Control Inteligente para el Servicio Crítico de un Sistema de Información en Línea Enmarcado en un Dominio de la ISO/IEC 27002

David Felipe Penagos Mosquera[a], Jaime Alberto Jurado Narvaéz[b], Siler Amador Donado[c], Ember Ubeimar Martinez Flor[d], Carlos Alberto Ardila Albarracín[e], Sara Donelly Garcés[f] ,

[a,b,c,d,e,f] Universidad del Cauca
Grupo de Tecnologías de la Información
Popayán, Cauca, Colombia
dfpenagos@unicauca.edu.co, jaimejn@unicauca.edu.co, samador@unicauca.edu.co, eumartinez@unicauca.edu.co, cardila@unicauca.edu.co, sgarces@unicauca.edu.co

Resumen. Se propone los pasos para la selección de un proceso crítico en una organización, la arquitectura para control inteligente enmarcado en la ISO/IEC 27002 que integre un mecanismo de medición de riesgos basado en OWASP, la selección de una técnica de inteligencia artificial que permita integrarse a la arquitectura propuesta y una arquitectura de red que facilite al control inteligente monitorear un servicio crítico de un sistema de información en línea.

Palabras Clave: OWASP, ISO/IEC 27002, Proceso Crítico, Riesgos Aplicaciones Web, Metodología de las Elipses, Medición de Riesgos, Vulnerabilidades, Seguridad de la Información.

1 Introducción

Los sistemas de información (SI) se han convertido en factor de gran importancia 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 el presente trabajo, propone la implementación de un control de seguridad de la norma técnica ISO/IEC 27002[1], que integre un mecanismo de medición del riesgo de seguridad basado en OWASP 2013[2] para un sistema de información en línea.

Para llegar a dicha implementación se partió de la identificación de un procedimiento de nivel crítico en una organización mediante la metodología de las elipses[3], posteriormente se determina la técnica de inteligencia artificial que mejor se adapte al control de la ISO/IEC 27002[1] seleccionado y la propuesta de una arquitectura que permita la implantación del control de inteligente.

Cabe resaltar que este trabajo fue financiado por Fondo Regional para la Innovación Digital en América Latina y el Caribe (FRIDA) por ser el "Proyecto Ganador de Subvenciones FRIDA 2013".

1.1 Antecedentes

Para el desarrollo del presente artículo se consultaron los siguientes trabajos relacionados con el tema objeto de estudio.

1.1.1 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. En ella se describen términos y definiciones, además de aportar las bases de la implementación de un Sistema de Gestión de la Seguridad de Información (SGSI)[4].

Entre esta familia de estándares se destacan la ISO/IEC 27001 (Sistemas de gestión de la seguridad de la información - SGSI)[5], 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)[1], 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)[5], que proporciona pautas para la gestión de riesgo de seguridad de la información.

1.1.2 OWASP

Es un proyecto iniciado en el año 2000. Está 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. 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 proyectos de documentación. Para el presente estudio sólo se utilizaron los siguientes proyectos de documentación: Guía de pruebas[2] y OWASP Top 10 - 2013[6]. Este último es un 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.

1.1.3 Vulnerabilidad en las aplicaciones web

La vulnerabilidad es una debilidad en una aplicación web. Esta puede ser un fallo en el diseño, un comportamiento inesperado o un error en la implementación. Ella 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[7].

1.1.4 Secuencia de comandos en sitios cruzados (en inglés “Cross Site Scripting” -XSS-)

Este es un ataque que inyecta 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, tokens de sesión 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[8].

Según algunos autores[6] existen tres tipos de fallas conocidas XSS: 1) Almacenadas, 2) Reflejadas y 3) basadas en DOM.

- *Persistente:* Consiste en almacenar código en las aplicaciones web para que se ejecute una vez la aplicación web se carga[9].

- *Reflejado:* Funciona modificando valores que las aplicaciones web pasa de una página a otra[10].

- *DOM:* Es un tipo de inyección que permite al atacante tomar el control de un DOM[11].

1.1.5 Detección de ataques XSS

Varios autores[12] se enfocan en determinar características en documentos Web y Uniform Resource Locator, por su equivalente en inglés (URL) que les permite clasificar los ataques usando técnicas de aprendizaje de máquina (en inglés Machine Learning)

Otros autores[13] 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.

Hay quienes[14] 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).

Otro grupo de investigadores[15] determinan características de los ataques XSS encontrados en URL y JavaScript, y apoyados en técnicas de Machine Learning (Naive Bayes, Árbol de decisión (generado con J48), Support Vector Machine) clasifican las páginas web en normal o malicioso.

Adicionalmente, se encuentran autores[16] que se enfocan en obtener características de la URL y del contenido de la página web. Usan Técnicas de

Machine Learning (Naive Bayes 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 Desarrollo e implementación

2.1 Selección del proceso crítico de la organización

El especialista Alberto G. Alexander (2007)[3] propone la metodología de las elipses como el método más apropiado para determinar el alcance de un SGSI, identificando los procesos, unidades y entidades externas en la organización.

Para seleccionar un proceso crítico en una organización se sugieren seguir estos pasos:

Paso 1: Definir el alcance de la organización siguiendo la sección 4.3 del estándar ISO/IEC 27001:2013[17].

Paso 2: Identificar los procesos de la organización, e identificar las unidades organizacionales y entidades externas. Para esto se sugiere aplicar la metodología de las elipses para crear la elipse de los procesos críticos de la organización.

Paso 3: Identificar y describir los activos de información como lo indica el control A.8.1.1 en la Tabla A.1. de la norma ISO/IEC 27001:2013[10]

Paso 4: Tasar los activos de la información como lo indica Alberto G. Alexander (2007). Además es necesario identificar un porcentaje de participación en cada proceso.

Paso 5: Definir el Sistema de información usando los resultados consignados del Paso 4; seleccionar el activo de información que tenga, el valor total alto de la tasación en caso de existir varios, seleccionar el que tenga el promedio de participación más alto con respecto a los procesos críticos.

Paso 6: Seleccionar el proceso crítico. Para aplicar este paso se requiere la figura de la elipse construida en el Paso 2. 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.

Paso 7: Seleccionar una metodología para la valoración del riesgo. El numeral 6.1.2 literal d de la norma ISO/IEC 27001:2013[10], exige que en la organización debe existir una metodología para la evaluación del riesgo de los activos. Se sugiere escoger la metodología de valoración de riesgo de OWASP[5], ya que esta facilita puntuar el riesgo en las aplicaciones web, lo que permite ahorrar tiempo y estimar adecuadamente la severidad de todos los riesgos, asegurando no distraerse en riesgos de menor importancia.

2.2 Relación entre ISO/IEC 27002 y OWASP Top 10 2013

Se filtró los controles de la ISO 27002:2013[3] aplicando los siguientes criterios:

- ❖ Los controles de ámbito administrativo no se tomaron en cuenta, por lo tanto no se tuvieron en cuenta los dominios del 5 al 8 y sus correspondientes controles.
- ❖ 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 su aplicabilidad al listado de OWASP. Al finalizar se obtiene una lista de controles que tiene relación con OWASP top-10[6] y la norma ISO/IEC 27002[3].

La Tabla 1 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[3], 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 1. 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	Top 10 OWASP 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			

ISO/IEC 27002:2013	Top 10 OWASP 2013									
	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
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			

ISO/IEC 27002:2013	Top 10 OWASP 2013									
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
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				

2.2.1 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[1] 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 2013, dándole la característica de escalar de manera más contundente a cualquiera de estos riesgos por medio de otros controles que lo complementen.

2.3 Propuesta del Control Inteligente

Este trabajo propone una arquitectura fundamentada en la propuesta de V. Teresius[18], a dicha propuesta se le hizo una adaptación. La Figura 1 presenta la vista general de la propuesta.

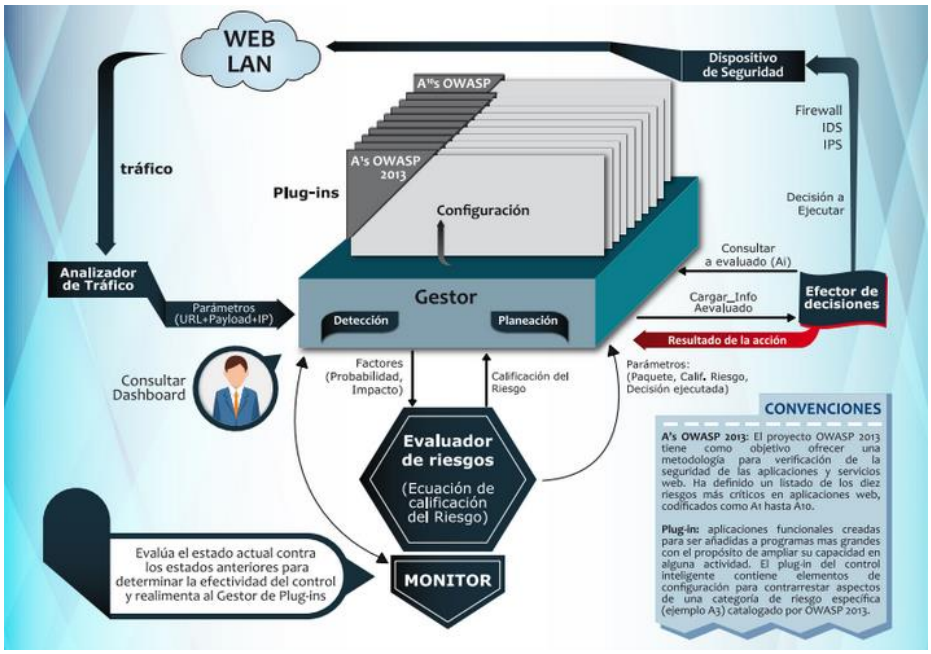

Fig. 1. Visión General de la propuesta.

La adaptación de la arquitectura (Figura 2) a partir de la propuesta de V. Teresius[18] 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.

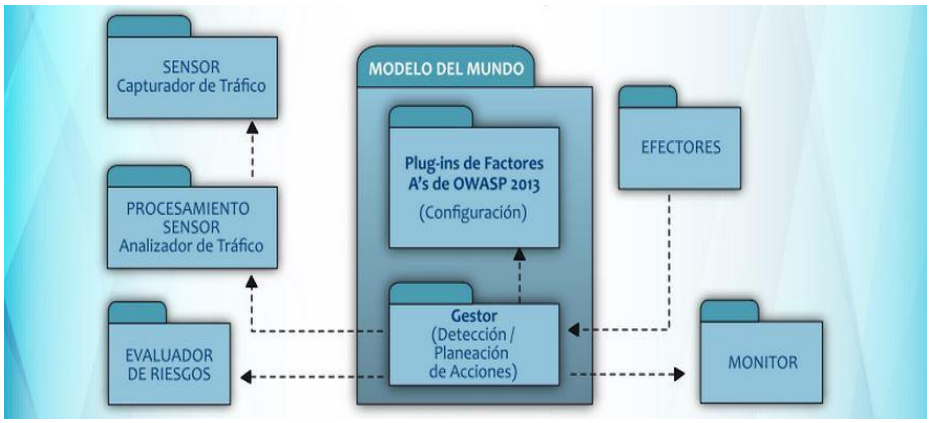

Fig 2. Vista Arquitectónica Control Inteligente.

A continuación se presenta una descripción general de cada uno de los componentes de la Figura 1 y la descripción de los módulos que se seleccionaron para el desarrollo de la arquitectura propuesta.

2.3.1 Módulo Capturador de Tráfico (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.

2.3.2 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.

2.3.3 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[6], 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.

2.3.4 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.

2.3.5 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.

2.3.6 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.

2.4 Detección Inteligente en el Control

Para seleccionar una técnica de inteligencia se decidió aplicar la metodología de CRISP-DM, compuesta por 6 fases: comprensión del negocio, comprensión de datos, preparación de datos, modelado, evaluación y desarrollo. A continuación se presenta lo que se realizó en cada una de ellas.

2.4.1 Fase I Comprensión del negocio

En esta fase se revisó la situación actual del riesgo de las aplicaciones web, y el Cross Site Scripting XSS (Secuencia de Comandos en Sitios Cruzados), 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[19].

También se encontró autores[13], [14], [15] y [16], quienes han 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.

El experto en seguridad considero que las escalas mínimas a tener en cuenta para de complejidad de un ataque eran: (NO_ATAQUE, BAJO, MEDIO, ALTO).

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.

2.4.2 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, 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[20] 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. 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).

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

2.4.3 Fase III Preparación de datos

A cada elemento de los conjuntos de datos de entrenamiento fue necesario realizarle un proceso de decodificación 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, por ser un estándar para el intercambio de información usado casi en todos los sistemas informáticos de la actualidad; 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. 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.

El "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.

Para seleccionar las características se utilizó la guía XSS Filter Evasion Cheat Sheet[21] 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[46] que sugiere el uso de listas blancas 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.

- *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.
- *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"[22].
- *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 %"
- *Característica 4. Search Comment.* En esta característica se identifica si existen comentarios de HTML dentro de una cadena de caracteres.
- *Característica 5. Search Document Cookie.* Esta característica verifica si existe el elemento document.cookie en una cadena de caracteres.
- *Característica 6. Search Document Location.* Esta característica explora si hay elementos DOM.
- *Característica 7. Search From Char Code.* Esta característica busca si hay elementos con la forma ".fromcharcode".
- *Característica 8. Search Functions.* Esta característica explora si existen funciones dentro de una cadena de caracteres.
- *Característica 9. Search Html Event Attributes.* Esta característica busca eventos "on" que pertenecen al HTML dentro de una cadena de caracteres.
- *Característica 10. Search Protocol.* Esta característica identifica si existen protocolos dentro de una cadena de caracteres.
- *Característica 11. Search Tags*. Esta característica verifica la existencia de elementos Tags dentro de una cadena de caracteres.
- *Característica 12. Control Character*. Esta característica, busca si existen caracteres de control no imprimibles, como salto de línea y retorno de carro.

2.3.4 Fase IV Modelado

En los artículos [13], [14], [15] y [16], los autores han usado diferente 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 Desiciones (generado con 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 (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 [23].

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

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.

Los resultados obtenidos, luego de aplicar los algoritmos fueron:

Tabla 2. Resultados de la Prueba Experimental

Algoritmo de aprendizaje de máquina	Instancias clasificadas correctamente	Porcentaje correctamente clasificado	Instancias No clasificadas correctamente	Porcentaje no clasificado correctamente	Promedio ponderado precisión	Promedio Roc área
J48	1559	97.8657%	34	2.1343%	97.9%	0.975
Naive Bayes	1424	89.3911%	169	10.6089%	89.8%	0.968
Perceptron Multicapa	1564	98.1795%	29	1.8205%	98.2%	0.999

De la tabla 2 se observa que el algoritmo de Naive Bayes, es la peor de las tres Técnicas de Machine Learning y el Perceptron Multicapa aparenta ser la mejor opción.

2.4.5 Fase V Evaluación de los Modelos

En esta etapa se evaluó cada modelo seleccionado en la prueba experimental 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 (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 3. Comparación del porcentaje de aciertos de los modelos construidos en la fase experimental

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%

En la tabla 3, se evidencia el porcentaje de acierto obtenido de los modelos generados en la fase experimental, con los cuatros conjunto de datos de entrenamiento. Pero en este caso, la efectividad del Perceptron Multicapa no fue la más alta.

Se escoge el modelo de clasificación J48 generado en la prueba experimental para hacer parte de la detección del control inteligente, 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.

3 Implantación del control

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.

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 3. Cabe aclarar que el servidor Control es el que hospeda al control de seguridad (prototipo).

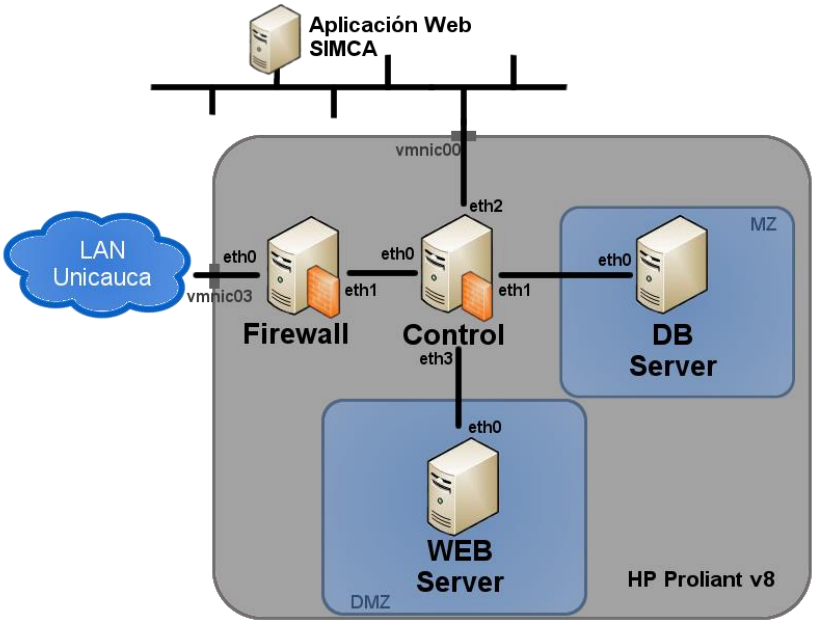

Fig.3. Arquitectura de red.

Las solicitudes las recibe la interfaz eth0 del servidor Firewall, la cual 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.

3.1 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.

3.1.1 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.

3.1.2 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.

3.1.3 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.

3.4 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 Conclusiones

Adaptar la metodología de las elipses en un conjunto de pasos, permite que de forma precisa, se determine el alcance, se seleccione el proceso más crítico y la aplicación web más importantes para la organización.

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. 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. A su vez, 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.

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.

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

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.

Agradecimientos

El desarrollo de este trabajo fue financiado en su totalidad por el Fondo Regional para la Innovación Digital en América Latina y el Caribe (FRIDA).

Los autores desean expresar sus agradecimientos por el apoyo recibido a todos los involucrados en el proyecto Control Inteligente para un Servicio Crítico de un Sistema de Información en Línea Enmarcado en un Dominio de la ISO/IEC 27002 y en especial a la División de Tecnologías de la Información y las Comunicaciones en la Universidad del Cauca, en especial a Jorge Martínez, Gonzalo Acte y Francisco Javier Terán.

Siler Amador Donado, Ember Ubeimar Martinez Flor, Carlos Alberto Ardila Albarracín y Sara Donelly Garcés agradecen a la Universidad del Cauca donde trabajan como docentes.

Referencias

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

[2] The OWASP Foundation, "OWASP Testing Guide v4." [Online]. Available: http://goo.gl/r9ccBe, 2015.

[3] Alberto G. Alexander, *Diseño De Un Sistema De Gestión De Seguridad De Información*. Bogotá: Alfa Omega editores, 2007.

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

[5] ISO/IEC 27005:2011, "Information technology - Security techniques - Information security risk management." 2011.

[6] The OWASP Foundation, "OWASP Top 10 2013." [Online]. Available: http://goo.gl/gjQNzs, 2013.

[7] Universidad Nacional Autónoma de México, "Punto Seguridad Defensa Digital, Ataques Web," vol. 5.

[8] OWASP, "Cross-site Scripting (XSS)." [Online]. Available: http://goo.gl/jXnKzO, 22-Apr-2014.

[9] WebSecurityDev, "Vulnerabilidad Cross-site scripting y sus Clases." [Online]. Available: http://goo.gl/cxNgP6.

[10] Angie Aguilar Domínguez, "¿Qué es y cómo opera un ataque de Cross-Site Scripting (XSS)?" [Online]. Available: http://goo.gl/wvYXYN.

[11] "XSS Hacking Tutorial." [Online]. Available: http://goo.gl/yLMpdV.

[12] 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," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, 2012, pp. 000702–000707.

[13] P. K. C. For, *Multiclass Classification of XSS Web Page Attack using Machine Learning Techniques S.Krishnaveni*. .

[14] Dr R.P Mahapatra, Ruchika Saini, Neha Saini, "A Pattern Based Approach to Secure Web Applications from XSS Attacks," *J. Comput. Technol. Electron. Eng.*, vol. 2.

[15] 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.

[16] S. S. Prof. D.D.Patil, “Document-based and URL-based Features for Automatic Classification of Cross-Site Scripting in Web Pages,” *IOSR J. Eng.*, vol. 3, pp. 11–18.
[17] *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.
[18] V. Teresius, “Reference architecture of intelligent system,” *Electron. Electr. Eng.*, vol. 63, no. 7, pp. 53–56, 2005.
[19] Antonio Rodríguez Romero, “Ataques XSS en Aplicaciones Web.” [Online]. Available: http://goo.gl/sRPEM4.
[20] “OWASP Xenotix XSS Exploit Framework.” [Online]. Available: http://goo.gl/tGlIRp.
[21] The OWASP Foundation, “XSS Filter Evasion Cheat Sheet.” [Online]. Available: http://goo.gl/JPr31g, 2015.
[22] “Unicode,” *Mozilla Developer Network*. [Online]. Available: http://goo.gl/9OLCMM. [Accessed: 05-Mar-2015].
[23] 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.