Control de seguridad para un servicio crítico de un sistema de información en línea, enmarcado en un dominio de ISO/IEC 27002, basado en medición de riesgos según OWASP

Trabajo de Grado

Jaime Alberto Jurado Narváez
David Felipe Penagos Mosquera

Director: Ing. Siler Amador Donado

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Grupo de Tecnologías de la Información (GTI).
Línea de Investigación: Seguridad Informática.
Popayán, 2015

AGRADECIMIENTOS

Agradezco a Dios y a mis padres por darme la vida, a mi familia, mi mamá Luz Dary Mosquera, a mi novia Rosmy Gaviria, a mis hermanos (Andrés y Paola) y mis primos por los ánimos y por el tiempo compartido, a mi director de tesis el ingeniero Siler Amador, a los ingenieros Carlos Ardila, Ember Martínez y Sara Garcés, por sus enseñanzas, dedicación, exigencia, paciencia y momentos compartidos, a mi profesores de colegio Esperanza Arboleda, Marco Mera, Luis Fernando Giraldo, a mis amigos de colegio, a mis amigos de Universidad, a Darío Corchuelo, Tatiana Solano, Laura Álvarez, Marcela Mera, Wilmer Camacho, Carlos Polindara, Edinson Castillo, Andrés Ruiz, Dany Cabrera, Jaime Jurado, a las personas que ya no están Tonguino, mis abuelos (Rosalbina y Marcos), a la familia Moncayo y muchas más personas a las que me gustaría nombrar, a todos quiero decirle que los quiero y les agradezco de corazón, por aportar sus enseñanzas y todo su ánimo, que ha permitido alcanzar muchos sueños y metas.

David Felipe Penagos Mosquera

Agradezco a mi familia por la paciencia y el amor que tuvieron a lo largo de mi formación, además de su apoyo incondicional en todo momento, en especial a mis padres por creer en mí y ayudarme en los momentos que más lo necesite. A mis hermanas por todo el cariño que me tienen y el cuidado que me han dado a lo largo de estos años. Al ingeniero Siler, por sus enseñanzas, los retos a los que nos hizo enfrentar para mejorar cada día, y por la confianza que nos dio al trabajar en este proyecto. A los ingeniero Carlos, Ember y Sara por apoyarnos en este proceso, ayudándonos a mejorar en cada momento. A la División de las Tic de la universidad del Cauca, por el soporte que nos brindaron para configurar el servidor de pruebas. A mis amigos que hicieron más ameno el paso por la universidad. Y a todas esas personas que de alguna u otra forma hicieron que mejorara cada día más.

Jaime Alberto Jurado

CONTENIDO

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE TABLAS

Pág.

INTRODUCCIÓN

Hoy, inmersos en un mundo de globalización y de avances tecnológicos, los sistemas de información – SI[1] – se han convertido en factor de gran importancia y desarrollo para todo tipo de organizaciones: industriales, comerciales, militares, asociativas, educativas, etc. Por consiguiente, es deber de la organización garantizar confidencialidad, integridad y disponibilidad de la información a sus usuarios.

Es por ello que con el presente trabajo de investigación: **Control de seguridad para un servicio crítico de un sistema de información en línea, enmarcado en un dominio de ISO/IEC27002, basado en medición de riesgos según OWASP**, se propone implementar un control de seguridad de la norma técnica ISO/IEC 27002, que integre un mecanismo de medición del riesgo de seguridad basado en OWASP (Open Web Application Security Project) e inteligencia artificial, para el servicio critico SIMCA (Sistema Integrado de Matricula y Control Académico de la Universidad del Cauca).

Para llegar a dicha implementación se partió de la identificación de un procedimiento de nivel crítico de SIMCA mediante la metodología de las elipses[2] para pasar a determinar la técnica de inteligencia artificial que mejor se adapte al control seleccionado y concluir con la implementación e implantación del control de seguridad según OWASP 2013. En otras palabras, el trabajo, primero, da a conocer los elementos de identificación del problema objeto de estudio, plantea un marco de referencia teórica sobre estándares de seguridad informática y a partir de éste, la propuesta de implementación del control de seguridad. Adicionalmente, en el capítulo de anexos se presenta, en forma detallada, el desarrollo de la metodología de las elipses, el prototipo del control de seguridad y una guía en Excel para hacer el análisis del riesgo.

---

[1] SI: Conjunto de elementos que interactúan entre sí con un fin común; que permite que la información esté disponible para satisfacer las necesidades en una organización. (http://goo.gl/y0esnu)

[2] Según Ph.D. Alberto G. Alexander la metodología de las elipses: “Es un método que permite identificar los distintos tipos de activos de información existentes dentro del alcance del modelo.” ALEXANDER, Alberto G. Diseño De Un Sistema De Gestión De Seguridad De Información. Bogotá. Alfa Omega, 2007.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCIÓN DEL PROBLEMA

Los sistemas de información (SI[3]) en los últimos años han apoyado las actividades en diferentes organizaciones y se han convertido en factor de gran importancia para estas [1]. El valor de un SI está en su eficacia porque facilita información oportuna, y en su eficiencia cuando lleva a la utilización de menores recursos tecnológicos, humanos y económicos. Los SI han permitido que la información sea fiable, exacta y esté disponible de manera oportuna para la toma de decisiones [2].

Las organizaciones han encontrado en las aplicaciones web[4], una forma efectiva y cómoda para gestionar y distribuir información de manera instantánea, esto ha tenido mucho éxito, principalmente porque sólo requieren un navegador web[5] (Google Chrome, Firefox, Opera, Internet Explorer) independientemente del sistema operativo, reducen costos porque no hace falta tener computadores muy potentes ni la compra de un software específico, no ocupan espacio porque están alojadas en la nube, y permiten estar disponibles día y noche a cualquier persona. Además, las aplicaciones web permiten compartir información con otros usuarios, y son usadas en el marketing y publicidad [3] como estrategia para ofrecer información a la organización sobre el comportamiento del usuario.

Por otro lado las aplicaciones web son activos de la organización[6], estos cada vez más numerosos: algunos son visibles y otros no, algunos son de más valor que otros, por tanto, si una organización desea subsistir, debe proteger sus activos frente a cualquier tipo de contingencia externa como interna [4].

Organizaciones cuyos servicios, básicamente estén dados a través de aplicaciones web, hoy, son el blanco de los ataques informáticos debido a la facilidad de acceso. Y si bien, existen guías y estándares que ayudan a gestionar la seguridad de la información, no se encuentran fácilmente mecanismos automatizados que ayuden

[3] SI: Conjunto de elementos que interactúan entre sí con un fin común; que permite que la información esté disponible para satisfacer las necesidades en una organización. (http://goo.gl/y0esnu)

[4] Aplicaciones Web: Programa o conjunto de programas que pueden ser vistos mediante un navegador web. (http://goo.gl/rkj9K)

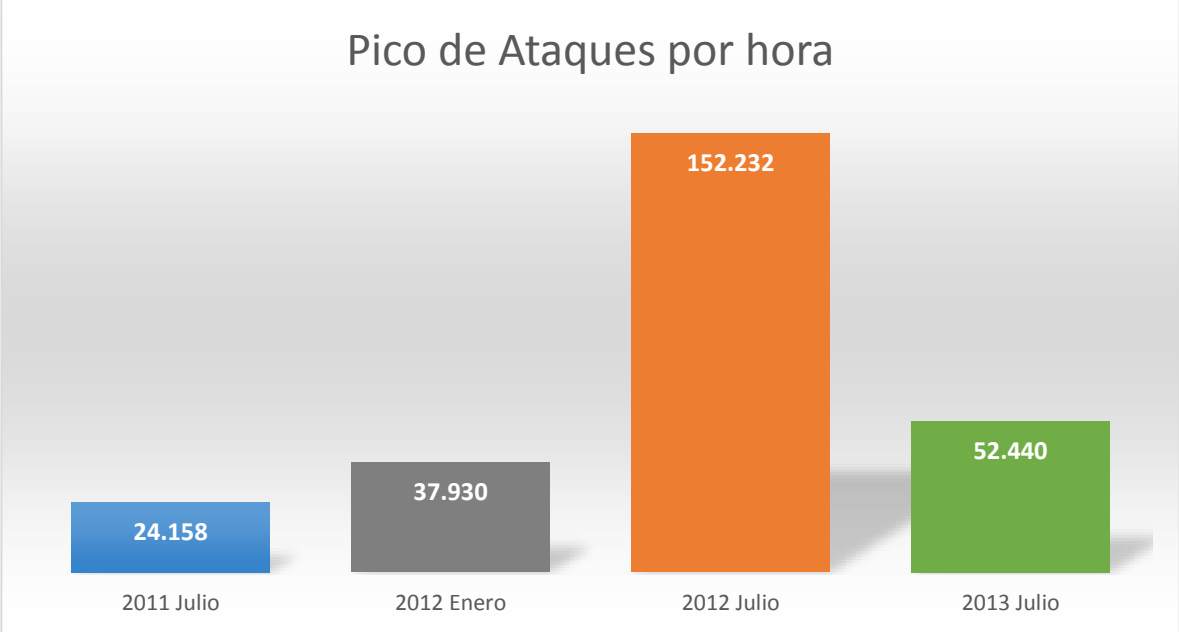
[5] Navegador Web: Programa que permite visualizar e interactuar con los sitios web. RAMOS, Alicia. Internet, características y evolución En: Aplicaciones Web (Novedad 2011). Madrid. p. 5

[6] Activo de la organización: Todo aquello (humano, tecnológico, software, etc.) que una organización considera importante o de valor. GRUPO ORGANIZACIÓN Y SISTEMAS UPTC. Clasificación de Activos de Información [diapositivas]. http://goo.gl/Da1HLG.

a identificar y a controlar en tiempo real toda esa serie de entradas maliciosas que llegan a las aplicaciones web de la organización. En consecuencia, todo esto pone en evidente vulnerabilidad a la organización. Vulnerabilidad que se puede ver desde diferentes aspectos: confidencialidad e información corporativa, confidencialidad de los datos de los clientes, aspecto financiero, para mencionar algunos. Y, en la actualidad, es obligación de la organización, como ya se dijo, garantizar seguridad de la información frente a cualquier tipo de contingencia.

Comprobar la seguridad es un elemento clave para la organización, OWASP (Open Web Application Security Project) [5] es un proyecto de código abierto de carácter independiente y no ligado a ningún fabricante, dedicado a determinar y combatir las prácticas que hacen inseguro al software, incrementando así el nivel de seguridad de las aplicaciones.

Figura 1. **Número de ataques máximos en una hora**



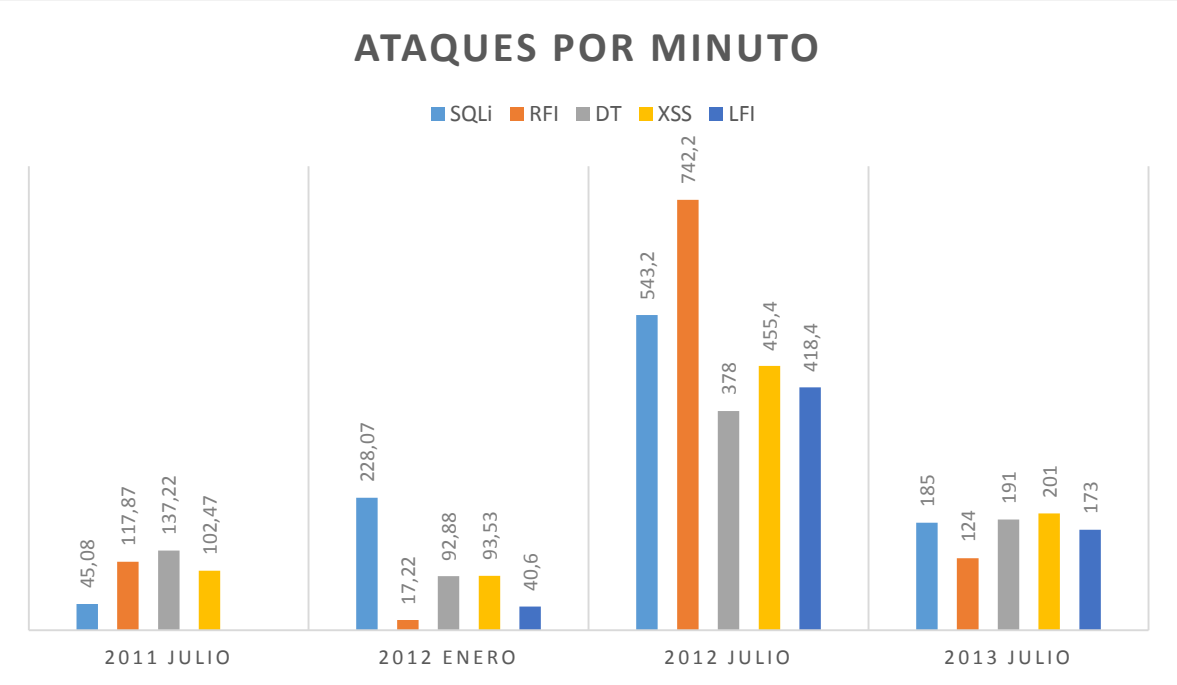
Fuente: Reportes de ataques a aplicaciones web de IMPERVA

Existen empresas especializadas en la protección de datos que anualmente reportan fallos de seguridad. El centro de defensa de aplicaciones de Imperva Application Defense Center[7] (ADC) publica el WAAR (Web Application Attack

[7] Imperva Application Defense Center: Es una organización de investigaciones de primera para el análisis de seguridad, descubrimiento de vulnerabilidades y la experiencia en el cumplimiento. (http://goo.gl/aAm28h)

Report) [6], un informe sobre ataques a aplicaciones web. En la Figura 1 se muestra como el número de ataques por hora hacia las aplicaciones web ha aumentado, pasando de 24.158 ataques por hora en el reporte de Julio de 2011 a 52.440 ataques por hora en el reporte de Julio de 2013. En los informes se listan que las amenazas más frecuentes como: inyección SQL (en inglés "SQL injection" –SQLi-), Remote File Inclusion (RFI), Directory Trasversal, Local File Inclusion (LFI) y Secuencias de Sitios Cruzados (en Inglés "Cross Site Scripting" -XSS-); en la Figura 2 se muestra los ataques por minutos de las amenazas listadas anteriormente y se observa que el XSS tiende a ser el mayor crecimiento con respecto a las otras amenazas en el año 2013.

Figura 2. **Número de ataques máximos en un minuto por tipo de ataque**



Fuente: Reportes de ataques a aplicaciones web de IMPERVA

Lo anterior plantea la necesidad de desarrollar aplicaciones mediante estándares que analicen en tiempo real, evalúen el riesgo de las aplicaciones web y sean capaces de tomar decisiones y controlar el riesgo.

Todo lo anterior lleva a plantear la siguiente pregunta de investigación:
¿Cómo optimizar la seguridad de un servicio crítico en un sistema de información en línea enmarcado en la ISO/IEC 27002 y basado en la medición de riesgo según OWASP?

1.2 JUSTIFICACIÓN

Durante el desarrollo y operación de un Sistema de Gestión de la Seguridad de la Información (SGSI) se realizan una serie de actividades como la identificación de activos, identificación de amenazas, estimación de impactos y vulnerabilidades, entre otras. Estas actividades se llevan a cabo con el objeto de estimar el riesgo, establecer políticas y controles. El diagnóstico que se realiza es dinámico puesto que cambia a lo largo del tiempo y la organización requiere realizar un monitoreo constante con el objeto de realizar ajustes al sistema [4]. El proyecto propone el diseño e implementación de un sistema de control de seguridad para un control del estándar ISO/IEC27002:2013 [7], que por su complejidad, nivel de precisión y dinamismo requiera la aplicación de conceptos como razonamiento, "intuición", el sentido común o el "aprendizaje".

Este proyecto tiene como beneficiario directo la Universidad del Cauca aportando al desarrollo del sistema de gestión de la seguridad que se viene adelantando a través del proyecto: "Implantación y certificación del Sistema de Gestión de Seguridad de la Información" y a las empresas de carácter público o privado que cuenten con un sistema de información en línea.

El desarrollo de la propuesta aborda la implementación de un control enmarcado en el estándar ISO/IEC 27002:2013 [7], y la aplicación de técnicas de inteligencia artificial, para la construcción de un sistema que combine procedimientos de monitorización, mecanismos de detección de amenazas y control para detectar, prevenir o corregir eventos e incidentes de seguridad, además determinar si las acciones realizadas para resolver las brechas de seguridad fueron efectivas. Incorporar el uso de una técnica de inteligencia artificial[8], permitirá integrar conocimiento en un sistema de control para que sea autónomo y ejecuten acciones correctivas adecuadas.

Cabe resaltar que este trabajo está enmarcado en el proyecto Control Inteligente para el servicio crítico de un sistema de información en línea enmarcado en un dominio de la ISO/IEC 27002, el cual es financiado por Fondo Regional para la Innovación Digital en América Latina y el Caribe (FRIDA)[9].

[8] Inteligencia Artificial: Conjunto de técnicas, algoritmos y herramientas que permiten resolver problemas a priori y que requieren cierto grado de inteligencia. GARCÌA, Alberto. Inteligencia Artificial. Fundamentos, práctica y aplicaciones. Madrid. RC Libros, 2012.

[9] FRIDA: Fondo Regional para la Innovación Digital en América Latina y el Caribe, dedicada a financiar proyectos de investigación que contribuyan al desarrollo de la sociedad de la información en la región. (http://programafrida.net/grant2013)

1.3 OBJETIVOS

1.3.1 Objetivo General

Implementar un control de seguridad de la norma técnica colombina NTC-ISO/IEC 27002, que integre un mecanismo de medición del riesgo de seguridad basado en OWASP 2013 e inteligencia artificial, para el servicio critico SIMCA (Sistema Integrado de Matricula y Control Académico de la Universidad del Cauca).

1.3.2 Objetivos Específicos

- Identificar un procedimiento de nivel crítico de SIMCA utilizando la metodología de las elipses[10].
- Seleccionar el control que permita disminuir el riesgo de seguridad de la información del procedimiento de nivel crítico de SIMCA.
- Determinar la técnica de inteligencia artificial que mejor se adapte al control seleccionado.
- Implementar[11] e Implantar[12] el control de seguridad según OWASP 2013.

[10] Metodología de las elipses: Método que permite identificar los distintos tipos de activos de información existentes dentro del alcance del modelo. ALEXANDER, Alberto G. Diseño De Un Sistema De Gestión De Seguridad De Información. Bogotá. Alfa Omega, 2007.

[11] Para el desarrollo de este trabajo implementar es tomar un control de la norma colombiana NTC-ISO/IEC 27002, aplicarle una técnica de inteligencia artificial y evaluar el riesgo.

[12] En este trabajo implantar es definir el proceso de instalación del prototipo.

2. MARCO TEÓRICO

2.1 SEGURIDAD DE LA INFORMACIÓN

Según Cano [8] “es la disciplina que habla de los riesgos, de las amenazas, de los análisis de escenarios, de las buenas prácticas y esquemas normativos, que nos exigen niveles de aseguramiento de procesos y tecnologías para elevar el nivel de confianza en la creación, uso, almacenamiento, transmisión, recuperación y disposición final de la información”.

“La Seguridad de la Información se encarga de garantizar la confidencialidad, integridad y disponibilidad de la información. Para alcanzar el objetivo se apoya en la Seguridad Informática, es decir la seguridad de la información será la encargada de “regular” y establecer las pautas a seguir para la protección de la información” [9].

2.2 SEGURIDAD INFORMATICA

Es una disciplina encargada de implementar las técnicas y/o pautas para la protección de la información por medio de dispositivos y herramientas computacionales. Su principal objetivo es intentar reducir las amenazas, encontrar un sistema informático seguro es imposible, por eso resulta de gran utilidad tomar medidas de seguridad adecuadas que ayuden a proteger la seguridad de los sistemas informáticos [10].

2.3 APLICACIONES WEB

Es un tipo de aplicación cliente/servidor que suele está formada por código HTML[13], donde el usuario interactúa con un cliente web (Navegador Web) para realizar solicitudes a un servidor. La comunicación entre cliente y servidor se da través de un protocolo estandarizado (HTTP) [11]. Este tipo de aplicaciones son muy populares por su facilidad de actualización y mantenimiento.

[13] HTML (Hyper Text Markup Language): Lenguaje de marcado para describir documentos web. (http://goo.gl/iVbs)

2.3.1 Vulnerabilidad en las Aplicaciones Web. Es una debilidad en la aplicación web, el cual puede ser un fallo en el diseño, un comportamiento inesperado o un error en la implementación, que permite a un atacante mediante el uso de técnicas de intrusión acceder a información privada (datos personales) o información importante de la aplicación web y perjudicar a los interesados de la aplicación [12].

2.3.2 Administración del riesgo en aplicaciones Web. Para hacer frente a una de las vulnerabilidades más comunes en las aplicaciones web, el Cross Site Scripting (XSS), Shar et al. [13] proponen un enfoque para removerlas de las aplicaciones web. Usando análisis y técnicas de asociación, se identifica las vulnerabilidades XSS potenciales en el código fuente, previniendo valores de entradas que puedan causar la ejecución de código malicioso. Se propone una aplicación que evalúa las aplicaciones web basado en el enfoque mencionado anteriormente y minimiza la intervención del usuario en la mitigación de estas vulnerabilidades.

K. Shar et al. [14] plantean la predicción de inyecciones de código, teniendo como objetivo el XSS y la inyección SQL. Mediante la observación de las medidas que se usan para este fin en las aplicaciones web, construyen modelos de predicción usando datos históricos, identificando patrones de estas vulnerabilidades. Se propone el prototipo de una herramienta, la cual se encarga de la recolección de datos y evaluar los modelos usados para la predicción. Como resultado en general encuentran un gran porcentaje acertado de las alertas de XSS e inyección SQL en las 8 aplicaciones web de código abierto que se usaron para las pruebas, con lo que muestra la utilidad y la efectividad de la predicción de vulnerabilidades.

Una comparación entre las técnicas para descubrir vulnerabilidades es el objetivo de Austin et al. [15] mediante el uso de algunos sistemas de registros para comparar cuatro (4) técnicas. Los resultados de este estudio, evidencia que ninguna técnica por sí sola puede detectar cada tipo de vulnerabilidad, por lo que usar solo una es insuficiente, además de que cada técnica sólo encuentra un subconjunto de vulnerabilidades. Basado en los resultados obtenidos y con el fin de encontrar la mayor cantidad de vulnerabilidades es recomendable utilizar en al menos las pruebas de penetración manual sistemáticas (en inglés “Systematic manual penetration testing”) y los análisis estáticos automatizados (en inglés “automated static analysis”).

Shahriar et al. [16] ofrecen una taxonomía y clasificación a partir de trabajos de monitoreo existente, el término taxonomía se considera como una “clasificación jerárquica”. En este estudio se clasifican los enfoques existentes en un conjunto de características comunes de la detección de ataques en línea. Se clasifica los

trabajos en ocho criterios identificados como los más comunes: aspecto de vigilancia, utilización estado del programa, mecanismo de aplicación, cambios ambientales, respuesta al ataque, cobertura de tipo de ataque, el lenguaje y el tipo de programa. Finalmente se realiza un siguiente nivel y se clasifica los trabajos basándose en sus tareas de control, esto incluye la operación del programa, el flujo de ejecución de código y origen, código estructura, integridad, valor deseado, e invariantes. Se observa que las técnicas de vigilancia varían significativamente de acuerdo con los criterios y aspectos anteriormente mencionados. Los resultados permiten diferenciar entre los enfoques de vigilancia existente y proporcionar a los profesionales e investigadores una guía para desarrollar herramientas de monitoreo con las características deseadas y elegir las herramientas adecuadas para sus necesidades.

Raspotnig et al. [17] comparan diferentes técnicas para la identificación de riesgo, el peligro y la amenaza de los sistemas informáticos. El propósito de la investigación fue evaluar si Safety (minimización del riesgo de ocurrencia -prevención) y Security (se encarga del control de incidentes) pueden complementarse mutuamente. Se encuentra que las técnicas de ambos campos pueden fortalecerse mutuamente, los resultados más evidentes es que, las técnicas de “Security” se pueden reforzar incluyendo mejores interesados y descripciones de la comunicación, mientras que las técnicas “Safety” pueden beneficiarse de una integración más estrecha entre la identificación del riesgo y el desarrollo de los procesos.

2.4 SERIE 27000

Es un conjunto de estándares desarrollado por ISO (International Organization for Standardization) e IEC (International Electrotechnical Commission) para gestionar la seguridad de la información de cualquier organización.

Para garantizar un mínimo de seguridad de la información a clientes, proveedores y socios, se hace primordial el uso de estándares que permitan establecer políticas, procedimientos y seleccionar controles de seguridad adecuados. Entre los estándares encargados de orientar a una organización que desee garantizar un mínimo de seguridad está la serie 27000, en ella se describen términos y definiciones que se emplean y aportan las bases de la implementación de un SGSI [18]. Entre esta familia se destacan la ISO/IEC 27001 (Sistemas de gestión de la seguridad de la información - SGSI) [19], la cual establece el marco de trabajo para definir un SGSI, y se centra en la gestión de la seguridad como un proceso continuo; la norma ISO/IEC 27002 (Código de buenas prácticas para la Gestión de la Seguridad de la Información) [7], la cual permite a las organizaciones mejorar la seguridad de su información y la norma ISO/IEC 27005 (Gestión de riesgos de

seguridad de la Información) [20], que proporciona pautas para la gestión de riesgo de seguridad de la información.

2.4.1 ISO/IEC 27001. Es la única norma internacional auditable y la principal de la serie 27000, la que contiene los requisitos necesarios para establecer, implantar, mantener y mejorar un sistema de gestión de la seguridad de la información (SGSI). La norma está concebida para garantizar la selección de controles de seguridad adecuados y proporcionales, que ayuden a gestionar y proteger los activos de la información en la organización, demostrando así a clientes, proveedores y accionistas la integridad de sus datos y su compromiso con la seguridad de la información [19]**.**

2.4.2 ISO/IEC 27002. Es una guía de implementación que permite a las organizaciones mejorar la seguridad de su información enfocada en los controles del Anexo A de la ISO/IEC 27001. Esta norma describe los dominios de control y mecanismos de control que se pueden ser implementados dentro la organización para minimizar los riesgos, detectados en el Análisis de Riesgos hasta un nivel asumible por la organización [7].

2.5 OWASP

Es un proyecto iniciado en el año 2000 conformado por una comunidad abierta de empresas, organizaciones educativas y particulares de todo el mundo, que crean artículos, metodologías, documentación, herramientas y tecnologías que pueden ser usadas libre y gratuitamente, esto con la misión de mostrar y hacer conciencia sobre la seguridad en las aplicaciones web. Usar OWASP permite a las organizaciones tomar mejores decisiones sobres sus riesgos de seguridad. Los proyectos OWASP se dividen en dos categorías principales, proyectos de desarrollo y de documentación. Se usaran son solo los siguientes proyectos de documentación: Guía de pruebas [21] y Top 10 - 2013 [22].

2.5.1 OWASP TOP 10 - 2013. Documento de concientización que tiene como finalidad dar a conocer a los desarrolladores de aplicaciones web y profesionales de seguridad, los riesgos más críticos en las aplicaciones Web.

Se describe a continuación las vulnerabilidades presentadas en el documento OWASP Top 10 - 2013 [22].

A1 – Inyección: “Las fallas de inyección, tales como SQL, OS, y LDAP, ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder datos no autorizados”.

A2 – Pérdida de Autenticación y Gestión de Sesiones: “Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, claves, token de sesión[14] o explotar otras fallas de implementación para asumir la identidad de otros usuarios”.

A3 – Secuencia de comandos en sitios cruzados (en inglés “Cross Site Scripting” -XSS-): Este es un ataque[15] que embebe código malicioso a través de las aplicaciones web. Ocurre en cualquier aplicación web donde se reciba información del usuario, lo que genera una salida sin la validación o codificación de la entrada. Los atacantes usan este tipo de ataque para enviar código malicioso a usuarios desprevenidos, el navegador no tiene manera de saber que el script no es confiable y lo ejecutara, debido a que el navegador piensa que el script viene de una fuente confiable, lo que le permite al atacante acceder a las cookies[16], tokens de sesión (robo de sesión[17]) y otra información sensible retenidas por el navegador, incluso pueden reescribir el contenido de una página HTML, permitiéndole al atacante poder realizar ataques de phishing[18] [23].

“Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso”.

Según algunos autores [22] existen tres tipos de fallas conocidas XSS: 1) Almacenadas, 2) Reflejadas y 3) basadas en DOM.

---

[14] Token de sesión: Conjunto de datos que es usado en comunicaciones de red para identificar una sesión. (http://goo.gl/Z3E3um).

[15] Ataque: Técnicas que usan los atacantes para aprovechar las vulnerabilidades en las aplicaciones. (https://goo.gl/2kG4oF).

[16] Cookie: Especie de contenedor de datos que los sitios web y el navegador utilizan con el propósito de recordar preferencia y configuraciones. (http://goo.gl/kq3Q).

[17] Sesión: Es intercambio de información entre dos o más dispositivos de comunicación, o entre un computador y un usuario. (http://goo.gl/7neMT).

[18] Phishing: Técnica de ingeniería social utilizada por los delincuentes para obtener información confidencial como nombres de usuario, contraseñas y detalles de tarjetas de crédito haciéndose pasar por una comunicación confiable y legítima. (http://goo.gl/zMAb).

Tipos de XSS

- **Persistente:** Consiste en almacenar código en las aplicaciones web para que se ejecute una vez la aplicación web se carga [24].

- **Reflejado:** Funciona modificando valores que las aplicaciones web pasa de una página a otra [25].

- **DOM**[19]**:** Es un tipo de inyección que permite al atacante tomar el control de un DOM [26].

A4 – Referencia Directa Insegura a Objetos: "Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder datos no autorizados".

A5 – Configuración de Seguridad Incorrecta: "Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación".

A6 – Exposición de Datos Sensibles: "Muchas aplicaciones web no protegen adecuadamente datos sensibles tales como números de tarjetas de crédito o credenciales de autenticación. Los atacantes pueden robar o modificar tales datos para llevar a cabo fraudes, robos de identidad u otros delitos. Los datos sensibles requieren de métodos de protección adicionales tales como el cifrado de datos, así como también de precauciones especiales en un intercambio de datos con el navegador".

A7 – Ausencia de Control de Acceso a las Funciones: "La mayoría de aplicaciones web verifican los derechos de acceso a nivel de función antes de hacer visible en la misma interfaz de usuario. A pesar de esto, las aplicaciones necesitan verificar el control de acceso en el servidor cuando se accede a cada función. Si las solicitudes de acceso no se verifican, los atacantes podrán realizar peticiones sin la autorización apropiada".

[19] DOM: Interfaz de programación de aplicaciones para documentos HTML y XML. Define el modo en que se accede y manipula el documento. (http://goo.gl/JukI2).

A8 – Falsificación de peticiones en sitios cruzados (en Inglés "Cross-Site Request Forgery" -CSRF -): "Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legítimas provenientes de la víctima".

A9 – Uso de componentes con vulnerabilidades conocidas: "Algunos componentes tales como las librerías, los *frameworks*[20] y otros módulos de software casi siempre funcionan con todos los privilegios. Si se ataca un componente vulnerable esto podría facilitar la intrusión en el servidor o una perdida seria de datos. Las aplicaciones que utilicen componentes con vulnerabilidades conocidas debilitan las defensas de la aplicación y permiten ampliar el rango de posibles ataques e impactos".

A10 – Redirecciones y reenvíos no validados: "Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios hacia otras páginas o sitios web, y utilizan datos no confiables para determinar la página de destino. Sin una validación apropiada, los atacantes pueden redirigir a las víctimas hacia sitios de *phishing* o *malware*[21], o utilizar reenvíos para acceder páginas no autorizadas".

2.5.2 Guía de pruebas. Es una guía centrada en las pruebas y listas de comprobación de seguridad que permiten garantizar que las aplicaciones web cumplen con lo básico de seguridad.

- **Metodología OWASP**

Para evaluar el riesgo en las aplicaciones web siguiendo la metodología de valoración de riesgo OWASP [21] se deben seguir los siguientes pasos:

Paso 1: Identificar un riesgo. Se identifican algunos riesgos o vulnerabilidades que se presenten en la aplicación. "El primer paso es recopilar información sobre los agentes que causan la amenaza, el ataque que utilizan, la vulnerabilidad

[20] Frameworks: Es un ambiente software reusable y universal que provee funcionalidades particulares para facilitar el desarrollor de aplicaciones software. Pueden incluir programas de ayuda, compiladores, librerías, herramientas y APIs (http://goo.gl/Z6zGV).
[21] Malware: Código Malicioso. (http://goo.gl/MuXPK).

involucrada, y el impacto de la explotación con éxito en la organización de un riesgo".

Paso 2: Factores para estimar la probabilidad de ocurrencia. Se identifican los factores para estimar la probabilidad de ocurrencia. Los factores se pueden dividir en dos grupos, el primer grupo de factores están relacionados con los agentes que causan la amenaza (Nivel de Conocimiento, Oportunidad, Motivación y Tamaño), el objetivo es estimar la probabilidad de ocurrencia de un ataque con éxito. Factores que afectan a la vulnerabilidad, el objetivo de estos es estimar la probabilidad de que la vulnerabilidad sea descubierta y explotada. Cada factor tiene un conjunto de opciones y cada opción tiene una evaluación de la probabilidad de cero a nueve asociado a cada uno.

Paso 3: Factores para estimar el impacto. Se debe tener en cuenta que existen dos tipos de impacto. El primero es un impacto técnico en la aplicación, su objetivo es estimar la magnitud del impacto en la aplicación si la vulnerabilidad fuera aprovechada, el otro es el impacto sobre el negocio, es el más importante, pero requiere un entendimiento profundo de que es lo importante para la organización.

Paso 4: Determinar la Severidad del Riesgo. En este paso con la Probabilidad de ocurrencia global y el impacto global se determina la severidad del riesgo.

Paso 5: Decidiendo que arreglar. Se obtendrá una lista priorizada de que arreglar, como regla general se deben arreglar los problemas que supongan un riesgo más severo. Arreglar los riesgos menos importantes no ayuda a reducir el riesgo global, incluso si son fáciles o baratos de arreglar.

Paso 6: Personalización del Modelo. Un modelo adaptado es probable que produzca mejores resultados para la organización, existen varias maneras de adaptar el modelo, como por ejemplo agregando factores que representen que es lo importante para la organización, personalizando las opciones con términos diferentes, cambiar los valores de cada opción o ponderando los factores.

2.6 AUDITORÍA DE SEGURIDAD E INTELIGENCIA ARTIFICIAL

Feng et al. [27] proponen un modelo de análisis de riesgos de seguridad para la información (SRAM) usando redes bayesianas (BNS) y optimización basada en colonias de hormigas (ACO), esto con el fin de identificar las relaciones causales

entre los factores de riesgo, analizar la complejidad y la incertidumbre de la propagación de la vulnerabilidad. Se desarrolla una Red bayesiana (BN) para definir simultáneamente los factores de riesgo y sus relaciones causales basándose en el conocimiento de casos observados y expertos de dominio. Luego se realiza el análisis de propagación de vulnerabilidad de seguridad, ACO les permitió determinar las trayectorias de propagación con la probabilidad más alta y el valor más grande del riesgo estimado. Se concluye que SRAM permite a las organizaciones establecer planes de gestión de riesgos de seguridad proactiva de los sistemas de información. Finalmente la eficacia de este modelo la demuestran a través de un caso de estudio.

La propuesta de Huang et al. [28] tiene como objetivo formular un modelo que pueda ser usado para el análisis de las vulnerabilidades de un producto software, o que pueda utilizarse como base para la mejora del mismo. Se usa un proceso analítico jerárquico difuso para construir un modelo para la toma de decisiones, el cual ayuda en la valoración de vulnerabilidades y mostrar el nivel en que están afectando la seguridad. A través de un caso de estudio, se muestra que el uso del modelo difuso para la toma de decisiones es práctico en la evaluación de vulnerabilidades y ayuda en la mejora del producto, además de que puede ser usado para futuras vulnerabilidades.

Para proveer de un buen nivel de seguridad en una organización, la auditoría juega un rol clave, pero el alto costo, tiempo y uso de recursos humanos para este fin, es un problema común. Kozhakhmet et al. [29] sugieren un sistema experto como una solución, para facilitar el proceso de auditoría y reducir el uso de los recursos mencionados anteriormente. Como resultado de esta investigación, se sugiere que los sistemas expertos aplicados en el campo de la seguridad informática es una técnica que permite emular la habilidad de toma de decisiones de un especialista, además permiten escribir las reglas en lenguaje natural, esto ayuda a la simplificación de la comunicación entre el experto en el área y el ingeniero de conocimiento.

Kozhakhmet et al. [30] tiene un enfoque a un sistema experto para facilitar las auditorías y reducir el uso de recursos, concluye que hay varios beneficios en el uso de estas técnicas en el campo de seguridad, además de las muchas áreas inexploradas para este.

Wang et al. [31] proponen la implementación de un prototipo, que integre la tecnología de agentes inteligentes con los servicios web para hacer uso de las ventajas de ambos. Luego de realizar la evaluación del prototipo los beneficios encontrados son los siguientes: *i.* Integración del Sistema. La gestión de

excepciones sistema es capaz de integrarse con aplicaciones de sistemas heredados. *ii.* Inteligencia. Los problemas de negocios complejos se pueden identificar y diagnosticado por un número de agentes inteligentes a través de sus propiedades, tales como la autonomía, la reactividad, pro actividad y capacidad social. *iii.* Escalabilidad. Es fácil añadir más funcionalidades de negocio del sistema mediante la adición de más agentes de servicios web. Finalmente *iv.* Reusabilidad, la arquitectura propuesta y los agentes son reutilizables para otras aplicaciones empresariales.

2.7 DETECCIÓN DE ATAQUES XSS

Nunan et al. [32] se enfocan en determinar características en documentos Web y Uniform Resource Locator, por su equivalente en inglés (URL[22]) que les permite clasificar los ataques usando técnicas de aprendizaje de máquina (en inglés Machine Learning)

En este trabajo Krishnaveni et al. [33] identifican cuatro posibles características maliciosas del XSS, obtenidas al realizar una investigación en diferentes sitios web y los algoritmos de Machine Learning (Naive Bayes, Arboles de Decisión y Perceptron Multicapa) aplicados para realizar una clasificación de los ataques XSS. Concluyen que el Perceptron Multicapa y Arboles de decisión tienen una alta tasa de precisión.

Mahapatra et al. [34] proponen un Framework de patrones para la prevención de ataques XSS, ellos usan expresiones regulares para encontrar las características que han observado en los ataques XSS y a partir de esta caracterización forman un grupo que lo autores denominan como patrones de expresiones regulares (en inglés Pattern From Regex).

Vishnu et al. [35] en este trabajo determinan características de los ataques XSS encontrados en URL y JavaScript, y apoyados en técnicas de Machine Learning (Naive Bayes, Arboles de Decisión (Algoritmo J48), Support Vector Machine) clasifican las páginas web en normal o malicioso.

Adicionalmente, Patil et al. [36] se enfocan en obtener características de la URL y del contenido de la página web. Usan Técnicas de Machine Learning (Naive Bayes

[22] URL: Usado para nombrar recursos en internet, con el propósito de asignar una dirección única a cada uno de dichos recursos. (http://goo.gl/aWHh3S).

and Support Vector Machines) mediante las cuales diseñan un modelo predictivo, para clasificar páginas web como XSS o No XSS. Estos autores observaron que Naive Bayes presenta menos costo computacional y logra un rendimiento cercano a la técnica de Support Vector Machines.

2.8 METODOLOGÍA CRISP-DM

Esta metodología [37] es utilizada para la explotación de datos, está estructurada en seis fases. A continuación se describe como se adaptó cada fase:

Fase de comprensión del negocio: En esta fase se entendió el negocio, se conoció los procesos y procedimientos de la organización, se obtuvo el inventario de los activos críticos, lo cual permitió identificar el proceso crítico de la organización.

Fase de comprensión de los datos: Una vez conocido el negocio, se efectuó una recopilación inicial de los datos con el fin de conocerlos y describirlos; esto llevó a tener un contacto inicial con el problema.

Fase preparación de los datos: Seleccionados los datos, estos se procesaron con el fin de obtener un conjunto de datos adecuado para ser usado en una herramienta de minería de datos.

Fase de modelado: Se realizó la selección y evaluación de las Técnicas de Machine Learning apropiadas, se escogió el algoritmo que mejor se ajustaba a los datos y se creó el modelo de clasificación, también en esta fase se implementó el prototipo y se integró con el modelo de clasificación creado.

Fase de evaluación: Se evaluó la funcionalidad del prototipo mediante pruebas unitarias y de integración.

Fase de despliegue: Esta fase describe la arquitectura de red usada para el funcionamiento del prototipo.

En la Figura 3 se presenta la adaptación de la metodología CRISP-DM, las fases usadas, acompañadas por tareas y salida.

Figura 3. **Tareas y salidas de la adaptación de la Metodología CRIDP-DM**

Fuente: Elaboración Propia

2.9. APORTES

Luego de realizar un trabajo detenido del estado del arte sobre el tema objeto de estudio, se observó que existen diferentes proyectos que tratan sobre uso de estándares, metodologías de riesgos, administración del riesgo, aplicaciones web accesibles mediante sistemas inteligentes, auditorías de seguridad con inteligencia artificial, pero no se evidencia una propuesta que aborde el problema planteado en esta investigación, sobre todo con el enfoque que se plantea. En consecuencia, se presentan los siguientes aportes:

- El desarrollo de un sistema que combine procedimientos de monitorización, detección de amenazas basándose en una relación factible entre la ISO/IEC 27002:2013 y el listado de los riesgos más críticos en aplicaciones web según OWASP 2013, tal como se puede evidenciar en el capítulo de implementación del presente trabajo.

- La construcción de un control de seguridad enmarcado en el dominio ISO/IEC 27002 que mide el riesgo en tiempo real, y toma decisiones que permiten disminuir el riesgo en el sistema de información en línea.

- Se definió una serie de pasos, basados en la metodología de las elipses, que permiten a una organización seleccionar un proceso crítico.

3. DISEÑO E IMPLEMENTACIÓN DE UN CONTROL DE SEGURIDAD SEGÚN OWASP 2013

En los numerales 3.1 y 3.2 se desarrolló la fase de compresión del negocio de la metodología de CRISP-DM.

3.1 IDENTIFICAR UN PROCESO DE NIVEL CRÍTICO USANDO LA METODOLOGÍA DE LAS ELIPSES

El especialista Alberto G. Alexander (2007) [1] propone la metodología de las elipses como el método más apropiado para determinar el alcance, identificando los procesos, unidades y entidades externas en la organización. Al finalizar la aplicación de la metodología de las elipses la organización estará lista para evaluar el riesgo.

Para mostrar ejemplos de los resultados durante cada etapa y probar el prototipo del control de seguridad se toma la Universidad del Cauca como caso de estudio.

- **Definir el alcance de la organización.** El alcance es fundamental ya que delimita las partes[23] de la organización. La sección 4.3 del estándar NTC-ISO-IEC 27001:2013 [38], es el punto de partida para establecer que está dentro y fuera del alcance, se debe considerar las partes interesadas (*stakeholders*), las interfaces[24] y dependencias entre las actividades realizadas por la organización y las realizadas por otras organizaciones (proveedores), en otras palabras "entender la organización y su contexto, y entender las necesidades y expectativas de las partes interesadas". El estándar ISO/IEC 27001:2013 [38]: 4.3 exige que el alcance esté documentado y disponible.

Se siguieron los siguientes pasos para aplicar la metodología de las elipses:

Paso 1. Identificar los procesos: Los procesos importantes de la organización se deben ubicar en la elipse concéntrica. A cada proceso se le deben identificar sus respectivos subprocesos, adicionalmente se debe documentar cada proceso (puede

[23] Según Alberto G. Alexander, Partes: Se refiere a cualquier componente que se pueda identificar y se tratado como independiente del resto. ALEXANDER, Alberto G. Diseño De Un Sistema De Gestión De Seguridad De Información. Bogotá. Alfa Omega, 2007.

[24] Según Alberto G. Alexander, Interfaces: Punto a través del que se comunican dos zonas. ALEXANDER, Alberto G. Diseño De Un Sistema De Gestión De Seguridad De Información. Bogotá. Alfa Omega, 2007.

consultarse la documentación en el Anexo 1). Los procesos que se identificaron fueron:

1. Matrícula Académica.
2. Matrícula Financiera.
3. Ajuste a la Matrícula.
4. Evaluación Docente.
5. Labor Docente.
6. Control Académico.
7. Inscripciones y Admisiones

Paso 2. Identificar unidades, organizacionales y entidades externas: Ubicar en la elipse intermedia las unidades organizacionales que tengan relación con alguno de los procesos y relacionar, a través de flechas, las distintas interacciones entre los procesos de la elipse concéntrica con las unidades y entidades externas de la organización. Las flechas indican el tipo de interacción y la direccionalidad que tiene el flujo de la información.

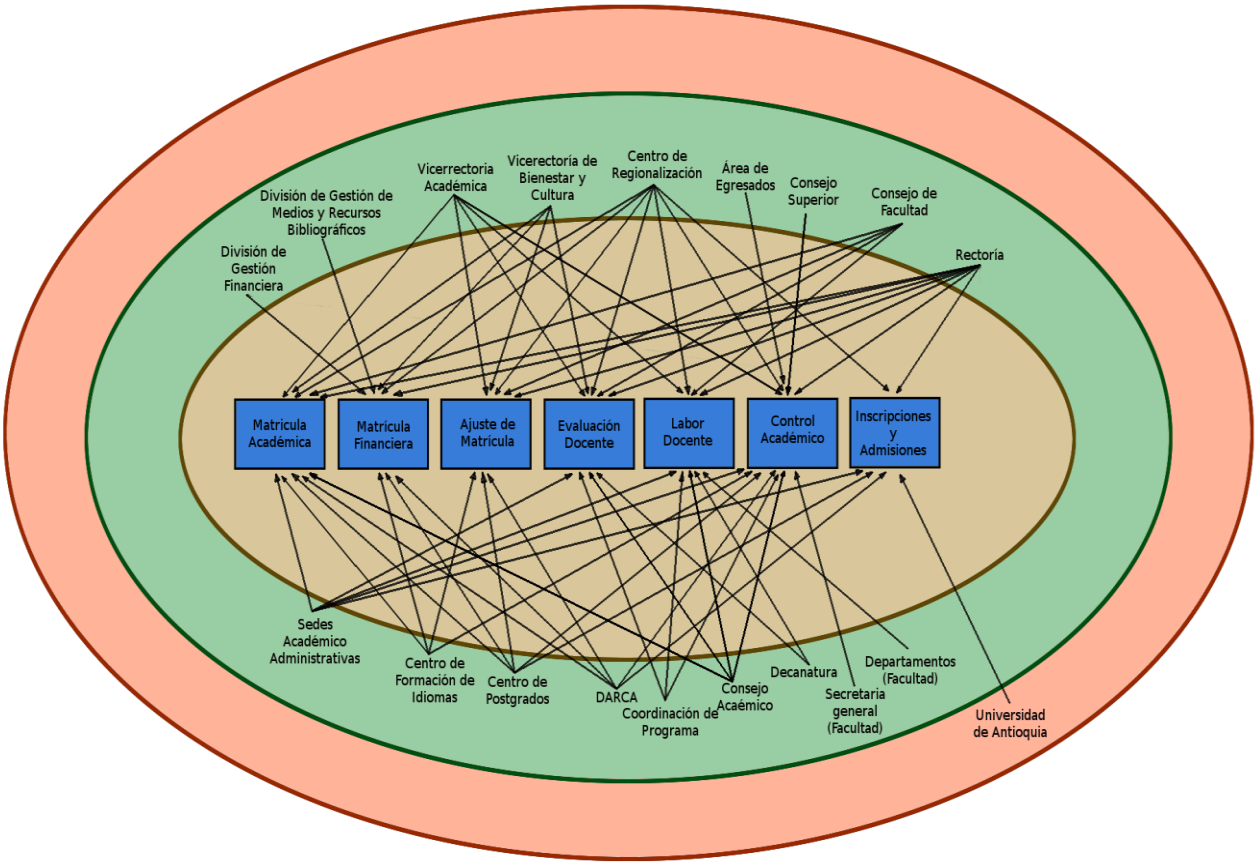
Se identificaron las siguientes unidades:

1. División de Gestión Financiera
2. División de Gestión de Medios y Recursos Bibliográficos
3. Vicerrectoría Académica
4. Vicerrectoría de Bienestar y Cultura
5. Centro de Regionalización
6. Centro de Posgrados
7. Área de Egresados
8. Consejo Superior
9. Consejo de Facultad
10. Rectoría
11. Consejo Académico
12. Secretaría General
13. División de Admisiones, Registro y Control Académico (DARCA)
14. Decanatura
15. Departamentos (adscritos a las Facultades)
16. Coordinación de Programa
17. Centro de Formación de Idiomas
18. Sedes Académico Administrativas

Sólo se identificó una entidad externa: la Universidad de Antioquia la cual se encarga de aplicar el examen de admisión cada semestre.

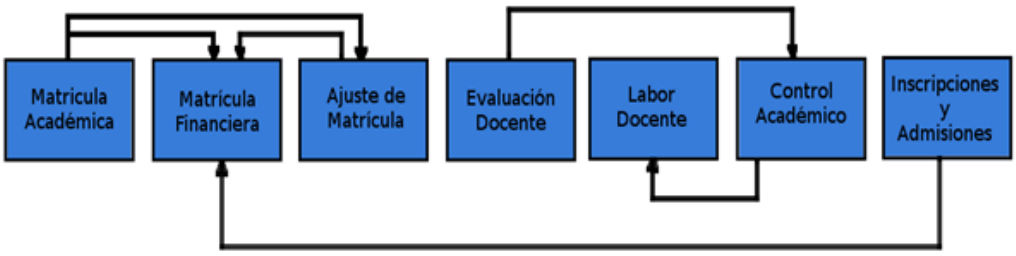
El resultado de aplicar la metodología de las elipses en la Universidad del Cauca se presenta a continuación en las figuras 4 y 5:

Figura 4**. Elipse procesos críticos de la Universidad del Cauca**



Fuente: Elaboración propia

Figura 5**. Interacciones entre procesos de la elipse**



Fuente: Elaboración propia

Paso 3. Identificar los activos de información: Con los usuarios, empleados o dueños de los procesos de la organización, se deben identificar cuáles son los activos de información vitales, como lo indica el control A.8.1.1 en la Tabla A.1. de la norma NTC-ISO-IEC 27001:2013 [38]. Debido a que tales activos conforman un conjunto muy extenso, sólo se relacionaron los activos de software (aplicaciones web) que tengan relación con la figura 4. También se deben identificar a los propietarios correspondientes, como lo exige el control A.8.1.2 en la Tabla 1 de la norma NTC-ISO-IEC 27001:2013 [38]. Una descripción completa de los activos de información puede consultarse en el Anexo 2.

A continuación se detalla nombre del activo de la información y el propietario.

Tabla 1**. Activos de información de la organización**

Activo de información	Propietarios
Sistema de Recursos Físicos (SRF Plus)	Oficina de Planeación
Sistema de Recursos Humanos (SRH)	División Financiera
Sistema Finanzas Plus (FPL)	División Financiera
Sistema de Información de Matrículas y Control Académico (SIMCA)	División de Admisiones, Registro y Control Académico
Sistema de Información y Gestión de Labor Académica (SIGELA) (Integrado en SIMCA)	Vicerrectoría Académica
Sistema para Automatización de Bibliotecas (Unicornio)	División de Bibliotecas
Sistema de ingresos y facturación (SQUID)	División Financiera
Sistema de solicitudes	División de Tecnologías de la Información y Comunicación

Fuente: Elaboración propia

Paso 4. Tasar los activos de la información: Luego de identificar los activos de información se procede a realizar su tasación, como lo indica Alberto G. Alexander (2007) [1], para esto cada activo se tasa, utilizando una escala de Likert, donde 1 significa “muy poco” y 5 “muy alto”. La pregunta que se debe efectuar para utilizar la escala es: ¿Cómo una falla en un determinado activo afecta la confidencialidad, la integridad y la disponibilidad?

Tabla 2**. Activos de información críticos del sistema**

ACTIVO DE LA INFORMACIÓN	Valoración de Activos				Porcentaje de relación con los procesos críticos								
	Tasación												
	C	I	D	**Total**	Propietario	P1	P2	P3	P4	P5	P6	P7	**Promedio del porcentaje de participación con los procesos críticos**
Sistema de Recursos Físicos (SRF Plus)	1	2	1	**1**	Oficina de Planeación	0	0	0	0	0	0	0	**0**
Sistema de Recursos Humanos (SRH)	2	1	1	**1**	División Financiera	2	0	0	0	10	5	0	**2**
Sistema Finanzas Plus (FPL)	1	2	2	**2**	División Financiera	2	30	15	0	10	15	20	**13**
Sistema de Información de Matrículas y Control Académico (SIMCA)	1	3	3	**2**	División de Admisiones, Registro y Control Académico (DARCA)	80	60	80	80	50	80	70	**71**
Sistema de Información y Gestión de Labor Académica (SIGELA) (Integrado en SIMCA)	1	3	3	**2**	Vicerrectoría Académica	20	2	20	10	40	0	0	**13**
Sistema para Automatización de Bibliotecas (Unicornio)	3	2	3	**3**	División de Bibliotecas	5	5	5	0	0	0	0	**2**
Sistema de ingresos y facturación (SQUID)	1	2	2	**2**	División Financiera	15	30	15	0	0	15	20	**14**

Fuente: Elaboración propia

Convenciones: C: Confidencialidad I: Integridad D: Disponibilidad

P1: Proceso de Matrícula Académica
P2: Proceso de Matrícula Financiera
P3: Proceso de Ajuste a la Matrícula
P4: Proceso de Evaluación Docente
P5: Proceso de Labor Docente
P6: Proceso de Control Académico
P7: Proceso de Inscripciones y admisiones

Adicionalmente a cada activo se le debe identificar un porcentaje de participación en cada proceso.

Paso 5. Definir el Sistema de información: Usando los resultados consignados en la Tabla 2, se puede seleccionar el activo de información apropiado para ser protegido. Es importante tener en cuenta los siguientes criterios: seleccionar los activos que tengan valor total alto y luego seleccionar el que tenga el promedio de participación más alto con respecto a los procesos críticos.

En nuestro caso de estudio, el Sistema de Información SIGELA actualmente está finalizando su integración a SIMCA y por tanto fue descartado en este análisis.

Los activos de información que tienen un valor muy alto son Sistema Finanzas Plus, SIMCA, y Unicornio. Se escoge SIMCA por ser el activo de información que tiene mayor porcentaje de participación con todos los procesos críticos de la metodología de las elipses, además de tener en cuenta la siguiente afirmación proporcionada por la organización "Uno de los sistemas de información más importantes en la institución es el SISTEMA INTEGRADO DE MATRÍCULA Y CONTROL ACADÉMICO – SIMCA" [39].

En el Anexo 3, se puede observar el documento del alcance aplicado al caso de estudio (Universidad del Cauca).

3.2 SELECCIÓN DEL PROCESO CRÍTICO

Para seleccionar el proceso crítico es necesario contar y usar la elipse que resulte de aplicar la metodología de las elipses. Para ilustrar cómo se debe llevar el proceso se usara la elipse de la Figura 4 y de ella se seleccionó un proceso crítico.

Para seleccionar dicho proceso crítico, se debe observar la relación entre los procesos y las dependencias de la organización. Se debe escoger el proceso con mayor número de relaciones con las dependencias de la organización.

En caso de que existan dos o más procesos que cumplan la condición anterior, tomar el proceso con mayor número de relaciones con otros procesos. Por ejemplo, se puede observar a través de la Figura 4, que Control Académico es el proceso con mayor número de relaciones de entrada, lo que permite afirmar que si falla, afectara a más dependencias en la organización.

Una vez seleccionado el proceso se debe contar con la aprobación de la organización, por eso resulta indispensable presentar y explicar los motivos que llevaron a la selección del proceso. En la organización que dio como resultado la Figura 4, se consultó a la dependencia encargada de la seguridad de la información (La División de Tecnologías de información - dependencia que hace parte de la organización), y se realizó la siguiente pregunta ¿Cuál consideran como el proceso más crítico entre los que se identificaron en la elipse: Matriculas académicas, Matricula financiera, Ajuste de matrículas, Evaluación docente, Control académico,

Inscripciones y Admisiones? La división de Tecnologías de información, determinó que el proceso de Control académico es para ellos el más importante, debido a que contiene información personal de estudiantes y docentes, notas y faltas de asistencia de los estudiantes.

Con los anteriores resultados se confirma que el proceso de Control Académico es el más crítico de la organización. Dentro de este proceso, se seleccionó el procedimiento "Ingresar, eliminar, modificar, notas y faltas", al ser considerado por la organización como el procedimiento más importante del proceso.

3.3 EVALUACIÓN DEL RIESGO

El riesgo es la probabilidad que se produzca un impacto[25] determinado en un activo o en toda la organización. Este impacto puede hacer que una amenaza explote una vulnerabilidad en particular causando pérdidas o daños. Conocer el riesgo al que están expuestos los activos le permite a una organización tomar medidas para disminuir, prevenir o estar protegida ante la posible ocurrencia de un riesgo. En otras palabras, el objetivo de la evaluación de riesgos es realizar un cálculo de las amenazas a los activos de la organización.

El modelo estándar de la valoración del riesgo es:

Riesgo = Probabilidad de ocurrencia x Impacto

El numeral *6.1.2 literal d* de la norma NTC-ISO-IEC 27001:2013 [38], exige que en la organización debe existir una metodología para la evaluación del riesgo de los activos.

La organización OWASP a partir del modelo estándar y de la colaboración de empresas y expertos en seguridad ha creado una metodología para la valoración de riesgo. La metodología de OWASP [21] es un sistema para puntuar el riesgo en las aplicaciones web, permite, a su vez, ahorrar tiempo y estimar adecuadamente la severidad de todos los riesgos; esto asegura no distraerse en riesgos de menor importancia.

[25] Impacto: Cambio adverso en el nivel de los objetivos del negocio logrados. ISO/IEC 27005:2011, "Information technology - Security techniques - Information security risk management." 2011.

Por lo anterior y dando cumplimiento al *numeral 6.1.2 literal d* se considera que la metodología para la valoración de riesgo de OWASP, es la más apropiada para medir el riesgo y ser implementada en el control de seguridad.

Escogida la metodología de valoración de riesgo de OWASP, se procedió a entenderla. Esta metodología usa el modelo estándar de la valoración del riesgo, y propone los siguientes grupos:

- Factores Relacionados al agente de amenaza
- Factores Relacionados con la Vulnerabilidad
- Factores al Impacto Técnico
- Factores relacionados al impacto del negocio.

Estos grupos permiten medir la probabilidad de ocurrencia y el impacto [21]. Cada grupo de factores mencionados anteriormente contiene una serie de factores; cada uno incluye una descripción de lo que se debe medir (por lo general una pregunta que se debe resolver) y una lista de posibles valores que puede tomar (cada factor sólo puede tomar un valor de la lista que le corresponda); los factores permiten calcular la probabilidad de ocurrencia y el impacto, para cada A del Top 10 - 2013 de OWASP.

La Guía de Pruebas de OWASP, también presenta dos métodos para definir los factores y capturar las respuestas. El método repetitivo, que requiere tener información previa de la información (históricos) que permita capturar las respuestas y el método informal que “calcula a ojo” los factores y las respuestas.

Debido a que en la organización (Universidad del Cauca) no se evidenció el almacenamiento de históricos de XSS (en las reuniones sostenidas con el personal del División de las TIC), se escogió el método informal, para definir los factores.

Con el apoyo del Top 10 - 2013 [22], la Guía Pruebas [21] de OWASP, información como: Página objetivo, complejidad del ataque, dirección IP de origen que fueron obtenidas del tráfico de red, la selección del método informal y reuniones con expertos, se definió el formato con una visión general que permite seleccionar el valor de algunos factores para cada A del Top 10 - 2013 [22]. En el presente trabajo se adaptó dicho documento para seleccionar los valores de los factores para el A3. (En el Anexo 4, puede consultarse el documento)

A continuación se describe como se definieron los factores para el cálculo del riesgo para el XSS.

Factor Nivel de conocimiento: Las formas para construir un payload[26] son infinitas porque depende de la creatividad del atacante. Para dar el valor a este factor se hizo por medio de una Técnica de Machine Learning con la cual se determinó la complejidad del ataque.

Los valores escogidos siguiendo la escala de cero a nueve (según OWASP) para la complejidad fueron los siguientes: BAJO igual a uno, MEDIO igual a cuatro, ALTO igual a siete.

Para el tipo de XSS los valores se determinaron de la siguiente forma: si es un ataque persistente, se le asigna un valor de nueve (9), si es un ataque de DOM se le da un valor de tres (3) y si es un reflejado el valor de seis (6).

Los valores de complejidad y tipo de XSS se promedian para obtener el factor de nivel de conocimiento.

Factor de Motivación: Para determinar el valor de este factor se tiene en cuenta el objetivo que es blanco de ataque, lo que permitió plantear lo siguiente:

La organización debe proporcionar un inventario de la aplicación web que especifique las páginas que tienen acceso a persistencia, las cuales se les asigno el valor de nueve en caso de que la pagina no esté en el inventario se le da un valor de uno.

Factor de Oportunidad: Para dar un valor a este factor se usó el nivel de complejidad del ataque obtenido por la Técnica de Machine Learning, con los siguientes valores: BAJO igual a uno, para MEDIO igual a cuatro, para ALTO igual a siete y el tipo de XSS con los siguientes valores: siete (7) para persistencia, siete (7) para DOM y nueve (9) para reflejado.

Se promediaron los valores de complejidad y tipo de XSS para obtener el factor de oportunidad.

[26] Payload: Acción o carga dañina que ejecuta un Ataque. (http://goo.gl/1jXiQl).

Factor de Tamaño: Para determinar el valor de este factor se usó la IP de origen del ataque, se definieron los siguientes rangos y se les asigno el siguiente valor:

Si la dirección IP de origen está dentro del rango de los administradores, el valor del factor es dos, si la dirección IP de origen fue dentro del rango de direcciones asignadas a los usuarios de intranet, el valor del factor es cuatro, si está en el rango asignado a los usuarios se le da al factor el valor de seis, si no está en ninguno de los rangos anteriores el valor del factor es nueve.

Factor de Detección de la intrusión: Para definir el valor de este factor se decidió hacerlo de la siguiente forma: Si la organización tiene una herramienta para detectar incidencias XSS el valor del factor es uno (1), si la organización lleva un registro de las incidencias XSS y las revisa el valor del factor será tres (3), si las incidencias se registra pero no se revisa el valor del factor es ocho (8) y si no registra ninguna incidencia de XSS, el valor del factor es nueve (9).

Para definir el valor de los siguientes factores se usó el Top 10 - 2013 [22], el cual establece lo siguiente: “El valor de MUY DIFUNDIDO a 0. El resto de riesgos se enmarcan entre difundidos a poco comunes (valores de 1 a 3)”. Esto, a su vez, permitió definir la siguiente escala:

Tabla 3. Escala de factores de riesgo

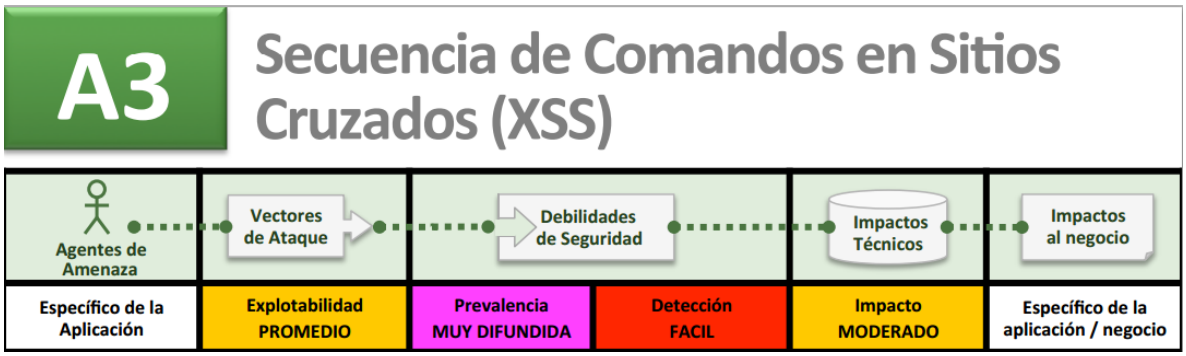
Factores de riesgo	Valor
MUY DIFUNDIDO	0
FACIL	1
PROMEDIO	2
MODERADO	2

Fuente: OWASP Top 10 - 2013

Esta escala es inversamente proporcional a la de la metodología para la valoración de riesgo de OWASP. Para usar los valores del Top 10 - 2013, fue necesario ajustarlos de manera proporcional a la escala de la metodología de valoración [21] de riesgo de OWASP de la siguiente forma:

“MUY DIFUNDIDO” se ajusta a nueve (9) en la metodología de valoración de riesgo de OWASP, “MODERADO” se ajusta a uno (1), “PROMEDIO” se ajusta a cinco (5), “FÁCIL” se ajusta a siete (7).

Figura 6**. Factores de riesgo Top 10 del A3**



Fuente: OWASP Top 10 - 2013

Con los valores ajustados a la metodología de valoración de riesgo de OWASP y los factores de riesgo del A3 (Figura 6) se les asigna valores a los siguientes factores.

Factor de Facilidad de Descubrimiento: El factor está definido en función de la facilidad para la detección del ataque XSS, que de acuerdo con la escala ajustada, se le asigna el valor de siete (7).

Factor de Facilidad de Explotación: El factor está definido en función de la facilidad para la explotabillidad de una vulnerabilidad con un ataque XSS; según la escala ajustada se le da el valor de cinco (5).

Factor de Conocimiento de la Vulnerabilidad: el factor está definido a identificar que tan conocida es la vulnerabilidad explotada por un XSS; según la escala ajustada se le da el valor de nueve (9) al factor.

Según los factores de riesgo del A3 – Figura 6, el impacto para el A3 es "MODERADO". Usando la escala ajustada se le da el valor de cinco (5) a todos los factores que hacen parte del impacto técnico: Factor de Perdida de confidencialidad, Factor de Perdida de integridad, Factor de pérdida de disponibilidad y Factor de pérdida de control de responsabilidad.

Para determinar los valores de los "*factores de impacto sobre el negocio*" fue necesario contar con el apoyo de la organización. Para ello se solicitó (a la organización) responder un cuestionario de selección múltiple con única respuesta.

Esto permitió definir el valor para los factores restantes. El cuestionario fue elaborado de acuerdo con la Guía de pruebas [26] de OWASP (ver Anexo 5).

A continuación se describe la interpretación del desarrollo del cuestionario mencionado.

Factor de Daño Financiero:

¿Cuánto daño financiero resultaría de la explotación de un ataque XSS?

Las opciones que podía escoger la organización fueron:

1. Menor al coste de arreglar la vulnerabilidad (1).
2. Leve efecto en el beneficio anual (3).
3. **Efecto significativo en el beneficio anual (7).**
4. Bancarrota (9).

La organización escogió la tercera opción. Lo cual quiere decir que para la organización el daño financiero es de consideración puesto que estaría causando un impacto significativo en el beneficio anual de la misma.

Factor de Daño sobre la reputación:

¿La explotación de una vulnerabilidad con un ataque XSS tendría por resultado un daño sobre la reputación del negocio?

Las respuestas que podían escoger fueron:

1. Daño mínimo (1).
2. Pérdida de las cuentas principales (4).
3. **Pérdida del buen nombre (5).**
4. Daño sobre la marca (9).

La organización escogió la tercera opción. Lo cual quiere decir que para la organización el daño de la reputación es de suma importancia, ya que estaría causando la pérdida del prestigio de la organización.

Factor de No conformidad:

¿Qué tanta exposición introduce un ataque XSS a la organización?

1. Violación leve (2).
2. Clara violación (5).
3. **Violación prominente (7).**

La organización escogió la cuarta opción. Lo cual quiere decir que para la organización es critica la exposición de datos cuando recibe un ataque XSS.

Factor de Violación de la privacidad:

¿Si se presenta un ataque XSS, qué tanta información que facilite la identificación personal podría ser revelada?

1. Un individuo (3).
2. **Cientos de personas (5).**
3. Miles de personas (7).
4. Millones de personas (9).

La organización escogió la segunda opción. Lo cual quiere decir que para la organización la violación de la identificación personal es de carácter leve, puesto que la develación de esta información no afectaría el desempeño de la organización.

3.4 SELECCIÓN DE UN CONTROL DE LA ISO/IEC 27002:2013

3.4.1 Relación entre ISO/IEC 27002:2013 y OWASP 2013. Se realizó un primer filtro de los controles de la ISO 27002:2013 [7] aplicando los siguientes criterios:

- No tomar en cuenta los controles que para ámbito administrativo (Dominios del 5 al 8).
- El control debe permitir ajustarse a una aplicación web.
- El control debe permitir ajustarse al listado de los 10 riesgos más críticos en aplicaciones web, según OWASP 2013.

Al resultado de aplicar el filtro anterior se obtiene un nuevo listado de controles, y posteriormente cada control se contrasta con el listado de OWASP, teniendo en cuenta el objetivo del control y si puede aplicar al listado de OWASP. Al finalizar se obtiene una lista de controles que tiene relación con OWASP Top 10 - 2013 [22] y la norma ISO/IEC 27002:2013 [7].

La Tabla 4 presenta el resultado de los posibles controles y las incidencias en el listado de OWASP, donde la columna izquierda hace referencia a los controles de la ISO/IEC 27002:2013 [7], los elementos de primer nivel hacen referencia a los dominios, los de segundo nivel a los objetivos de control y finalmente los de tercer nivel a los controles. En las columnas de la derecha están representados los riesgos del listado de OWASP (A's) y una X para indicar que el control incide en el A correspondiente.

Tabla 4**. Relación entre los controles que pueden aplicarse a un sistema de información en línea de la norma ISO/ IEC 27002 y el listado de los riesgos más críticos según OWASP 2013**

ISO/IEC 27002:2013	OWASP Top 10 - 2013									
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
9. CONTROL DE ACCESOS.										
9.1 Requisitos de negocio para el control de accesos.										
9.1.1 Política de control de accesos.		x		x			x			
9.1.2 Control de acceso a las redes y servicios asociados.		x		x			x			
9.2 Gestión de acceso de usuario.										
9.2.1 Gestión de altas/bajas en el registro de usuarios.				x			x			
9.2.2 Gestión de los derechos de acceso asignados a usuarios.				x			x			
9.2.3 Gestión de los derechos de acceso con privilegios especiales.				x						
9.2.4 Gestión de información confidencial de autenticación de usuarios.		x				x				
9.2.5 Revisión de los derechos de acceso de los usuarios.		x		x			x			
9.2.6 Retirada o adaptación de los derechos de acceso		x		x			x			
9.3 Responsabilidades del usuario.										
9.3.1 Uso de información confidencial para la autenticación.		x		x			x			
9.4 Control de acceso a sistemas y aplicaciones.										
9.4.1 Restricción del acceso a la información.				x			x			
9.4.2 Procedimientos seguros de inicio de sesión.		x								
9.4.3 Gestión de contraseñas de usuario.		x								
9.4.4 Uso de herramientas de administración de sistemas.				x			x			
9.4.5 Control de acceso al código fuente de los programas.				x						

10. CIFRADO.										
10.1 Controles criptográficos.										
10.1.1 Política de uso de los controles criptográficos.		x		x						
10.1.2 Gestión de claves.		x								
12. SEGURIDAD EN LA OPERATIVA.										
12.2 Protección contra código malicioso.										
12.2.1 Controles contra el código malicioso.	x		x					x		
12.6 Gestión de la vulnerabilidad técnica.										
12.6.1 Gestión de las vulnerabilidades técnicas.	x	x	x	x	x	x	x	x	x	x
12.6.2 Restricciones en la instalación de software.									x	
12.7 Consideraciones de las auditorías de los sistemas de información.										
12.7.1 Controles de auditoría de los sistemas de información.		x			x	x				
13. SEGURIDAD EN LAS TELECOMUNICACIONES.										
13.1 Gestión de la seguridad en las redes.										
13.1.1 Controles de red.		x				x				
13.1.2 Mecanismos de seguridad asociados a servicios en red.		x		x						
13.1.3 Segregación de redes.		x		x			x			
13.2 Intercambio de información con partes externas.										
13.2.1 Políticas y procedimientos de intercambio de información.		x				x				x
13.2.2 Acuerdos de intercambio.		x				x				x
13.2.3 Mensajería electrónica.		x				x				
13.2.4 Acuerdos de confidencialidad y secreto.						x				
14. ADQUISICIÓN, DESARROLLO Y MANTENIMIENTO DE LOS SISTEMAS DE INFORMACIÓN.										
14.1 Requisitos de seguridad de los sistemas de información.										
14.1.1 Análisis y especificación de los requisitos de seguridad.					x					
14.1.2 Seguridad de las comunicaciones en servicios accesibles por redes públicas.		x				x				
14.1.3 Protección de las transacciones por redes telemáticas.						x				x
14.2 Seguridad en los procesos de desarrollo y soporte.										
14.2.1 Política de desarrollo seguro de software.	x	x	x	x	x	x	x	x	x	x
14.2.2 Procedimientos de control de cambios en los sistemas.					x					
14.2.3 Revisión técnica de las aplicaciones tras efectuar cambios en el sistema operativo.									x	
14.2.4 Restricciones a los cambios en los paquetes de software.					x				x	
14.2.5 Uso de principios de ingeniería en protección de sistemas.		x		x						x
14.3 Datos de prueba.										
14.3.1 Protección de los datos utilizados en pruebas.						x				

Fuente: Elaboración propia.

3.4.2 Selección del control aplicable al proceso crítico. De los controles comparados, se puede destacar el control 12.6.1 Gestión de las vulnerabilidades técnicas de la Norma ISO/IEC 27002:2013 [7] que se constituye en la base del control de seguridad objeto de estudio. Esto, debido a la capacidad de incidir de forma global en los diez riesgos más críticos según OWASP, dándole la característica de escalar de manera más contundente a cualquiera de estos riesgos por medio de otros controles que lo complementen.

El control de seguridad tiene como objetivo uno de estos riesgos, por lo que se seleccionó un del listado de OWASP, el A3 - Secuencia de Comandos en Sitios Cruzados (XSS). La falta de la validación de las entradas[27] suministradas por el usuario es una de las vulnerabilidades más explotadas por el XSS, lo que convierte a este ataque en el más difundido en las aplicaciones web, además de ser encontrado con relativa facilidad por medio de análisis de código o pruebas [22]. Teniendo en cuenta estas dos características, este riesgo se convierte en el objetivo para el control de seguridad que se aplicó a través del presente trabajo.

Para lograr más efectividad contra el XSS, se complementó el control de seguridad ya mencionado con un control más de la Norma ISO/IEC 27002:2013 [7]: el 12.2.1 - Controles contra el código malicioso, que controla tipos de riesgos como el A1 – Inyección, el ya mencionado A3 – XSS y A8 - Falsificación de Peticiones en Sitios Cruzados (CSRF), teniendo como enfoque el XSS, como se mencionó anteriormente.

En resumen, se tienen los controles 12.2.1 y 12.6.1 de la Norma ISO/IEC 27002:2013 [7], para implementarlos como el control de seguridad objeto del presente estudio, y el riesgo A3 - Secuencia de Comandos en Sitios Cruzados (XSS) como objetivo del control.

3.5 DISEÑO LOGICO DEL CONTROL

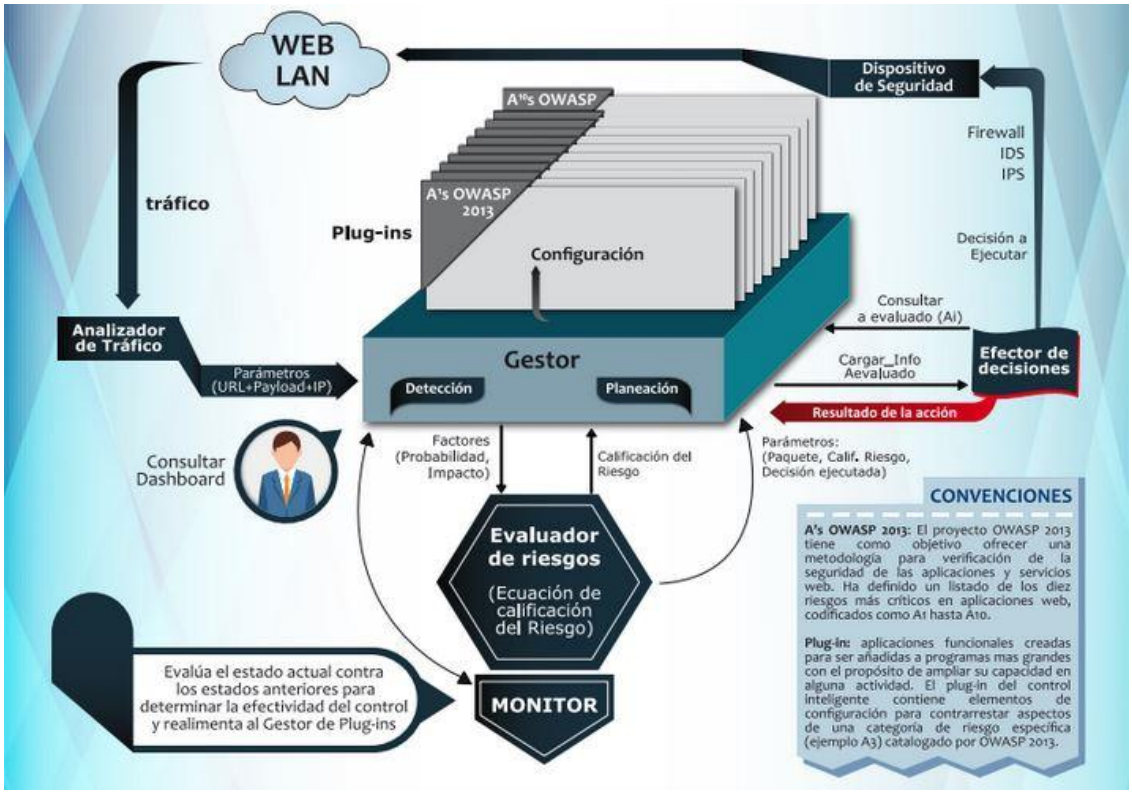
Para este trabajo, al estar enmarcado en el proyecto “Control Inteligente para el servicio crítico de un sistema de información en línea enmarcado en un dominio de la ISO/IEC 27002”, financiado por Fondo Regional para la Innovación Digital en América Latina y el Caribe (FRIDA), se propuso una arquitectura fundamentada en

[27] En este contexto las entradas hacen referencia a la información suministrada por un usuario, generalmente a través de un dispositivo de entrada, como por ejemplo: teclado, mouse u otros dispositivos. (http://goo.gl/Sa6hUU)

la propuesta de V. Teresius [40]. A dicha propuesta se le hizo una adaptación para la implementación del citado proyecto.

El diseño lógico del control muestra claramente la estructura, en sí, del control. Se parte, entonces, de la visión general del proyecto (Figura 7), se presenta la adaptación que se hizo para la implementación del mismo, se hace una descripción general de cada uno de los componentes de la arquitectura y la descripción de los módulos que se seleccionaron para el desarrollo de la arquitectura propuesta.

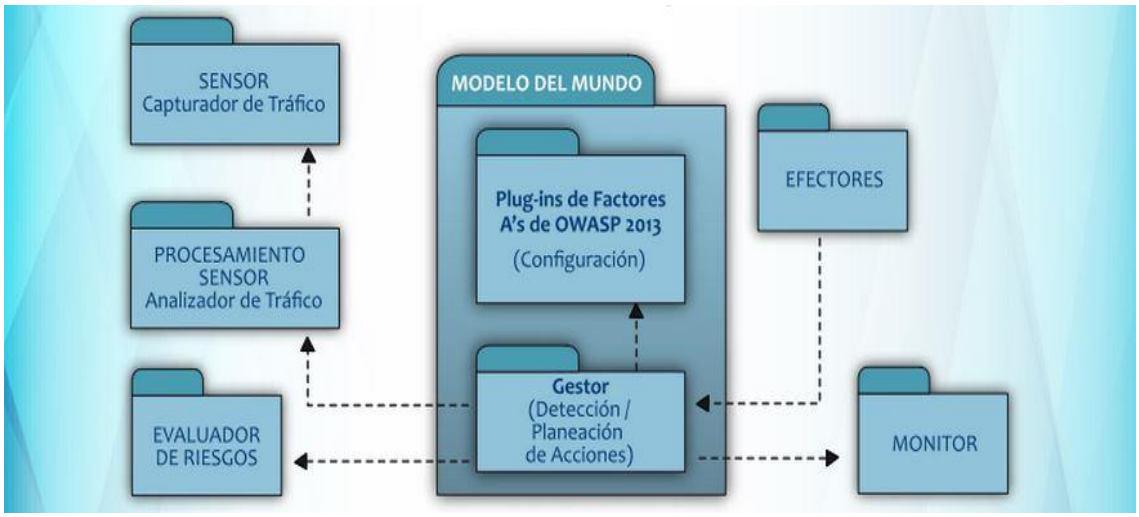
Figura 7. **Visión general del proyecto**.



Fuente: Grupo del Proyecto “Control Inteligente para el servicio crítico de un sistema de información en línea enmarcado en un dominio de la ISO/IEC 27002”

La adaptación de la arquitectura (Figura 8) a partir de la propuesta de V. Teresius [40] permitió que cada módulo estuviera especializado en una tarea concreta, que la intercomunicación se facilitara y que todos los módulos cooperaran para alcanzar una meta común.

Figura 8. **Arquitectura software del control**



Fuente: Grupo del Proyecto "Control Inteligente para el servicio crítico de un sistema de información en línea enmarcado en un dominio de la ISO/IEC 27002"

A continuación se describe de forma general cada uno de los módulos.

MÓDULO CAPTURADOR DE TRAFICO (CT): Este módulo es el encargado de capturar el tráfico de red (paquete), que va dirigido hacia una dirección IP y puerto que se le especifique. El tráfico de red que cumple las condiciones definidas anteriormente es enviado al módulo Analizador de Tráfico.

MÓDULO ANALIZADOR DE TRAFICO (AT): Este módulo es el encargado de obtener la cabecera HTTP, de los elementos enviados por el módulo Capturador de Tráfico y obtener la URL, payload e IP de origen para enviarlos al módulo Gestor.

MÓDULO GESTOR (GE): Este módulo es el encargado de determinar si la URL enviada por el módulo Analizador de Tráfico, contiene un ataque del OWASP Top 10 - 2013, mantener la comunicación entre los módulos Evaluador de Riesgo (ER), Monitor (MR), y el Efector de Decisiones (ED), tomar una decisión si existe un riesgo por un ataque que pertenezca al OWASP Top 10 - 2013, además debe administrar los Plug-Ins construidos en base al OWASP Top 10 - 2013.

MÓDULO EVALUADOR DE RIESGO (ER): Este módulo con los datos recibidos del Gestor, se encarga de generar una calificación en una de las cinco opciones

definidas por OWASP (Crítico, Alto, Medio, Bajo, Nota) y le entrega la valoración del riesgo al Gestor.

MÓDULO MONITOR (MR): Este módulo recibe del gestor la información de la URL, el A detectado, el valor del riesgo calculado a partir del ataque y la decisión tomada y los almacena. También evalúa el estado actual de control contra los estados anteriores y le envía el resultado de esa evaluación al Gestor.

MÓDULO EFECTOR DE DECISIONES (ED): Este módulo es el encargado de recibir del Gestor (La categoría de riesgo, el A de OWASP y calificación de riesgos) y ejecute todas las acciones proporcionas por el módulo Gestor buscando disminuir el riesgo en la Aplicación Web, mediante dispositivos de seguridad como el Firewall, IDS, IPS, Proxy y entre otros que estén disponibles.

Usa la integración de dos patrones [41], el patrón decorador (Decorator Pattern) y el patrón constructor (Builder Pattern), lo que le permite al patrón decorador agregar funcionalidades a un objeto en tiempo de ejecución y el patrón builder agrega las funcionalidades necesarias al efector para que sea capaz de ejecutar las acciones de un plan al construir las funcionalidades necesarias según lo requiera el plan.

3.5.1 Módulos de la propuesta. Se propuso un prototipo basado en un control de la ISO/IEC27002:2013 [7] que tome información del tráfico dirigido hacia una aplicación web, realice una medición de riesgo para ataques XSS y tome acciones que permitan optimizar la seguridad de dicha aplicación.

Siguiendo la arquitectura propuesta se desarrollaron los siguientes módulos: Capturador de Tráfico, Analizador de Tráfico, Gestor, Evaluador de Riesgos.

Para la comunicación entre módulos se usó el patrón fachada, a continuación se describe a profundidad cada módulo.

3.5.1.1 Módulo Capturador de Tráfico. Este módulo está encargado de identificar los dispositivos Ethernet disponibles, también es el responsable de transformar el tráfico de la red en objetos (paquetes) y con el usó de las librerías Libpcap para un sistema operativo Linux o WinPcap para un sistema operativo Windows, obtener el tráfico Ethernet dirigido hacia el Sistema de Información en línea que se monitorea, es también responsable de enviar los paquetes al módulo Analizador de tráfico, para ser revisados.

Funciones del módulo:

- Capturar el tráfico de red y convertirlo en un objeto Pcap.
- Enviar el tráfico capturado al Analizador de Tráfico
- Identificar los dispositivos Ethernet (Tarjetas de Red)

3.5.1.2 Módulo Analizador de Tráfico. Este módulo se encargar del análisis de los paquetes enviados por el módulo de Capturador de tráfico.

En cada paquete que se recibe, se verifica la existencia de la cabecera http, para obtener la URL, la cual se decodifica mediante un conjunto de métodos creados para tal fin y por medio de librería que implementa listas blancas, se determina si el tráfico es sospechoso.

En caso de no estar la cabecera http o que el tráfico no sea sospechoso se descarta el paquete.

Otra función de este módulo es obtener por medio de expresiones regulares el vector característico[28] de un tráfico sospecho y que es usado luego por una Técnica de Machine Learning para clasificar el ataque.

Si un tráfico se considera sospechoso, la siguiente información es enviada al Gestor: “Timestap[29], URL decodificada, IP de origen, URL, contenido del Payload, vector característico”.

[28] Para el desarrollo de este trabajo un vector característico, es una cadena de caracteres formado por doce elementos, cada elemento representa si está presente o no, los grupos del numeral 4.5.6.
[29] Timestamp: Secuencia de caracteres que denotan fecha y hora. (http://goo.gl/a75tte).

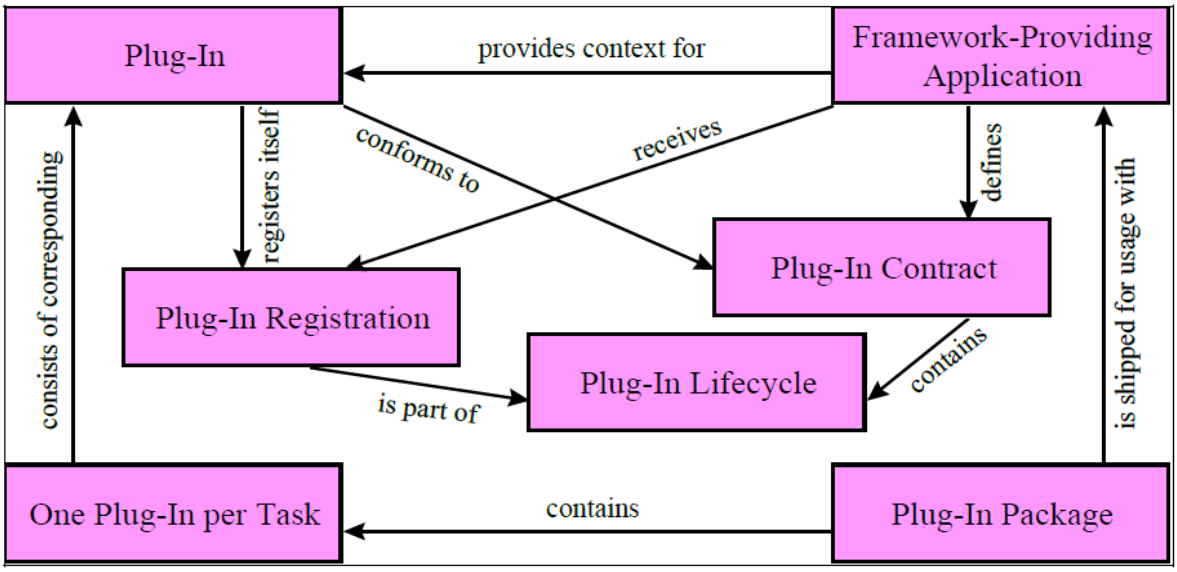
Funciones del Módulo:

- ❖ Decodificar la URL
- ❖ Determinar si una URL o Payload contienen elementos no permitidos (tráfico sospechoso)
- ❖ Obtener el vector característico de un tráfico sospechoso.
- ❖ Enviar al Gestor información obtenida de un tráfico sospechoso.

3.5.1.3 Módulo del Gestor. Este es el módulo central del control, se encarga de la comunicación entre los módulos (Evaluador de Riesgo, Analizador de Tráfico) y así apoyar las tareas del control.

Este módulo también se encarga de la gestión de los Plug-In, esta gestión se creó teniendo como base los patrones para Plug-Ins (“Patterns for Plug-Ins”) de Klaus Marquardt [42], quien define como debe ser un Plug-In (Patrón 1), como registrarlo (Patrón 4), y como usarlo (Patrón 5).

Figura 9. **Patrones para Plug-Ins**



Fuente: MARQUARDT, Klaus: Patterns for Plug-Ins [43]

La Figura 9, presenta el esquema general para el desarrollo de un Plug-Ins propuesta por Klaus Marquardt.

También se encarga de procesar las solicitudes del Módulo Analizador de Tráfico, para detectar si el tráfico sospechoso pertenece a un A del OWASP 2013.

Por último también se encarga de crear un plan de acciones que tendrá en cuenta: el valor del riesgo, el tipo de ataque, los dispositivos de seguridad (como firewall o IDS), o dispositivos para informar (como él envió de un correo o mensajes de texto).

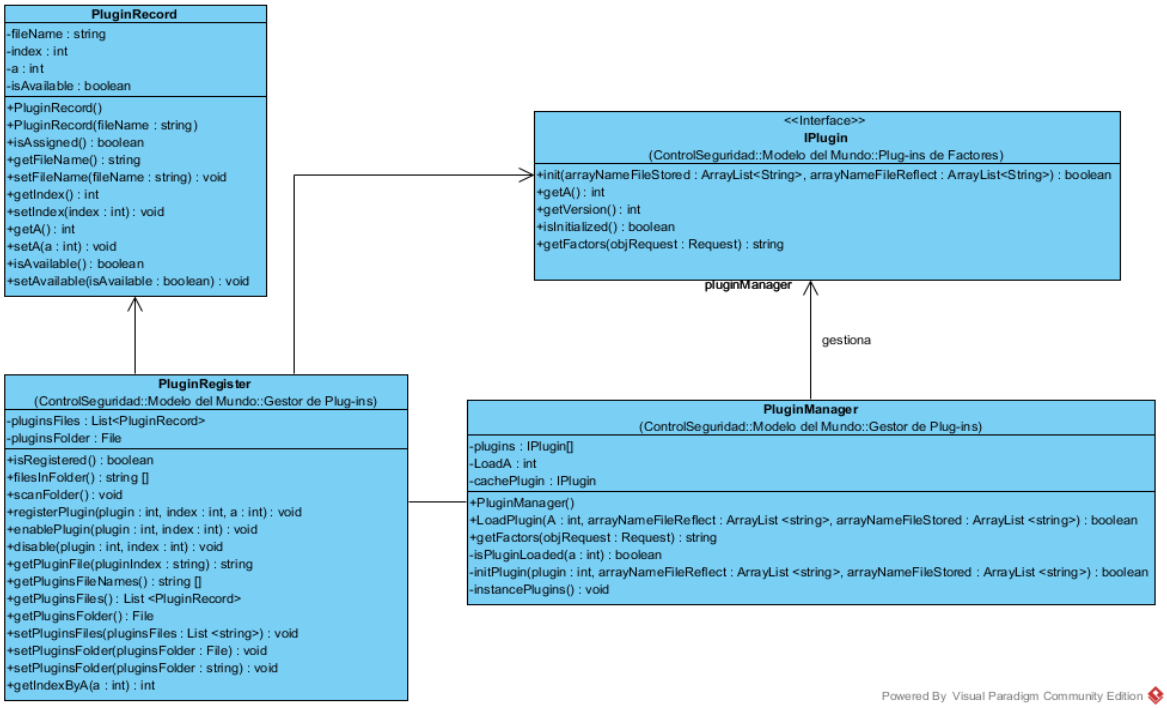
Funciones:

- La detección de un ataque XSS apoyado de una Técnica de Machine Learning.
- La configuración apoyada en Plug-Ins especializados.
- La planeación de las acciones a tomar para la protección del sitio monitoreado.
- La validación del estado del dispositivo Ethernet (Activo o Desactivado).
- Informar a la bitácora los errores.
- Enviar los resultados a la bitácora
- La creación y carga del inventario de página de la organización

- **Plug-In.** El Plug-In no hace parte de la arquitectura del control de seguridad, fue pensado como un elemento de soporte que permite extender las funcionalidades del control en tiempo de ejecución. Los Plug-Ins se encargan de administrar los valores asociados a los factores de cada A de OWASP.

  En la Figura 10, se presenta el diagrama de clases diseñado para administrar los plugins dentro del control de seguridad y que resulto de la adaptación de los Patrones para Plug-Ins de Klaus Marquardt [43].

Figura 10. **Adaptación de los Patrones para Plug-Ins para el control de seguridad**



Fuente: Elaboración Propia

Para este trabajo se categorizó el A3, lo que sólo permitió construir el Plug-In para esta categoría de riesgo.

3.5.1.4 Módulo Evaluador de Riesgo. Este módulo se encarga del cálculo del riesgo basado en la metodología de valoración del riesgo, especificada en la guía de pruebas de OWASP [21].

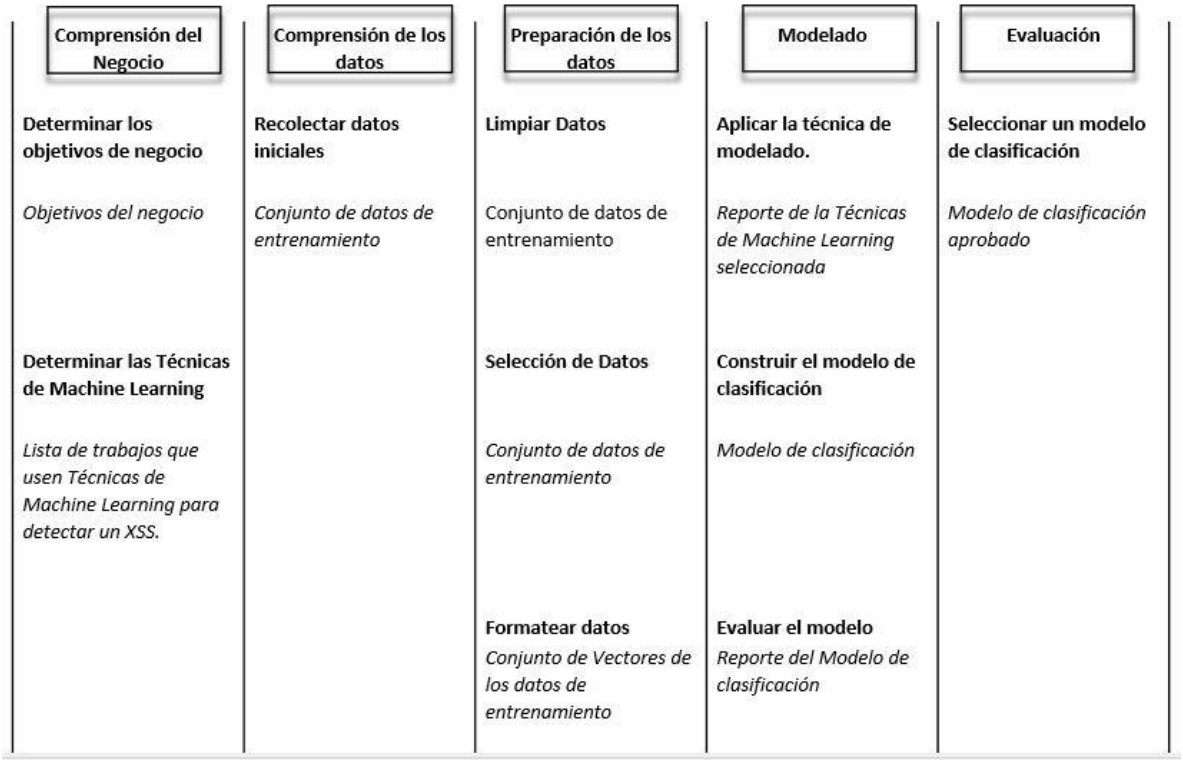
Se recibe del Módulo Gestor los valores de cada uno de los factores que se especificaron en el numeral 3.3 y se calcula el riesgo. El riesgo calculado es enviado al gestor, el cual lo usara como insumo para crear un plan de acciones.

3.6 DETECCIÓN INTELIGENTE DEL CONTROL DE SEGURIDAD

Para seleccionar una técnica de inteligencia se decidió adaptar la metodología de CRISP-DM, compuesta por 6 fases, las fases seleccionadas fueron: comprensión del negocio, comprensión de datos, preparación de datos, modelado y evaluación.

En la Figura 11 se presenta la adaptación de la metodología CRISP-DM para seleccionar una técnica de inteligencia artificial, las fases usadas, acompañadas por tareas y salida.

Figura 11. **Tareas y salidas de la adaptación de la Metodología CRIDP-DM para seleccionar una técnica de inteligencia artificial.**



Fuente: Elaboración Propia.

A continuación se presenta lo que se realizó en cada fase seleccionado.

3.6.1 Fase I. Comprensión del negocio. En esta fase se revisó la situación actual del riesgo de las aplicaciones web, para esto se usó el proyecto OWASP Top 10 - 2013, en el numeral 2.3.2 se explica que el Cross Site Scripting XSS (Secuencia de Comandos en Sitios Cruzados), es un ataque que aprovecha la falta de mecanismos de validación de la información ingresada en una aplicación web. Lo que permite al atacante ejecutar secuencias de códigos en el navegador de la víctima **[44]**.

En [33], [34], [35] y [36] es identificado y caracterizado los ataques XSS, con el uso de diferentes Técnicas de Machine Learning han logrado determinar si una URL o el contenido de un sitio web contiene un ataque XSS o no.

Con lo anterior se propone los siguientes objetivos: Identificar y determinar la complejidad (NO_ATAQUE, BAJO, MEDIO, ALTO) de un ataque XSS por medio de una técnica de Machine Learning y seleccionar el modelo de clasificación que use por lo menos el 70% de las características definidas.

3.6.2 Fase II. Comprensión de datos. Se exploró un sitio web denominado XSSED (http://www.xssed.com/), en el que se reporta al público incidencias de XSS en las aplicaciones web. Con el fin de recolectar datos iniciales, el 04 de septiembre de 2014, se usó el sistema operativo KaliLinux y el comando wget para descargar todas las páginas del sitio XSSED. Se construyó una aplicación que buscó, obtuvo y almacenó en un archivo, la URL del ataque XSS de cada página web del sitio XSSED descargado. Sólo se tuvo en cuenta aquellas que tenían en su interior "Category" XSS. Es importante aclarar esto debido a que también se reportan incidencias de redirección ("Category: Redirect") las cuales no son consideradas ataques XSS, este proceso permitió tener un primer conjunto de datos de entrenamiento ("Conjunto de datos de entrenamiento 36").

Se revisó la herramienta OWASP Xenotix XSS Exploit Framework. Según Cristian [45] afirma que se trata de una herramienta muy eficiente para detectar vulnerabilidades que pueden ser explotadas por ataques XSS, debido a que dispone de 1.593 ejemplares de ataques, desde los básicos, hasta los capaces de saltar un WAF[30]. Para obtener estos datos se usó el sistema operativo KaliLinux y la herramienta TCPDUMP para capturar y almacenar en un archivo todo el tráfico que salía de herramienta OWASP Xenotix XSS Exploit Framework y que tenía como destino sitio web vulnerable de pruebas Damn Vulnerable Web Application (DVWA).

[30] Waf: Dispositivo que puede ser hardware o software que analiza el tráfico web. (https://goo.gl/ocu9z).

Con los ejemplos de ataques obtenidos al usar TCPDUMP se construye un segundo conjunto de datos de entrenamiento (Conjunto de datos de entrenamiento 1500).

3.6.3 Fase III. Preparación de datos. A cada elemento de los conjuntos de datos de entrenamiento fue necesario realizarle un proceso de decodificación[31] que consistió en dejar todos los caracteres del conjunto de datos de entrenamiento en un solo tipo de codificación, para el caso se escogió ASCII; para obtener información adicional de esta codificación visitar la siguiente página web: http://www.elcodigoascii.com.ar. Este proceso de decodificación fue pensado para tener una mayor velocidad en la búsqueda de características.

Se revisó la forma cómo están construidos los ataques sobre el "Conjunto de datos de entrenamiento 36" y el "Conjunto de datos de entrenamiento 1500"; en esta última se encontró mayor variedad en los ataques: ataques comunes y no comunes, por lo cual se decidió usar el "Conjunto de datos de entrenamiento 1500" para construir el modelo de clasificación y usar el "Conjunto de datos de entrenamiento 36" para verificar el modelo de clasificación.

Para seleccionar las características se utilizó la guía XSS Filter Evasion Cheat Sheet [46] diseñada con el fin de que las organizaciones prueben, que tan vulnerables son sus aplicaciones web a los ataques y se siguieron las recomendaciones de Del Moral [47] que sugiere el uso de listas blancas[32] para mitigar el XSS. Teniendo en cuenta estas sugerencias, y haciendo un revisión sobre los conjuntos de datos de entrenamiento 36 y 1500, se identificaron las siguientes características (atributos) necesarias para determinar si un ejemplar es un ataque XSS o no.

3.6.3.1 Característica 1. Html Character Entity Name. Esta característica se identifica teniendo como referencia la presencia de caracteres reservados en HTML Entity Name. Estos tienen la siguiente estructura: *"&entity_name;"*.

Tabla 5**. Ejemplos de Html Character Entity Name**

Resultado	Descripción	Entity Name
<	Signo Menor que	&lt;
&	Signo &	&amp;

Fuente: Elaboración propia

[31] Decodificar: Aplicar inversamente las reglas de su código a un mensaje codificado para obtener la forma primitiva de este. (http://goo.gl/QjGaWt)
[32] Lista Blanca: Contiene una lista de los elementos aceptados en un HTML. (http://goo.gl/rjubq5).

3.6.3.2 Característica 2. Html Character Entity Number. En esta característica están los caracteres reservados en HTML Entity Number; presentan la siguiente estructura: "*&#entity_number;"*.

Tabla 6**. Ejemplos de Html Character Entity Number**

Resultado	Descripción	Entity Number
<	Signo Menor que	<
&	Signo &	&

Fuente: Elaboración propia

3.6.3.3 Característica 3. Codificación Unicode. En esta característica están los elementos más comunes de la codificación Unicode, que tengan la notación: *"UHex".*

Tabla 7**. Ejemplos de Caracteres en Unicode**

Resultado	Descripción	Unicode
<	Signo Menor que	U003C
&	Signo &	U0026

Fuente: Elaboración propia

3.6.3.4 Característica 4. Codificación Hexadecimal. En esta característica están los elementos HTML codificados de la siguiente forma: *"%Hex".*

Tabla 8**. Ejemplos Caracteres en Hexadecimal**

Resultado	Descripción	Unicode
<	Signo Menor que	%3C
&	Signo &	%26

Fuente: Elaboración propia

3.6.3.5 Característica 5. Search Comment. Se identificó este característica debido a que se observó el uso de comentarios HTML en los ataques *"<!--…-->"* para evadir los filtros.

3.6.3.6 Característica 6. Search Document Cookie. Esta característica aparece debido a que JavaScript tiene la función *"document.cookie()"* la cual es muy usada por los atacantes para obtener las cookies de una sesión de una aplicación web.

3.6.3.7 Característica 7. Search Document Write. Esta característica se identifica porque JavaScript tienen la función *"document.write()"* la cual le permite a un atacante modificar el documento HTML.

3.6.3.8 Característica 8. Search From Char Code. Se establece esta característica al observar que varios ataques usan la función *"String.fromCharCode()"* la cual a partir de números crea una secuencia de caracteres en Unicode.

3.6.3.9 Característica 9. Search Functions. Debido a que muchos ataques de XSS usan funciones, se identifica esta característica.

3.6.3.10 Característica 10. Search On. Los eventos HTML permiten al atacante inyectar código malicioso. Estos eventos se caracterizan porque su estructura empieza por "on", lo que permitió establecer esta característica.

3.6.3.11 Característica 11. Search Protocol. Los ataques tienen por objetivo enviar la información obtenida al atacante, por esto aparece esta característica con el fin de identificar los protocolos más comunes de internet.

3.6.3.12 Característica 12. Search Tags[33]. Los Tags son el principal elemento usado para realizar ataques XSS, esto llevó a la creación esta característica.

3.6.3.13 Característica 13. Return Carriage. Debido a que el *retorno de carro* no permitía visualizar adecuadamente la cadena de caracteres que contiene un ataque XSS, se adicionó esta característica para determinar si en el ataque estaba presente.

3.6.3.14 Característica 14. New Line. Esta característica se identificó por el mismo motivo de la Característica 13. Return Carriage, esto debido a que el *salto de línea* no permitía visualizar adecuadamente una cadena.

[33] Tag: Marca que se dejan en un texto para que luego sean interpretadas, sobre el mismo texto marcado. Por ejemplo, el lenguaje HTML es interpretado y mostrado por un navegador. (http://goo.gl/BlSt91).

Mientras se identificaban las características, se observó que el conjunto de datos de entrenamiento 1500" presentaba mayor variedad en los ataques: ataques comunes y no comunes, por lo cual se decidió usar el "Conjunto de datos de entrenamiento 1500" para construir el modelo de clasificación y usar el "Conjunto de datos de entrenamiento 36" para verificar el modelo de clasificación.

La "Conjunto de datos de entrenamiento 36" se dividió en cuatro nuevas conjuntos de datos de entrenamiento: "Conjunto de datos de entrenamiento 36-1", "Conjunto de datos de entrenamiento 36-2", "Conjunto de datos de entrenamiento 36-3" y "Conjunto de datos de entrenamiento 36-4", esto permitió evaluar y seleccionar el modelo de clasificación construido. En el Anexo 6, se encuentran los conjuntos de datos de entrenamiento, obtenidos inicialmente.

Una vez definidas las características, se creó una aplicación la cual mediante expresiones regulares identificó cada elemento de la Conjunto de datos de entrenamiento 1500 con un "1" si presenta la característica y "0" en caso contrario. Es de anotar que el vector de características (vector característico) siempre va a tener un tamaño fijo de catorce elementos.

3.6.4 Fase IV. Modelado. Esta fase tuvo como objetivo identificar Técnicas de Machine Learning usadas para identificar ataques XSS y obtener datos iniciales de ataques XSS.

En [33], [34], [35] y [36] se han usado diferentes Técnicas de Machine Learning, para determinar si una URL o el contenido de un sitio web contiene un ataque XSS o no. Para este trabajo y teniendo en cuenta los anteriores artículos se escogieron las Técnicas de Machine Learning: Arboles de Decisión (Clasificador J48), Naive Bayes y Perceptron Multicapa (clasificadores) y se adaptó el método que siguieron los autores, no sólo para determinar si es un ataque de XSS o no, sino también para inferir el nivel de complejidad con el que un ataque XSS está construido.

Se usó la herramienta de minería de datos WEKA (Waikato Environment for Knowledge Analysis en Español "entorno para análisis del conocimiento de la Universidad de Waikato") (http://www.cs.waikato.ac.nz/ml/weka/) para experimentar y construir los modelos de clasificación entrenados con los clasificadores J48, Naive Bayes y Perceptron Multicapa. Para los clasificadores J48 y Naive Bayes se usó la configuración predeterminada proporcionada por WEKA. Para el Perceptron Multicapa de la configuración proporcionada por WEKA, sólo se modificó la capa de entrada: con el número de variables independientes (características) y se agregó

una capa intermedia formada por el 85% de la capa de entrada, siguiendo las recomendaciones de algunos autores [48].

Todos los modelos de clasificación son construidos con la técnica de validación cruzada[34] para garantizar la independencia entre datos de entrenamiento y prueba.

Para cada prueba experimental se usó entrenamiento supervisado, el cual consistió en conocer para cada vector característico del "Conjunto de datos de entrenamiento 1500" la clasificación (NO_ATAQUE, BAJO, MEDIO, ALTO) de un experto en seguridad informática; para clasificar el experto sólo tuvo la definición de las características y los vectores característico del "Conjunto de datos de entrenamiento 1500".

Con los vectores característicos y las clasificaciones del experto se generó un archivo ".arff" con la estructura requerida para ser leído por WEKA, por cada prueba experimental realizada.

A continuación se describen las pruebas experimentales que se realizaron con las Técnicas de Machine Learning, lo que permitió escoger una de ellas, para construir un modelo de clasificación.

3.6.4.1 Prueba Experimental Uno. Para esta primera prueba se le pidió al experto clasificar el "Conjunto de datos de entrenamiento 1500" sólo en dos grupos (ALTO y BAJO). Se usó la configuración por defecto de los clasificadores (algoritmos de clasificación) para tener valores de referencia con la clasificación más básica.

Los resultados de usar los modelos de clasificación con los algoritmos: J48, Naive Bayes y Perceptron Multicapa se presentan en la tabla 9:

[34] Validación Cruzada: Técnica utilizada para evaluar los resultados y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Esta técnica es muy utilizada en proyectos de inteligencia artificial para validar modelos generados. (http://goo.gl/jPYLu2).

Tabla 9**. Resultados de la Prueba Experimental Uno**

Algoritmo de aprendizaje de máquina	Instancias clasificadas correctamente	Porcentaje correctamente clasificado	Instancias No clasificadas correctamente	Porcentaje no clasificado correctamente
J48	1593	100%	0	0%
Naive Bayes	1592	99.9372%	1	0.0628%
Perceptron Multicapa	1593	100%	0	0%

Fuente: Elaboración propia

De la tabla 9 se puede observar que Naive Bayes fue el único que se equivocó, mientras que la clasificación de los otros dos fue perfecta.

En el Anexo 7, se da a conocer los resultados de cada una de las pruebas experimentales realizadas - Prueba Experimental Uno, Dos, Tres, Cuatro, Cinco, Seis - las cuales presentan los modelos de clasificación generados y que pueden ser visualizados usando la herramienta WEKA y se presenta una imagen con la descripción de los resultados de cada algoritmo.

3.6.4.2 Prueba Experimental Dos. Luego el experto clasificó los datos en tres grupos, (ALTO, MEDIO, BAJO), los resultados obtenidos fueron los siguientes.

Tabla 10**. Resultados de la Prueba Experimental Dos**

Algoritmo de aprendizaje de máquina	Instancias clasificadas correctamente	Porcentaje correctamente clasificado	Instancias No clasificadas correctamente	Porcentaje no clasificado correctamente
J48	1560	97.9284%	33	2.0716%
Naive Bayes	1507	94.6014%	86	5.3986%
Perceptron Multicapa	1570	98.5562%	23	1.4438%

Fuente: Elaboración propia

De la tabla 10 se observa que el algoritmo de Naive Bayes, sigue siendo el peor de los tres algoritmos seleccionados, y se empieza a evidenciar una superioridad en el Perceptron Multicapa.

3.6.4.3 Prueba Experimental Tres. Se le solicitó al experto nuevamente clasificar los datos, pero en cuatro grupos (ALTO, MEDIO, BAJO, NO_ATAQUE). Los resultados se aprecian en la tabla que aparece a continuación.

Tabla 11**. Resultados de la Prueba Experimental Tres**

Algoritmo de aprendizaje de máquina	Instancias clasificadas correctamente	Porcentaje correctamente clasificado	Instancias No clasificadas correctamente	Porcentaje no clasificado correctamente
J48	1578	99.0584%	15	0.9416%
Naive Bayes	1555	97.6146%	38	2.3854%
Perceptron Multicapa	1587	99.6234%	6	0.3766%

Fuente: Elaboración propia

En la tabla 11 se continúa presentando una alta tasa de acierto en todas las Técnicas de Machine Learning y se sigue evidenciando una superioridad en el algoritmo del Perceptron Multicapa.

3.6.4.4 Prueba Experimental Cuatro. Se agregan al archivo “Conjunto de datos de entrenamiento 1500” 1138 no ataques, para un total de 2731 datos; esto se realizó sólo para observar el comportamiento de los algoritmos. Los resultados obtenidos se presentan en la tabla 12:

Tabla 12**. Resultados de la Prueba Experimental Cuatro**

Algoritmo de aprendizaje de máquina	Instancias clasificadas correctamente	Porcentaje correctamente clasificado	Instancias No clasificadas correctamente	Porcentaje no clasificado correctamente
J48	2715	99.4141%	16	0.5859%
Naive Bayes	2602	95.2765%	129	4.7235%
Perceptron Multicapa	2719	99.5606%	12	0.4394%

Fuente: Elaboración propia

En la tabla 12, se comprobó que agregar datos de una sola característica (NO_ATAQUE), el porcentaje de acierto se mantuvo alto y el Perceptron Multicapa siguió siendo la mejor Técnica de Machine Learning.

3.6.4.5 Prueba Experimental Cinco. En el numeral 3.6.4.3 se generaron tres modelos de clasificación con resultados muy superiores a los esperados. Esta prueba se generó buscando verificar si cumple con el objetivo de estar usando el 70% de las características seleccionadas.

Para determinar los algoritmos que permitieran verificar cuáles grupos están siendo usados, se siguió el "Tutorial de Weka 3.6.0" de la universidad Carlos III de Madrid, creado por Aler [49].

El autor [49] recomienda usar las siguientes combinaciones de algoritmos *"InfoGain con Ranker", "CfsSubsetEval*y con Greedy Stepwise", "*ClassifierSubsetEval* con *PART* y con Greedy Stepwise"). Para todos los algoritmos se usa validación cruzada de 5 iteraciones:

- **Evaluación de las 14 características seleccionadas**

3.6.4.5.1 Primera evaluación de las 14 características. Usando la herramienta WEKA se escogieron los siguientes algoritmos:

Attribute Evaluator: *InfoGain*
Selected Method: *Ranker*

El autor[49] menciona que Ranker ordena los atributos de mayor a menor importancia.

Tabla 13**. Resultados primera evaluación de las 14 características.**

Orden Según Ranker	Características (Atributos)
1	SearchTags
2	SearchFunctions
3	Codificacion_HEXADECIMAL
4	SearchOn
5	Codificacion_Unicode
6	HTMLName
7	SearchProtocol
8	NewLine
9	ReturnCarriage
10	SearchComment
11	SearchDocumentWrite
12	SearchDocumentCookie
13	SearchFromCharCode
14	HTMLNumber

Fuente: Elaboración propia

En la tabla 13 se pueden apreciar los resultados obtenidos, y se observa, también, el orden de prioridad que le da el algoritmo Ranker a los atributos.

3.6.4.5.2 Segunda evaluación de las 14 características. En la herramienta WEKA, se seleccionaron los siguientes algoritmos para verificar el uso de las características en cinco iteraciones con validación cruzada:

Attribute Evaluator: *CfsSubsetEvaly*.
Selected Method: Greedy Stepwise*.*

Tabla 14**. Resultados segunda evaluación de las 14 características.**

Numero de Iteraciones	Características (Atributos)
0 (0 %)	HTMLNumber
4 (80 %)	HTMLName
1 (20 %)	Codificacion_Unicode
5 (100 %)	Codificacion_HEXADECIMAL
0 (0 %)	SearchComment
0 (0 %)	SearchDocumentCookie
0 (0 %)	SearchDocumentWrite
0 (0 %)	SearchFromCharCode
5 (100 %)	SearchFunctions
5 (100 %)	SearchOn
0 (0 %)	SearchProtocol
5 (100 %)	SearchTags
2 (40 %)	ReturnCarriage
0 (0 %)	NewLine

Fuente: Elaboración propia

En la tabla 14 se presentan los datos obtenidos con la herramienta WEKA, en la primera columna se colocó el uso de las características en cada iteración y su porcentaje.

3.6.4.5.3 Tercera Evaluación de los 14 Grupos. Para esta evaluación se escogió la siguiente combinación de algoritmos:

Attribute Evaluator: *ClassifierSubsetEval* y como **Classifier**: *PART*
Selected Method: Greedy Stepwise*.*

Tabla 15**. Resultados tercera evaluación de las 14 características**

Número de Iteraciones	Características (Atributos)
0 (0 %)	HTMLNumber
5 (100 %)	HTMLName
5 (100 %)	Codificacion_Unicode
5 (100 %)	Codificacion_HEXADECIMAL
0 (0 %)	SearchComment
0 (0 %)	SearchDocumentCookie
0 (0 %)	SearchDocumentWrite
0 (0 %)	SearchFromCharCode
5(100 %)	SearchFunctions
5(100 %)	SearchOn
5(100 %)	SearchProtocol
5(100 %)	SearchTags
0(0 %)	ReturnCarriage
0(0 %)	NewLine

Fuente: Elaboración propia

En la tabla 15, se observó que sólo 7 de las 14 características (50%), son tenidas en cuenta.

En las pruebas realizadas en los numerales 3.6.4.5.2 y 3.6.4.5.3, se detectó que las 14 características seleccionadas no cumplían con el objetivo de usar al menos el 70% de las características, lo que hizo necesario revisar las características y realizar otra prueba experimental.

3.6.4.6 Prueba Experimental Seis. Al observar que existían características que no se usaban y para dar cumplimiento al objetivo, fue necesario repetir el proceso de caracterización de los ataques XSS, lo que dio como resultado las siguientes modificaciones:

- HTML Name y Html Number pasan a ser parte de un solo grupo y se le da como nombre “HTML Code”.
- Los grupos “Return Carriage” y “New Line” pasan a formar el grupo “Control Character”.

- El grupo “SearchDocumentWrite” se amplía, pasa de buscar solo “Document.Write” a buscar cualquier “Document.”, diferente de “Document.cookie” y todo lo relacionado con “Location”, este nuevo grupo pasa a ser llamando “Search Document Location”.

Se solicitó al experto nuevamente categorizar cada elemento del “Conjunto de datos de entrenamiento 1500” en los siguientes grupos: (ALTO, MEDIO, BAJO, NO_ATAQUE).

Estas modificaciones llevaron a tener nuevas características (Atributos), los cuales se describen a continuación:

3.6.4.6.1 Característica 1. HTML Codes. Esta nueva característica es la unión de elementos HTML Name (&Name) y HTML Number (&#Hexadecimal), se activa si encuentra cualquiera de los dos elementos.

3.6.4.6.2 Característica 2. Codificación Unicode. Conjunto de caracteres universal que pretende incluir todos los caracteres necesarios para cualquier sistema de escritura del mundo, está definido por números hexadecimales y un prefijo U, por ejemplo: “U+0041 representa la letra A”[50].

3.6.4.6.3 Característica 3. URL Encoding. Se activa si encuentra valores numéricos del 0 al 9 y las letras A, B, C, D, E, F precedidos del símbolo %, por ejemplo: “%25 representa el carácter %”

3.6.4.6.4 Característica 4. Search Comment. En esta característica se identifica si existen comentarios de HTML dentro de una cadena de caracteres.

3.6.4.6.5 Característica 5. Search Document Cookie. Esta característica verifica si existe el elemento document.cookie en una cadena de caracteres.

3.6.4.6.6 Característica 6. Search Document Location. Esta característica explora si hay elementos DOM.

3.6.4.6.7 Característica 7. Search From Char Code. Esta característica busca si hay elementos con la forma ".fromcharcode".

3.6.4.6.8 Característica 8. Search Functions. Esta característica explora si existen funciones dentro de una cadena de caracteres.

3.6.4.6.9 Característica 9. Search Html Event Attributes. Esta característica busca eventos "on" que pertenecen al HTML dentro de una cadena de caracteres.

3.6.4.6.10 Característica 10. Search Protocol. Esta característica identifica si existen protocolos dentro de una cadena de caracteres.

3.6.4.6.11 Característica 11. Search Tags. Esta característica verifica la existencia de elementos Tags dentro de una cadena de caracteres.

3.6.4.6.12 Característica 12. Control Character. Esta característica, busca si existen caracteres de control no imprimibles, como salto de línea y retorno de carro.

Los resultados obtenidos, luego de aplicar los algoritmos fueron:

Tabla 16**. Resultados de la Prueba Experimental Seis**

Algoritmo de aprendizaje de máquina	Instancias clasificadas correctamente	Porcentaje correctamente clasificado	Instancias No clasificadas correctamente	Porcentaje no clasificado correctamente
J48	1559	97.8657%	34	2.1343%
Naive Bayes	1424	89.3911%	169	10.6089%
Perceptron Multicapa	1564	98.1795%	29	1.8205%

Fuente: Elaboración propia

Con las nuevas características, se siguió observando que Naive Bayes, en precisión, es la peor de las tres Técnicas de Machine Learning y que el Perceptron Multicapa aparenta ser la mejor opción.

Con los nuevos atributos de esta prueba experimental, era necesario conocer el porcentaje de uso, lo cual llevó a repetir el proceso de evaluación de las características.

- **Evaluación de las 12 características seleccionados**. En esta sección se procedió a evaluar si la herramienta WEKA cumple con la condición de usar el 70% de los atributos. Para ello se repite el proceso que se realizó en el numeral 3.6.3.

3.6.4.6.13 Primera evaluación de las 12 características.

Attribute Evaluator: *InfoGain*
Selected Method: *Ranker*

Los resultados obtenidos y clasificados de mayor a menor, fueron:
Tabla 17**. Resultados primera evaluación de los 12 características**

Orden Según Ranker	Características (Atributos)
1	SearchOn
2	SearchTags
3	SearchFunctions
4	Codificacion_Unicode
5	SearchProtocol
6	Codificacion_HEXADECIMAL
7	SearchComment
8	HTMLCodes
9	SearchDocumentLocation
10	SearchDocumentCookie
11	SearchFromCharCode
12	ControlCharacter

Fuente: Elaboración propia

En la tabla 17, se puede evidenciar el orden de prioridad que le asigna el algoritmo de Ranker a los atributos.

3.6.4.6.14 Segunda evaluación de las 12 características.

Attribute Evaluator: *CfsSubsetEvaly*.
Selected Method: Greedy Stepwise*.*

Tabla 18**. Resultados segunda evaluación de las 12 características.**

Numero de Iteraciones (%)	Características
0 (0 %)	HTMLCodes
5 (100 %)	Codificacion_Unicode
3 (60 %)	Codificacion_HEXADECIMAL
5 (100 %)	SearchComment
4 (80 %)	SearchDocumentCookie
0 (0 %)	SearchDocumentLocation
0 (0 %)	SearchFromCharCode
5 (100 %)	SearchFunctions
5 (100 %)	SearchOn
5 (100 %)	SearchProtocol
5 (100 %)	SearchTags
4 (80 %)	ControlCharacter

Fuente: Elaboración propia

En la tabla 18, se observó que el 75% de los grupos fueron usados en las 5 iteraciones.

3.6.4.6.15 Tercera evaluación de las 12 características.

Attribute Evaluator: *ClassifierSubsetEval* y como **Classifier**: *PART*
Selected Method: Greedy Stepwise*.*

En la tabla 19 que se presenta a continuación, se evidencio que la herramienta está usando más del 80% de los atributos, sólo estaban dos características en cero, se decide no retirar estas características porque en el “Conjunto de datos de entrenamiento 36”, se observó una alta presencia de “document.cookie” y caracteres que representan salto de línea y/o un retorno de carro.

Tabla 19**. Resultados tercera evaluación de las 12 características.**

Número de Iteraciones	Características
5(100 %)	HTMLCodes
5(100 %)	Codificacion_Unicode
5(100 %)	Codificacion_HEXADECIMAL
4(80 %)	SearchComment
0(0 %)	SearchDocumentCookie
4(80 %)	SearchDocumentLocation
5(100 %)	SearchFromCharCode
5(100 %)	SearchFunctions
5(100 %)	SearchOn
5(100 %)	SearchProtocol
5(100 %)	SearchTags
0(0 %)	ControlCharacter

Fuente: Elaboración propia

Con los resultados de 3.6.4.6.14 y 3.6.4.6.15 se observó que la herramienta está usando más del 70% de los grupos, cumpliendo uno de los objetivos propuestos.

Los algoritmos usados en la prueba experimental seis tienen un alto porcentaje de cierto al ser capaces de clasificar los datos en (ALTO, MEDIO, BAJO, NO_ATAQUE). Por lo anterior se decidió usar los modelos de clasificación de la Prueba Experimental 6, para la fase de evaluación de los modelos.

3.6.5 Fase V Evaluación de los Modelos. En esta etapa se evaluó cada modelo seleccionado en la prueba experimental 6 para decidir cuál escoger. Para ello se usó los “Conjunto de datos de entrenamiento (XSS) 36-1”, “Conjunto de datos de entrenamiento (XSS) 36-2”, “Conjunto de datos de entrenamiento (XSS) 36-3” y “Conjunto de datos de entrenamiento (XSS) 36-4”. En cada elemento de los cuatro conjuntos de datos de entrenamiento se obtuvo el vector característico conformado por las doce características seleccionadas a partir de la prueba experimental 6 y estos fueron entregados al experto, quien realizó el proceso de clasificación (ALTO, MEDIO, BAJO, NO_ATAQUE).

Se evaluó cada una de los cuatro conjuntos de datos de entrenamiento con los modelos (Arboles de Decisión (Clasificador J48), Naive Bayes, Perceptron Multicapa), para así obtener la respuesta del modelo.

Finalmente se compara la respuesta obtenida por cada modelo con la respuesta esperada por el experto.

Los resultados del porcentaje de aciertos para cada modelo y “Conjunto de datos de entrenamiento 36-1, 36-2, 36-3 y 36-4” se pueden observar a través de la tabla que aparece a continuación.

Tabla 20**. Comparación del porcentaje de aciertos de los modelos construidos en la fase experimental 6**

Número Conjunto de datos de entrenamiento	Cantidad de elementos	% Aciertos J48	%Aciertos Naive Bayes	% Aciertos Perceptron Multicapa
Uno	9034	87%	79%	85%
Dos	9033	87%	79%	85%
Tres	9033	87%	78%	85%
Cuatro	9035	87%	78%	84%

Fuente: Elaboración propia

En la tabla 20, se evidencia el porcentaje de acierto obtenido de los modelos generados en la fase experimental seis, con los cuatros conjunto de datos de entrenamiento. Pero en este caso, la efectividad del Perceptron Multicapa no fue la más alta, como lo había hecho durante las seis pruebas experimentales.

En el Anexo 8, se encuentran los conjuntos de datos de entrenamientos con la calificación esperada y el resultado de evaluar cada Conjunto de datos de entrenamiento por los modelos con los algoritmos (J48, Naive Bayes, Perceptron Multicapa).

Se escoge el modelo de clasificación generado con el algoritmo J48 creado en la prueba experimental seis para hacer parte de la detección del control de seguridad, por haber sido el modelo que obtuvo el mayor porcentaje de aciertos en esta fase sobre las Técnicas de Machine Learning, Naive Bayes y Perceptron Multicapa. Con esta selección se da cumplimento a los objetivos fijados en el numeral 3.6.1.

3.7 IMPLEMENTACIÓN DEL CONTROL DE SEGURIDAD.

3.7.1 Metodología de desarrollo XP[51]. Es un sistema sencillo para la producción rápida, enfocada en proporcionar un sistema que cubra las necesidades inmediatas del cliente.

3.7.1.1 Ciclo de Vida de XP. Las iteraciones en XP son relativamente cortas, entre más rápido se entregue lo desarrollado al cliente, más retroalimentación se obtendrá. Se describen las fases que se usaron en la construcción del prototipo.

3.7.1.1.1 Fase de Exploración: Se exploró las tecnologías de desarrollo (JAVA, C# y Python) y librerías para la captura de tráfico, y se definió a grandes rasgos las características del prototipo.

3.7.1.1.2 Fase de Planteamiento: Se definió los requisitos a ser implementados por iteraciones, se crea la primera versión de la arquitectura del control de seguridad, lo que permitirá cumplir el desarrollo del prototipo.

3.7.1.1.3 Fase de producción: Se diseñó y ejecutó las pruebas unitarias de funcionamiento del sistema, mientras se incorporaban nuevas funcionalidades al sistema.

3.7.1.1.4 Fase de Muerte: En esta fase se evaluó el cumplimiento de los requisitos y de la arquitectura; adicionalmente se generó la documentación del sistema del prototipo.

3.7.1.2 Artefactos. Se consideran los siguientes artefactos mínimos para el desarrollo:

- Documento de especificación de requerimientos.
- Diagrama de Clases
- Prototipo del Software
- Documentación del código del prototipo

3.7.2 Aplicación de la Metodología. El proyecto consistió en construir un prototipo de un control de seguridad apoyado de una técnica de inteligencia artificial que, a partir del tráfico de red recibido en una red Ethernet, mida el riesgo según lo indica la metodología de valoración del riesgo de OWASP y tome decisiones en tiempo real para disminuir el nivel de riesgo.

3.7.2.1 Fase de Exploración de la metodología XP

❖ **Especificación de requerimientos.** Con el apoyo del equipo de trabajo del proyecto Control Inteligente para el servicio crítico de un sistema de información en línea enmarcado en un dominio de la ISO/IEC 27002:2013, se desarrolló un documento de requerimiento que detalla el comportamiento del prototipo (requerimientos funcionales) y describe los requerimientos no funcionales y restricciones del prototipo. En el Anexo 9 se incluye el documento de especificación de requerimientos que se usó para la construcción del prototipo.

3.7.2.2 Fase de Planeación. Esta fase describe los pasos a tener en cuenta para el desarrollo del prototipo y la definición de responsabilidades.

La tabla que aparece a continuación, representa el número de la iteración donde se realizaron las tareas de un módulo.

Tabla 21**. Iteraciones realizadas para la construcción del prototipo.**

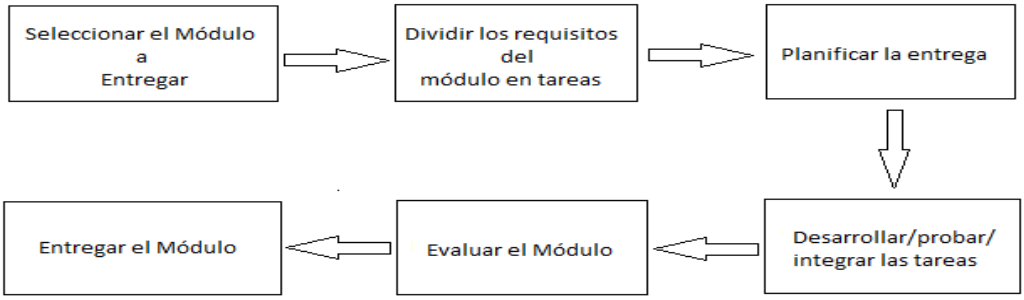
No. de iteración	Funcionalidad	Responsable
1	Módulo Capturador de Tráfico.	David Penagos
1	Módulo Analizador de Tráfico	David Penagos
2, 3, 4, 5	Módulo Gestor de Plugins	David Penagos, Jaime Jurado
2	Módulo Gestor-Detector	David Penagos
4	Módulo Gestor-Configuración	Jaime Jurado
4	Módulo Plugin A3	David Penagos, Jaime Jurado
5	Módulo Gestor-Planeador	Jaime Jurado
4	Módulo Evaluador de Riesgo	David Penagos, Jaime Jurado
2,3	Módulo Monitor	Desarrollador 1
4,5	Módulo Efector	Desarrollador 2

Fuente: Elaboración Propia

Debido a que los requerimientos del documento de requerimientos están bien detallados, se decidió obviar las historias de usuario y se toma cada ítem del documento como una serie de tareas.

La figura 12, describe el proceso que se realizó para entregar cada módulo.

Figura 12**. Ciclo de entrega de los módulos**

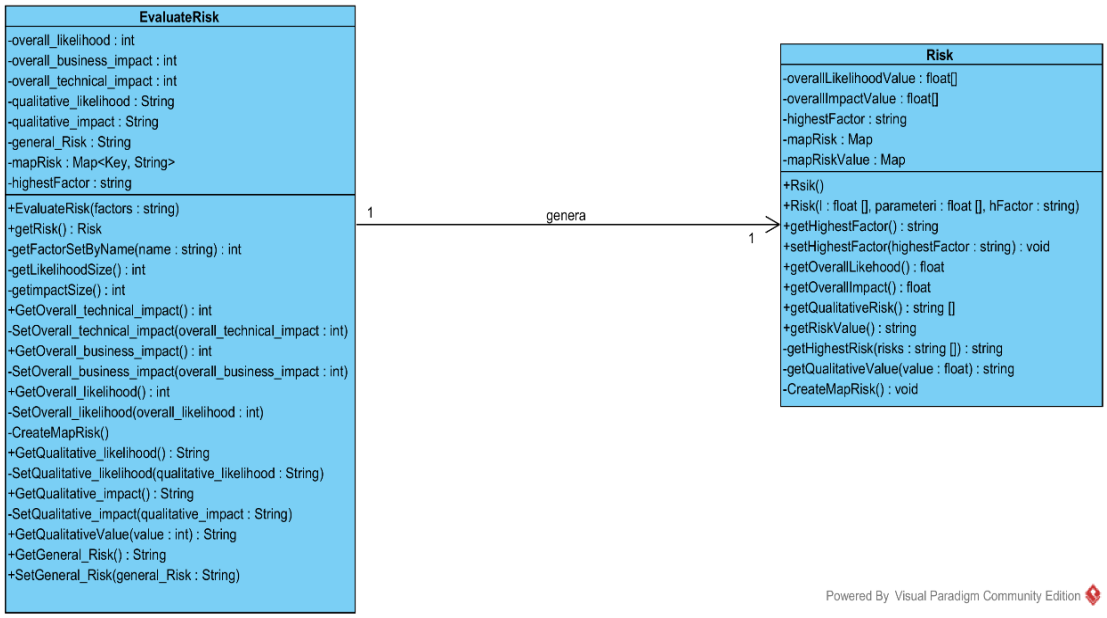


Fuente: Elaboración Propia.

3.7.2.3 Fase de Producción

- **Diagrama de clases**. En la herramienta Visual Paradigm y con el apoyo del libro de patrones de Gamma [41], se realizó el diseño de los diagramas de clases, en la figura 13 se puede observar el diagrama de clases del Evaluador de Tráfico. En el Anexo 10, se encuentran los diagramas de clases construidos para el prototipo.

Figura 13**. Diagrama de clases del Evaluador de Riesgos**



Fuente: Elaboración propia

3.7.2.4 Fase de Muerte. Esta fase determina la verificación de cumplimiento de los requisitos, lo que llevó a finalizar la construcción del prototipo, implantar el prototipo y generar la documentación. En el Anexo 11 se incluye la documentación del código del prototipo.

3.7.3 Herramientas y librerías de desarrollo empleadas. Mediante reuniones realizadas con el grupo del proyecto de FRIDA, se optó por JAVA como herramienta de desarrollo; como motor de base de datos se escogió MONGO, la cual es una base de datos documental. Adicionalmente, se describen cada una de las librerías que se usó en la construcción del prototipo.

- **JAVA.** Lenguaje de programación orientado a objetos y flexible que permite crear aplicaciones que pueden ser ejecutadas en cualquier sistema operativo que soporte la máquina virtual de JAVA. Se caracteriza por ser portable, de arquitectura neutral, robusto, seguro, con soporte para múltiples hilos y para desarrollo distribuido.

- **NetBeans.** Entorno de desarrollo de código abierto para Java, con soporte para desarrollo web.

- **Librería JNETPCAP.** Librería de código abierto que captura el tráfico de red por un dispositivo Ethernet y lo serializa como objetos. Disponible para Sistemas operativos (Linux y Windows) x86 y x64.

- **Librerías Antisamy y JSOUP.** Librería de código abierto que verifica si una cadena contiene sólo elementos permitidos definidos (Listas Blancas).

- **Librería Log4j.** Librería que se encarga de almacenar en archivos la depuración y los errores de la aplicación.

- **Librería WEKA.** Librería de código abierto que contiene algoritmos de Machine Learning, usada para crear el modelo del algoritmo J48.

- **Librería JAXB.** Librería de código abierto que se encarga de serializar XML a objetos y viceversa.

- **Librería JMS.** Librería de código abierto que se encarga de enviar y recibir objetos o texto por la red.

- **Mongo.** Sistema de base de datos de código abierto orientado a guardar estructura de documentos tipo JSON[35].

- **Quartz.** Librería de código abierto para Java encarga de ejecutar tareas de forma repetitiva en una fecha y hora específica.

3.8 EVALUACIÓN [52], [53]

En esta fase se desarrolló la fase de evaluación de acuerdo con la metodología de CRISP-DM. Para ello, se realizaron pruebas unitarias y de integración al prototipo, con el fin de descubrir defectos en los componentes individuales y comprobar que el sistema funciona como se ha especificado.

Los diseños de casos de prueba se realizaron con el enfoque de caja negra con el fin de demostrar que las funciones del prototipo son operativas, que las entradas se aceptan correctamente y producen una salida adecuada.

De los tipos de métodos de caja negra se usó el de partición equivalente, mediante el cual se busca un conjunto de estados válidos o inválidos para las entradas y análisis de valores límites. Este método se usó para seleccionar valores límites de las entradas y salidas, con el fin de aumentar la eficiencia de la prueba.

3.8.1 Pruebas Unitarias. Este tipo de prueba fue pensada para verificar el funcionamiento de aislado del software, se centra en verificar la lógica y el funcionamiento interno. Se diseñaron pruebas unitarias para los siguientes módulos:

Analizador de Tráfico, Evaluador de Riesgo, Efector de Decisiones y Bitácora.

Para verificar el funcionamiento del Gestor se realizaron las pruebas unitarias sobre los componentes que lo conforman (Detector, Configuración y Planeador).

Para conocer más sobre los casos de prueba ver Anexo 12, Pruebas Unitarias.

[35] JSON (Javascript Object Notation): Formato ligero de intercambio de datos que no requiere uso de XML. (http://json.org/).

3.8.2 Pruebas de Integración. Estas pruebas fueron pensadas con el fin de comprobar que los componentes individuales (módulos) realmente funcionan juntos. Se diseñaron pruebas de integración para los siguientes módulos:

Analizador de tráfico con el componente Detector del Gestor, Evaluador de riesgos con el componente de configuración del Gestor, el Gestor con Bitácora.

Sólo se integraban los módulos que pasaban con éxito las pruebas unitarias. Para conocer en más detalle los casos y datos de pruebas usados ver el Anexo 12, Pruebas de Integración.

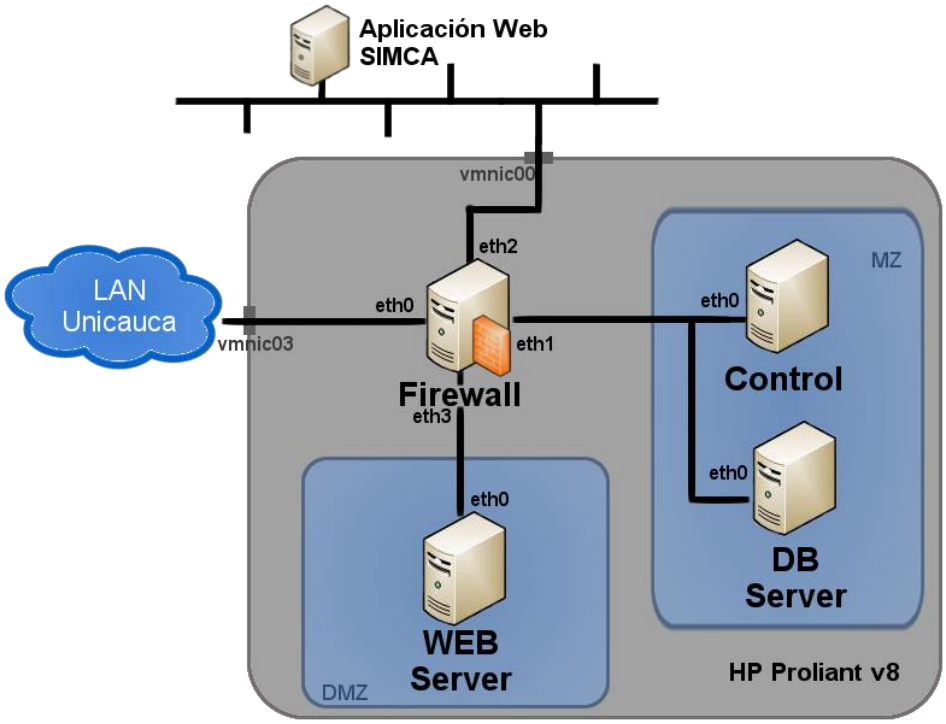
4. IMPLANTACIÓN DEL CONTROL

En esta sección se desarrolla la última fase de la metodología de CRISP-DM. En esta fase se desarrolla todo lo relacionado con configuración de red y la instalación del control. Se muestra de forma general la arquitectura diseñada para el proyecto "Control Inteligente para el servicio crítico de un sistema de información en línea enmarcado en un dominio de la ISO/IEC 27002", financiado por FRIDA.

4.1 ARQUITECTURA DE RED

Para que el control funcione adecuadamente, debe estar funcionando en una arquitectura en donde el tráfico dirigido a la Aplicación Web (SIMCA) transite por el Control y así realizar un análisis para posteriormente tomar una decisión si es necesaria, tal como lo muestra la figura 14. Cabe aclarar que el servidor Control es el que hospeda al control de seguridad (prototipo).

Figura 14**. Arquitectura de red primera versión**



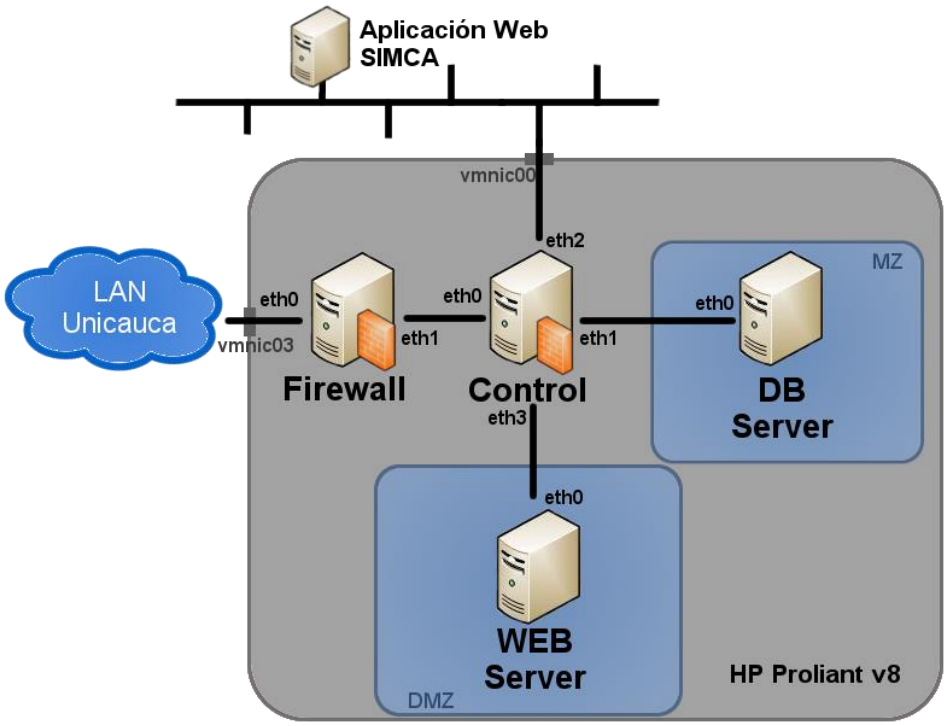
Elaboración Propia

Para lograr esto se configuró el servidor de Firewall con algunas reglas para redirigir el tráfico al control. De esta manera los paquetes entran a la interfaz eth0 del servidor Firewall, la cual está en puente con la interfaz física vmnic03. Luego se redirige hacia el Control usando la interfaz eth1 para realizar el análisis y las acciones pertinentes, se devuelve el paquete al Firewall para que este lo redirija hacia la Aplicación Web por la interfaz eth2 (que está en puente con la interfaz física vmnic00) finalizando el recorrido.

Se encontró un problema con este diseño, las solicitudes que debían dirigirse hacia la Aplicación Web, se quedaron en un ciclo, debido a que el paquete siempre regresaba al Control, por lo que la Aplicación Web nunca recibió la solicitud y por lo tanto no hubo la respuesta esperada.

Por medio de asesoramiento de la División de las TIC de la Universidad del Cauca, se cambió la arquitectura de red (ver Figura 15). Ahora el servidor Control tiene un corta fuegos, para regular el tráfico hacia la Aplicación Web y demás servidores, y el servidor Firewall gestiona el acceso hacia el Control.

Figura 15**. Arquitectura de red segunda versión**



Fuente: Elaboración Propia

Las solicitudes las recibe la interfaz eth0 del servidor Firewall, al igual que el diseño anterior, está en puente con la interfaz física vmnic03. El Firewall por medio de reglas, redirige los paquetes usando la interfaz eth1 para que pasen por el servidor Control. El Control recibe los paquetes por la interfaz eth0, el prototipo analiza y toma las decisiones que considera necesarias y el servidor redirige los paquetes hacia la Aplicación Web usando la interfaz eth2, la cual está en puente con vmnic03. De esta manera la aplicación recibe solicitudes previamente analizadas por el prototipo.

4.1.1 Componentes de la topología de red. Se usó un servidor HP Proliant ML310e Gen8 v2 con un procesador Intel Xeon E3-1240v3 Quad Core de 3.40GHz y 8MB de memoria cache, 2 módulos DDR3 de 8GB de memoria RAM de 1.600 MHz cada uno y un disco duro de 2 TB de 7.200 RPM, capaz de soportar la virtualización de al menos cuatro sistemas operativos Linux sin interfaz gráfica para cumplir con las funcionalidades de los componentes en el segundo diseño de red que se tomó como definitivo. Se instaló en cada una de las máquinas virtualizadas el Debían 7.8.

4.1.1.1 Servidor Firewall. A este servidor se le asignó los siguientes recursos de hardware:

CPU: 2 núcleos virtuales
Memoria RAM: 3 GB
Disco Duro: 300 GB
Tarjetas de Red: 2 adaptadores de red virtuales VMNET3

Se configuró la interfaz eth0 en puente con la interfaz física vmnic03 del servidor HP Proliant, con una dirección IP fija de la red de la organización, y la segunda interfaz eth1 se configuró como una red virtual 192.168.2.0/24.

4.1.1.2 Servidor Control**.** Este servidor fue configurado con los siguientes recursos de hardware:

CPU: 8 núcleos virtuales
Memoria RAM: 8 GB
Disco Duro: 500 GB
Tarjetas de Red: 4 interfaces de red virtuales VMNET3

Se configuró la interfaz eth2 en puente con la interfaz física vmnic00 del servidor HP Proliant y así pueda dirigir el tráfico para ser analizado por el prototipo, este, a su vez, redirige hacia la Aplicación Web.

La interfaz eth0 se configura con la IP 192.168.121.2.2/24 de la red interna 192.168.2.0/24, de esta manera hay una conexión con el servidor Firewall, el cual re direcciona las solicitudes a la Aplicación Web.

La interfaz eth1 tiene asignada la IP 192.168.6.1/24 de la red desmilitarizada (DMZ) 192.168.6.0/24 por la cual le da acceso a la aplicación de administración del prototipo; por último, la interfaz eth3 tiene la IP 192.168.9.1/24 de la red militarizada (MZ) 192.168.9.0/24 la cual hospeda al servidor de base de datos DB Server.

4.1.1.3 Servidor de base de datos DB Server. A este servidor se le asignó los siguientes recursos de hardware:

CPU: 2 núcleos virtuales
Memoria RAM: 2 GB
Disco Duro: 300 GB
Tarjetas de Red: Interfaz de red VMNET3

Para este servidor sólo es necesaria una interfaz, eth0, la cual se configuró con la IP 192.168.9.253/24 de la red MZ.

4.1.1.4 Servidor para la administración del control Web Server. Este servidor contó con la siguiente asignación de recursos de hardware:

CPU: 2 núcleos virtuales

Memoria RAM: 3 GB
Disco Duro: 300 GB
Tarjetas de Red: Interfaz de red VMNET3

Se configura la interfaz eth0 de este servidor con la IP 192.168.6.254/24 de la red DMZ.

4.2 CONFIGURACIÓN DE LOS SERVIDORES

4.2.1 Servidor Firewall. Para la configuración de este servidor se utilizó el sistema de corta fuegos (por su nombre en inglés - Firewall -) vinculado al kernel del sistema operativo. Este es el encargado de dirigir las solicitudes permitidas hacia el Sistema operativo donde se encuentra instalado el control de seguridad.

El control de seguridad debe ser capaz de configurar el firewall, para ello es necesario tener instalado JDK Java versión 8.020 o superior y realizar la siguiente configuración.

Los elementos a continuación mencionados se encuentran en el Anexo 11. Firewall Principal.

- Copiar el archivo Firewall.jar a "/etc/"
- Copiar el archivo firewall_security a la ubicación "/etc/init.d/"
- Dar permisos de lectura y escritura al archivo firewall_security con el siguiente comando: "sudo chmod +x /etc/inid.d/firewall_security"
- Iniciar la aplicación que permite configurar el firewall como un servicio se usa: "sudo /etc/init.d/firewall_security start", el usuario requiere tener permisos para iniciar un servicio

4.2.2 Servidor control. Este servidor es el encargado de contener el Control de Seguridad y un Firewall de segundo nivel, y para ello es necesario tener instaladas las siguientes librerías:

- apt-get install build-essential.
- apt-get install flex.
- apt-get install byacc flex.
- tcpdump versión 1.5.3 o superior.
- JDK Java versión 8.020 o superior.

Se detalla cómo deben ser instalados el control de seguridad y el firewall de segundo nivel:

4.2.2.1 Instalación del Control de Seguridad. Se detalla a continuación cómo se debe ubicar e iniciar el control de seguridad. Los elementos desarrollados en este trabajo se encuentran en el Anexo 13. Control de Seguridad.

- Poner el archivo SecurityControlOWASP.jar, dentro de la carpeta "/usr/SecurityControl".

- Copiar el archivo securitycontrol a la siguiente ubicación: "/etc/init.d/"

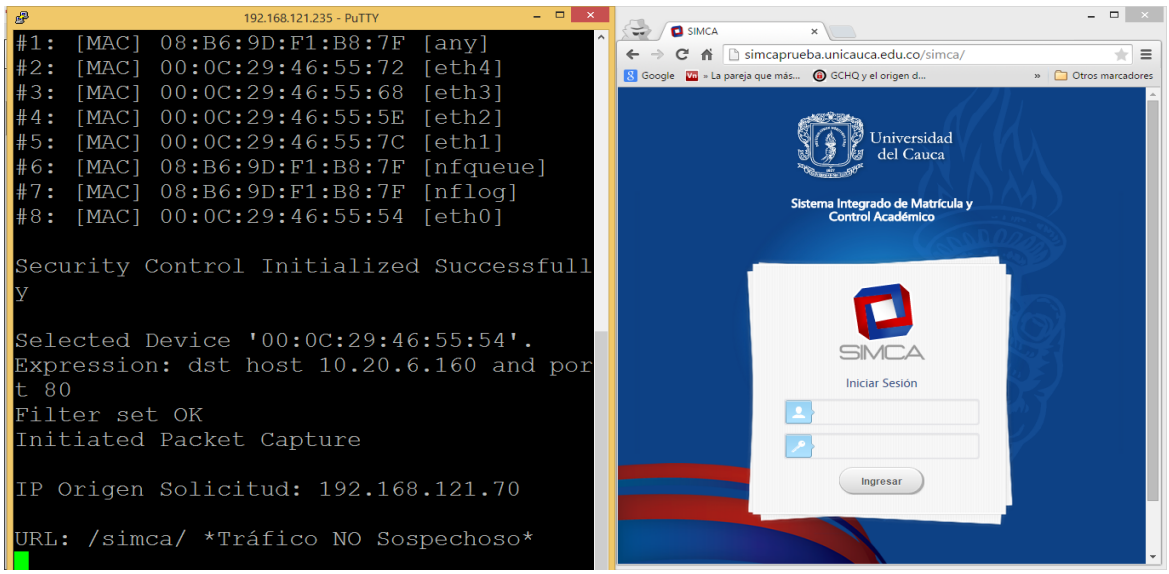
- Dar permisos al archivo securitycontrol con el siguiente comando: "sudo chmod +x /etc/init.d/"

- Modificar el archivo "/etc/rc.local" y agregar la siguiente línea: "/etc/init.d/securitycontrol start"

Para iniciar el control de seguridad como servicio se usa el siguiente comando: "sudo service securitycontrol start", el usuario requiere tener permisos para iniciar un servicio.

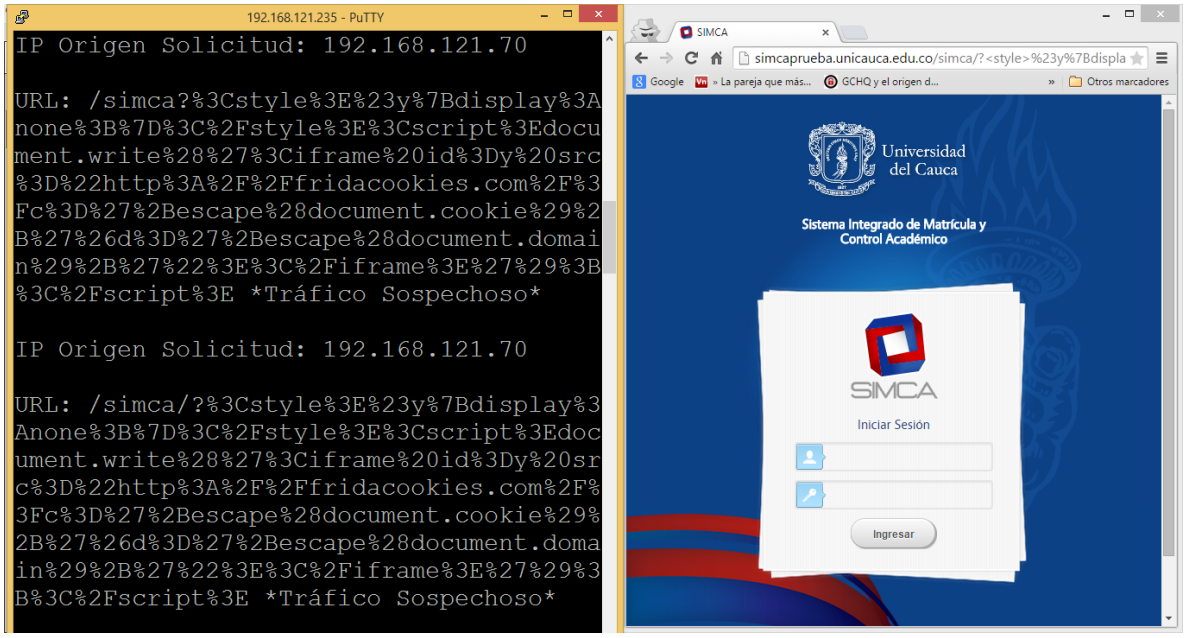
En las figuras 16 y 17 se presentan algunas capturas de pantalla del control de seguridad en funcionamiento, para observar más captura de pantalla del funcionamiento del control, ver el Anexo 14.

Figura 16. **Primera captura de pantalla del funcionamiento del control de seguridad.**



Fuente: Elaboración Propia

Figura 17 **Segunda captura de pantalla del funcionamiento del control de seguridad.**



Fuente: Elaboración Propia

4.2.2.2 Instalación del Firewall de Segundo Nivel. El procedimiento para la instalación del firewall de segundo nivel es el que se detalla a continuación.

- Copiar el archivo firewallControl a la ubicación “/etc/init.d/”

- Dar permisos de lectura y escritura al archivo firewall_security con el siguiente comando: “sudo chmod +x /etc/inid.d/firewallControl”

- Iniciar el firewall de segundo nivel como servicio se usa lo siguiente: “sudo /etc/init.d/ firewallControl start”, el usuario requiere tener permisos para iniciar un servicio

4.2.3 Servidor de Base de datos. Este servidor contiene dos bases de datos (una relacional y la otra documental), la base de datos relacional fue usada para almacenar los datos de los usuarios que tienen permitido usar la aplicación web del control de seguridad. La base de datos documental es usada para almacenar la información del control de seguridad.

4.2.3.1 Instalación de MongoDB. Para la instalación se debe seguir los siguientes pasos:

Paso 1: Primero importar la llave pública usada por el gestor de paquetes del sistema, mediante el siguiente comando:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
```

Paso 2: Crear los siguientes archivos para una correcta instalación:

```
echo "deb http://repo.mongodb.org/apt/debian "$(lsb_release -sc)"/mongodb- org/3.0 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.0.list
```

Paso 3: Actualizar la base de datos local de paquetes:

```
sudo apt-get update
```

Paso 4: Instalar los paquetes de MongoDB:

```
sudo apt-get install -y mongodb-org
```

Paso 5: Iniciar el servicio de MongoDB:

```
sudo service mongod start
```

- **Configuración de MongoDB**. Seguir los siguientes pasos para configurar MongoDB:

Paso 1: Se creó el usuario root y se le asignó privilegios con el siguiente comando:

```
use admin
db.createUser( { user: “admin”
        pwd: "admin",
        roles: [ "userAdminAnyDatabase",
                "dbAdminAnyDatabase",
                "readWriteAnyDatabase"

] } )
```

Paso 2: Luego de crear el usuario root, se usó lo siguiente para conectarse como root:

```
mongo --port 27017 -u admin -p admin --authenticationDatabase admin
```

Paso 3: Conectado como root se creó una base de datos con el nombre records_prueba usando lo siguiente:

```
use records_prueba
```

Paso 4: Se creó un usuario para que gestione la base de datos creada el paso 3.

```
db.createUser(
{
        user: "usrfrida",
        pwd: "frida2015",
        roles: [
        { role: "readWrite", db: " records_prueba " }
        ]
}
)
```

Paso 5: Finalmente se habilitó MongoDB para aceptar conexiones remotas, se elimina en el archivo **"mongodb.conf"** la siguiente línea:

```
bind_ip 127.0.0.1
```

4.2.3.2 Instalación de MySQL. Para la instalación de MySQL se debe seguir los siguientes pasos:

- apt-get install mysql-server
- apt-get install mysql-client

- **Configuración de MySQL**. Para la configuración se debe seguir los siguientes pasos:

Paso 1: Se debe conectar como usuario root, debido a que éste cuenta con todos los privilegios, usando la siguiente línea:

```
mysql –u root –p
```

Paso 2: Se debe crear la base de datos que servirá para gestionar la información de los usuarios de la aplicación web correspondiente al monitor de bitácora. Lo anterior se realiza con el siguiente script:

```
CREATE DATABASE dbautenticacion;
USE dbautenticacion;
CREATE TABLE IF NOT EXISTS `groups` (
```

```
`groupid` varchar(20) NOT NULL,
`description` varchar(255) DEFAULT NULL,
PRIMARY KEY (`groupid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `users` (
`userid` varchar(255) NOT NULL,
`password` varchar(255) NOT NULL,
`email` varchar(255) NOT NULL,
`nombres` varchar(255) NOT NULL,
PRIMARY KEY (`userid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `users_groups` (
`groupid` varchar(20) NOT NULL,
`userid` varchar(255) NOT NULL,
PRIMARY KEY (`groupid`,`userid`),
KEY `fk_users_groups_userid` (`userid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `users_groups`
 ADD CONSTRAINT `fk_users_groups_groupid` FOREIGN KEY
(`groupid`) REFERENCES
`groups` (`groupid`),

 ADD CONSTRAINT `fk_users_groups_userid` FOREIGN KEY (`userid`)
REFERENCES `users`
(`userid`);
```

Paso 3: Luego se debe crear un usuario al que se le asignan privilegios de consulta, inserción, actualización y eliminación de registros en la base de datos creada en el paso anterior (base de datos dbautenticacion):

```
CREATE USER 'usrfrida'@'localhost' IDENTIFIED BY 'frida2015';

GRANT SELECT,INSERT,UPDATE,DELETE ON `dbautenticacion`.* TO
'usrfrida'@'%' IDENTIFIED BY 'frida2015';
```

Paso 4: Por último se habilitó el servidor MySQL para que acepte conexiones remotas. Lo anterior se realizó borrando la siguiente línea en el archivo “my.cnf” ubicado en “/etc/mysql/”:

```
bind-address = 127.0.0.1
```

4.2.4 Servidor Web. Este servidor contiene una aplicación web, encargada de realizar la configuración inicial del control de seguridad y permitir al usuario observar, si existe un ataque XSS, qué nivel de riesgo tiene ese ataque y qué decisión se ejecutó para disminuir el riesgo; también le permite al usuario conocer si ha ocurrido o no un error en el control de seguridad.

Se requiere tener instalado previamente JDK Java Versión 8**.**20 o posterior.

4.2.4.1 Instalación del Servidor Glassfish 4

Paso 1: Descargar los archivos de instalación de GlassFish 4.1 para Debian, a continuación se presenta como se realizó el proceso de instalación desde la terminal del sistema operativo:

- cd /opt
- wget http://dlc.sun.com.edgesuite.net/glassfish/4.1/release/glassfish-4.1.zip
- unzip glassfish-4.1.zip
- cd /opt/glassfish4/glassfish/bin
- ./asadmin start-domain
- ./asadmin set --user admin server.jms-service.jms-host.default_JMS_host.admin-password=admin
- asadmin> change-admin-password --user admin
- Enter admin password> [VACIO]
- Enter new admin password> NuevoPassword
- Enter new admin password again> NuevoPassword

Paso 2: Se configuró la seguridad del panel de administración de GlassFish con lo siguiente:

- cd /opt/glassfish4/glassfish/bin
- ./asadmin --host localhost --port 4848 enable-secure-admin

4.2.4.2 Configuración del Sevidor GlassFish

Paso 1: Con GlassFish 4.1 instalado, se debe crear el recurso para la comunicación con el control de seguridad (Una Cola de Mensajes). Para tal fin se debe usar los siguientes comandos:

- cd /opt/glassfish4/glassfish/bin
- ./asadmin
- asadmin> create-jms-resource --restype javax.jms.Queue TestQueue

Paso 2: Se reinicia el dominio de MongoDB con el siguiente comando:

- ./asadmin restart-domain

Paso 3: Se debe crear un “JDBC Connector Pool” en el panel de administración con las siguientes propiedades:

- Name: MySQLPool
- Resource Type: javax.sql.DataSource
- Database Vendor: MySQL
- serverName 192.168.121.55
- port: 3306
- databaseName: Dbautenticacion
- user: usrfrida
- password: Frida2015

Paso 4: Se debe crear un DataSource con nombre “jdbc/mysql” y asociarlo con el pool creado anteriormente.

Paso 5: Crear un JDBCRealm, con el siguiente nombre “jdbcrealm” y las siguientes propiedades:

- datasource-jndi jdbc/mysql
- user-table users
- user-name-column userid
- password-name-column password
- group-table users_groups
- group-name-column groupid
- jaas-context jdbcRealm
- digest-algorithm SHA-256

4.2.4.3 Despliegue de la aplicación. Para realizar el despliegue de la aplicación se deben seguir los pasos que se enumeran continuación:

Paso 1: Se accedió al panel de control por medio de un navegador y la siguiente URL: 192.168.121.55:4848.

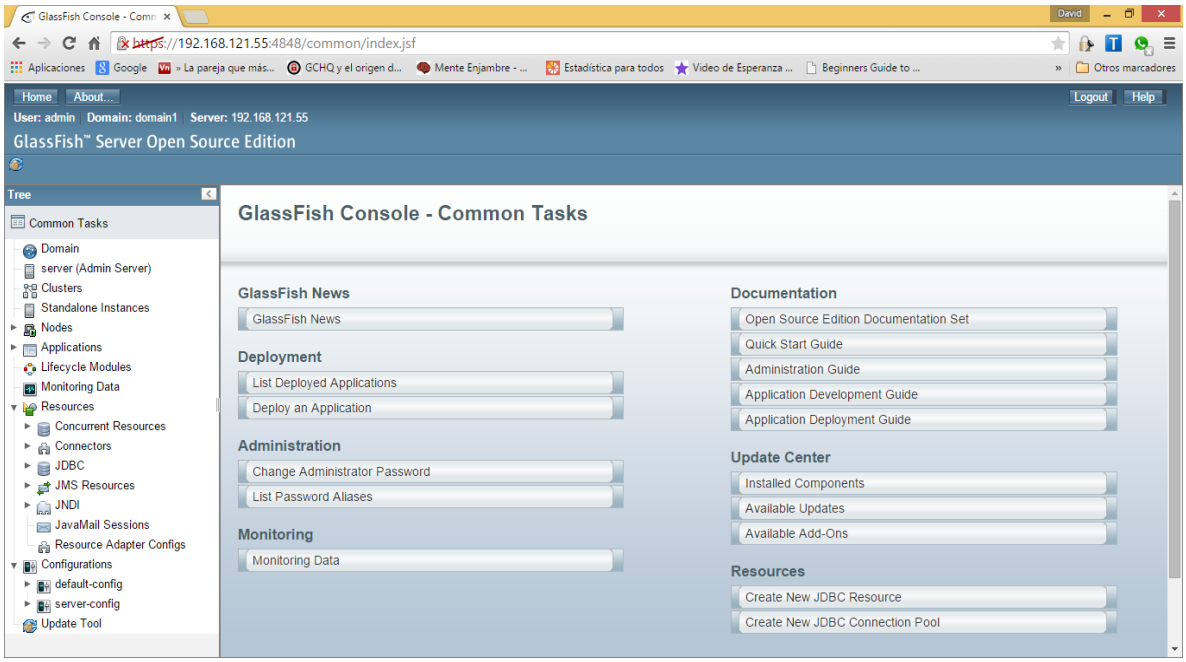
Se ingresó con los siguientes datos:

- usuario: admin
- password: admin

Paso 2: Se seleccionó la pestaña aplicaciones y se escogió la opción deploy, este cargo un asistente de configuración en el que se seleccionó el archivo "*.war"* ubicado en el Anexo 11. Administrador del control, una vez finalizó el asistente la aplicación se encontraba desplegada.

En la figura se presenta un ejemplo la interfaz que permite configurar el despliegue del archivo "*.war"*

Figura 18. **Interfaz de configurador del servidor Glassfish.**



Fuente: Servidor Glassfish.

CONCLUSIONES

El uso de estándares como la ISO 27002 y OWASP Top 10 - 2013, permiten a las organizaciones optimizar la seguridad en las aplicaciones web. De la relación realizada entre estos estándares se escogió el control “12.6.1. Gestión de las vulnerabilidades técnicas”, por ser uno de los controles de la ISO 27002 que se relaciona con todos los riesgos de OWASP Top 10 - 2013 y que el control puede, perfectamente, ser aplicado a una aplicación web. Adicionalmente, para medir el riesgo de una aplicación se usó la valoración de riesgos propuesta por OWASP y el tráfico de red. Esto permitió generar un documento que presenta una visión general de los aspectos que deben ser tenidos en cuenta en cada factor de la valoración de riesgo.

Para construir el control de seguridad que optimice la seguridad de las aplicaciones web, se adaptó la propuesta de V. Teresius [40]. A ésta, se le incluyó el patrón Plug-In dentro del diseño de la arquitectura del prototipo con el fin de tercerizar el Plug-In de cada A. Esto permitió centrarse en la construcción de un sistema base, y permitir que este crezca en funcionalidad a medida que se caractericen y agreguen más Plug-Ins.

Se considera que los algoritmos J48, Naive Bayes y Perceptron Multicapa son adecuados para determinar un XSS, debido a que estos presentaron un porcentaje de instancias correctamente clasificadas, superior al 89%.

Mediante una técnica de inteligencia artificial arboles de decisión con el clasificador J48 incorporada al prototipo se optimizó la detección de tráfico para una aplicación web. Esto permitió obtener un alto porcentaje de acierto – 87% – con varios conjuntos de datos de entrenamiento.

Definir e implantar una arquitectura de red, permitió tener un ambiente real controlado para la realización de las pruebas de integración de los módulos implementados, estas pruebas fueron superadas en su totalidad con éxito, lo que demuestra que los módulos implementados se están comunicando.

Con el desarrollo de las pruebas unitarias y de integración se pudo evidenciar la viabilidad de la implementación e implantación del prototipo.

TRABAJOS A FUTURO

En este trabajo sólo se desarrolló el proceso de entendimiento del A3, uno de los diez A del Top 10 de OWASP, lo que deja abierto la posibilidad de realizar trabajos con los otros nueve A del Top 10 - 2013 de OWASP.

Se recomienda que cada A, tenga su propia técnica de inteligencia artificial presente en su Plug-In.

Una vez se tenga caracterizados los otros A del Top 10 - 2013 de OWASP, se recomienda se use una técnica de inteligencia artificial, que incluya las características de todos los ataques para construir un único modelo para que determine si un tráfico es sospechoso.

Se sugiere escalar a una ontología para la toma de decisiones que esté basada en el conocimientos de expertos en seguridad, y relacione las recomendaciones del listado de Top 10 - 2013 de OWASP y la ISO/IEC 27002:2013

LECCIONES APRENDIDAS

La metodología de las elipses permitió seleccionar el proceso Control Académico como el más crítico en la organización, por la cantidad de dependencias relacionadas con él, y la aplicación web (SIMCA) como el activo de información de más importancia para la organización dado la gran participación con el proceso seleccionado.

Para analizar el tráfico de red, se planteó una topología de red con un solo firewall para controlar las solicitudes hacia la aplicación web. Este diseño no funcionó debido a que las solicitudes no llegaban a su destino. En consecuencia, hubo necesidad de diseñar una nueva topología con dos firewalls para que el segundo se encargue de redireccionar las solicitudes a la aplicación

Por otra parte, la versión actual de la librería JNETPCAP versión 1.4.r1425 para Linux no reconoce las tarjetas de red virtuales, por lo cual fue necesario descargar el código fuente, corregir el error y volver a compilar la librería para las arquitecturas x86 y x64 de Linux.

El presente trabajo muestra que las Técnicas de Machine Learning se pueden utilizar no sólo para determinar si un ataque es o no XSS, sino también para inferir su nivel de complejidad.

REFERENCIAS BIBLIOGRAFICAS

[1] A. G. Alexander, *Diseño De Un Sistema De Gestión De Seguridad De Información*. Bogotá: Alfa Omega, pp. 39-47, 2007.

[2] C. de P. Heredero, J. J. L. H. Agius, S. M. R. Romero, and S. M. Salgado, *Organización y transformación de los sistemas de información en la empresa*. ESIC Editorial, pp. 20-33, 2012.

[3] P. A. López, *Seguridad informática*. Editex, pp. 8-23, 2010.

[4] J. Areitio Bertolín, *Seguridad de la información. Redes, informática y sistemas de información*. Editorial Paraninfo, pp. 21-25, 2008.

[5] The OWASP Foundation, "The Open Web Application Security Project." [Online]. Available: https://goo.gl/hCYz7C.

[6] Imperva, "Imperva Web Application Attack Report." [Online]. Available: http://goo.gl/DTfp6k, 2013.

[7] ISO/IEC 27002:2013, "Information technology - Security techniques - Code of practice for information security controls." 2013.

[8] Cano, Ph.D. Jeimy J., "[ISACA] La Gerencia de la Seguridad de la Información: Evolución y Retos Emergentes".

[9] P. J. Gonzalez, "¿Seguridad Informática o Seguridad de la Información?".

[10] M. del P. Ramos and A. G. Hurtado, *Seguridad Informática ED.11 Paraninfo*. Editorial Paraninfo, pp. 21, 2011.

[11] S. L. Mora, *Programación de aplicaciones web: Historia, principios básicos y clientes web*. Editorial Club Universitario, pp. 47-59, 2002.

[12] Universidad Nacional Autónoma de México, "Punto Seguridad Defensa Digital, Ataques Web".

[13] L. K. Shar and H. B. K. Tan, "Automated removal of cross site scripting vulnerabilities in web applications," *Inf. Softw. Technol.*, vol. 54, no. 5, pp. 467–478, May 2012.

[14] L. K. Shar and H. B. K. Tan, "Predicting SQL injection and cross site scripting vulnerabilities through mining input sanitization patterns," *Inf. Softw. Technol.*, vol. 55, no. 10, pp. 1767–1780, Oct. 2013.

[15] A. Austin, C. Holmgreen, and L. Williams, "A comparison of the efficiency and effectiveness of vulnerability discovery techniques," *Inf. Softw. Technol.*, vol. 55, no. 7, pp. 1279–1288, Jul. 2013.

[16] H. Shahriar and M. Zulkernine, "Taxonomy and classification of automatic monitoring of program security vulnerability exploitations," *J. Syst. Softw.*, vol. 84, no. 2, pp. 250–269, Feb. 2011.

[17] C. Raspotnig and A. Opdahl, "Comparing risk identification techniques for safety and security requirements," *J. Syst. Softw.*, vol. 86, no. 4, pp. 1124–1151, Apr. 2013.

[18] "ISO 27000." [Online]. Available: http://goo.gl/0n2pVn.

[19] ISO/IEC 27001:2005, "Tecnología de la Información - Técnicas de seguridad - Sistemas de gestión de la seguridad de la información." 2005.

[20] ISO/IEC 27005:2011, “Information technology - Security techniques - Information security risk management.” 2011.
[21] The OWASP Foundation, “OWASP Testing Guide v4.” [Online]. Available: http://goo.gl/r9ccBe, 2015.
[22] The OWASP Foundation, “OWASP Top 10 2013.” [Online]. Available: http://goo.gl/gjQNzs, 2013.
[23] OWASP, “Cross-site Scripting (XSS).” [Online]. Available: http://goo.gl/jXnKzO, 22-Apr-2014.
[24] WebSecurityDev, “Vulnerabilidad Cross-site scripting y sus Clases.” [Online]. Available: http://goo.gl/cxNgP6.
[25] A. Aguilar Domínguez, “¿Qué es y cómo opera un ataque de Cross-Site Scripting (XSS)?” [Online]. Available: http://goo.gl/wvYXYN.
[26] “XSS Hacking Tutorial.” [Online]. Available: http://goo.gl/yLMpdV.
[27] N. Feng, H. J. Wang, and M. Li, “A security risk analysis model for information systems: Causal relationships of risk factors and vulnerability propagation analysis,” *Inf. Sci.*, vol. 256, pp. 57–73, Jan. 2014.
[28] C.-C. Huang, F.-Y. Lin, F. Y.-S. Lin, and Y. S. Sun, “A novel approach to evaluate software vulnerability prioritization,” *J. Syst. Softw.*, vol. 86, no. 11, pp. 2822–2840, Nov. 2013.
[29] K. Kozhakhmet, G. Bortsova, A. Inoue, and L. Atymtayeva, “Expert System for Security Audit Using Fuzzy Logic,” in *Midwest Artificial Intelligence and Cognitive Science Conference*, 2012, p. 146.
[30] K. T. Kozhakhmet, G. K. Bortsova, and L. B. Atymtayeva, “Some Issues of Development of Intelligent System for Information Security Auditing,” presented at the World Congress on Engineering 2012 Vol II, London, U.K, 2012.
[31] M. Wang, H. Wang, D. Xu, K. K. Wan, and D. Vogel, “A web-service agent-based decision support system for securities exception management,” *Expert Syst. Appl.*, vol. 27, no. 3, pp. 439–450, Oct. 2004.
[32] A. E. Nunan, E. Souto, E. M. dos Santos, and E. Feitosa, “Automatic classification of cross-site scripting in web pages using document-based and URL-based features,” 2012, pp. 000702–000707.
[33] S. Krishnaveni and K. Sathiyakumari, *Multiclass Classification of XSS Web Page Attack using Machine Learning Techniques S.Krishnaveni*. .
[34] D. R. Mahapatra, R. Saini, and N. Saini, “A Pattern Based Approach to Secure Web Applications from XSS Attacks,” *J. Comput. Technol. Electron. Eng.*, vol. 2.
[35] B. A. Vishnu and K. P. Jevitha, “Prediction of Cross-Site Scripting Attack Using Machine Learning Algorithms,” in *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing*, New York, NY, USA, 2014, pp. 55:1–55:5.
[36] P. D. D. Patil and S. Sarwade, “Document-based and URL-based Features for Automatic Classification of Cross-Site Scripting in Web Pages,” *IOSR J. Eng.*, vol. 3, pp. 11–18.
[37] “Metodología CRISP-DM para minería de datos.” [Online]. Available: http://goo.gl/ONnyxB.

[38] *NTC-ISO-IEC 27001. Tecnología de la información. Técnicas de seguridad. Sistemas de gestión de la seguridad de la información.*, 2013th–12th–20th ed. ICONTEC, 2013.
[39] Universidad del Cauca, “Informe Final Acreditación Institucional.” [Online]. Available: http://goo.gl/jiorIY, 2012.
[40] V. Teresius, “Reference architecture of intelligent system,” *Electron. Electr. Eng.*, vol. 63, no. 7, pp. 53–56, 2005.
[41] R. H. Gamma, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.
[42] K. Marquardt, “Patterns for Plug-Ins,” in *Proceedings of the 4th European Conference on Pattern Languages of Programms (EuroPLoP ’1999), Irsee, Germany, July 7-11, 1999*, 1999, pp. 203–232.
[43] D.-A. Manolescu, M. Voelter, and J. Noble, *Pattern Languages of Program Design 5*. Addison-Wesley Professional, 2006.
[44] A. Rodríguez Romero, “Ataques XSS en Aplicaciones Web.” [Online]. Available: http://goo.gl/sRPEM4.
[45] “OWASP Xenotix XSS Exploit Framework.” [Online]. Available: http://goo.gl/tGlIRp.
[46] The OWASP Foundation, “XSS Filter Evasion Cheat Sheet.” [Online]. Available: http://goo.gl/JPr31g, 2015.
[47] L. G. del Moral, *Curso de Ciberseguridad y Hacking Ético 2013*. Punto Rojo Libros, 2014.
[48] S. Karsoliya, “Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture,” *Int. J. Eng. Trends Technol.*, vol. 3, no. 6, pp. 713–717, 2012.
[49] R. Aler, “Tutorial Weka 3.6.0.” [Online]. Available: http://goo.gl/vWBCEQ.
[50] “Unicode,” *Mozilla Developer Network*. [Online]. Available: http://goo.gl/9OLCMM. [Accessed: 05-Mar-2015].
[51] “Extreme Programming XP.” Available: http://www.extremeprogramming.org, 03-Mar-2015.
[52] I. Sommerville, *Ingeniería del software*. Pearson Educación, pp. 503-513, 2005.
[53] R. S. Pressman, *Ingeniería del software: un enfoque práctico*. Mc Graw Hill, 2005.

ANEXOS

Ver documento adjunto.