPS 15
1.3 Análisis de las soluciones similares existentes. 16
1.3.1 Snort 17
1.3.2 Suricata 17
1.3.3 Bro 17
1.4 Conclusiones del análisis de los IDPS 18
1.5 Ambiente de desarrollo 18
1.5.1 Metodología de desarrollo. 18
1.5.2 Herramienta de modelado de software. 19
1.5.3 Lenguaje de Programación 19
1.5.4 Entorno de desarrollo integrado 20
1.5.5 Gestor de base de datos 20
1.5.6 Librería 20
Conclusiones del capítulo 20
CAPÍTULO 2. PROPUESTA DE SOLUCIÓN 22
2.1 Descripción de la propuesta de solución 22
2.2 Modelo de Dominio 23
2.3 Actor del sistema 23
2.4 Requisitos del sistema 24
2.4.1 Requisitos Funcionales 24
2.5 Casos de Uso del Sistema 25

2.5.1 Diagrama de Casos de Uso del Sistema 26
2.6 Descripción de los Casos de Uso del Sistema 26
2.6.1 Especificación de casos de uso 26
2.7 Requisitos No Funcionales 30
2.8 Arquitectura del sistema. 32
2.9 Estructura de la aplicación 33
2.10 Modelo de Base de datos. 34
2.11 Diagrama de Despliegue 35
Conclusiones del capítulo 36
CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN 37
3.1 Patrones de Diseño 37
3.1.1 Patrones de Diseño GRASP 37
3.1.2 Patrones de Diseño GOF 38
3.2 Pruebas 39
3.2.2 Prueba de Aceptación 40
3.2.3 Diseño de Casos de Prueba 40
3.3 Resultados de las Pruebas 41
Conclusiones del capítulo 42
Conclusiones 43
Recomendaciones 44
Bibliografía 45
Referencias 49
Anexos 1
Glosario de términos 5

Índice de figuras

Figura 1: Funciones de los IDS(7) 16
Figura 2: Propuesta de solución. 22
Figura 3: Modelo conceptual. 23
Figura 4: Diagrama de casos de uso del sistema. 26
Figura 5: Arquitectura basada en Plugins. 32
Figura 6: Descripción de la Arquitectura. 33
Figura 7: Estructura de la aplicación 34
Figura 8: Modelo de base de datos. 35
Figura 9: Diagrama de despliegue. 36
Figura 10: Prueba Unitaria a la aplicación. 40
Figura 11: Resultados de Pruebas 42

Índice de tablas

Tabla 1: Descripción del actor del sistema. 24
Tabla 2: Requisitos funcionales. 25
Tabla 3: Descripción del caso de uso Gestionar Reglas. 26
Tabla 4: Descripción del caso de uso Gestionar Configuración. 28
Tabla 5: Requisito No funcional – Usabilidad. 30
Tabla 6: Requisito No funcional – Confiabilidad. 31

Introducción

Con el surgimiento de la Informática como ciencia y su aplicación en actividades humanas se concibe una nueva forma de ver el mundo, surge la llamada sociedad de la información, la cual sufre una constante y progresiva transformación que la impulsa cada vez más al desorden tecnológico. Es por ello que se hace necesaria la utilización de técnicas o métodos de protección para controlar el crecimiento y la integridad de la información.(1)

La seguridad informática es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático. No existe técnica, ni software que permita asegurar la protección de un sistema (2). Entre las herramientas conocidas que contribuyen a la seguridad de un sistema se encuentran los programas antivirus, los cortafuegos o firewalls, la encriptación de la información y el uso de contraseñas.

El antivirus es un programa que ayuda a proteger a los softwares contra la mayoría de los virus, que puedan infectaren un ordenador. Entre los principales daños que pueden causar los virus y los ataques de red están: la pérdida de rendimiento del microprocesador, borrado de archivos, alteración de datos, información confidencial expuesta a personas no autorizadas y la desinstalación del sistema operativo. Los antivirus son uno de los puntos de apoyo básicos de un sistema de seguridad personal, al lado de firewalls. Normalmente, los antivirus monitorizan actividades de virus en tiempo real y hacen verificaciones periódicas, o de acuerdo con la solicitud del usuario.(3)

El firewall o cortafuegos es una herramienta de seguridad cuya misión es analizar y bloquear cualquier intento de conexión peligrosa con el sistema, habitualmente, desde internet. Los cortafuegos se pueden aplicar a nivel de hardware, de software, o mixto, y cumple un papel trascendente en un entorno cada vez más dependiente del mundo conectado a internet. Los cortafuegos, a diferencia de los antivirus, funcionan en base a un acotado número de reglas que definen su ámbito de actuación. Y al igual que estos, pueden ser configurados para que por defecto, sean permisivos, o por defecto, sean restrictivos.(4)

Actualmente internet es una herramienta utilizada por muchas personas, incluyendo empresas, universidades y el gobierno de los diferentes países; junto a estas personas existen los usuarios malintencionados, su principal objetivo atacar las redes de ordenadores, dejando inutilizados los servidores o invadiendo su privacidad. Los expertos en seguridad informática se encargan de mitigar estos ataques y en la medida de lo posible diseñar nuevas arquitecturas contra estos ataques. En los últimos años la seguridad en la red se ha convertido en una prioridad, ya que cada vez hay más usuarios malintencionados que buscan cualquier vulnerabilidad para romper la seguridad, ya sea por superarse a sí mismo o por beneficios económicos.(5)

El aseguramiento de la integridad y seguridad debería ser aplicado a los sistemas de computación y datos. El internet ha facilitado el flujo de la información, desde personal hasta financiera. Los usuarios maliciosos buscan objetivos vulnerables tales como sistemas no actualizados, sistemas infectados con troyanos y redes ejecutando servicios inseguros. Las alarmas son necesarias para notificar a los administradores y a los miembros del equipo de seguridad que ha ocurrido una entrada ilegal para que así estos puedan responder en tiempo real a la amenaza. Se han diseñado los sistemas de detección y prevención de intrusos como tales sistemas de notificación (IDPS).(6)

La detección de intruso es el proceso de monitoreo de los eventos que ocurren en un sistema o red de computadoras analizándolos en busca de indicios de posibles incidentes, que son violaciones de política de seguridad informática, políticas de uso aceptable, o prácticas de seguridad estándar. Un sistema de detección de intruso (IDS) es un software que automatiza el proceso de detección de intruso. Un sistema de prevención de intrusión (IPS) es un software que tiene todas las capacidades de un sistema de detección de intruso y también puede intentar detener posibles incidentes.(7)

La empresa cubana de seguridad informática, Segurmática, tiene como línea principal la comercialización de productos antivirus para los sistemas operativos Linux [1] y Microsoft Windows [2]. Segurmática también ofrece otro grupo de servicios relacionados con la seguridad informática y otros productos de software. Dicha empresa desarrolló el software Segurmática Antivirus, con el objetivo de lograr una mayor seguridad y la protección ante ataques de red.

Segurmática Antivirus incorpora las principales prestaciones de los programas antivirus de uso común, como son: una interfaz amigable con diversas acciones y opciones de configuración, lo cual permite una interacción más eficiente entre el usuario y la aplicación. Brinda un mecanismo de protección basado fundamentalmente en la protección permanente y la búsqueda de códigos malignos a demanda, lo cual permite combatir de manera efectiva la actividad relacionada con códigos malignos. Presenta un proceso de actualización de las bases de definiciones de programas malignos de acuerdo con la frecuencia de ejecución y la vía de descarga, incluida el sitio web de Segurmática. El almacenamiento de las estadísticas de funcionamiento, cuarentena, información de códigos malignos detectados y acciones tomadas es otro de los módulos con que cuenta el producto. Además brinda la posibilidad de conectarse a un servidor corporativo para la administración remota.(8)

La empresa Segurmática firmó con el Centro de Telemática (TLM) de la Universidad de las Ciencias Informáticas (UCI) un convenio de unidad docente, juntos han desarrollado varias soluciones. En el centro de TLM se está desarrollando actualmente la interfaz visual y por consola para el antivirus

[1] Linux, sistema operativo de software libre, (no es propiedad de ninguna empresa o persona).
[2] Microsoft Windows es un sistema operativo, un conjunto de programas que posibilita la administración de los recursos de una computadora.

SAVUnix para Linux. Este software presenta varias funcionalidades, como son la protección permanente, actualización, cuarentena, exclusiones, estadísticas, servidor corporativo y búsquedas. Pero una de las debilidades que presenta SAVUnix es que no garantiza la seguridad ante ataques de red y los usuarios no pueden definir sus propias reglas para denegar conexiones. Representando una debilidad en la seguridad de las estaciones de trabajo o entornos personales, por lo que puede traer consigo:

- Daño a los sistemas informáticos con la intención de dejarlos inutilizados y, por tanto, dejen de prestar sus servicios como habían hecho hasta ese momento.
- Acceso a datos y ficheros no autorizados.
- Utilización inadecuada de determinados servicios por parte de usuarios no autorizados.
- Creación de nuevas cuentas de usuario con privilegios administrativos, que faciliten posteriores accesos al sistema comprometido.
- Modificación o destrucción de archivos y documentos guardados en sistemas informáticos.

Teniendo en cuenta la situación problemática planteada se identifica el siguiente **problema a resolver**: ¿Cómo contribuir en la seguridad de las computadoras ante posibles ataques de red?

Por lo que se delimita como **objeto de estudio:** la protección ante ataques de red. Para dar solución al problema de la investigación se establece como **objetivo general:** Desarrollar el módulo de protección contra ataques de red para el software Segurmática Antivirus para Linux que permita bloquear posibles ataques de red basado en reglas previamente configuradas por el usuario. Enmarcado en el **campo de acción:** la protección de ataques de red para Segurmática Antivirus para Linux. Para el cumplimiento del objetivo trazado, se han propuesto las siguientes **tareas de investigación**:

1. Elaborar el marco teórico-metodológico referente a los conceptos asociados a la protección contra ataques de red.
2. Seleccionar las diferentes herramientas y metodologías de desarrollo de software que faciliten el desarrollo del módulo para la protección contra ataques de red.
3. Desarrollar un módulo de protección contra ataques de red para el software Segurmática para Linux.
4. Validar la propuesta de solución desarrollada, para comprobar su correcto funcionamiento.

Métodos Teóricos

- **Análisis - Síntesis:** permitió analizar las teorías, documentos, posibilitando la extracción de los elementos más importantes que se relacionan con el objeto de estudio. A partir de la información y documentación estudiada sobre el tema, en diferentes fuentes, se facilitó la obtención de un conocimiento general durante la investigación.
- **Modelación:** es una reproducción simplificada de la realidad, permitió descubrir y analizar las relaciones y cualidades del objeto de estudio y propuesta de solución. Se utilizó en la realización

de los diagramas y artefactos generados como parte del desarrollo de la investigación.

- **Histórico - Lógico:** permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Su objetivo en una investigación es constatar teóricamente cómo ha evolucionado un determinado fenómeno en un período de tiempo, en toda su trayectoria o en un fragmento temporal de la lógica de su desarrollo. Permitió realizar un análisis profundo sobre la evolución y comportamiento de los sistemas de prevención y detección de intrusos durante las últimas décadas.

Métodos Empíricos

- **Observación:** se empleó para analizar el funcionamiento de los IDS, en la percepción selectiva de las restricciones y propiedades del sistema. Se utilizó para la determinación de la problemática que da origen a la investigación, mediante la interacción con las aplicaciones existentes. La observación que se realiza es estructurada ya que previamente se establecen los aspectos que se desean observar sistematizando los detalles más significativos para la investigación.

Técnica de Investigación

- **Análisis de documentos:** para apreciar los aspectos importantes contenidos en los documentos que aparecen en Internet sobre las características de los IDS, todos de actualidad y confiables, lo que posibilitan llegar a conclusiones certeras del tema objeto de estudio.

La estructura del documento se caracteriza por la presencia de: introducción, tres capítulos, conclusiones, bibliografías y los anexos. El contenido de cada capítulo se describe a continuación:

Capítulo I, Fundamentación Teórica: se explican los diferentes conceptos asociados al problema que permiten comprender la solución propuesta, así como las diferentes soluciones existentes en el mundo relacionadas con la protección de software. Además, se describen y caracterizan las herramientas a utilizar para la propuesta de solución.

Capítulo II, Características y diseños del sistema: se especifican los artefactos generados correspondientes a los diferentes flujos de trabajo según la metodología de desarrollado utilizada, que garantizan un mejor entendimiento y una correcta implementación de la propuesta de solución.

Capítulo III, Implementación y pruebas del sistema: se especifican las acciones asociadas al análisis, diseño e implementación del módulo de protección de software, así como las posibles pruebas para comprobar el correcto funcionamiento del componente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

En este capítulo se exponen algunos principios o axiomas que rigen a la protección de ataques de red. Se detalla el marco teórico del objeto de estudio y varias de las propuestas alternativas existentes en el mercado que permiten brindar solución a la problemática identificada. Se detallan además las herramientas y tecnologías que se emplean en el desarrollo de dicho componente.

1.1 Conceptos asociados al dominio del problema.

1.1.1 Ataques informáticos

Según Leandro Alegsa: Ataque informático: “intento organizado y deliberado de una o más personas para causar daño o problemas a un sistema informático o red. Los ataques en grupo suelen ser hechos por bandas de piratas informáticos por diversión, para causar daño, buenas (relativamente buenas) intenciones, espionaje, obtención de ganancias, etc. Los blancos preferidos suelen ser los sistemas de grandes corporaciones o estados, pero ningún usuario de internet u otras redes está exento”(9).

1.1.2 Seguridad informática.

Según Charly Brown la Seguridad informática: se refiere a las características y condiciones de sistemas de procesamiento de datos y su almacenamiento, para garantizar su confidencialidad, integridad y disponibilidad. (10)

1.1.3 Sistema de prevención de intruso (IPS)

Un sistema de prevención de intrusos (IPS) consiste en un conjunto de acciones predefinidas que tienen como objetivo prevenir actividades sospechosas que provienen tanto de las redes externas/internas como del mismo host de una manera proactiva y eficaz.(11)

1.1.4 Sistema de detección de intrusiones (IDS)

El término IDS (Sistema de detección de intrusiones) hace referencia a un mecanismo que, sigilosamente, escucha el tráfico en la red para detectar actividades anormales o sospechosas, y de este modo, reducir el riesgo de intrusión.(12)

Otra definición de sistema de detección de intrusos: es un componente más dentro del modelo de seguridad de una organización. Consiste en detectar actividades inapropiadas, incorrectas o anómalas desde el exterior-interior de un sistema informático.(13)

1.2 Descripción general del objeto de estudio.

La seguridad en la red se ha convertido en una prioridad, porque cada vez hay más usuarios malintencionados que buscan cualquier vulnerabilidad para romper la seguridad, ya sea por superarse a sí mismo o por beneficios económicos. Existen distintos tipos de ataques informáticos, se puede diferenciar en primer lugar entre los ataques activos, que producen cambios en la información y en la situación de los recursos del sistema, y los ataques pasivos, que se limitan a registrar el uso de los recursos y/o a acceder a la información guardada o transmitida por el sistema. Seguidamente se presenta una relación más detallada de los principales tipos de ataques contra redes y sistemas informáticos(14):

- ✓ Actividades de reconocimiento de sistemas estas actividades directamente relacionadas con los ataques informáticos: si bien no se consideran ataques como tales ya que no provocan ningún daño, persiguen obtener información previa sobre las organizaciones y sus sistemas informáticos, realizando para ello un escaneo de puertos para determinar qué servicios se encuentran activos o un reconocimiento de versiones de sistemas operativos y aplicaciones.
- ✓ Detección de vulnerabilidades en los sistemas: este tipo de ataques tratan la documentación de las posibles vulnerabilidades de un sistema informático, para a continuación desarrollar alguna herramienta que permita explotarlas fácilmente (herramientas conocidas popularmente como “exploits”).
- ✓ Robo de información mediante la interceptación de mensajes: ataques que tratan de interceptar los mensajes de correo o los documentos que se envían a través de redes de ordenadores como Internet, vulnerando de este modo la confidencialidad del sistema informático y la privacidad de sus usuarios.

Los IDPS se enfocan principalmente en identificar posibles incidentes, un IDPS podría detectar cuando un atacante ha comprometido con éxito un sistema explotando una vulnerabilidad en el sistema. Estos sistemas pueden monitorear las transferencias de archivos e identificar los que podrían ser sospechosos. Como se mencionó anteriormente los IDPS están compuestos por dos grandes grupos los IDS y los IPS los cuales se detallan continuación.

Alrededor de la década del 1960, los sistemas financieros comenzaron a introducir la práctica de la auditoría en sus procesos para inspeccionar datos y verificar la existencia de fraudes o errores en sistemas. Sin embargo, surgieron algunas cuestiones: ¿qué debería ser detectado?, ¿cómo analizar lo que se descubrió y cómo proteger los diversos niveles de habilitación de seguridad en una misma red sin comprometer la seguridad? Entre 1984 y 1986, Dorothy Denning y Peter Neumann desarrollaron un

primer modelo de IDS[3], un prototipo nombrado como IDES (Sistema Especialista en Detección de Intrusión).(15)

Un IDS protege a un sistema contra ataques, malos usos y compromisos. Puede también monitorear la actividad de la red, auditar las configuraciones de la red y sistemas por vulnerabilidades, analizar la integridad de los datos. Dependiendo de los métodos de detección que seleccione utilizar. Un IDS y las funciones que proporciona, es clave para determinar cuál será el tipo apropiado para incluir en una política de seguridad de computación. Esta sección discute los conceptos detrás de los IDSes, las funcionalidades de cada tipo de IDS y la aparición de los IDSes híbridos, que emplean varias técnicas de detección y herramientas en un sólo paquete.

Algunos IDSes están basados en conocimiento, lo que alerta a los administradores de seguridad antes de que ocurra una intrusión usando una base de datos de ataques comunes. Alternativamente, existen los IDS basados en comportamiento, que hacen un seguimiento de todos los recursos usados buscando cualquier anomalía, lo que es usualmente una señal positiva de actividad maliciosa. Algunos IDSes son servicios independientes que trabajan en el fondo y escuchan pasivamente la actividad, registrando cualquier paquete externo sospechoso. Otros combinan las herramientas de sistemas estándar, configuraciones modificadas y el registro detallado, con la intuición y la experiencia del administrador para crear un kit poderoso de detección de intrusos. Evaluando las diferentes técnicas de detección de intrusos lo ayudará a encontrar aquella que es adecuada para su organización. (6)

Los tipos más importantes de IDSes mencionados en el campo de seguridad son conocidos como IDSes basados en host y basados en red. Los IDS basado en host es el más completo de los dos, que implica la implementación de un sistema de detección en cada host individual. Sin importar en qué ambiente de red resida el host, estará protegido. Un IDS basado en la red filtra los paquetes a través de un dispositivo simple antes de comenzar a enviar a host específicos. Los IDSes basados en red a menudo se consideran como menos completos puestos que muchos host en un ambiente móvil lo hacen indisponible para el escaneo y protección de paquetes de red.(6)

Existen dos claras familias importantes de IDS:

- El grupo N-IDS (Sistema de detección de intrusiones de red), que garantiza la seguridad dentro de la red.
- El grupo H-IDS (Sistema de detección de intrusiones en el host), que garantiza la seguridad en el host.

[3] IDS: Sistema de detención de intruso.

Un N-IDS necesita un hardware exclusivo. Este forma un sistema que puede verificar paquetes de información que viajan por una o más líneas de la red para descubrir si se ha producido alguna actividad maliciosa o anormal. El N-IDS pone uno o más de los adaptadores de red exclusivos del sistema en modo promiscuo. Este es una especie de modo "invisible" en el que no tienen dirección IP; tampoco tienen una serie de protocolos asignados. Es común encontrar diversos IDS en diferentes partes de la red. Por lo general, se colocan sondas fuera de la red para estudiar los posibles ataques, así como también se colocan sondas internas para analizar solicitudes que hayan pasado a través del firewall o que se han realizado desde dentro.

El H-IDS se encuentra en un host particular. Por lo tanto, su software cubre una amplia gama de sistemas operativos como Windows, Solaris[4], Linux, HP-UX[5], Aix[6] y otras distribuciones libres. El H-IDS actúa como un servicio estándar en el sistema de un host. Tradicionalmente, el H-IDS analiza la información particular almacenada en registros y también captura paquetes de la red que se introducen/salen del host para poder verificar las señales de intrusión (como ataques por denegación de servicio, puertas traseras, troyanos, intentos de acceso no autorizado, ejecución de códigos malignos o ataques de desbordamiento de búfer).(12)

Los IDS es un sistema que se encargará de supervisar el comportamiento de una red para detectar e informar cualquier intrusión no autorizada, lo que puede afectar la integridad de la red. También está el IPS, una herramienta muy similar que detecta intrusiones que tiene la capacidad de bloquear o impedir el acceso después de su detección. Un IDS supervisa la red para detectar cuándo un sistema realiza actividades sospechosas al examinar el tráfico de red y las llamadas realizadas en el sistema. Mientras que el firewall se establecerá cuando una conexión entre dos computadoras a través de Internet no cumpla con las políticas de seguridad establecidas para el entorno de red. El antivirus puede controlar cuándo un dispositivo o un servidor de archivos en particular intentan realizar actividades maliciosas que pueden afectar la seguridad de su información.(16)

Los IPS presentan una mejora importante sobre las tecnologías de cortafuegos tradicionales, al tomar decisiones de control de acceso basados en los contenidos del tráfico, en lugar de direcciones IP o puertos.(17) La prevención de intrusos se realiza:

- Comparando las firmas de las actividades sospechosas con las firmas de las actividades ya conocidas y que se incluyen en un fichero de identificadores.
- Para que las tareas de protección contra intrusiones sean efectivas, un sistema IPS debe disponer de un sistema de actualización continuo mediante el cual, el fichero que contiene los

[4] Solaris es un sistema operativo de tipo Unix desarrollado desde 1992 inicialmente por Sun Microsystems y actualmente propiedad de Oracle Corporation.

[5] HP-UX es la versión de Unix desarrollada y mantenida por Hewlett-Packard.

[6] AIX es un sistema operativo UNIX System V propiedad de IBM.

identificadores de intrusiones se actualizará en todo momento.

- Un sistema de prevención de intrusos puede estar compuesto por software, hardware o la combinación de ambos elementos.

Según su ubicación, existen distintos tipos de IPS:

- **IPS de red:**
    - Tienen como objetivo proteger segmentos enteros de la red o zonas a las que tienen acceso.
    - Capturan paquetes del tráfico de red (sniffers) y los analizan en busca de patrones que puedan suponer algún tipo de ataque.
    - Si se han ubicado correctamente en la red, son capaces de analizar grandes redes, aunque su impacto sobre el tráfico es por lo general mínimo.
    - Utilizan un dispositivo de red configurado en modo promiscuo; es decir, son capaces de ver y analizan todos los paquetes que circulan por un segmento de red, aunque estos no vayan dirigidos a un determinado equipo).
    - Trabajan no solo a nivel TCP/IP, sino que también lo pueden hacer a nivel de aplicación.
- **IPS de host:**
    - Fueron los primeros IDS (Intrusion Detection System) desarrollados por la industria de seguridad informática.
    - Se limitan a proteger un sólo equipo.
    - Monitorizan gran cantidad de eventos y actividades con una gran precisión, determinando de esta manera qué procesos y usuarios se involucran en una determinada acción.
    - Recaban información del sistema como ficheros, logs y recursos para su posterior análisis en busca de posibles incidencias dentro del propio sistema, en modo local.

Existen grandes diferencias entre los IDS e IPS. Mientras que el primero es un software que automatiza el proceso de detección de intrusos, el segundo es un software de prevención de intrusión, que tiene como objetivo impedir posibles ataques. Así que uno trabaja de manera reactiva e informativa, mientras que el IPS disminuye el riesgo de comprometimiento de un ambiente(15).

1.2 Funcionamiento de los IDS e IPS

Seguidamente se analiza el funcionamiento de los IDS e IPS con el objetivo de lograr una mejor comprensión del tema.(7). (Ver Figura1)

- IDS: En resumen, el IDS es un software el cual analiza y detecta supuestos intrusos en la red o un ordenador, este sistema está basado en sensores virtuales, que permiten monitorear el tráfico de la

red, para así evitar dichos atacantes, pero este también permite analizar el comportamiento y el contenido de la red no solo el tráfico. El IDS, no solo analiza el tráfico de la red, sino su comportamiento y contenido, también suelen tener una base de datos con un conjunto información del atacante.

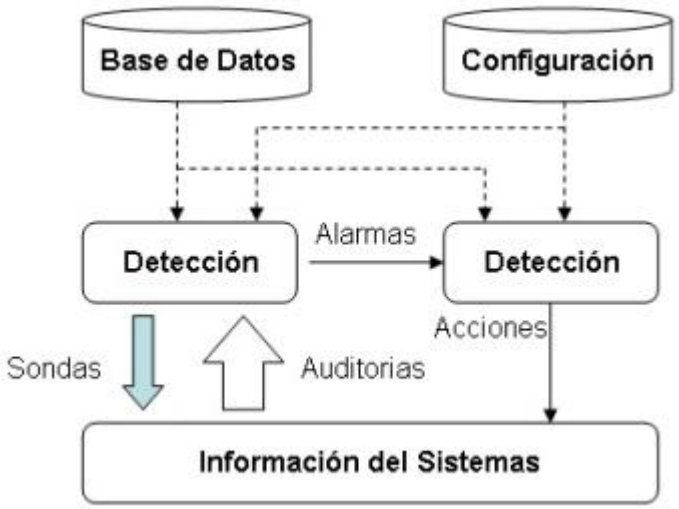


Figura 1: Funciones de los IDS(7)

- Los sistemas IPS no utilizan direcciones IP como lo hacen los firewalls, su función es poner normas o reglas que permiten restringir el acceso a usuarios, aplicaciones y a host siempre y cuando se detectan que estos están realizando actividades malintencionadas o transfiriendo código malicioso en el tráfico de la red.(18)

1.3 Análisis de las soluciones similares existentes.

Entre los softwares de protección instalados en las computadoras para garantizar la seguridad, se encuentra una amplia variedad de opciones de fabricantes y proveedores. Entre ellos, hay tres que parecen ser los más frecuentemente mencionados: IPDS, firewall y antivirus.

Los antivirus permitirán la detección de códigos maliciosos. Una buena solución de antivirus también debe detectar cuándo un archivo tiene algún tipo de comportamiento malicioso para no permitir la ejecución, y así prevenir el daño o el robo de información.

El firewall es una herramienta de seguridad que le permite controlar el tráfico de la red. Por lo general, filtran el tráfico de red entre Internet y un dispositivo en particular, y pueden operar de dos maneras diferentes: permitiendo todos los paquetes de red y solo bloqueando a algunos sospechosos; o negando todos los paquetes, solo permitiendo aquellos que se consideran necesarios.(16)

Es muy importante estar debidamente capacitado en cómo trabajar contra estas actividades maliciosas y crear conciencia entre los usuarios sobre las nuevas amenazas. De lo contrario, si los usuarios no tienen cuidado con la información que proporcionan en Internet o las contraseñas que están usando, pueden perder sus cuentas e información. El uso responsable de la información y los dispositivos permite que los entornos de trabajo sean más productivos con diferentes tecnologías de una manera más segura.

La evolución de los sistemas IDS convencionales basados en librerías de firmas hacia la nueva generación de IPS constituye un ejemplo paradigmático del cambio de enfoque defensivo a una aproximación preventiva a la seguridad en red. Los IPS combinan múltiples funcionalidades, como firewall, IDS, detección de anomalías de protocolo, antivirus, valoración de vulnerabilidades y filtrado de contenidos. De esta forma, consiguen proteger automáticamente de los ataques antes de que hayan impactado en la red, poniendo el énfasis en la prevención y en la automatización.(19)

Seguidamente se analizan algunos IDS e IPS para lograr una mejor comprensión del tema. Con este análisis se pretende identificar cuál o cuáles de las soluciones responden al problema de la investigación.

1.3.1 Snort

Sistema de detección de intrusiones basado en red, no están diseñados para detener un ataque, aunque sí pueden generar ciertos tipos de respuesta ante estos. El comportamiento de Snort se establece a partir de un archivo de configuración que responde al nombre de snort.conf. En este archivo se especifican las distintas opciones que delimitan cómo se comportará Snort y de qué forma trabajará.(20) Una de las funcionalidades más completas y poderosas de Snort, es la capacidad de definir reglas. En ciertos casos, es necesario configurar determinadas alertas para determinados tipos de paquetes. Es común, que frente a una vulnerabilidad en algún equipo crítico, será necesario solventar la situación mediante la configuración de alguna regla hasta que la vulnerabilidad sea solucionada(21).

1.3.2 Suricata

Herramienta IDS de arquitectura distinta, se comporta de la misma manera que Snort y usa las mismas firmas. De hecho, es capaz de funcionar sobre Snort.(20).

Suricata es un motor de detección de amenazas de red libre, madura, rápida y sólido, de código libre y abierto. El motor Suricata es capaz de detección de intrusión en tiempo real (IDS), prevención de intrusiones en línea (IPS), monitoreo de seguridad de red (NSM) y procesamiento de capa sin conexión. Suricata inspecciona el tráfico de la red utilizando reglas potentes, extensas y lenguaje de firma, y tiene un poderoso soporte de secuencias de comandos para la detección de amenazas complejas(22).

1.3.3 Bro

Software Bro-IDS o también llamado simplemente Bro, es un IDS basado en anomalías como en firmas. El tráfico capturado generará una serie de eventos. Por ejemplo, un evento podría ser un inicio de sesión

de usuario, conexión a servicio web. Bro es utilizado principalmente para el Intérprete de Políticas Script. Con su propio lenguaje de administración (Bro-Script). (20)

1.4 Conclusiones del análisis de los IDPS

El estudio realizado a los sistemas anteriormente descritos ha aportado una visión más clara de la línea a seguir durante el desarrollo del presente trabajo investigativo. Se han observado características comunes en estos sistemas garantiza y contribuye en la seguridad del host, entre las que se destacan:

- ✓ Vigilar y analizar las actividades de los usuarios y del sistema.
- ✓ Revisar las configuraciones del sistema y de las vulnerabilidades, mediante la utilización de reglas definidas por los usuarios.
- ✓ Evaluar la integridad de los archivos críticos del sistema.
- ✓ Análisis estadísticos para los modelos anormales de la actividad, mostrados en forma de notificación.

Así como un conjunto de elementos que no serían factibles al aplicar una de estas soluciones al problema planteado, entre los que se encuentran:

- ✓ Están desarrolladas para el trabajo en plataforma .NET.
- ✓ No cuenta con una interfaz gráfica de usuario (GUI). Por lo que no es posible que el propio usuario gestione sus reglas.

Tanto las ventajas como desventajas que ofrecen estos IDSes contribuyen para el desarrollo del presente trabajo. La propuesta de solución a desarrollar tendrá como premisas para su funcionamiento las ventajas mencionadas anteriormente, lo que contribuirá a un mejoramiento del sistema. Dentro de las características con las que contará el sistema se encuentran:

- ✓ Bloquear posibles ataques de red basado en reglas previamente configuradas por el usuario.
- ✓ El componente se podrá ejecutar en sistemas operativos Linux, logrando que sea un software libre.

1.5 Ambiente de desarrollo

Para lograr el desarrollo del módulo de protección de ataque de red se hace necesario tener conocimientos de las tecnologías a utilizar para alcanzar el objetivo propuesto. Además, con fines de lograr una mayor integración con la nueva versión del producto que se está desarrollando, y por las características que presentan, se seleccionan las siguientes tecnologías de desarrollo.

1.5.1 Metodología de desarrollo.

Toda metodología debe ser adaptada a las características de cada proyecto donde se utiliza por lo que

se decide realizar una adaptación de la metodología AUP que se adaptara al ciclo de vida definido para la actividad productiva de la UCI y de esta forma asegurar la calidad del software que se produce. De las cuatro fases de AUP (Inicio, Elaboración, Construcción y Transición) la AUP-UCI decide mantener la de Inicio, donde se realiza un estudio con el cliente para obtener información acerca del alcance, tiempo costo y esfuerzo del desarrollo y se decide si es viable pasar a la fase de Ejecución. En la fase de Ejecución se integran las fases de Elaboración, Construcción y Transición donde se ejecutan las actividades requeridas para desarrollar el software, se modela el negocio, se obtienen los requisitos, se elabora la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es entregado al cliente y se realizan capacitaciones sobre el uso del software.

En la metodología AUP-UCI se decide añadir una nueva fase, Cierre, en la cual se analizan los resultados del producto y se llevan a cabo todos los procedimientos formales del cierre del proyecto. Se escoge AUP-UCI como metodología de desarrollo dado que el proyecto lo desarrollan un integrantes que ejecuta diversos roles, con tiempo limitado para el desarrollo de la solución por lo que es necesario un enfoque ágil. Se emplean las tres fases definidas: de inicio, ejecución y cierre; y de los escenarios permitidos, el 01 que define el desarrollo del proyecto mediante el empleo de casos de uso para la representación del sistema.

1.5.2 Herramienta de modelado de software.

Se decide utilizar la herramienta profesional Visual Paradigm Suite en su versión 8.0 para representar el modelado de los procesos del sistema.

Esta versión permite desarrollar software de manera eficiente, rápida y de forma colaborativa. Visual Paradigm Suite 8.0 soporta todas las necesidades de diseño y modelado a lo largo del ciclo de vida de desarrollo de software, es una herramienta que ayuda a construir aplicaciones de calidad, de manera más rápida, óptima y más barata.(23)

1.5.3 Lenguaje de Programación

C++ es un lenguaje orientado a objeto, los códigos escritos en C++ ocupan menos memoria, son más rápidos en comparación con otros lenguajes en tiempo de ejecución y permite la programación multihilo, características que son muy importantes en la aplicación a desarrollar ya que se requiere de un sistema que permita bloquear posibles ataques de red. Posee varias características como:

- Es el lenguaje de programación de propósito general asociado al sistema operativo UNIX.
- Es un lenguaje de medio nivel.
- Posee una gran portabilidad.
- Se utiliza para la programación de sistemas: construcción de intérpretes, compiladores, editores de texto.(24)

1.5.4 Entorno de desarrollo integrado

QtCreator ha sido desarrollado para ser un entorno de desarrollo integrado (IDE) multiplataforma adaptado a las necesidades de los desarrolladores de Qt. El editor de código avanzado de QtCreator ofrece compatibilidad con la edición del lenguaje C++, ayuda sensible al contexto y finalización de código.(25)

Se utiliza el QtCreator como entorno de desarrollo en su versión 5.0, teniendo en cuenta las facilidades que ofrece para la edición de código en C++, para el diseño de las interfaces y para las herramientas de gestión de versiones y proyectos.

1.5.5 Gestor de base de datos

Para la gestión de bases de datos se utiliza el gestor SQLite ya que se necesita un medio para almacenar configuraciones simples y logos de la aplicación. No es un requisito importante ni tiene gran impacto en la solución, un archivo serializado podría servir, pero se selecciona una base de datos SQLite para aprovechar las ventajas del lenguaje SQL. Es un sistema de gestión de base de datos relacional, ligero, fácil de utilizar, muy confiable y libre. Este es de dominio público que implementa una pequeña librería de aproximadamente 500kB programada en lenguaje C. A diferencia del sistema de gestión de base de datos clientes-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. (26)

1.5.6 Librería

Libpcap es una biblioteca Open Source escrita en C que ofrece al programador una interfaz desde la que capturar paquetes en la capa de red. Además, Libpcap es perfectamente portable entre un gran número de sistemas operativos lo que la convierte en una herramienta importante que permite 'escuchar' la Red.(27) Se utiliza para ejecutar las configuraciones y reglas que sean definidas por el usuario para proteger su sistema ante posibles ataques de red.

Conclusiones del capítulo

Con el estudio de las soluciones similares existentes en el mundo y los conceptos asociados a la protección de software, se ha obtenido una visión más clara de los elementos y características que se deben tener en cuenta durante el diseño y construcción de un componente para la protección de ataques de red para el software Segurmática Antivirus para Linux.

Por lo que se llega a las siguientes conclusiones:

- ✓ La descripción general del objeto de estudio ayudó a comprender cuáles fueron las primeras técnicas de protección utilizadas. Además de evidenciar cuál es la técnica que por sus

características convendría utilizar en el desarrollo del módulo de protección de ataques de red para el software Segurmática Antivirus para Linux.

- ✓ La investigación realizada a las soluciones existentes de IDS permitió concluir que estos no podían ser utilizados como solución al problema por las desventajas que presentan, ya que en su mayoría presentan un elevado costo y están destinados a productos específicos.

- ✓ Los IDS analizados cuentan con características comunes que se pueden tener en cuenta en la implementación del componente, enfocándolas en las necesidades actuales del proyecto.

- ✓ Se caracterizaron las herramientas y tecnologías a utilizar en el desarrollo de la aplicación enunciándose de cada una de ellas las principales características y las facilidades que ofrecen al equipo de desarrollo, favoreciendo la posible integración con sistemas que lo soporten y garantizando futuras actualizaciones del sistema.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

En el presente capítulo se realiza la descripción de la propuesta de solución y se describen los principales requerimientos de la aplicación desarrollando los artefactos definidos en la metodología seleccionada, además se explica el uso de los patrones de diseño y el patrón arquitectónico seleccionado para la implementación. Por otra parte, se especifican cada uno de los casos de uso del sistema definidos según los requisitos identificados. Se elabora el diagrama de clases y se diseña el modelo de la base de datos creada a partir del diagrama de clases persistentes.

2.1 Descripción de la propuesta de solución

La propuesta de solución para resolver la problemática planteada, consiste en un software para la protección contra ataques de red, que le permite al usuario la definición de su propia lista de reglas. Desde desactivar la protección hasta el bloqueo completo de acceso a la red y debe ser capaz de bloquear la comunicación con el equipo que intente el ataque.

Figura *2*: Propuesta de solución.

El usuario al acceder al módulo de protección contra ataque de red, puede seleccionar diferentes reglas como activar modo sigiloso, tiempo de conexión, bloquear IP atacante, también permitirá al usuario gestionar las notificaciones, mediante la interfaz gráfica de usuario, que se comunica con el servicio mediante una llamada IPC, el servicio es el encargado de consultar la información en la base de datos y a su vez tiene incluida la librería Libcap que realizará todo el filtrado y análisis de los paquetes que viajan por la red.

2.2 Modelo de Dominio

El modelo de dominio se crea con el fin de representar el vocabulario y los conceptos clave del dominio del problema. Identifica las relaciones y atributos entre todas las entidades comprendidas en el ámbito del dominio del problema. Este a su vez proporciona una visión estructural del dominio que puede ser complementado con otros puntos de vista dinámicos, como el modelo de casos de uso.(28)

A continuación, se muestra el modelo de dominio de la propuesta de solución a desarrollar. Tener en cuenta que solo se adiciona el módulo de protección contra ataques de red.

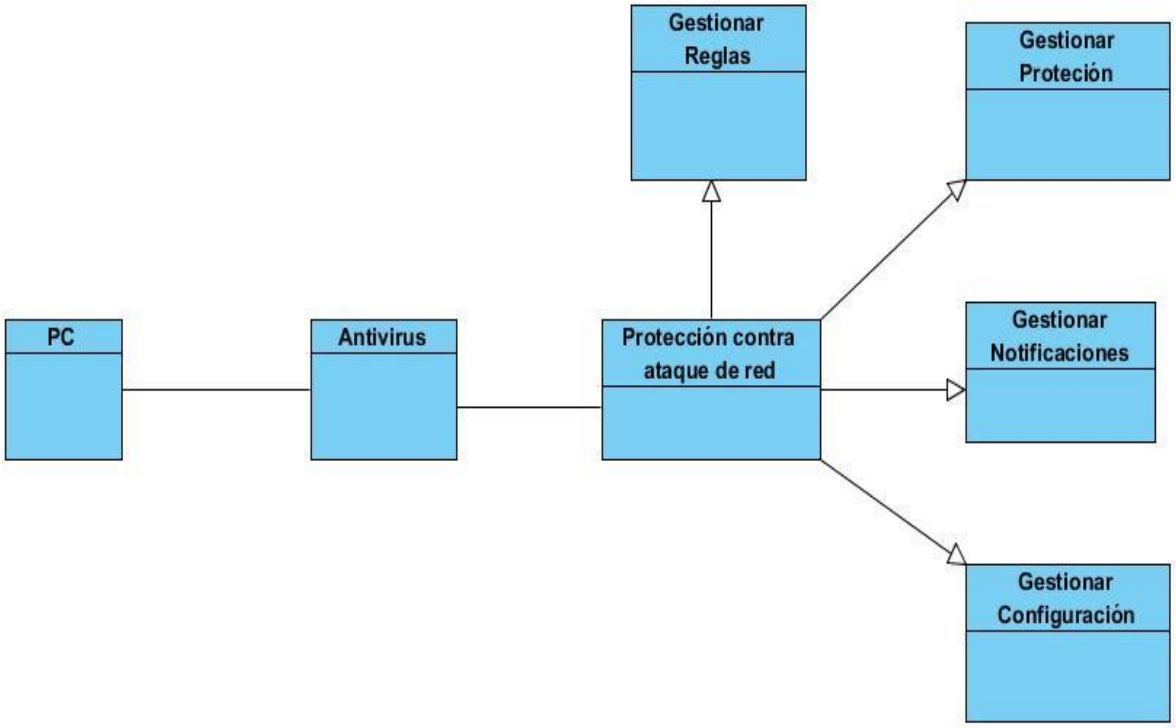


Figura *3*: Modelo conceptual.

2.3 Actor del sistema

A continuación se describe el actor del sistema, las tareas y responsabilidades que cumple.

Tabla 1: Descripción del actor del sistema.

Actor	Descripción
Usuario	Es el máximo responsable de definir la lista de reglas, de definir su propia lista de reglas, desde desactivar la protección hasta el bloqueo completo de acceso a la red, puede bloquear la comunicación con el equipo que intenta el ataque.

2.4 Requisitos del sistema

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan la necesidad de los clientes, para que un sistema ayude a resolver un determinado problema(29). A continuación de desarrolla los artefactos definidos por la metodología AUP-UCI utilizada para guiar el proceso de desarrollo de la propuesta de solución.

2.4.1 Requisitos Funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas dependen del tipo de software que se desarrolle(29).

Para dar solución a la problemática identificada y como parte de la propuesta de solución se identificaron 11 requisitos funcionales que se detallan a continuación, para ello se utiliza el artefacto descriptivo de requisitos definidos en la metodología.

Tabla *2*: Requisitos funcionales.

Nº	Nombre	Descripción	Complejidad	Prioridad
RF1 – RF4	Gestionar Reglas	Debe permitir al administrador adicionar, eliminar, modificar y mostrar reglas.	Alta	Alta
RF5	Gestionar configuración	Debe permitir al administrador activar o desactivar la protección, habilitar o deshabilitar las Notificaciones acerca de ataques de red, Bloquear la IP atacante, Modo Sigiloso (Stealth).	Media	Alta
RF6 – RF9	Gestionar notificaciones	Debe permitir al administrador mostrar, eliminar, exportar notificaciones	Baja	Media
RF10 – RF11	Gestionar protección	Desarrollo de las funciones que se deben ejecutar cuando se Activa la protección y cuando se Detiene la protección.	Alta	Alta

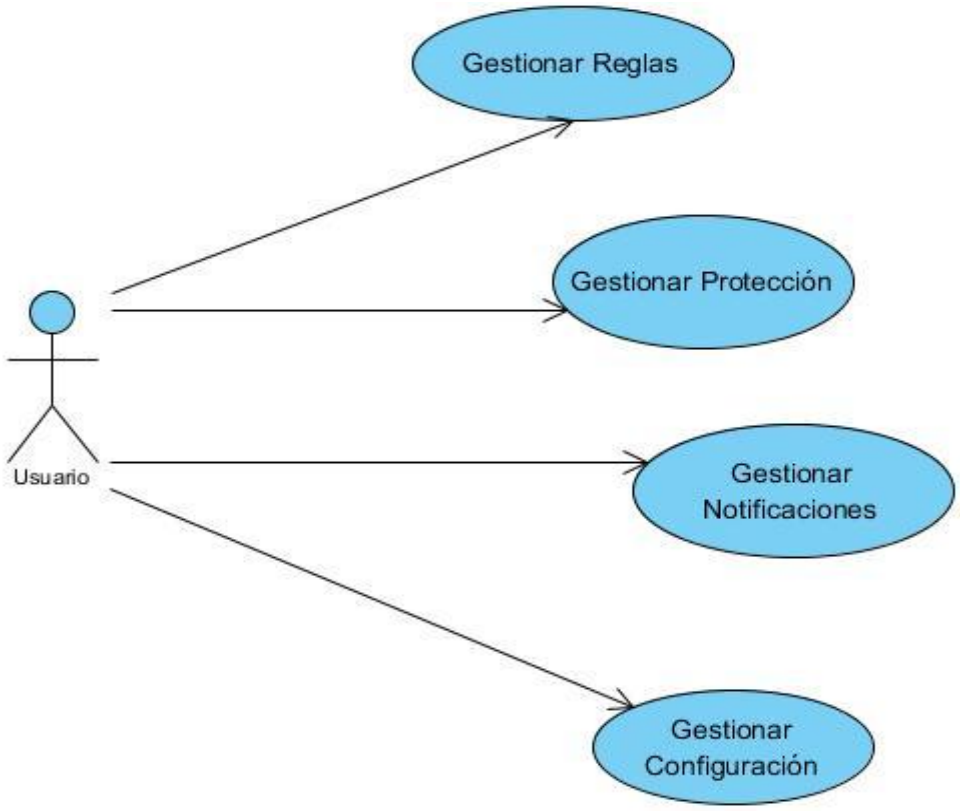
2.5 Casos de Uso del Sistema

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los diagramas de casos de uso muestran la relación entre los actores y los casos de uso en un sistema, una relación es una conexión entre los elementos del modelo. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

2.5.1 Diagrama de Casos de Uso del Sistema

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Además es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Figura *4*: Diagrama de casos de uso del sistema.



2.6 Descripción de los Casos de Uso del Sistema

La descripción de los casos de uso permite comprender mejor el funcionamiento de un sistema. Estos muestran cómo debería reaccionar el mismo ante una entrada del usuario, así como la descripción de las funcionalidades con las que cuenta.

2.6.1 Especificación de casos de uso

CU1. Gestionar Reglas

Tabla *3*: Descripción del caso de uso Gestionar Reglas.

Objetivo	Gestionar Reglas
Actores	Usuario
Resumen	El caso de uso comienza cuando el usuario ejecuta el Gestionar Regla, luego se muestra una interfaz donde permita adicionar, eliminar, modificar y mostrar las reglas.

Complejidad	**Alta**
Prioridad	**Alta**
Precondiciones	El usuario debe de haber seleccionado la opción.
Postcondiciones	Se adicionan, se eliminan, se modifican o se muestran las reglas.

Flujo de eventos

Flujo básico Gestionar Reglas

	Actor	Sistema
1.	1.1 Selecciona en la jerarquía del enfoque, la opción que desea realizar, adicionar, eliminar, modificar o mostrar las reglas.	
2.		2..1 • Si decide adicionar una regla, ir a la opción: - "Adicionar Regla" • Si decide eliminar una regla, ir a la opción: - "Eliminar Regla" • Si decide modificar una regla, ir a la opción: - "Modificar Reglas"
3.		3.1 Verificar que los campos estén llenos. 3.2 Verificar que los datos introducidos estén correctos.
		4.1 Almacena los datos de la regla. Finalizando así el Caso de Uso.

Sección 1: "Adicionar Reglas**"**

Flujo básico

	Actor	Sistema
1.	1.1 Selecciona la opción "Adicionar Regla "	
2.		2.1 Muestra los siguientes campos a introducir: • Host • Net • Port
3.	3.1 Luego de haber insertado los datos solicitados por el sistema los campos presiona "Aceptar"	
4.		4.1 Verifica que todos los campos estén llenos. 4.2 Verificar que los datos introducidos estén correctos.
5.		5.1 Adicionar la regla nueva en el sistema. Finalizando así el Caso de Uso.

Sección 2: "Eliminar Reglas**"**

Flujo básico

	Actor	Sistema

1.	1.1 Selecciona la opción “Eliminar Regla “ 1.2 Debe seleccionar la regla que desea eliminar	
2.		2.1 Debe de eliminar de la tabla y de la base de dato la regla que desea eliminar el usuario.
3.	3.1 Luego de haber eliminado las reglas, presiona el botón Aceptar	
4.		4.1 Debe de guardar todas las modificaciones realizadas por el usuario
Sección 2: “Modificar Reglas**”**		
Flujo básico		
1	1.1 Selecciona la opción “Modificar Regla “	
2		2.1 El sistema debe de permitir al usuario modificar los campos de las reglas que desea modificar
3	3.1 Luego de haber modificado las reglas, presiona el botón Aceptar	
4		4.1 Debe de guardar todas las modificaciones realizadas por el usuario

Relaciones	**CU incluidos**	No aplica
	CU extendidos	No aplica
Requisitos no funcionales	No aplica	
Asuntos pendientes	No aplica	

Prototipo elemental de interfaz gráfica de usuario

CU2 Gestionar Configuración

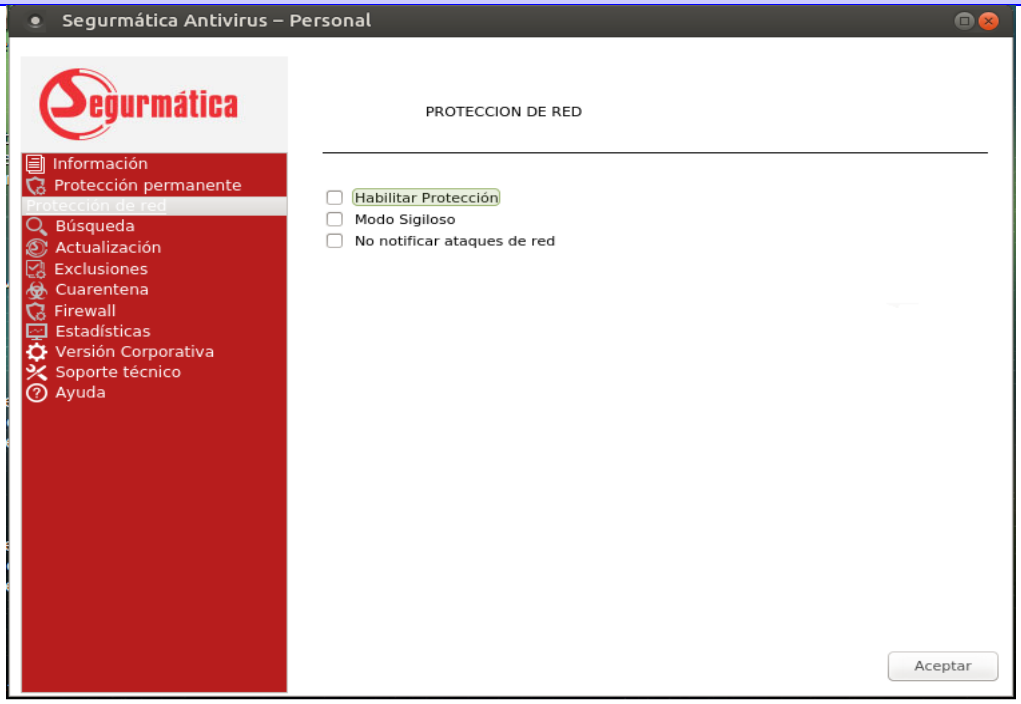
Tabla 4**:** Descripción del caso de uso Gestionar Configuración.

Objetivo	Gestionar Configuración
Actores	Administrador
Resumen	El caso de uso comienza cuando el usuario ejecuta la opción Gestionar configuración, el cual debe de permitir al usuario activar o desactivar la protección, habilitar o deshabilitar las Notificaciones acerca de ataques de red, Bloquear la IP atacante, Modo Sigiloso (Stealth).

Complejidad	**Media**
Prioridad	**Alta**
Precondiciones	El usuario debe de haber seleccionado una de las siguientes opciones: ✓ Modo sigiloso ✓ No notificar ataques de red ✓ Bloquear IP atacante
Postcondiciones	Activar o desactivar la protección, habilitar o deshabilitar las Notificaciones acerca de ataques de red, Bloquear la IP atacante, activar o desactivar el modo Sigiloso (Stealth).

Flujo de eventos		
Flujo básico Gestionar Configuración		
	Actor	**Sistema**
•	1.1 Selecciona en la jerarquía del enfoque, la opción que desea gestionar el administrador; Activar o desactivar la protección, habilitar o deshabilitar las notificaciones acerca de ataques de red, Bloquear la IP atacante, activar o desactivar el modo Sigiloso (Stealth).	
1.		2..1 Si decide Activar o desactivar la protección ir a la opción: - "Activar o desactivar la protección" Si decide habilitar o deshabilitar las notificaciones ir a la opción: - "Habilitar o deshabilitar notificaciones" Si decide Bloquear la IP atacante ir a la opción - "Bloquear la IP atacante" Si decide activar o desactivar el modo Sigiloso ir a la opción: - "Modo Sigiloso (Stealth)."
•		3.1 Almacena los datos de la regla. Finalizando así el Caso de Uso.
Sección 1: "Gestionar Configuración "		
Flujo básico		
	Actor	**Sistema**
1.	1.1 El usuario activa la opción de Modo Sigiloso.	
2.		2.1 Solo permitirá las actividades de red que son iniciadas por el usuario. El resto de las acciones como conexiones remotas hacia la PC, acceso a puertos, etc. no estarán permitidas.
3.	Para guardar las configuración presiona el botón Aceptar.	
4.		Almacena los cambios en la base de datos. Finalizando así el Caso de Uso.
Flujos alternos		
Nº Evento <Condición que dio lugar a la extensión>		
	Actor	**Sistema**
1.		

Relaciones	**CU incluidos**	No aplica
	CU extendidos	No aplica
Requisitos no funcionales	No aplica	
Asuntos pendientes	No aplica	
Prototipo elemental de interfaz gráfica de usuario		



2.7 Requisitos No Funcionales

Los requisitos no funcionales (RNF) como su nombre lo indica, son aquellos requerimientos que se refieren a las propiedades del sistema, como el tiempo de respuesta, la capacidad de almacenamiento y otros aspectos como el diseño, aspectos éticos, de seguridad entre otros. De forma alternativa, definen las restricciones del sistema como la capacidad entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema (30).

Tabla *5*: Requisito No funcional – Usabilidad.

Atributo de Calidad	RNF Usabilidad.
Sub-atributos/Sub-características	• Idiomas distintos. • Habilidades mínimas de usuarios. • Documentación de usuarios.
Objetivo	• Garantizar facilidad de uso por personas que no hablen el idioma (español) del país donde el producto es creado. • Garantizar que el sistema pueda ser usado por personas con un mínimo de conocimientos sobre el sistema operativo GNU/Linux. • Garantizar documentación detallada de cada una de las funcionalidades del sistema.

Origen	Usuario del sistema.
Artefacto	El sistema.
Entorno	Operación normal.
1. a Selección de funcionalidad.	
Seleccionar funcionalidad a utilizar.	Ejecución correcta de la funcionalidad seleccionada.
2, a Consulta del manual de usuarios.	
Seleccionar funcionalidad a utilizar.	Visualización del manual de usuarios.
Medida de respuesta	
Disponibilidad del sistema y manual de usuarios, para cada funcionalidad y en el idioma seleccionado.	

RNF: Confiabilidad

Tabla 6: Requisito No funcional – Confiabilidad.

Atributo de Calidad	RNF Confiabilidad.
Sub-atributos/Sub-características	• Disponibilidad del sistema. • Ejecución de funcionalidades.
Objetivo	• Garantizar que el sistema esté disponible. • Garantizar que el sistema ejecute justo las funcionalidades solicitadas por el usuario.
Origen	Usuario del sistema.
Artefacto	El sistema.
Entorno	Operación normal.
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
1. a Ejecutar sistema.	
Ejecutar el sistema o funcionalidad a utilizar.	Disponibilidad del sistema, ejecutando la(s) funcionalidad(es) que sea(n) solicitada(s).
Medida de respuesta	
Disponibilidad del sistema y ejecución de las funcionalidades solicitadas.	

RNF: Eficiencia

Tabla 7: Requisito No funcional – Eficiencia.

Atributo de Calidad	RNF Eficiencia.
Sub-atributos/Sub-características	• Tiempo de respuesta. • Utilización de recursos.
Objetivo	• Garantizar la respuesta del sistema en el tiempo requerido. • Garantizar valores de consumo de memoria RAM adecuados.
Origen	Usuario del sistema.
Artefacto	El sistema.
Entorno	Operación normal.
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
1. a Ejecutar sistema.	
Ejecutar el sistema o funcionalidad a utilizar.	Ejecución del sistema o funcionalidad a utilizar en un tiempo no mayor a los 5s. Dicha ejecución no

	debe implicar un consumo de memoria RAM superior a 50%.
Medida de respuesta	
Tiempo de respuesta del sistema no mayor de 5s.	

2.8 Arquitectura del sistema.

La arquitectura de un sistema constituye un modelo relativamente pequeño e intelectualmente comprensible de cómo está estructurado el sistema, cómo trabajan juntos sus componentes y la relación entre ellos. La arquitectura destaca las decisiones iniciales relacionadas con el diseño que tendrán un impacto profundo en todo el trabajo de la ingeniería del software que le sigue y lo que también resulta importante, en el éxito final del sistema.(31)

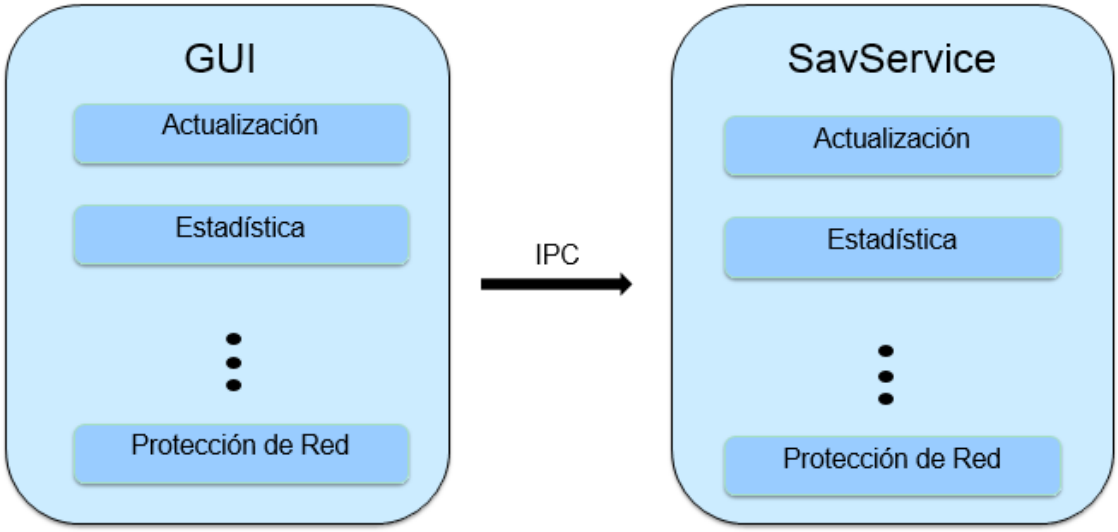


Figura 5: Arquitectura basada en Plugins.

El servidor de control de Antivirus (SavService), contiene los componentes de los servicios del antivirus Segurmática, este se debe de ejecutar con permisos del administrador y estar funcionando correctamente para que los subproyectos que dependan se ejecuten y funcionen correctamente. Cada uno de estos plugin del subproyecto SavService se relaciona directamente con el plugin asociado a dicho servicio en consola y en la interfaz gráfica de usuario (GUI).

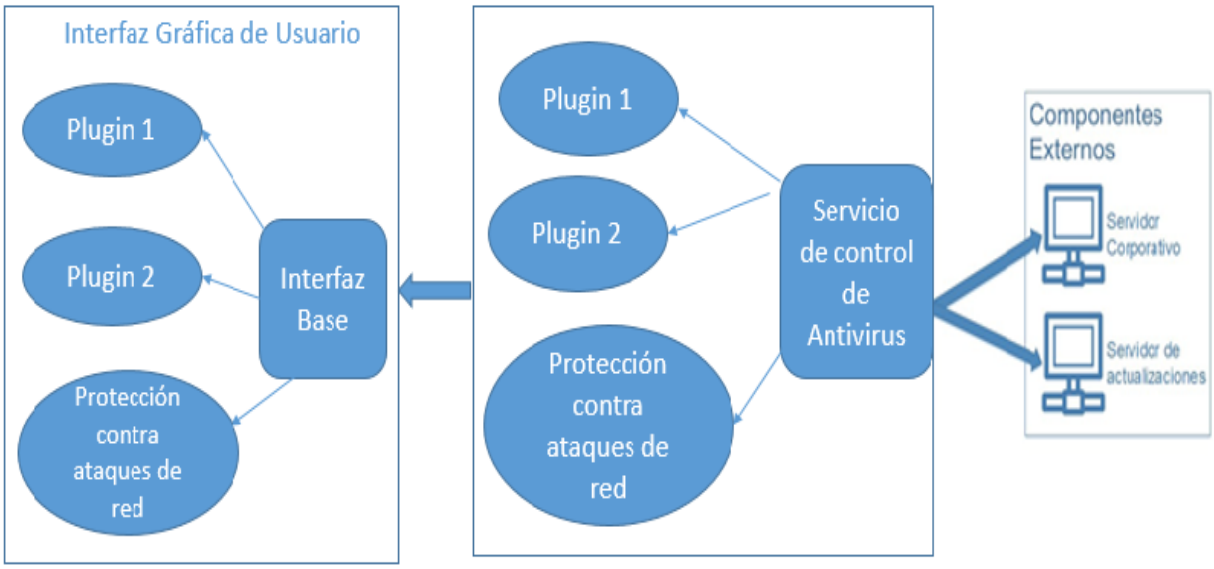


Figura 6: Descripción de la arquitectura.

2.9 Estructura de la aplicación

La solución se encuentra dividida en subproyectos, los cuales se presentan a continuación en la Figura 6. Donde:

- **CLI**: Subproyecto de interfaz de línea de comandos (consola).
    - CLI: Aplicación principal de la interfaz.
    - Plugins: Plugins de la interfaz.
- **GUI**: Subproyecto de interfaz gráfica de usuario.
    - GUI: Aplicación principal de la interfaz.
    - Plugins: Plugins de la interfaz.
    - SharedInterfaces: Interfaces para la comunicación entre la aplicación principal y sus plugins.
- **SavService**: Subproyecto de servicio
    - SAVService: Aplicación principal del servicio antivirus.
    - Plugins: Plugins de los servicios del antivirus.
- **SavCore**: Subproyecto de componentes base comunes para el resto de los subproyectos. Contiene interfaces, implementaciones genéricas y modelos útiles. Se emplea durante la compilación de los otros subproyectos. Una vez compilados este subproyecto ya no es necesario y no forma parte de la aplicación compilada.

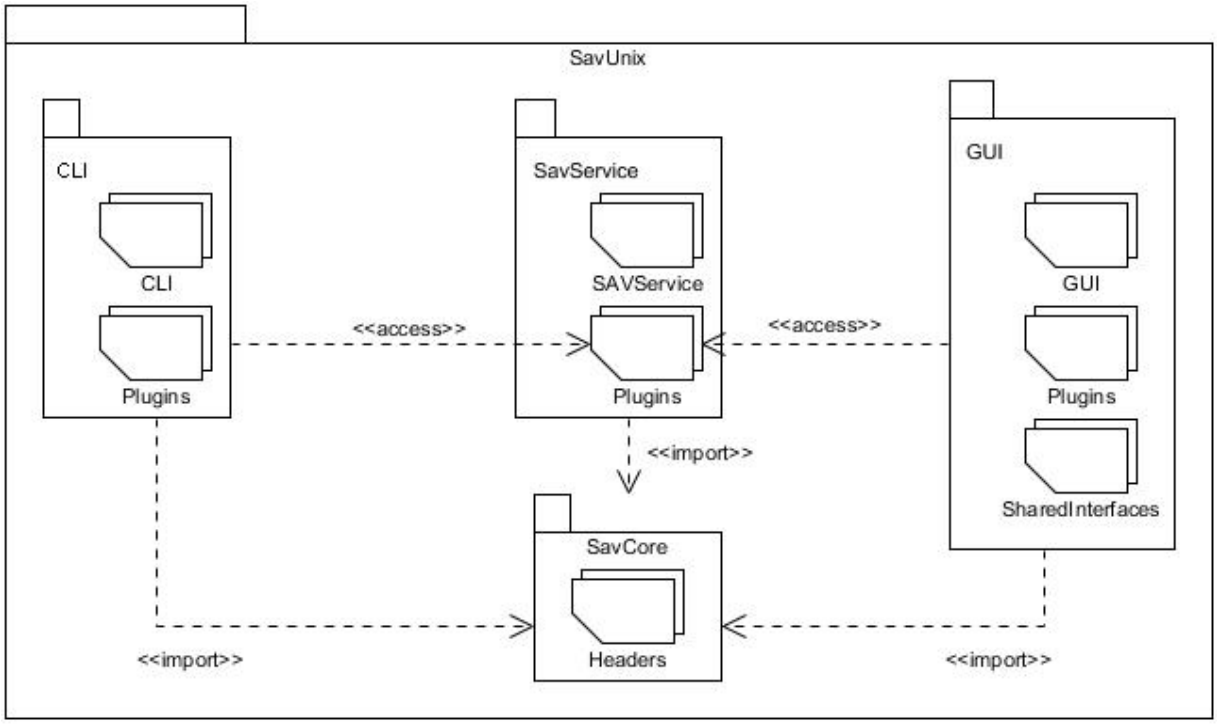


Figura 7: Estructura de la aplicación

La comunicación con SavService se realiza mediante IPC[7], una vez que la aplicación esta compilada. Cuando se compila el subproyecto SavCore deja de existir, ya que los componentes que requieren los otros subproyectos y que están contenidos en él se transfieren a estos al compilarlos.

2.10 Modelo de Base de datos.

El diagrama de modelo de base de datos es un conjunto de conceptos que permite describir los datos, las relaciones entre ellos, la semántica y las restricciones de consistencia. El modelo entidad-relación es uno de los enfoques de representación de datos más utilizado debido a su simplicidad y legibilidad. Mediante ella se almacena las configuraciones y reglas definidas por el usuario y que luego mediante el servicio se podrán realizar utilizando la librería Libcap.

[7] IPC: Inter-Process Communication (comunicación entre procesos).

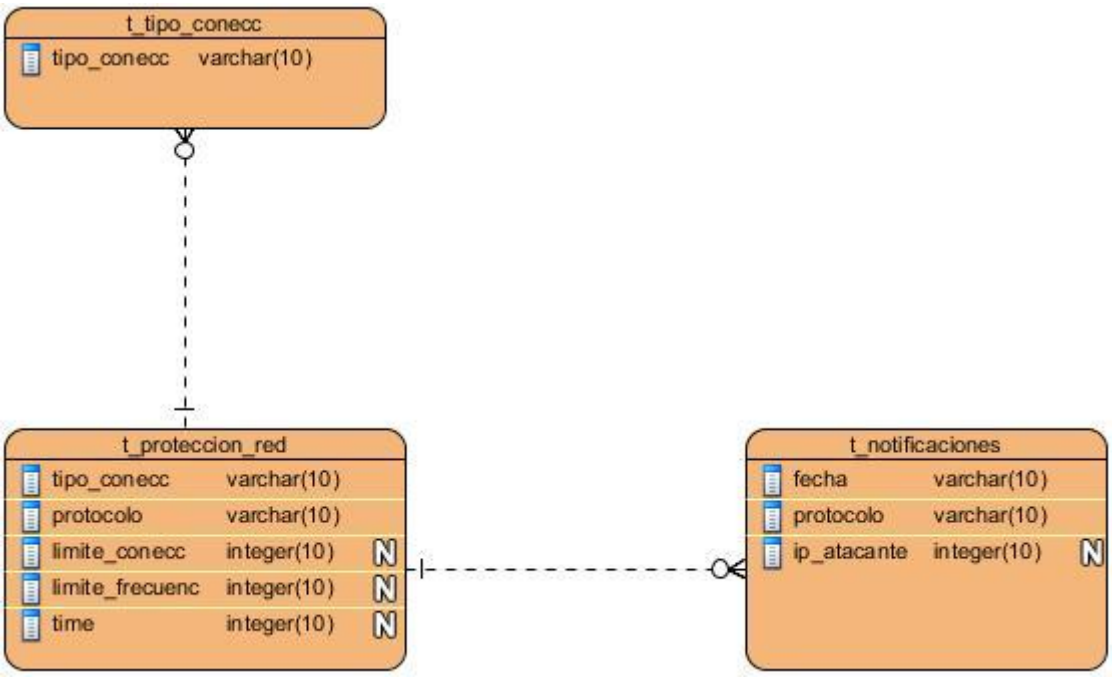


Figura 8: Modelo de base de datos.

2.11 Diagrama de Despliegue

El diagrama de despliegue ayuda a modelar el aspecto físico de un sistema de software orientado a objetos. Modela la configuración del tiempo de ejecución en una vista estática y visualiza la distribución de componentes en una aplicación. En la mayoría de los casos, implica el modelado de las configuraciones de hardware junto con los componentes de software que perduraron.(32)

A continuación, se muestra el diagrama de despliegue en el cual se describe el despliegue físico de la información generada por el programa de software en los componentes de hardware.

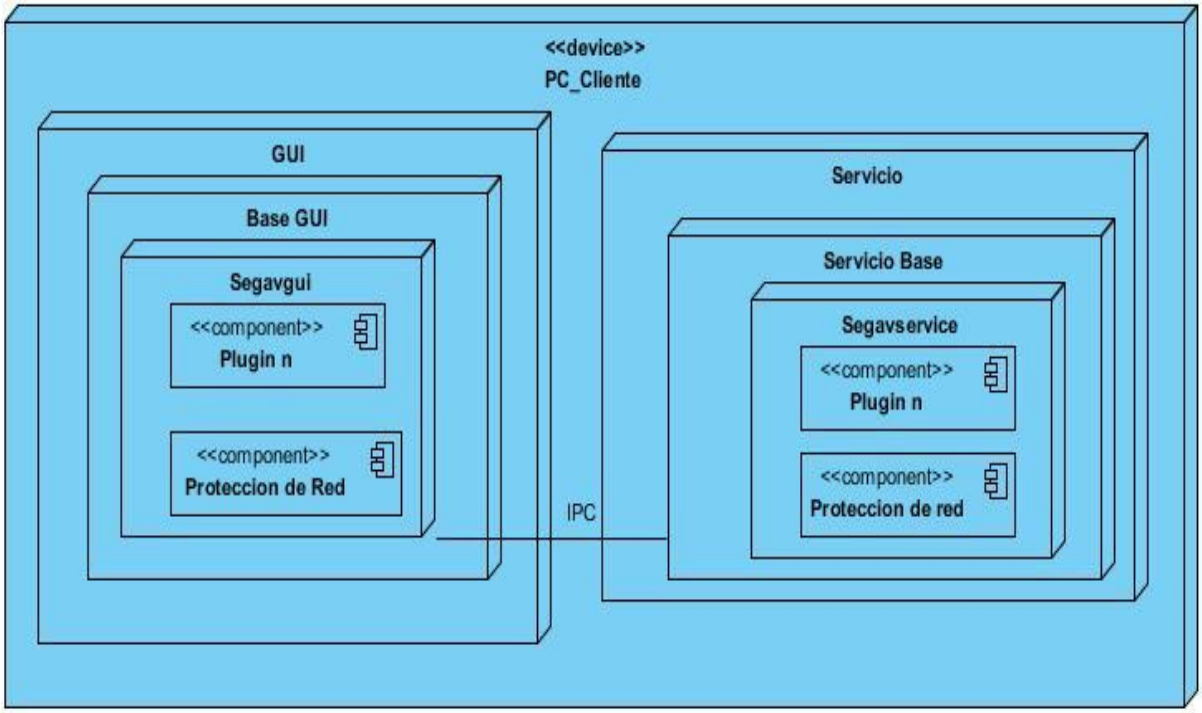


Figura 9: Diagrama de despliegue.

Conclusiones del capítulo

1. Mediante la representación del modelo de dominio, su pudo identificar las entidades relacionadas con el sistema y las relaciones entre ellas.
2. El levantamiento de los requisitos funcionales de la aplicación permitió dar respuesta a las necesidades del problema, cómo el sistema debe responder ante cada solicitud del usuario en cada momento. Se identificaron los requisitos no funcionales los que definieron las restricciones del mismo.
3. La definición de la propuesta de solución del problema, detallándola con la ayuda de los artefactos propuestos por la metodología AUP-UCI, posibilitó estructurar todo el proceso de desarrollo.
4. La realización de los diagramas de componentes permitió comprender la interacción entre los componentes que conforman el sistema y mostrar las dependencias que existen entre los mismos.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Partiendo del resultado del análisis y diseño, en el presente capítulo se realizará la validación de la propuesta de solución a partir de los métodos y técnicas definidos. Se ejecutarán pruebas funcionales para validar el correcto funcionamiento de la herramienta desarrollada, con este objetivo se diseñan los casos de pruebas pertinentes y se realizan pruebas unitarias para valorar la factibilidad de la herramienta desarrollada.

3.1 Patrones de Diseño

Los patrones de diseño facilitan un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. Tienden a ser independientes de los lenguajes y paradigmas de programación y su aplicación no afecta necesariamente al sistema completo, pero sí a un subsistema o parte del mismo.

Para la definición de las clases del sistema, es importante la revisión de algunos patrones que permiten realizar un diseño adecuado y consistente. (34)

3.1.1 Patrones de Diseño GRASP

Los patrones GRASP (Patrones Generales de Asignación de Responsabilidades) son patrones de diseño que se usan para asignar responsabilidades a una clase. A continuación se muestran los más utilizados en la solución propuesta(34).

Experto: Asignar una responsabilidad a la clase más competente en información, la clase cuenta con la información necesaria para cumplir la responsabilidad. Es el principio básico de asignación de responsabilidades que suele utilizarse en el diseño orientado a objetos.

Las clases TestPlugin contarán con la información necesaria para cumplir cada una las responsabilidades que le corresponden. De este modo se obtiene un diseño con mayor cohesión y un bajo acoplamiento.

Controlador: Se aplica para realizar las asignaciones en cuanto al manejo de los eventos del sistema y definir sus operaciones.(35)

Las clases testplugin.cpp y tespluginsvc.cpp son las responsables de atender los eventos del sistema en cuanto a los eventos de los agentes y los canales de comunicación.

Bajo acoplamiento: Soporta el diseño de clases más independientes. Asigna las responsabilidades de forma tal que las clases se comuniquen con el menor número de clases que sea posible.

A la clase testplugin.h se les asignan responsabilidades de forma tal que solo se comuniquen con las clases que se encargan de integrar, con el uso de este patrón se fortalece la reutilización de código y se disminuye la dependencia entre las clases.

Alta cohesión: Asignar una responsabilidad de modo que la unión se mantenga a gran escala. Asignar a las clases responsabilidades que trabajen sobre una misma área de aplicación y que no tengan mucha complejidad. Mejoran la claridad y facilidad con que se entiende el diseño.(36)

La clase testplugin.cpp se destinan a responsabilidades específicas que pertenecen a la misma área de la aplicación y que no tengan mucha complejidad.

3.1.2 Patrones de Diseño GOF

Los patrones GOF (Gand of Four, Banda de los Cuatro) son patrones de diseño que definen una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular(37). Estos patrones se dividen en tres categorías: los creacionales, los estructurales y los de comportamiento.(38) Seguidamente se presentan los patrones empleados para el desarrollo de la solución propuesta.

- **Creacionales:** los patrones creacionales abstraen el proceso de creación de instancias y analizan los detalles de cómo los objetos son creados o inicializados.
- **Estructurales:** los patrones estructurales se ocupan de cómo las clases y objetos se combinan para formar grandes estructuras y proporcionan nuevas funcionalidades.
- **Comportamiento:** los patrones de comportamiento están relacionados con los algoritmos y la asignación de responsabilidades entre los objetos. Son utilizados para organizar, manejar y combinar comportamientos.

Seguidamente se presentan los patrones empleados para el desarrollo de la solución propuesta:

Creacionales

- ✓ Singleton: Este patrón consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella.

Estructurales

- ✓ Fachada: Este patrón se emplea para brindar una interfaz que abstrae completamente al usuario de la complejidad de los procesos. Este patrón es evidenciado en la clase TestPlugingvi.cpp, la cual solo contiene y presenta al usuario final las funcionalidades que son de interés para él.

- ✓ Decorador (decorator): Patrón que responde a la necesidad de añadir dinámicamente funcionalidades a un objeto. Es usado para utilizar funciones de otros módulos o del SavService en el módulo protección, permitiendo no tener que crear estas funciones nuevamente sino decorarlas de acuerdo con la necesidad que se tenga.

3.2 Pruebas

El desarrollo del software ha de ir acompañado de alguna actividad que garantice la calidad del software, la prueba es un elemento crítico para ello. Es por ello que se deben incorporar acciones que evalúen la calidad del producto que se está desarrollando. Dentro del proceso de desarrollo de un software, el flujo de trabajo de prueba, es mediante el que se puede validar que las suposiciones hechas en el diseño y los requerimientos se estén cumpliendo satisfactoriamente, por lo que se encarga de verificar que el producto funcione como se diseñó y que los requerimientos se cumplan adecuadamente. Este flujo de trabajo brinda soporte para encontrar, documentar y solucionar defectos en el sistema(39).

3.2.1 Pruebas Unitarias

Al desarrollar un nuevo software una de las pruebas que no se deben pasar por alto son las pruebas unitarias, específicamente la técnica de pruebas de caja blanca. Para realizar este tipo de pruebas el personal debe estar familiarizado en el uso de herramientas con este fin, además de conocer el lenguaje de programación en el que se está desarrollando el sistema. Las pruebas unitarias permiten conocer si el funcionamiento de un módulo de código es correcto, con el fin de garantizar que cada módulo funcione por separado de la manera correcta. El objetivo de estas pruebas es alejar determinadas partes del código y demostrar que no tienen errores, probándole de este modo al programador que la solución está libre de errores lógicos de programación, esto se evidencia si el probador introduce determinados datos al sistema y los valores que se obtienen son los esperados.

En la actualidad existen una gran cantidad de herramientas que tienen como objetivo la realización de estas pruebas. CppUnit es un ejemplo de estas, y precisamente es la herramienta seleccionada para realizar las pruebas unitarias a la aplicación. A continuación en la figura se evidencia un ejemplo del resultado de las pruebas.

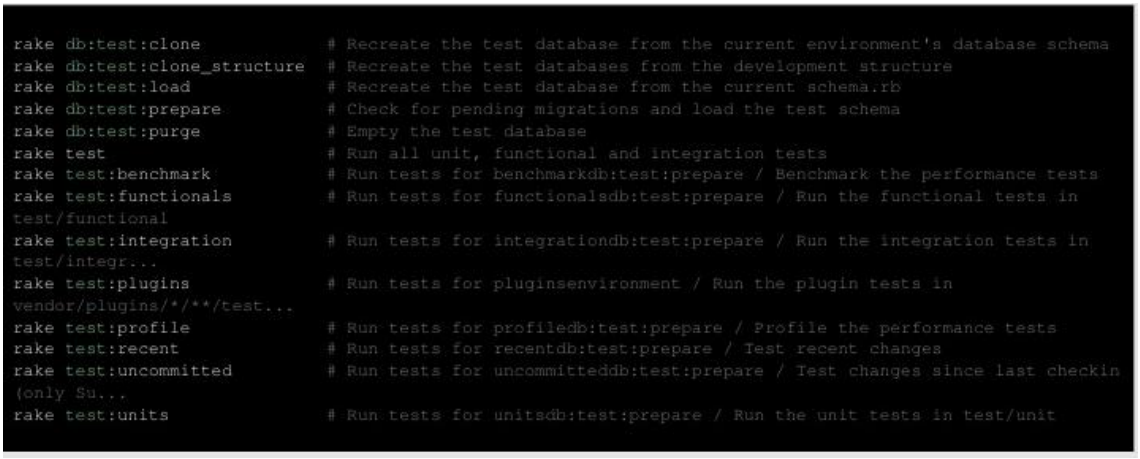


Figura 10: Prueba Unitaria a la aplicación.

3.2.2 Prueba de aceptación

Las pruebas de aceptación, también llamadas pruebas funcionales son supervisadas por el cliente basándose en los requerimientos tomados de las descripciones de requisitos funcionales. Son pruebas de caja negra, que representan un resultado esperado de determinada transacción con el sistema. Para que una descripción de requisito funcional se considere aprobada, deberá pasar todas las pruebas de aceptación elaboradas para dicha descripción(40). Estas pruebas de aceptación se realizan para evaluar el grado de calidad del software de acuerdo a todos los aspectos relevantes que intervienen en el sistema.

3.2.3 Diseño de casos de prueba

Los casos de prueba son la forma de verificar las diversas funcionalidades existentes en un producto de software descritas en el formato de los Casos de Uso desarrollados para cumplir un objetivo en particular o una función esperada (43).

Debe verificar si el producto satisface los requerimientos del usuario y se comporta como se desea, tal y como se describe en la especificación de los requerimientos.

Caso de pruebas del caso de uso: Gestionar configuración.

Tabla 8: Caso de prueba del Caso de uso Gestionar configuración.

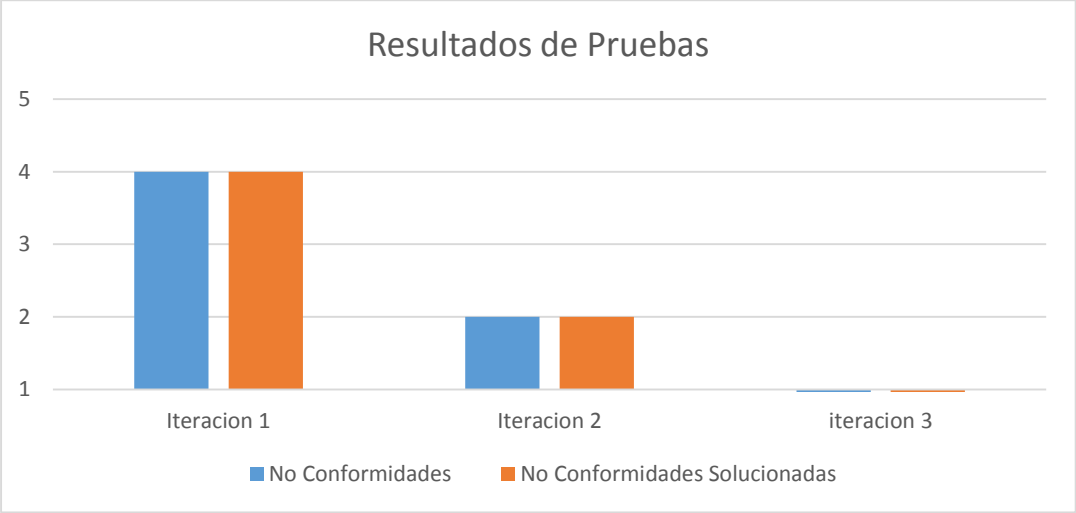
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar Modo sigiloso	El usuario debe seleccionar el modo sigiloso.	Se selecciona el modo sigiloso	Seleccionar Modo Sigiloso el cual permite las actividades de red que son iniciadas por el usuario y el resto de las acciones como conexiones remotas hacia su PC, acceso a puertos, no estarán permitidas.
EC 1.2 Seleccionar No notificar ataques de Red	El usuario Selecciona No notificar ataques de Red.	El sistema no mostrara ninguna notificación de ataques de red	El sistema no mostrará ninguna notificación de ataques de red
EC 1.3 Seleccionar la opción Bloquear IP atacante	El usuario Selecciona la opción Bloquear IP atacante durante el tiempo que determine	El sistema bloqueará el IP atacante durante el tiempo definido por el usuario	El sistema Bloqueará el IP atacante

3.3 Resultados de las Pruebas

Durante la fase de prueba se realizaron tres iteraciones al Módulo de Protección contra ataque de red. En la primera iteración se detectó que la herramienta para gestionar las reglas no validaba correctamente los campos y permitía la entrada de caracteres no alfanuméricos en el campo de texto, además permitía la entrada de caracteres extraños en el campo de texto perteneciente al tipo de Host. Estas inconformidades fueron corregidas y luego se volvió a realizar otra iteración de las pruebas para el caso de uso Gestionar Reglas, arrojando resultados satisfactorios.

Se diseñaron un total de 4 diseños de casos de prueba de aceptación divididos en 3 iteraciones. Para una primera iteración se detectaron 4 no conformidades, en la segunda iteración se detectaron 2 no conformidades y en la tercera iteración realizada, no se detectaron no conformidades y fueron corregidas las no conformidades detectadas con anterioridad.

Figura 11: Resultados de Pruebas



Conclusiones del capítulo

En este capítulo se realizó una descripción de la implementación y las pruebas realizadas al sistema. En el mismo se generaron los artefactos necesarios para la implementación y las pruebas del sistema, por lo que se puede llegar a las siguientes conclusiones:

- ✓ Se identificaron los patrones de diseño GRASP y GOF utilizados en el desarrollo de la aplicación, permitiendo facilitar la asignación de responsabilidades logrando un diseño de software que sirva de apoyo a la implementación del sistema.
- ✓ Mediante las pruebas de caja negra seleccionadas para validar el cumplimiento de los requisitos funcionales establecidos en el análisis del sistema, se lograron encontrar problemas funcionales presentes en el sistema.
- ✓ Con la realización de las pruebas de caja negra y sus resultados satisfactorios en la última iteración de prueba se puede concluir que el software no presenta ningún error funcional, por lo que su etapa de desarrollo culminó exitosamente.

Conclusiones

Con la realización de la presente investigación se adquirieron y pusieron en práctica los conocimientos necesarios para el desarrollo del módulo de Protección contra ataques de red para el software Segurmática antivirus para Linux, de esta forma se le dio cumplimiento al objetivo general, así como a las tareas de la investigación, arribando a las siguientes conclusiones:

- ✓ El desarrollo de la investigación dio lugar a la creación de las bases para el desarrollo de la presente investigación.
- ✓ El estudio de los conceptos asociados al dominio del problema ayudó en gran medida a la comprensión de los principales términos que se manejarían durante la investigación.
- ✓ La realización de un estudio a las soluciones similares existentes ayudó a determinar las principales características positivas y negativas con que cuentan los sistemas actuales de protección, tomando como premisas para el desarrollo del módulo las características positivas de estos sistemas.
- ✓ El análisis de las tecnologías y herramientas de desarrollo de software determinó la selección de las mismas para el desarrollo del sistema, por las características que presentaban además de permitir una mayor integración con los sistemas.
- ✓ La descripción y el diseño del módulo de Protección contra ataques de red, permitió crear las bases para la implementación de dicho sistema, evaluando y restringiendo cada posibilidad de error, para lograr un sistema completamente funcional.
- ✓ Las pruebas realizadas al módulo de Protección contra ataques de red, ayudaron a corregir inconformidades que no se tuvieron en cuenta durante la implementación, las mismas ayudaron a refinar el producto, liberándolo de cualquier inconformidad.

.

Recomendaciones

Para futuras investigaciones se recomienda:

- Investigar técnicas y mecanismos que permitan detectar posibles ataques de red basado en patrones de comportamiento para ser integrados a la solución desarrollada en la presente investigación.
- Integrar los módulos de protección contra ataques de red y cortafuegos en el software Segurmática Antivirus para Linux, para obtener un antivirus más completo y se garantice en mayor medida la prevención de ataques de red y seguridad de los sistemas.

Bibliografía

Informática Aplicada a las Ciencias Sociales. Historia de la Informática. Rafael Barzanallana. UMU. [online]. [Accessed 21 November 2017]. Available from: http://www.um.es/docencia/barzana/IACCSS/Historia-de-la-informatica.html

Definición de seguridad informática - Qué es, Significado y Concepto. [online]. Available from: https://definicion.de/seguridad-informatica/

¿Qué es un antivirus ? [online]. Available from: https://www.informatica-hoy.com.ar/software-seguridad-virus-antivirus/Que-es-un-antivirus.php

Qué diferencia a un antivirus de un firewall y de un IDS. [online]. [Accessed 6 December 2017]. Available from: https://www.pabloyglesias.com/antivirus-firewall-e-ids/

Ataques a las redes : Listado de diferentes ataques a las redes de ordenadores. [online]. [Accessed 31 October 2017]. Available from: https://www.redeszone.net/2010/11/03/ataques-a-las-redes-listado-de-diferentes-ataques-a-las-redes-de-ordenadores/

Detección de intrusos. [online]. [Accessed 5 December 2017]. Available from: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-es-4/ch-detection.html

SCARFONE, Karen. *Guide to Intrusion Detection and Prevention Systems (IDPS)* [online]. 2007. Available from: https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=50951.

Historia | IDS Comercial. [online]. [Accessed 25 May 2018]. Available from: http://www.ids.com.mx/nuestra-empresa/historia

Segurmática Antivirus | Segurmática. [online]. Available from: http://www.segurmatica.cu/segavprod

Consecuencias de un ataque DDoS. [online]. [Accessed 1 December 2017]. Available from: http://www.aratecnia.es/ataque-ddos-consecuencias-seguridad-informatica/

Definición de ataque informático. [online]. [Accessed 7 November 2017]. Available from: http://www.alegsa.com.ar/Dic/ataque_informatico.php

Definición de Seguridad Informática | Gestión de riesgo en la seguridad informática. [online]. [Accessed 7 November 2017]. Available from: https://protejete.wordpress.com/gdr_principal/definicion_si/

¿A qué se denomina Sistema de Prevención de Intrusos o IPS? - Soporte Técnico Panda Security. [online]. [Accessed 25 May 2018]. Available from: https://www.pandasecurity.com/usa-es/support/card?id=31452

Sistema de detección de intrusiones (IDS). [online]. [Accessed 7 November 2017]. Available from: http://es.ccm.net/contents/162-sistema-de-deteccion-de-intrusiones-ids

Seguridad Informatica / IDS - Detección de Intrusos en Tiempo Real. [online]. [Accessed 21 November 2017]. Available from: http://www.segu-info.com.ar/proteccion/deteccion.htm

GÓMEZ VIEITES, Álvaro. TIPOS DE ATAQUES E INTRUSOS EN LAS REDES INFORMÁTICAS. . P. 13.

VANESSA, Viñes Sanjuan. *Analisis de los sistemas de prevencion de intruso* [online]. 2004. Proyecto Fin de Carrera. Available from: http://deim.urv.cat/~pfc/docs/pfc375/d1126516530.pdf

IDS, Firewall and Antivirus: what you need to have installed? [online]. [Accessed 6 December 2017]. Available from: https://www.welivesecurity.com/2015/04/30/ids-firewall-antivirus-need-installed/

Presupuesto participativos (PPs) e Instituciones Participativas (IPs) en Brasil: Criterios (y marco) para la evaluación de experiencias y casos - ProQuest. [online]. [Accessed 6 June 2018]. Available from: https://search.proquest.com/openview/1908f04d6d3bf72673709e9f4ad6ef2a/1?pq-origsite=gscholar&cbl=2046214

IPS vs. IDS: Mejor prevenir que curar | | CIO. [online]. [Accessed 5 June 2018]. Available from: http://www.ciospain.es/archive/ips-vs-ids-mejor-prevenir-que-curar

Mejores IDS Opensource para Detección de Intrusiones – Proteger mi PC. [online]. [Accessed 11 January 2018]. Available from: https://protegermipc.net/2017/02/22/mejores-ids-opensource-deteccion-de-intrusiones/

Suricata | Open Source IDS / IPS / NSM engine. [online]. [Accessed 13 February 2018]. Available from: https://suricata-ids.org/

What's New in Visual Paradigm? [online]. [Accessed 6 December 2017]. Available from: https://www.visual-paradigm.com/whats-new/

Lenguaje de programación C++ | Aprendiendo Arduino. [online]. [Accessed 5 June 2018]. Available from: https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/

Libraries & APIs, Tools and IDE | Qt. [online]. [Accessed 3 June 2018]. Available from: https://www.qt.io/qt-features-libraries-apis-tools-and-ide/

SQLite Home Page. [online]. [Accessed 3 June 2018]. Available from: https://www.sqlite.org/index.html

OpenLibra | Aprendiendo a programar con Libpcap. [online]. [Accessed 23 May 2018]. Available from: https://openlibra.com/es/book/aprendiendo-a-programar-con-libpcap

CRAIG, Larman. *Una introducción al análisis y diseño orientado a objetos y al proceso unificado.*

Ingeniería de requisitos | Marco de Desarrollo de la Junta de Andalucía. [online]. [Accessed 19 January 2018]. Available from: http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/ingenieria/ingenieria-requisitos

Técnicas para Identificar Requisitos Funcionales y No Funcionales - Metodología Gestión de Requerimientos. [online]. [Accessed 23 February 2018]. Available from: https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales

PRESSMAN, Roger S. *Software Engineering Chapter 9: Architectural Design*.

Una Metodología para el Modelado de Sistemas de Ingeniería Orientad... [online]. [Accessed 10 April 2018]. Available from: http://www.redalyc.org/html/925/92513102003/

UML Modeling - Unified Modeling Language Tool. [online]. [Accessed 26 February 2018]. Available from: https://www.visual-paradigm.com/VPGallery/diagrams/index.html

¿Qué es un Patrón de Diseño? [online]. [Accessed 10 May 2018]. Available from: https://msdn.microsoft.com/es-es/library/bb972240.aspx

POLO, Uasaolo. *Patrones Grap*.

Alta cohesión y bajo acoplamiento - Diseño de Software | El Blog de Julio Pari. [online]. [Accessed 26 May 2018]. Available from: http://juliopari.com/alta-cohesion-y-bajo-acoplamiento-diseno-de-software/

Un modelo formal de patrones orientados a objetos. [online]. [Accessed 26 May 2018]. Available from: http://sedici.unlp.edu.ar/handle/10915/22146

Carlos A. Guerrero, Johanna M. Suárez, Luz E. Gutiérrez**.** *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web*. [online]. [Accessed 10 May 2018]. Available from: https://scielo.conicyt.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext

Tipos de pruebas de software. [online]. [Accessed 15 May 2018]. Available from: https://es.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software

Comparativa práctica de las pruebas en entornos tradicionales y ágiles. [online]. [Accessed 24 May 2018]. Available from: http://www.redalyc.org/html/922/92217159004/

MISAS, Arango. *Una caja negra a luz de las redes neuronales*.

Diogo Thimoteo da Cunha, Rafaela Ribeiro de Brito Métodos para aplicar las pruebas de aceptación para la alimentación escolar: validación de la tarjeta lúdica. [online],ISSN 0717-7518. [Accessed 24 May

2018]. Available from: https://scielo.conicyt.cl/scielo.php?pid=S0717-75182013000400005&script=sci_arttext

Casos de Uso vs. Casos de Prueba - Testeando Software. [online]. [Accessed 15 May 2018]. Available from: https://testeandosoftware.com/casos-de-uso-vs-casos-de-prueba/

Toro, A., & Jiménez, B. Metodología para la Elicitación de Requisitos de Sistemas Software Versión 2.1. InformeNTécnico LSI-2000-10. Facultad de Informática y Estadística. Sevilla, España (2000).

Guerrero, A., & Suárez, J. *Patrones de diseño para el desarrollo de aplicaciones Web - Construyendo software con buenas prácticas*. (Sic) Editorial Ltda. Colombia (2010).

Gracia, J. Patrones de Diseño. Ingeniero Software, Análisis y Diseño [en línea] (2005), http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php.

Cristian L. Vidal, Rodolfo F Schmal, Sabino Rivero y Rodolfo H. Villarroel. *Extensión del Diagrama de Secuencias UML (Lenguaje de Modelado Unificado) para el Modelado Orientado a Aspectos*. Revista Información Tecnológica. ISSN: 0718-0764. [En línea]. Vol. 23(6), (2012). https://scielo.conicyt.cl/pdf/infotec/v23n6/art07.pdf.

Bahit, Eugenia. 2010. eByte. Introducción al patrón arquitectónico MVC. [En línea] 2010. http://www.eugeniabahit.com/mvc/.

Referencias

1. Informática Aplicada a las Ciencias Sociales. Historia de la Informática. Rafael Barzanallana. UMU. [online]. [Accessed 21 November 2017]. Available from: http://www.um.es/docencia/barzana/IACCSS/Historia-de-la-informatica.html

2. Definición de seguridad informática - Qué es, Significado y Concepto. [online]. Available from: https://definicion.de/seguridad-informatica/

3. ¿Qué es un antivirus? [online]. Available from: https://www.informatica-hoy.com.ar/software-seguridad-virus-antivirus/Que-es-un-antivirus.php

4. #MundoHacker: Qué diferencia a un antivirus de un firewall y de un IDS. [online]. [Accessed 6 December 2017]. Available from: https://www.pabloyglesias.com/antivirus-firewall-e-ids/

5. Ataques a las redes : Listado de diferentes ataques a las redes de ordenadores. [online]. [Accessed 31 October 2017]. Available from: https://www.redeszone.net/2010/11/03/ataques-a-las-redes-listado-de-diferentes-ataques-a-las-redes-de-ordenadores/

6. Detección de intrusos. [online]. [Accessed 5 December 2017]. Available from: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-es-4/ch-detection.html

7. SCARFONE, Karen. *Guide to Intrusion Detection and Prevention Systems (IDPS)* [online]. 2007. Available from: https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=50951.

8. Historia | IDS Comercial. [online]. [Accessed 25 May 2018]. Available from: http://www.ids.com.mx/nuestra-empresa/historia

9. Segurmática Antivirus | Segurmática. [online]. Available from: http://www.segurmatica.cu/segavprod

10. Consecuencias de un ataque DDoS. [online]. [Accessed 1 December 2017]. Available from: http://www.aratecnia.es/ataque-ddos-consecuencias-seguridad-informatica/

11. Definición de ataque informático. [online]. [Accessed 7 November 2017]. Available from: http://www.alegsa.com.ar/Dic/ataque_informatico.php

12. Definición de Seguridad Informática | Gestión de Riesgo en la Seguridad Informática. [online]. [Accessed 7 November 2017]. Available from: https://protejete.wordpress.com/gdr_principal/definicion_si/

13. ¿A qué se denomina Sistema de Prevención de Intrusos o IPS? - Soporte Técnico Panda Security. [online]. [Accessed 25 May 2018]. Available from: https://www.pandasecurity.com/usa-es/support/card?id=31452

14. Sistema de detección de intrusiones (IDS). [online]. [Accessed 7 November 2017]. Available from: http://es.ccm.net/contents/162-sistema-de-deteccion-de-intrusiones-ids

15. Seguridad Informatica / IDS - Detección de Intrusos en Tiempo Real. [online]. [Accessed 21 November 2017]. Available from: http://www.segu-info.com.ar/proteccion/deteccion.htm

16. GÓMEZ VIEITES, Álvaro. TIPOS DE ATAQUES E INTRUSOS EN LAS REDES INFORMÁTICAS. . P. 13.

17. VANESSA, Viñes Sanjuan. *Analisis de los sistemas de prevencion de intruso* [online]. 2004. Proyecto Fin de Carrera. Available from: http://deim.urv.cat/~pfc/docs/pfc375/d1126516530.pdf

18. IDS, Firewall and Antivirus: what you need to have installed? [online]. [Accessed 6 December 2017]. Available from: https://www.welivesecurity.com/2015/04/30/ids-firewall-antivirus-need-installed/

19. Presupuesto Participativos (PPs) e Instituciones Participativas (IPs) en Brasil: Criterios (y marco) para la evaluación de experiencias y casos - ProQuest. [online]. [Accessed 6 June 2018]. Available from: https://search.proquest.com/openview/1908f04d6d3bf72673709e9f4ad6ef2a/1?pq-origsite=gscholar&cbl=2046214

20. IPS vs. IDS: Mejor prevenir que curar | | CIO. [online]. [Accessed 5 June 2018]. Available from: http://www.ciospain.es/archive/ips-vs-ids-mejor-prevenir-que-curar

21. Mejores IDS Opensource para Detección de Intrusiones – Proteger mi PC. [online]. [Accessed 11 January 2018]. Available from: https://protegermipc.net/2017/02/22/mejores-ids-opensource-deteccion-de-intrusiones/

22. Suricata | Open Source IDS / IPS / NSM engine. [online]. [Accessed 13 February 2018]. Available from: https://suricata-ids.org/

23. What's New in Visual Paradigm? [online]. [Accessed 6 December 2017]. Available from: https://www.visual-paradigm.com/whats-new/

24. Lenguaje de programación C++ | Aprendiendo Arduino. [online]. [Accessed 5 June 2018]. Available from: https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/

25. Libraries & APIs, Tools and IDE | Qt. [online]. [Accessed 3 June 2018]. Available from: https://www.qt.io/qt-features-libraries-apis-tools-and-ide/

26. SQLite Home Page. [online]. [Accessed 3 June 2018]. Available from: https://www.sqlite.org/index.html

27. OpenLibra | Aprendiendo a programar con Libpcap. [online]. [Accessed 23 May 2018]. Available from: https://openlibra.com/es/book/aprendiendo-a-programar-con-libpcap

28. CRAIG, Larman. *Una introducción al análisis y diseño orientado a objetos y al proceso unificado.*

29. Ingeniería de requisitos | Marco de Desarrollo de la Junta de Andalucía. [online]. [Accessed 19 January 2018]. Available from: http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/ingenieria/ingenieria-requisitos

30. Técnicas para identificar requisitos funcionales y no funcionales - Metodología Gestión de Requerimientos. [online]. [Accessed 23 February 2018]. Available from: https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales

31. PRESSMAN, Roger S. *Software Engineering Chapter 9: Architectural Design*.

32. Una Metodología para el Modelado de Sistemas de Ingeniería Orientad... [online]. [Accessed 10 April 2018]. Available from: http://www.redalyc.org/html/925/92513102003/

33. UML Modeling - Unified Modeling Language Tool. [online]. [Accessed 26 February 2018]. Available from: https://www.visual-paradigm.com/VPGallery/diagrams/index.html

34. ¿Qué es un Patrón de Diseño? [online]. [Accessed 10 May 2018]. Available from: https://msdn.microsoft.com/es-es/library/bb972240.aspx

35. POLO, Uasaolo. *Patrones Grap*. [no date].

36. Alta cohesión y bajo Acoplamiento - Diseño de Software | El Blog de Julio Pari. [online]. [Accessed 26 May 2018]. Available from: http://blog.juliopari.com/alta-cohesion-y-bajo-acoplamiento-diseno-de-software/

37. Un modelo formal de patrones orientados a objetos. [online]. [Accessed 26 May 2018]. Available from: http://sedici.unlp.edu.ar/handle/10915/22146

38. Carlos A. Guerrero, Johanna M. Suárez ,Luz E. Gutiérrez**.** *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web*. [online]. [Accessed 10 May 2018]. Available from: https://scielo.conicyt.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext

39. Tipos de pruebas de software. [online]. [Accessed 15 May 2018]. Available from: https://es.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software

40. Comparativa práctica de las pruebas en entornos tradicionales y ágiles. [online]. [Accessed 24 May 2018]. Available from: http://www.redalyc.org/html/922/92217159004/

41. MISAS, Arango. *Una caja negra a luz de las redes neuronales*.

42. Diogo Thimoteo da Cunha, Rafaela Ribeiro de Brito Métodos para aplicar las pruebas de aceptación para la alimentación escolar: validación de la tarjeta lúdica. [online],ISSN 0717-7518. [Accessed 24 May 2018]. Available from: https://scielo.conicyt.cl/scielo.php?pid=S0717-75182013000400005&script=sci_arttext

43. Casos de uso vs. Casos de prueba - Testeando Software. [online]. [Accessed 15 May 2018]. Available from: https://testeandosoftware.com/casos-de-uso-vs-casos-de-prueba/

Anexos

Descripciones de casos de uso

CU Gestionar Notificaciones

Objetivo	Gestionar Notificaciones
Actores	Administrador
Resumen	Por defecto el programa informa al usuario cada vez que se produce un intento de ataque a su PC. Se muestra un mensaje con la información sobre el tipo de ataque, la dirección IP del atacante y el puerto local (si es posible determinarlo). Como esta notificación es solo para referencia se puede deshabilitar marcando “No notificar ataques de red”.
Complejidad	Baja
Prioridad	Media
Precondiciones	Debe estar deshabilitado la opción de “No notificar ataques de red”.
Postcondiciones	El sistema debe de permitirle al usuario eliminar, mostrar e imprimir las notificaciones.

Flujo de eventos		
Flujo básico Gestionar Notificaciones		
	Actor	**Sistema**
1.	1.1 Selecciona en la jerarquía del enfoque, la opción que desea gestionar, adicionar, eliminar, modificar o mostrar las notificaciones	
2.		2.1 Si decide Mostrar todas las notificaciones, ir a la opción: - “Mostrar Notificación” • Si decide eliminar una notificación, ir a la opción: - “Eliminar Notificación” • Si decide exportar todas las notificaciones, ir a la opción: - “Exportar Notificaciones”
3.		3.1 Almacena los datos de las notificaciones. Finalizando así el Caso de Uso.
Sección 1: “Eliminar Notificaciones”		
Flujo básico		
	Actor	**Sistema**
1.	1.1 El usuario selecciona la opción Eliminar Notificación	
2.		2.1 Debe eliminar la notificación seleccionada por el usuario.
3.	3.1 Luego de haber eliminado las notificaciones	

	debe de presionar el botón Aceptar	
4.		4.1 Debe de guardar todas las configuraciones realizadas por el usuario.
Sección 1: “Mostrar Notificaciones”		
Flujo básico		
	Actor	**Sistema**
1.	1.1 Selecciona la opción de Mostrar Notificaciones	
2.		2.1 Debe de mostrar todas las notificaciones que existan guardadas.
Sección 1: “Exportar Notificaciones”		
1	1.1 Selecciona la opción de Exportar Notificaciones	
2		2.1 Debe de exportar las notificaciones a un documento.

Relaciones	**CU incluidos**	No aplica
	CU extendidos	No aplica
Requisitos no funcionales	No aplica	
Asuntos pendientes	No aplica	

Prototipo elemental de interfaz gráfica de usuario

Casos de pruebas

- ✓ **Caso de pruebas del caso de uso: Gestionar Reglas**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar Adicionar regla	El usuario debe de llenar los campos de la regla	Adicionar la nueva regla a la tabla de tipo de conexión	El sistema debe de guardar las nuevas reglas en la base de date. Y mostrarla en la tabla de conexiones.
EC 1.2 Seleccionar Eliminar Regla	El usuario debe de seleccionar la regla que desea eliminar	El sistema debe de eliminar la regla que el usuario seleccionó	El sistema elimina la regla seleccionada por el usuario de la tabla y de la base de datos
EC 1.3 Seleccionar Guardar Reglas	El usuario desea guardar las reglas	El sistema guardará todas las reglas que existan	El sistema guardará todas las reglas que existan en la base de datos

- ✓ **Caso de pruebas del caso de uso: Gestionar Notificaciones**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar Mostrar Notificaciones	El usuario desea ver las notificaciones	Adicionar la nueva regla a la tabla de tipo de conexión	El sistema debe de guardar las nuevas reglas en la base de date. Y mostrarla en la tabla de conexiones.
EC 1.2 Seleccionar Eliminar Notificaciones	El usuario debe de seleccionar la notificación que desea eliminar	El sistema debe de eliminar la notificación que el usuario seleccionó	El sistema elimina la notificación seleccionada por el usuario de la tabla y de la base de datos
EC 1.3 Seleccionar Guardar Notificaciones	El usuario desea guardar las notificaciones	El sistema guardará todas las notificaciones que existan	El sistema guardará todas las notificaciones que existan en la base de datos

✓ **Caso de pruebas del caso de uso: Gestionar Protección**

Escenario	**Descripción**	**Respuesta del sistema**	**Flujo central**
EC 1.1 Seleccionar Activar protección	El usuario desea activar la protección	Iniciar la protección del sistema	El sistema debe de guardar configuraciones y activar la protección contra ataques de red.
EC 1.2 Seleccionar Desactivar Protección	El usuario desea desactivar la protección	El sistema debe desactivar la protección contra ataques de red	El sistema debe de guardar configuraciones y desactivar la protección contra ataques de red.

Glosario de términos

Antivirus: Software capaz de detectar y prevenir el accionar de los programas malignos o, en su defecto, combatirlos o minimizar su impacto.

Ataque informático: es un método por el cual un individuo, mediante un sistema informático, intenta tomar el control, desestabilizar o dañar otro sistema informático (ordenador, red privada, etcétera).

Interfaz visual. Es la encargada de garantizar la interacción entre los usuarios y el Servicio antivirus. Permite visualizar y modificar la configuración del antivirus; activar o desactivar el mecanismo de protección permanente; iniciar, ver el progreso y detener la actualización del antivirus y la búsqueda de códigos malignos; mostrar y eliminar las estadísticas del antivirus; así como mostrar y manipular la información de la cuarentena y copias de seguridad del antivirus.

IP: dirección física de la PC.

Interfaz por consola: es la encargada de garantizar la interacción de los usuarios y el Servicio antivirus mediante la consola. Permite ejecutar acciones que son especificadas al usuario mediante la consola, entre ellas: posibilidad de buscar códigos malignos y actualizar la base de datos de códigos malignos, además permite la eliminación de objetos que estén contaminados, así como seleccionar la opción de descontaminarlos. También permite consultar una Ayuda del sistema.

Interfaz visual: es la encargada de garantizar la interacción entre los usuarios y el Servicio antivirus. Permite visualizar y modificar la configuración del antivirus; activar o desactivar el mecanismo de protección permanente; iniciar, ver el progreso y detener la actualización del antivirus y la búsqueda de códigos malignos; mostrar y eliminar las estadísticas del antivirus; así como mostrar y manipular la información de la cuarentena y copias de seguridad del antivirus.

Prototipo: Maqueta visual funcional o no de la futura aplicación. Este puede ser una imagen o una aplicación software que simule funcionalidades del software.

Requisito: Una condición o capacidad que debe poseer un producto o componente de producto para satisfacer un contrato, estándar, especificación, u otros documentos obligatorios formales.

Segurmática Antivirus: Software antivirus orientado a la protección contra el accionar de los programas malignos en sistemas operativos de Microsoft Windows.

Sistema de detención de intruso (IDS): hace referencia a un mecanismo que, sigilosamente, escucha el tráfico en la red para detectar actividades anormales o sospechosas, y de este modo, reducir el riesgo de intrusión.