
**CENTRO DE CIENCIAS BÁSICAS**

**DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN**

# TESIS

IDENTIFICACIÓN Y DISEÑO DE LAS REGLAS DE DETECCIÓN DE AMENAZAS DE INEGI PARA UN WEB APPLICATION FIREWALL

## PRESENTA

JUANA TERESITA DE JESÚS BONILLA ROSALES

**PARA OBTENER EL GRADO DE MAESTRA EN INFORMÁTICA Y TECNOLOGÍAS COMPUTACIONALES**

**TUTORES**

Dra. María Dolores Torres Soto
Dra. Aurora Torres Soto

**INTEGRANTE DEL COMITÉ TUTORAL**

Dr. Cesar Velázquez

AGUASCALIENTES, AGS., 4 DE MAYO DE 2018

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

**M EN C JOSE DE JESUS RUIZ GALLEGOS**
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como Tutor designado del estudiante **JUANA TERESITA DE JESUS BONILLA ROSALES** con **ID 44555** quien realizó el trabajo de tesis titulado: **IDENTIFICACIÓN Y DISEÑO DE LAS REGLAS DE DETECCIÓN DE AMENAZAS DE INEGI PARA UN WEB APPLICATION FIREWALL** y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que ella pueda proceder a imprimirlo, así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 4 de mayo de 2018.

Dra. María Dolores Torres Soto
Tutor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría de Investigación y Posgrado
c.c.p.- Jefatura del Depto. de Sistemas de Información
c.c.p.- Consejero Académico
c.c.p.- Minuta Secretario Técnico

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

**M EN C JOSE DE JESUS RUIZ GALLEGOS**
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como Tutor designado del estudiante **JUANA TERESITA DE JESUS BONILLA ROSALES** con **ID 44555** quien realizó el trabajo de tesis titulado: **IDENTIFICACIÓN Y DISEÑO DE LAS REGLAS DE DETECCIÓN DE AMENAZAS DE INEGI PARA UN WEB APPLICATION FIREWALL** y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que ella pueda proceder a imprimirlo, así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 4 de Mayo de 2018.

Dra. Aurora Torres Soto
Tutor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría de Investigación y Posgrado
c.c.p.- Jefatura del Depto. de Sistemas de Información
c.c.p.- Consejero Académico
c.c.p.- Minuta Secretario Técnico

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

**M EN C JOSE DE JESUS RUIZ GALLEGOS**
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como Tutor designado del estudiante **JUANA TERESITA DE JESUS BONILLA ROSALES** con **ID 44555** quien realizó el trabajo de tesis titulado: **IDENTIFICACIÓN Y DISEÑO DE LAS REGLAS DE DETECCIÓN DE AMENAZAS DE INEGI PARA UN WEB APPLICATION FIREWALL** y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que ella pueda proceder a imprimirlo, así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 17 de Mayo de 2018.

Dr. César Eduardo Velázquez Amador
Asesor de Tesis

c.c.p.- Interesado
c.c.p.- Secretaría de Investigación y Posgrado
c.c.p.- Jefatura del Depto. de Sistemas de Información
c.c.p.- Consejero Académico
c.c.p.- Minuta Secretario Técnico

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

**JUANA TERESITA DE JESÚS BONILLA ROSALES**
**MAESTRÍA EN INFORMÁTICA Y TECNOLOGÍAS COMPUTACIONALES**
**P R E S E N T E.**

Estimada alumna:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **"IDENTIFICACIÓN Y DISEÑO DE LAS REGLAS DE DETECCIÓN DE AMENAZAS DE INEGI PARA UN WEB APPLICATION FIREWALL",** hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

**A T E N T A M E N T E**
Aguascalientes, Ags., a 18 de mayo de 2018
*"Se lumen proferre"*
**EL DECANO**

**M. en C. JOSÉ DE JESÚS RUÍZ GALLEGOS**

c.c.p.- Archivo.

# AGRADECIMIENTOS

Agradezco a INEGI por el apoyo proporcionado para realizar mi investigación, sobre todo al Grupo de Ingeniería en Sistemas o GIS que proporcionaron los datos para la realización del estudio realizado.

Agradezco a la universidad autónoma de Aguascalientes por todo el apoyo y conocimientos obtenido durante el transcurso de la maestría y sobre todo a mis tutoras y cotutores que me han estado ayudando a la realización de mi trabajo sobre la detección de reglas de identificación de amenazas.

Así como al maestro Osvaldo Diaz por el tiempo invertido para conocer mejor el tema de estudio y asesorarme para la realización del proyecto de tesis, así como de la información proporcionada para la realización de la tesis.

Y sobre todo un agradecimiento a mi familia por la paciencia y el apoyo para poder hacer cada una de las investigaciones y mi esposo por entenderme, así como ayudarme con la investigación de ciertas funciones que se necesitaron para la creación de la herramienta.

# DEDICATORIAS

Dedico esta investigación a mi futuro hijo, esposo, familia y amigos que han estado apoyándome en cada momento que ha transcurrido de la maestría.

Esta investigación me ha dejado grandes enseñanzas sobre el uso de los WAFs dentro de las empresas, y sobre todo la importancia que el INEGI debe de darles, ya que es uno de los filtros que permitirá denegar servicios no autorizados.

Dejando un legado sobre futuras investigaciones relacionadas con el tema debido a que solo aborda una parte de la gran cantidad de información que se relaciona con la investigación, como ampliar las amenazas a que sea todo el Top 10 de OWASP o desarrollar mediante métodos heurísticos y estadísticos mejores herramientas que permitan no solo detectar lo del top 10 sino que también futuras amenazas no cubiertas por este organismo, así como crear automáticamente los archivos de configuración necesarios para el MODSECURITY.

# INDICE GENERAL

# ÍNDICE DE TABLAS

# ÍNDICE DE IMAGENES

# RESUMEN EN ESPAÑOL

El presente documento habla sobre los Cortafuegos de Aplicaciones Web o Web Application Firewall (WAF), así como de la importancia de aplicarlos en las empresas, y de obtener de manera precisa las reglas de detección a las amenazas que acechan a las empresas.

Actualmente, un tema en boca de todos es la seguridad informática y la seguridad de la información. ¿Para qué sirve? ¿En dónde se aplican y que se tiene que hacer para decir que una empresa se encuentra segura?

La seguridad informática y seguridad de la información son dos áreas muy distintas, pero con un objetivo en común: proporcionar confiabilidad a los datos sensibles manejados por una organización.

La seguridad de la información es aquella disciplina basada en los principios de integridad, disponibilidad y confidencialidad, es decir que la información no se pierda y sea legible a las personas correctas o autorizadas. En otras palabras, sólo las personas con la autorización tienen acceso al uso de la información.

Este trabajo defiende el uso de los cortafuegos de aplicaciones en las empresas y la mejora que éstas pueden tener al atacar las amenazas más comunes establecidas en el top 10 de amenazas lanzado por el Proyecto Abierto de Seguridad en Aplicaciones Web u Open Web Application Security Project (OWASP en inglés), este organismo permite compartir información y herramientas para incrementar la seguridad web en las empresas.

Como se ha comentado anteriormente los Cortafuegos de Aplicaciones Web o WAF son importantes para la prevención de ataques a las aplicaciones web existentes, sin importar como estén realizadas. En el capítulo 3 se profundizará más acerca sobre los cortafuegos de aplicación, creación de reglas para la detección de amenazas de seguridad informática y las técnicas empleadas para crear reglas acordes a las necesidades de las empresas gubernamentales, en este caso INEGI.

En el capítulo 4 se muestra a detalle la metodología realizada para la creación de reglas de detección de amenazas, mediante el insumo obtenido del laboratorio de pentesting. Las cuáles serán implementadas por el Grupo de Ingeniería en Sistemas o GIS del INEGI. Estas son el producto de la herramienta desarrollada durante la investigación y son visualizadas como un reporte que facilita al GIS auditar mejor las reglas.

# RESUMEN EN INGLÉS

This document tells about Web Application Firewall o WAF, also the importancy about to implement inside in company's and obtain rules of identification more exact to identify threats that attack to the company.

The informatic security and information security are a topic that a lot people speaking. ¿What is the focus? ¿Where can I use and how to do for saying an enterprise is healthy?

The information security and informatic security are two areas so different, but with a focus in common: bring reliability in sensitive data that a company can exploit.

The information security is a discipline based on integrity, availability and reliability, with these principles the information is difficult to lost because the information is comprehensible for right people, in other words, people authorize logging to the information.

This work advocates the web application firewalls used in the enterprises and the performance defending for threats so typical for the vulnerabilities top 10 of the Open Web Application Security Project (OWASP), this organism allowed to share information and tools to increment the web security in company's

A Web Application Web (WAF) are very valuable to prevent attacks over web applications in use for they it's not important how is realized. In the chapter 3, you can know more about firewalls, how to create rules for detecting informatic security's threats and techniques employees for building rules according to the necessities of governmental company's, in this case INEGI.

The chapter 4 tells about the methodology realized to create the threats detection rules, through the information recollected at pentesting laboratory. These rules are implemented for the Systems Engineering Group of INEGI. These are the result of the tools developed during the research and are they are showed how a report that help the group to audit better the rules.

# 1. INTRODUCCIÓN

## 1.1 ANTECEDENTES

Hoy en día, las empresas manejan información sensible, como datos financieros e información del negocio a través del internet, mediante aplicaciones web o redes sociales. Esto, con la finalidad de abarcar más mercado o agilizar procesos en distintas áreas.

En el pasado, las empresas usaban páginas web con información visible para todo el mundo, es decir, la información manejada en las páginas web era sólo de uso informativo, por lo que la comunicación entre el usuario y el servidor de la empresa era inexistente (Manrique Maldonado,2015).

Conforme fue avanzando la tecnología, la interacción de los usuarios con las aplicaciones web, permitió el intercambio de información personal y financiera a través de internet mediante los servicios ofrecidos en está. Además, la interacción de los usuarios fue más dinámica, ya que permitió la construcción y transformación de la información del contenido web. A esto se le conoce como Web 2.0. Por lo tanto, a la etapa donde no había comunicación entre usuario y aplicaciones Web, se le conoce como Web 1.0 (Manrique Maldonado,2015).

Debido a que ahora la interacción entre usuarios y servidor es muy intensa, las empresas e investigadores en el mundo, han volteado a poner mayor énfasis en la seguridad de la información.

La seguridad ha ido incrementando desde 2003 debido a los ataques realizados en Estados Unidos; esto ha provocado que las empresas no escatimen esfuerzos en la implementación de medidas necesarias en el ámbito de seguridad, porque han observado que es algo que le compete a cualquiera y no solo a la milicia (Higuera, 2013).

Actualmente, existen empresas que se encargan de difundir el tema de seguridad de la información en aplicaciones Web, como es el caso de OWASP (Open Web Application Security Project o Proyecto Abierto de Seguridad en Aplicaciones Web), que es un proyecto abierto para la difusión de seguridad en aplicaciones Web. OWASP pone a disposición de las empresas marcos de referencia enfocados a la seguridad, un top 10 de riesgos informáticos; que se actualiza constantemente para ser tomado en cuenta por los desarrolladores de software (Prince, 2013).

Además, ofrece herramientas que permite a las empresas conocer las vulnerabilidades existentes en sus aplicaciones Web. Estas herramientas deben ser manejadas por profesionales o personas con conocimiento profundo en el tema (OWASP, 2017).

También existen empresas que se han encargado de desarrollar software para bloquear la entrada a las amenazas existentes, este software es llamado "Cortafuegos de Aplicaciones Web" o Web Application Firewall (WAF). Los WAF existen en dos vertientes; como aplicación o como dispositivo; algunos ejemplos de empresas que han desarrollado un WAF genérico se puede visualizar en la Tabla 1, esta tabla visualiza algunos cortafuegos de aplicaciones comerciales tanto de paga como de accesos libre.

*Tabla 1. Cuadro comparativo de cortafuegos de aplicación Web (WAF)*

| Cita | WAF Comercial | Ventajas | Desventajas |
|---|---|---|---|
| **(Radware, 2017)** | Radware's AppWall | Posicionado dentro del cuadrante de Gartner<br>Completa cobertura, incluyendo las amenazas citadas por OWASP<br>Protección (eXtensible Markup Language o Lenguaje de Marcas Extensible) XML y aplicaciones Web<br>Prevención de ataques día cero. | No mencionan costos, se debe solicitar presupuesto. |
| **(MODSECURITY, 2017)** | MODSECURITY | Permite el monitoreo, logueo y control de acceso.<br>Código abierto<br>Se puede tener acceso a las reglas y código para su adaptación<br>Aprende | Sólo modo pasivo |
| **(Imperva,2017b)** | ThreadRadar | Revisión de código en tiempo de ejecución<br>Filtrado de tráfico.<br>Inteligencia Crowd-source<br>Detecta clientes botnet o robots en la red y ataques de denegación de servicios (ddos)<br>Prevención de Fraude | Tiene costo |
| **(Privacyware, 2017)** | ThreatSentry | Motor basado en un comportamiento neural.<br>Instalación multiservidor<br>Monitoreo centralizado | Costo por servidor de 649 dólares |
| **(Imperva, 2017a)** | SecureSphere | Cubre las vulnerabilidades mencionadas en el Top 10 de OWASP | |

**Fuente:** Elaboración propia.

*Tabla 2. Cuadro comparativo de investigaciones relacionadas*

| Cita | Título del estudio | Problema | Metodología utilizada | Descripción | Contribución |
|---|---|---|---|---|---|
| **(Torrano Giménez, 2015)** | Estudio de técnicas estocásticas y máquina aprendizaje para la detección de anomalías basadas en ataques Web. | Mejorar la brecha entre los ataques y la defensa de éstos. Para así, mejorar la seguridad de las aplicaciones Web. | -Métodos estocásticos- Algoritmos de aprendizaje máquina.<br>-Técnica estadística<br>-Cadena de Markov<br>-Árboles de decisiones | En la tesis se habla del diseño de Cortafuegos de Aplicaciones Web mediante tres técnicas: La estadística, Cadena de Markov y árboles de decisiones. Comparando en cada técnica las siguientes variables: tasa de falsos positivos, detección de amenazas y rapidez. | El uso de técnicas estadísticas produce una mejor tasa de detección, mayor rendimiento y menor número de casos de falsos positivo. Técnicas estocásticas son menos efectivas en cuánto a la detección de amenazas y el número de falsos positivos es mayor. |
| **(Torrano-Gimenez, Perez-Villegas, & Alvarez, 2009)** | Cortafuegos de Aplicaciones Web basado en autoaprendizaje de anomalías | Implementar una versión de prueba de un WAF basado en anomalías. | Aprendizaje basado en anomalías. | Explica cómo implementar un WAF basado en anomalías donde las reglas de lo que es correcto se encuentran alojados en un XML | Éxito obtenido, pero con un gran número de falsos positivos. |

| | | | | | |
|---|---|---|---|---|---|
| **(Nguyen, Torrano - Gimenez, Alvarez, Franke, & Petrović, 2013)** | Mejorando la efectividad de los Cortafuegos de Aplicaciones Web mediante la selección de características genéricas | Obtener estadísticas de un WAF dentro de un alto nivel de tráfico HTTP. | Mediciones de Características genéricas seleccionadas en un alto nivel de tráfico HTTP. Usando un conjunto de datos ECML/PKDD-2007 | Investigación que permite conocer sobre el comportamiento de los Cortafuegos de Aplicaciones Web con características específicas que pueden ser genéricas dentro un conjunto de datos eliminando los conjuntos repetidos e inservibles. | Del conjunto de datos usado se obtuvo un 63% de características irrelevantes. |

**Fuente:** Elaboración propia

Por otra parte, existen diversas hipótesis o teorías sobre el uso de Metaheurísticas dentro del área de seguridad, en la Tabla 2, podemos ver algunas investigaciones relacionadas con el uso de diferentes métodos para identificar reglas de detección de amenazas y como han querido generalizarlas.

Para resolver la problemática actual, las empresas deben desarrollar medidas de prevención de robo de información e implementar buenas prácticas en el desarrollo de las aplicaciones enfocadas a la Web, si la empresa realiza aplicaciones Web; en el caso de que la empresa contrate a terceros para el desarrollo de sus aplicaciones Web, deberá de exigir que se cuente con las prácticas necesarias de seguridad.

Dentro de las medidas contempladas para la prevención de robo o acceso a la información sensible, se encuentra el uso de dispositivos que permiten cerrar los puertos abiertos (Cortafuegos físicos), los cuales pueden ser usados por los atacantes y así acceder a la información ajena.

Otro mecanismo de seguridad es el de los Sistemas de Detección de Intrusiones o IDS (Intrussion Detection Systems), hardening robusto, zona desmilitarizada, Cortafuegos de Aplicaciones Web.

Es importante también contar con buenas prácticas durante el desarrollo del software.

Hay que tomar en cuenta que, para tener un buen nivel de seguridad, se debe hacer uso de varias técnicas que permitan la protección de los datos; lo que indica que el uso de una sola técnica de protección no garantiza que se logre un buen nivel de seguridad de la información.

Para contar con un nivel de seguridad adecuado, se deben de cubrir dos puntos: físico (Cerrar puertos no utilizables, implementación de técnicas de hardening) y lógico (uso de Cortafuegos de Aplicaciones Web, antivirus, entre otros ejemplos).

La aplicación de un cortafuego físico y uno de aplicación Web (WAF) es una opción para cubrir ambos aspectos, pero el uso de reglas de detección no acorde con la empresa y la inexperiencia de la configuración de ambos cortafuegos hacen que estas medidas de seguridad no sean suficiente para una adecuada protección de datos.

Se puede decir que el uso de un cortafuegos se ha convertido en una opción de seguridad común, pero compleja, provocando una alta tasa de falsos positivos (identificación de una amenaza informática cuando no lo es), de manera que resulta poco óptima su implementación (Nomura & Salzetta, 2016).

Aunque con la adecuada identificación de reglas se puede tener una protección total en conjunto con otras técnicas de seguridad.

## 1.2 PROBLEMA.

A causa de la interacción de la información empresa-cliente, se han realizado más ataques a nivel informático para la obtención de los datos sensibles manejados por las empresas.

Esto ha causado que en otros países estén enfatizando las precauciones en el área de seguridad de la información. Esto ha contribuido a desarrollar profesionistas enfocados en el área.

En México, las empresas, ya sean, micros, pequeñas, medianas, o grandes, privadas o de gobierno; tienen un conocimiento muy escaso del área o implementan medidas muy pobres sobre los diferentes métodos de protección de los datos sensibles, debido a que no cuentan con la tecnología, los procesos y el recurso humano necesarios para implementarlos.

El Instituto Nacional de Estadística y Geografía (INEGI) es un organismo público de gobierno, dedicado a procesar información estadística y geográfica.

La información recopilada por INEGI se obtiene a través de diferentes medios como: Web, papel y dispositivos móviles. La información obtenida vía Web es transmitida por medio de internet y concentrada en un banco de datos, que es procesado para su explotación; por lo cual es de vital importancia cuidar la información y tratarla con los mejores mecanismos de seguridad para evitar robos o pérdidas.

Actualmente, la empresa cuenta con la tecnología y recursos humanos para la implementación de un Cortafuegos para Aplicaciones Web, pero no se ha podido implementar debido a miedos sobre el uso de la tecnología, a falta de procesos, y desconocimientos de las reglas necesarias para la detección de amenazas acorde con la empresa, este es un problema que no queda allí, pues se provocan muchos falsos positivos, es decir, agregan al cortafuego de aplicaciones, direcciones que son de confianza y no permite el acceso a la información requerida, causando malestar a las empresas que requieren la información y tienen convenios confiables con INEGI.

## 1.3 RELEVANCIA Y JUSTIFICACIÓN.

Como ya se ha mencionado, México es uno de los países más vulnerables en cuestiones de seguridad, esto es debido a varios factores como:

- Mano de obra no especializada en el área.
- Recursos insuficientes en cuestión de tecnología de seguridad.
- Desconocimiento de las diferentes medidas de seguridad entre los empleados.
- Manejo de políticas de seguridad mal diseñadas.
- Mal uso de la tecnología.
- Procedimientos incorrectos en el manejo de la información.

Por lo que es importante, implementar una serie de procesos y procedimientos que permitan a las instituciones de gobierno tanto públicas como privadas, el manejar los tres ámbitos en los que la seguridad influye que son nivel de hardware, software y humano (Ramírez Castro, 2017).

El uso adecuado de la tecnología, así como el uso de reglas de seguridad para la información transferida permitirá a las empresas que la información manejada deje de ser vulnerable ante los hackers. Día a día surgen nuevas tecnologías y se buscan nuevas formas de acceso.

Otro elemento importante de una buena herramienta de seguridad consiste en evitar los falsos positivos, que provocan la falta de confianza en comunicaciones sin malicia, pero la regla de seguridad no permite que la comunicación se concrete porque la identifica como una amenaza, haciendo que la empresa tenga pérdida de confianza por las empresas con un convenio de intercambio de comunicación.

## 1.4 OBJETIVOS.

### 1.4.1 General.

Implementar un mecanismo informático que permita identificar las vulnerabilidades y diseñar las reglas de detección de amenazas de INEGI para su implementación en un Web Application Firewall.

### 1.4.2 Específicos.

1. Identificar los factores necesarios para el manejo de reglas de detección de amenazas del instituto.
2. Desarrollar un mecanismo de automatización para la creación de reglas de detección de amenazas para el Instituto.
3. Establecer un marco de trabajo para el proceso de creación de reglas de detección de amenazas.

## 1.5 PREGUNTAS.

¿Es factible crear un mecanismo que permita la creación de reglas de detección de INEGI?
¿Es factible establecer un marco de trabajo para el proceso de creación de reglas de detección de amenazas de INEGI?
¿Es factible identificar los factores necesarios para manejar las reglas de detección de amenazas de INEGI?

## 1.6 HIPOTESIS.

Hipótesis 1: Un mecanismo de detección de reglas permitirá crear las reglas de detección de INEGI.

Hipótesis 2: La identificación de factores permitirá el manejo de las reglas de detección de amenazas de INEGI.

Hipótesis 3: Una metodología o marco de trabajo para establecer las reglas de detección de INEGI facilitará la comprensión de todo el proceso de identificación y diseño de reglas.

## 1.7 DESCRIPCIÓN DE TESIS.

La presente tesis permite diseñar las reglas de detección de amenazas relacionadas con el INEGI; las cuales sirven como base para que el experto las implemente en el Cortafuego de Aplicaciones Web utilizado por INEGI.

La recolección de material para la implementación de las reglas se realiza dentro de un programa automatizado que clasifica las vulnerabilidades encontradas en los proyectos (estos datos son proporcionados por un software que permite la detección de vulnerabilidades) y así obtener la información necesaria para generar las reglas que el experto necesita implementar en el cortafuego, reduciendo así de manera positiva los falsos positivos generados por las reglas que el programa trae de inicio.

En el capítulo 1- Introducción, se detalla la problemática detectada en INEGI, el por qué es importante investigar el tema, los trabajos que se han realizado sobre el tema, se establecen las preguntas que guían el proceso de investigación y las hipótesis que se tienen del problema.

En el capítulo 2 – Marco teórico se pueden encontrar términos sobre seguridad, ¿qué es una amenaza?, ¿Qué es una vulnerabilidad?, además se hace mención del top ten de vulnerabilidades y las definiciones sobre lo que es un cortafuego, demostrando así la importancia de éstos en una empresa ya sea pública, privada o de gobierno. Así como, lo que es MODSECURITY y la manera en que funcionan las reglas dentro de este tipo de cortafuego de aplicación web.

En el capítulo 3-Metodología, se puede encontrar a detalle la metodología utilizada para llegar a encontrar las reglas de detección de amenazas en INEGI y el funcionamiento del sistema para detectar estas reglas.

En el capítulo 4- Resultados, se describen los resultados obtenidos al aplicar la metodología descrita en el capítulo 3.

# 2. MARCO TEÓRICO.

## 2.1 SEGURIDAD (CONCEPTOS Y GENERALIDADES).

Antes de entrar de lleno al tema de interés, se verán algunos conceptos básicos que ayudarán al entendimiento de los términos utilizados a lo largo del capítulo.

Un concepto importante es ***seguridad,*** la cual se refiere a la capacidad de estar libre de peligro. Dentro del área computacional también existe este término. La seguridad se divide en dos ramas:

1. ***Seguridad de la información,*** que ve por la protección de la información, es decir, se asegura que la información no sea sustraída, eliminada y accedida por personas no autorizadas. En otras palabras, la **seguridad de la información** se encarga de cumplir tres aspectos o características básicas: ***integridad, confidencialidad y disponibilidad***. Si se llegara a violar alguno de estos tres principios se puede decir que la información no es segura (UNAM,2017).
2. **Seguridad *informática*** es la aplicación de normas o reglas tanto a nivel de software y hardware que evita que personas ajenas o con fines mal intencionados accedan a la información sensible que se estén manejando (UNAM,2017).

**La seguridad informática** permite protegerse de **amenazas**, que es un riesgo que puede afectar un objetivo en particular, puede ser un objeto, una persona, información, entre otras. Y se divide en dos tipos: amenazas naturales y las provocadas por el hombre.

1. **Las amenazas naturales** son aquellas que son causadas por desastres naturales como incendios, huracanes, inundaciones. Esto puede pasar en cualquier momento, pero hay que tomarlos en cuenta para prevenir perdidas costosas (Vacca, 2007b).

2. **Las amenazas provocadas por el hombre** son aquellas que el hombre realiza para su beneficio. En el mundo de la computación estas amenazas buscan la obtención de la información de manera ilegal ya sea por juego, venganza o un beneficio económico. Dentro de esta categoría se consideran; los fraudes cibernéticos, robo de identidad, entre otros (Vacca, 2007b).

OWASP define un ***ataque*** como las técnicas que utilizan los atacantes para descubrir las vulnerabilidades de las páginas Web.

Las **vulnerabilidades** son agujeros o espacios que los atacantes (dícese de las personas que quieren hacer mal uso de la información) encuentran dentro de las páginas y

mediante esos agujeros acceden a la información de manera ilegal, provocando a veces daños a la información sin haberse contemplado (Vacca, 2007b) .

En la tabla 3 podemos visualizar las vulnerabilidades más comunes que afectan a miles de empresas alrededor del mundo y son listadas de acuerdo con el daño que causan, este un ***top ten*** de vulnerabilidades que es publicado por (OWASP,2017).

*Tabla 3. Tabla de vulnerabilidades informáticas versión 2013 de OWASP.*

| Código | Riesgo | Relevante para el estudio |
|---|---|---|
| **A1** | Inyección | * |
| **A2** | Perdida de autenticación y gestión de sesiones | |
| **A3** | *Sentencias de comandos en sitios cruzados (XSS)* | * |
| **A4** | *Pérdida de control de los accesos* | |
| **A5** | *Configuración de seguridad incorrecta* | |
| **A6** | *Exposición a datos sensibles* | |
| **A7** | *Insuficiente protección a los ataques* | |
| **A8** | *Falsificación de peticiones en sitios cruzados(CSRF)* | |
| **A9** | *Uso de componentes con vulnerabilidades conocidas* | |
| **A10** | *APIs desprotegidas* | * |

**Fuente:** Basada en (OWASP,2017)

Un **cortafuego** puede ser un dispositivo o aplicación, que se encarga de negar el paso de cualquier amenaza; éstos se dividen en dos:

1. Los **físicos**, que bloquean los puertos inactivos (es decir, los canales de comunicación que no son usados, un ejemplo es el puerto 80 usado para tráfico Web) que los atacantes pueden usar para la obtención de la información de manera ilícita.

2. Los **Cortafuegos de Aplicaciones Web o WAF** por sus siglas en inglés. Catalogan la información de intranet de la red pública mediante reglas base, es decir, instrucciones que permiten dejar pasar información a la red interna o intranet. Mediante la correcta configuración de dichas políticas; éstos logran evitar los ataques de terceros (Mayer, Wool, & Ziskind, 2005).

Los WAF están diseñados para proteger a los servidores de aplicaciones Web de ataques basados en Web. Este tipo de cortafuegos permite que las respuestas sean analizadas antes de ser enviadas a su destino, permitiendo el aprendizaje de sus ataques y previniendo ataques a futuro (McMillan,2009).

De acuerdo con estudios realizados, la implementación de un WAF no es muy utilizado en la industria debido a que no se adapta a los diferentes entornos de trabajo (Hannes Holm & Mathias Ekstedt, 2013).

Pero su importancia radica en las reglas o políticas de detección, las cuales deben ser configuradas de manera correcta, para no comprometer el acceso a la información.

El funcionamiento del cortafuego de aplicaciones se describe a continuación (Maskey, Jansen, Guster, & Hall, 2007).

1.- Recibe las entradas provenientes de internet.
2.- Coteja las entradas recibidas con las reglas de detección implementadas por el WAF.
3.- Clasifica las entradas en *Permitidas* o *Denegadas*.
4.- Deniega al acceso a las entradas con la clasificación de denegada.

El funcionamiento más detallado lo podemos ver en la Imagen 1, la cual describe lo que pasa cuando el cortafuego de aplicaciones es implementado y la amenaza quiere acceder a los datos sensibles.

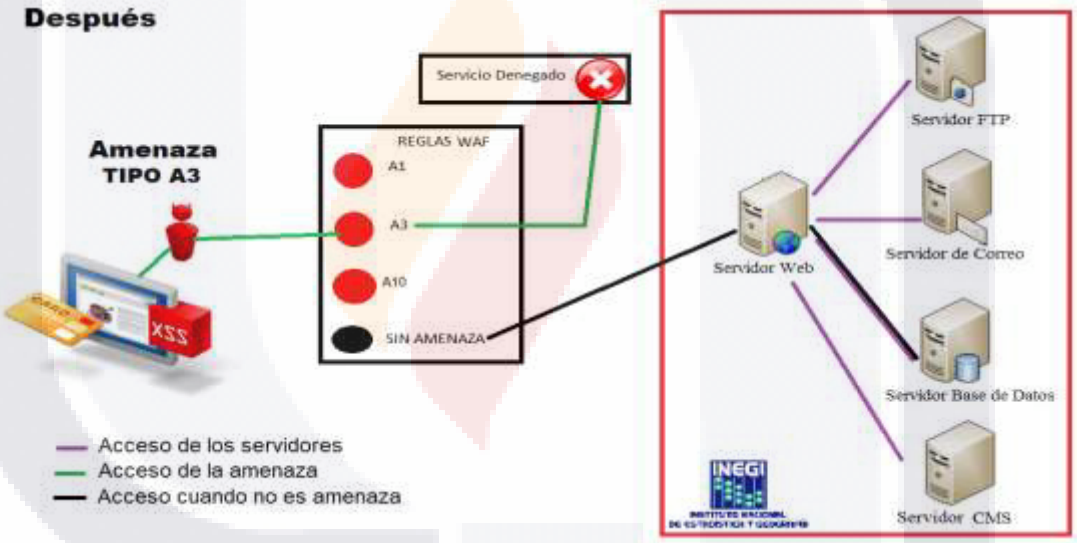


*Imagen 1. Funcionamiento del cortafuego de aplicaciones (WAF)*

Para poder evaluar el ataque se deben de considerar algunas variables como: el tipo de vulnerabilidad, que tan conocida es la vulnerabilidad, la severidad de la vulnerabilidad y los mecanismos de seguridad usados para evitar la vulnerabilidad.

Para poder comprender mejor en que beneficia el uso de un cortafuego de aplicaciones Web dentro de una empresa, se realizó una Tabla 4 donde se escriben las ventajas y desventajas que algunos autores han detectado.

*Tabla 4. Ventajas y desventajas de los Cortafuegos de Aplicaciones Web*

| Ventajas | Desventajas |
|---|---|
| **Evita ataques de terceros** | Dificultad en configurar el WAF |
| **Mediante el uso de reglas deniega el acceso de paquetes no deseados** | Reglas de protección duplicadas |
| **Se controla el acceso en una máquina.** | Constante ajuste de reglas de protección |
| **No se necesita modificar la aplicación** | No adaptable a todos los entornos |
| **Actúa antes de que la amenaza llegue a la aplicación** | |
| **Toma en cuenta el comportamiento del usuario y las sesiones** | |
| **Se basa en un histórico de ataques conocidos** | |
| **Se configura el servidor de aplicaciones Web** | |
| **Protege a cualquier aplicación Web** | |

**Fuente**: Basada en (Pałka & Zachara, 2011).

Existen diversas herramientas de WAF los cuales podemos implementar en una empresa, algunas de ellas fáciles de manejar y en otras es necesario capacitación para su uso (Vacca, 2007a).

Muchas investigaciones sobre seguridad de aplicaciones Web están basadas en OWASP. Como lo es la aplicación de Akana que es un sistema que detecta vulnerabilidades en el software (Akana, 2015) o (Cerullo, 2010) dice cómo implementar las reglas de OWASP para pruebas de aplicaciones Web.

Son pocas las tecnologías basadas en OWASP para la creación de un cortafuego de aplicaciones Web, por lo que se tiene mucho que descubrir para llegar a tener un firewall que se desarrolle con estás reglas (PR Newswire, 2013).

Si se quiere conocer cuáles son los mejores Cortafuegos de Aplicaciones Web, se puede consultar el cuadrante mágico de Gartner, el cual permite visualizar los mejores WAFs a nivel comercial, de acuerdo a (D'Hoinne & Neiva, 2016) Gartner distribuye los diferentes Cortafuegos de Aplicaciones Web en cuadrantes los cuales son: visionarios, líderes, retadores y jugadores del mercado; los que se sitúan en el cuadrante de líderes son los mejores catalogados y los cuales tienen muchas oportunidades, los que se posicionan en los de jugadores de mercado son aquellos que no tienen mucha oportunidad de crecimiento si no mejoran cualidades. En la Imagen 2, podemos ver la versión más reciente del cuadrante de Gartner.


*Imagen 2. Cuadrante mágico sobre WAFs comerciales (Gartner, 2017)*

## 2.2 MODSECURITY

MODSECURITY es una herramienta de código abierto que permite el monitoreo, logueo y control de acceso en tiempo de ejecución; esta aplicación tiene varios módulos, entre los más destacados se encuentra el módulo de Cortafuegos de Aplicaciones Web, el cual permite defenderse del tráfico peligroso que se encuentra en la Web y provee poderosas reglas que permiten el bloqueo de las amenazas detectadas (TRUSTWAVE, 2017).

Otra definición de MODSECURITY dice que es un cortafuego de aplicaciones que permite la entrada y salida de datos permitiendo así denegar el acceso a los datos que considera una amenaza, esto lo hace basado en reglas ya pre-configuradas (Magnus,2009).

MODSECURITY funciona mediante expresiones regulares y conjuntos de reglas para filtrar los ataques a los sitios de la máquina donde se encuentra instalado, este módulo es basado en OWASP y se describirán a continuación los comandos para la manipulación de las reglas y personalizarla de acuerdo con el negocio (Hostalia, 2017).

La estructura de una regla en MODSECURITY es la siguiente:

Directiva / proceso / acciones

En la directiva va el comando a aplicar, ya sea SecRule, SecRequestModeAccess, entre otros, estas directivas se pueden visualizar en la Tabla 5, es decir la forma de tratar la información.

En proceso va REQUEST HEADERS o algún otro proceso que indica en donde va a buscar la amenaza, ver Tabla 6, es decir en donde se localiza y como validar la amenaza, esto incluye las expresiones regulares o ciertos patrones que deben coincidir para que se cumpla o no se cumpla una amenaza, también puede ser un conjunto de direcciones Web que cumplen con cierto criterio.

Las acciones permiten definir que se hará en caso de cumplir con la expresión regular que se está monitoreando.

Un ejemplo es el siguiente, lo que está en negritas la directiva y lo que está en cursiva la directiva lo de texto normal son las acciones:

**SecRule** REQUEST *HEADERS* :User-Agent "xxx.2.6.5" "log,drop"  (Hostalia, 2017).

*Tabla 5. Tabla de directivas de MODSECURITY.*

| Directiva | Descripción |
|---|---|
| SecRequestBodyAccess | Puede tomar los valores de on y off, dependiendo del valor le indica al módulo cuando acceder al cuerpo de las peticiones. On permite acceder a la información incrementando la memoria de la máquina. Off permite bloquear el acceso al cuerpo de peticiones. |
| SecRequestBodyInMemoryLimit | Permite indicar el tamaño en bytes que será reservado en memoria RAM para almacenar los valores del cuerpo de peticiones. |
| SecRequestBodyLimit | Permite indicar el tamaño máximo de bytes permitidos para los buffers de los cuerpos de las peticiones, en caso de subirse un archivo se le debe de indicar el tamaño del archivo a subir. |
| SecRequestBodyNoFilesLimit | Directiva similar a la anterior, pero con la diferencia de que no se toma en cuenta la existencia de aplicaciones con subida de archivos. |
| SecResponseBodyAccess | Permite activar o desactivar el procesamiento de MODSECURITY para las respuestas generadas por el servidor. Puede tomar los valores off y on. |
| SecResponseBodyLimit | Permite establecer el límite n bytes de las respuestas generadas por el servidor. |
| SecResponseMimeType | Permite indicar los tipos MIME que pueden ir en las respuestas del servidor, es decir permite indicar lo que se permite o no se permite en el módulo. |
| SecRule | Examina las cabeceras de manera que cumpla con un valor dado definido en las expresiones regulares y ejecutando una acción en caso de cumplirse. |

Fuente: Basada en (Hostalia, 2017).

*Tabla 6. Tabla de procesos de las reglas de MODSECURITY.*

| Proceso | Descripción |
|---|---|
| Request Headers | Intercepta la petición realizada por el cliente a partir de las cabeceras y el cuerpo para enviarse al motor de reglas de MODSECURITY. Se realiza antes de procesar la petición. |
| Request Body | Importante porque permite el análisis de la petición interceptada, de manera que si cumple alguna regla procede a ejecutarla. |
| Response Headers | Se realiza cuando la petición es generada y ya se han analizado las cabeceras, de manera que permite inspeccionar el cuerpo de la respuesta. |
| Response Body | Se realiza el análisis del cuerpo de la respuesta de manera que el motor de reglas tome la decisión más adecuada. |
| Logging | Indica la forma de logueo y no puede ser bloqueada. |

**Fuente:** Basada en (Hostalia, 2017).

En la Imagen 3 podemos ver mejor cómo funciona el algoritmo de MODSECURITY para detectar amenazas y de que se conforma.

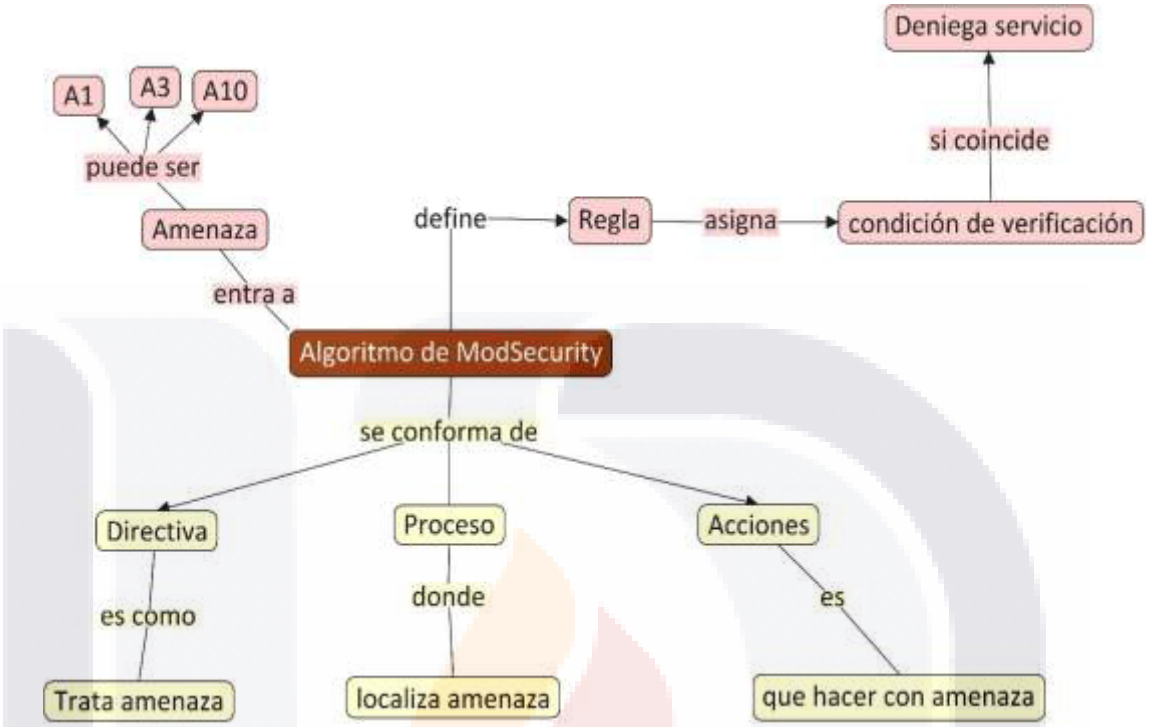


*Imagen 3. Funcionamiento de MODSECURITY*

## 3. METODOLOGÍA.

En la Imagen 4, se puede ver un diagrama del proceso de investigación, a continuación, detallaremos e incluiremos los resultados del proceso de investigación, la metodología encontrada para obtener las reglas de detección de amenazas del instituto, imágenes del reporte lanzado por la aplicación y la metodología ideal a la que se quiere llegar en un futuro.


*Imagen 4. Proceso de investigación*

El proceso de investigación consta de varias fases, listadas a continuación:

1. *Fase Inicial*. En esta fase incluye la investigación de los WAFs comerciales y la selección del WAF que se acopla a los requerimientos del instituto, ver Imagen 5.


*Imagen 5. Fase Inicial*

2. *Fase de Investigación.* En esta fase se expone el marco de trabajo establecido y la metodología usada, para obtener las reglas de detección de amenazas, ver Imagen 6.

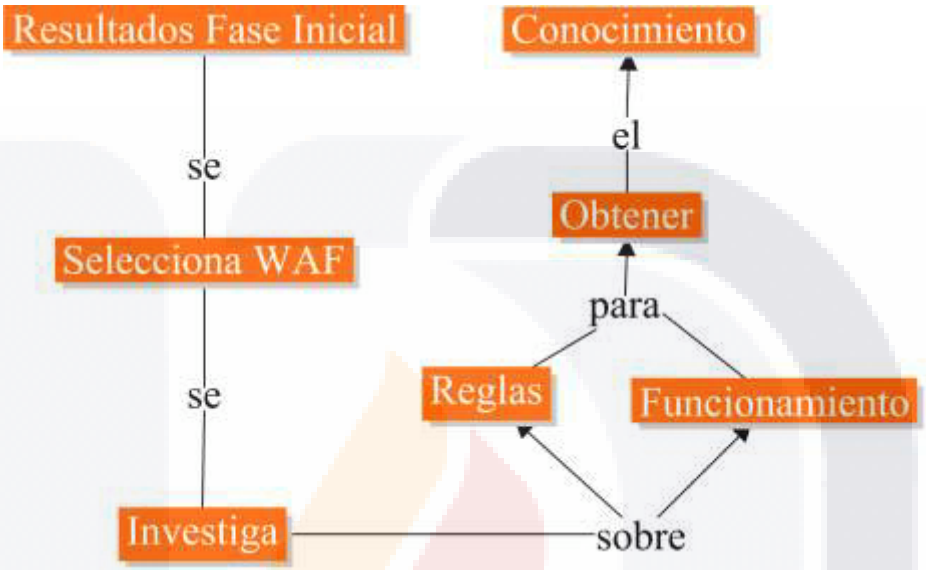


*Imagen 6. Fase de Investigación*

3. *Fase de Desarrollo.* En la cual se detalla con más claridad el desarrollo de la herramienta que permite crear las reglas de detección de amenazas que necesita el instituto ver Imagen 7.


*Imagen 7. Fase de Desarrollo*

4. *Fase Final.* En esta fase el reporte obtenido en la fase de desarrollo le permite al grupo de ingeniería en sistemas crear las reglas que podrán ser implementadas en el WAF implementado en INEGI, ver Imagen 8.


*Imagen 8. Fase Final*

## 3.1 SELECCIÓN DEL WAF.

En esta fase se pretende comparar los mejores WAFs comerciales existentes en el mercado y que cumplen con ciertos criterios como lo son: bajo costo, capacidad para personalizar reglas de configuración y que este certificado a nivel internacional; esto para poder elegir el que más se adecua al instituto.

En la Tabla 7 se observa que la mayoría de los WAFs tienen un costo, donde Qualys y Fortinet son de los más costosos por su tecnología usada, lamentablemente sólo están certificados en niveles de protección que no cubren los requisitos del instituto, pero son muy buenas opciones porque tienen aprendizaje de amenazas y están basados en el top ten de OWASP.

En el caso de Barracuda, este proveedor posee grandes características que los otros no tienen; su desventaja es en el costo y el tipo de certificado, en el anexo1 podemos ver los costos proporcionados por Barracuda, esto con índole informativa.

El mejor de todos es MODSECURITY, porque no tiene un costo elevado, es de código abierto y el certificado es generalizado. Por lo tanto, es una de las mejores opciones que puede tener el instituto para poder ser implementado.

*Tabla 7. Comparativo de características sobre WAFs comerciales*

| Características | Barracuda | Clodflare | Cisco ACE | SecureSphere | Qualys | Fortinet | ModSecurity |
|---|---|---|---|---|---|---|---|
| Presentación | Aplicación y Dispositivo | Aplicación | Dispositivo | Aplicación | Aplicación | Dispositivo | Aplicación |
| Escaner de vulnerabilidades y parcheo virtual agregado | X | | | X | X | X | X |
| Seguridad Web | X | X | X | X | X | X | X |
| Opciones de desarrollo | | | | X | | X | X |
| Autenticación | X | X | X | X | X | X | X |
| Administración y Reportes | X | | X | X | X | X | X |
| OWASP Top 10 | X | X | | X | X | X | X |
| Casos Exitosos en gobierno | | | | X USA | X COSTARICA | X MEX | |
| Configuración Personalizada de Reglas | | X | | X | X | | X |
| Protección a maquinas virtuales | X | | | X | X | X | X |
| Protección a la nube (azure) | X | | | X | X | X | X |
| Certificación | ICSA | PCI | PCI | ICSA | PCI | ICSA | OWASP |
| Aprendizaje de amenazas | | X | X | X | X | X | X |
| Dentro del cuadrante de Gartner | X | X | | X | | X | X |
| Costo | X | X | X | X | X | X | |
| Uso de crawler para identificación de amenazas | | X | | | | | |

**Fuente:** Elaboración propia

## 3.2 DISEÑO DE LA METODOLOGÍA.

En esta fase se pretende diseñar cómo será el proceso para encontrar las reglas de detección de amenazas dentro del instituto, se obtuvo dos diseños: el marco de trabajo que se debe de seguir dentro del instituto y la metodología usada para la detección de las reglas.

El marco de trabajo presentado en la Imagen 9, es una forma más general de cómo se debe de trabajar con la información obtenida por el laboratorio de pentesting y la relación que hay con el WAF usado en INEGI, la cual da pie a más investigaciones a futuro.

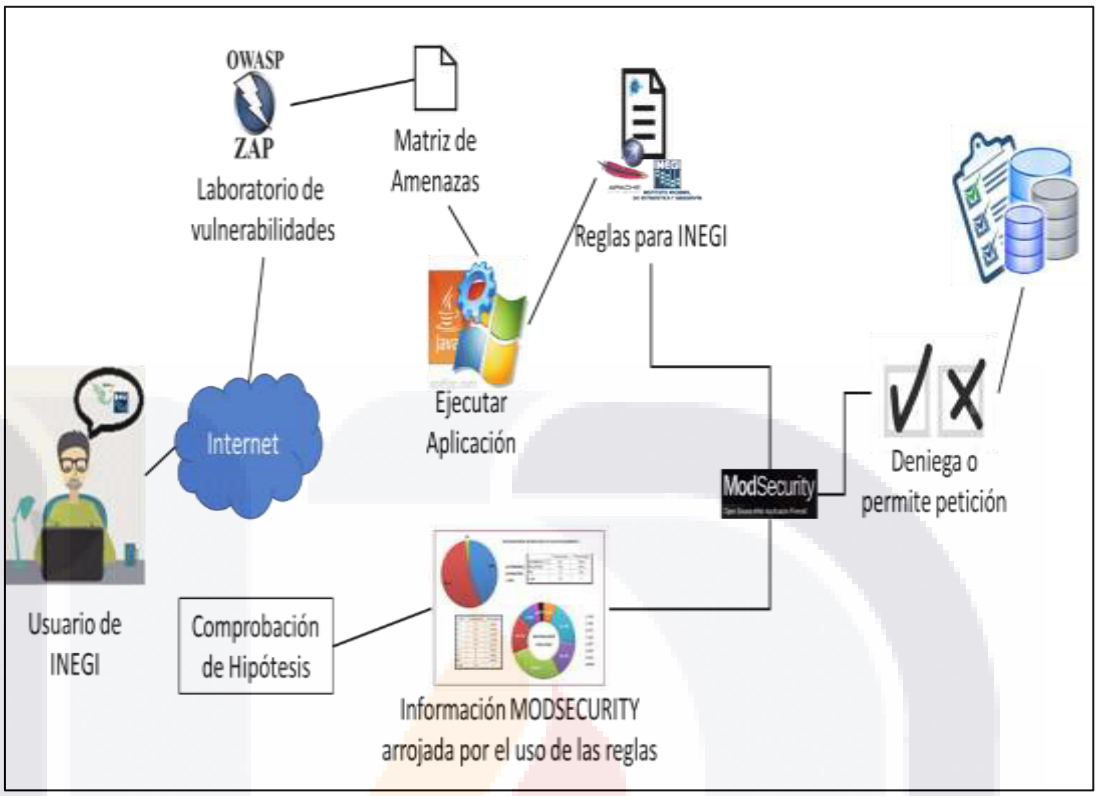


*Imagen 9. Metodología ideal de detección de reglas*

Para un mayor entendimiento, se investigaron los Cortafuegos de Aplicaciones Web existentes en el mercado, (aquellos con mayor renombre) y ubicados dentro del cuadrante de Gartner (Esquema realizado por Gartner para expresar las empresas o softwares mejores posicionados en el mercado), ya que el estar dentro del cuadrante indica una buena referencia para la selección de Cortafuegos de Aplicaciones Web. Además, se determinaron las características que pudieran ser relevantes para cada aplicación, y las innovaciones que ofrece cada solución.

La metodología usada para la creación de las reglas de detección de amenazas consiste en:

- *Procesamiento de insumos.* Consiste en procesar los insumos que se obtienen del laboratorio de vulnerabilidades o pentesting (aplicaciones que son utilizadas para obtener las vulnerabilidades de las páginas web). Este proceso involucra la creación de la matriz de amenazas, el cálculo de la ponderación y conteo de las amenazas que involucran a la url o url de cada proyecto. En la creación de la matriz de amenazas se verifica que existan las amenazas: A1-Inyección, A3-Sentencias de comandos en sitios cruzados y A10- APIs desprotegidas. Estas amenazas se seleccionaron por ser las que, con mayor probabilidad, pueden

atacar al instituto, además de ser un grupo de gran interés. Además, se hace el conteo de todas las amenazas que afectan a cada URL. Este procesamiento puede ser visualizado en la Imagen 10.

- Nota: Existen otras vulnerabilidades que son mencionadas en la Tabla 3, las cuales no son incluidas en la investigación como un tema de estudio, pero son importantes mencionar para el conocimiento del lector, así como pueden ser la base de trabajos futuros.

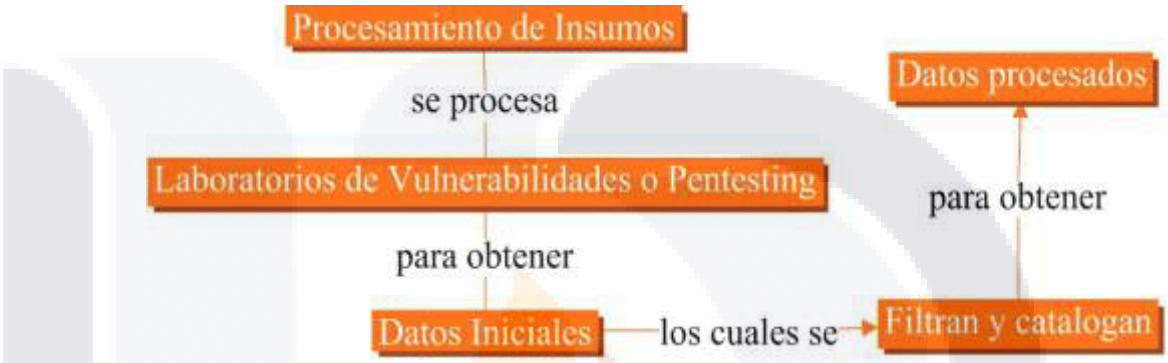


*Imagen 10. Procesamiento de Insumos*

- *Cálculos para la obtención de la matriz de amenazas.* El proceso permite obtener los cálculos necesarios para obtener la matriz de amenazas, los cuales son la obtención de la ponderación y el total de amenazas por url en cada proyecto. Lo anterior permite la clasificación de las amenazas de acuerdo con la ponderación de la clasificación resultante del cálculo. Ver Imagen 11.

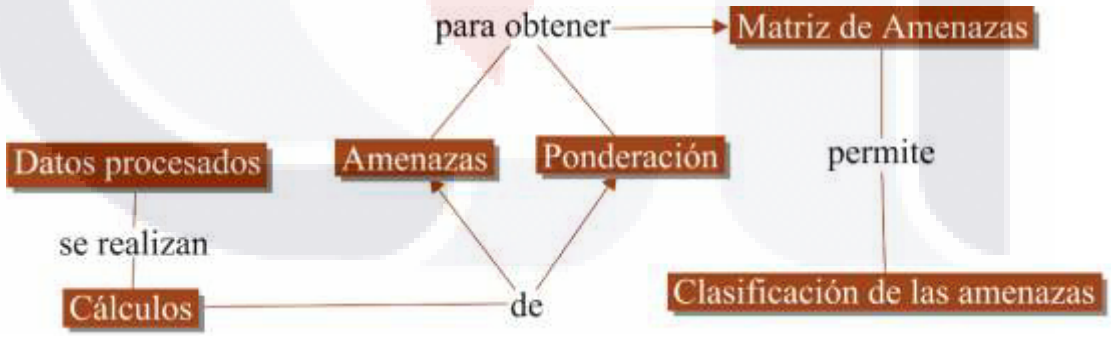


*Imagen 11. Cálculos para la obtención de la matriz de amenazas.*

- *Obtención de reglas de detección de amenazas. En* la que las reglas se visualizan en forma de reporte y pueden ser exportadas en PDF para su mejor comprensión, este reporte le sirve al Grupo de Ingeniería de Software para poder crear la regla o reglas que serán agregadas al archivo de configuración del MODSECURITY, este reporte es visualizado mediante la herramienta, la

cual ya obtiene las reglas detectadas con la información procesada, ver Imagen 12.

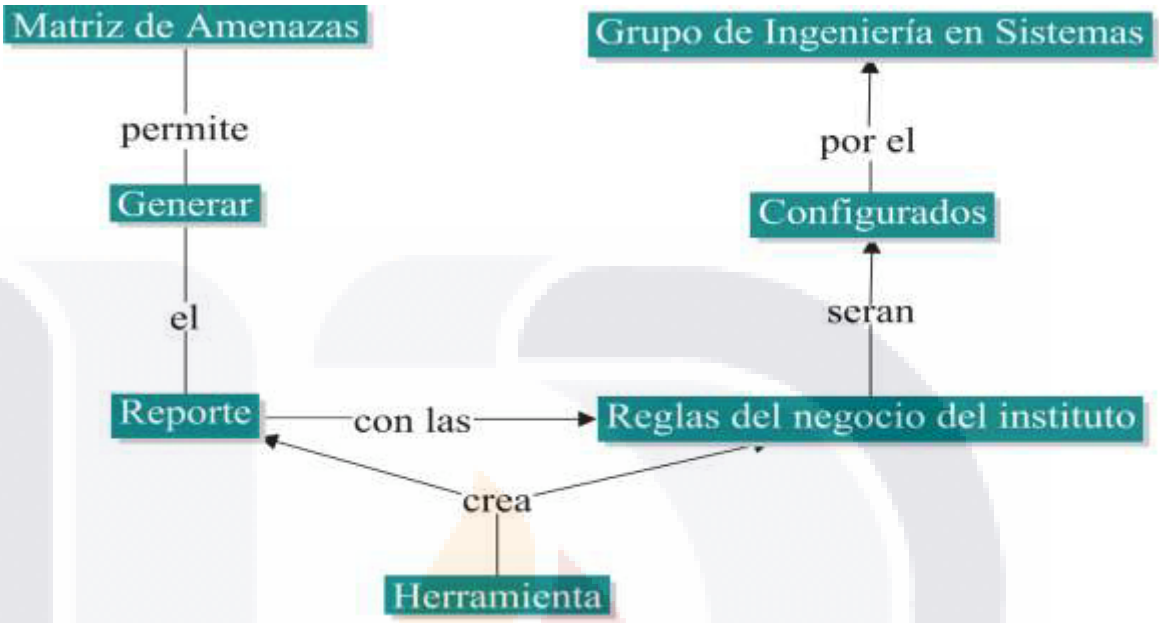


*Imagen 12. Obtención de las reglas de detección de amenazas*

## 3.3 DESARROLLO DE LA HERRAMIENTA.

### 3.3.1 Creación de Matriz de Amenazas.

Para la creación de la matriz de amenazas se contemplaron los datos obtenidos del laboratorio de vulnerabilidades simulados, proporcionando solamente los proyectos con los que cuenta actualmente el instituto, de estos proyectos se crea un catálogo filtrado de los datos en crudo de todos los proyectos y las urls proporcionados, obteniendo de este proceso solo valores únicos correspondiente a cada catálogo y asignando a cada uno de los catálogos un identificador único.

Este identificador creado es el que va en la matriz de amenazas tanto en la columna de IdProyecto e IdUrl respectivamente. Una vez realizado este proceso, se buscan las amenazas de interés como son: inyección, sentencias de comandos en sitios cruzados y APIs desprotegidas y se transforma a A1, A3 y A10 según sea el caso, esta información se filtra y se crean tres campos con A1, A3 y A10 para asignar 1 cuando encuentre cualquiera de las amenazas, en caso de que no se encuentre se asignaría 0, (se usa numeración binaria partiendo del código obtenido) validando también que sea de la misma url y proyecto, eliminando la repetida, y asegurando tener solo una url con las tres amenazas en la misma fila, ya sea que se encuentren o no, esto se hace con cada url hasta que termina de leer toda la información el archivo original, los campos del que consta la matriz de amenazas se puede ver en la Tabla 8.

*Tabla 8. Tabla de campos de matriz de amenazas*

| Nombre del Campo | Descripción |
|---|---|
| IdEmpresa | Identificador de la empresa. |
| Empresa | Empresa de donde vienen los proyectos a revisar. |
| IdProyecto | Identificador del proyecto. |
| Proyecto | Proyecto de la empresa en uso. |
| IdUrl | Identificador de la URL. |
| Url | Dirección Web que proporciona información relacionada con el proyecto manejado. |
| IdAmenaza | Identificador de Amenaza. |
| Amenaza | Nombre de la vulnerabilidad que afecta a la url |
| A1 | Amenaza 1: contiene 1 si detecta inyección y 0 si no. |
| A3 | Amenaza 3: Contiene 1 si detecta una vulnerabilidad de Sitio cruzado. |
| A10 | Amenaza 10: Contiene 1 si detecta una vulnerabilidad de API desprotegida. |
| Ponderacion | Es la sumatoria de las amenazas de $2^n$ si la amenaza es 1. (desde n=2 hasta 0) |
| Reprocesos | Es la sumatoria de las amenazas encontradas en esa URL. |

**Fuente:** Elaboración propia

En la Tabla 9 podemos observar un ejemplo de cómo sería la matriz de amenazas por proyecto y en la Tabla 10 la matriz de amenazas con los datos ya procesados por proyecto, pero sin el campo de ponderación ya que el proceso de obtención se describe más adelante.

*Tabla 9. Ejemplo de la matriz de amenazas por proyecto*

| Proyecto | Inyección (A1) | Referencia Cruzada (A3) | APIs desprotegidas (A10) |
|---|---|---|---|
| TIPO 1 | HIGH | MEDIUM | LOW |
| TIPO 2 | Sin Amenaza | Sin Amenaza | Sin Amenaza |
| TIPO 3 | Sin Amenaza | MEDIUM | LOW |
| TIPO 4 | Sin Amenaza | Sin Amenaza | LOW |
| TIPO 5 | Sin Amenaza | MEDIUM | LOW |

**Fuente:** Elaboración propia

Se tomó en cuenta solamente las vulnerabilidades de inyección, sentencia de comandos por referencia cruzada y APIs desprotegidas, debido a la gran cantidad de información que se tenía que considerar, así como el tiempo del que se disponía para la elaboración de la tesis.

De acuerdo con el grupo de ingeniería de software de INEGI las amenazas están clasificadas en HIGH, MEDIUM y LOW; donde HIGH son las amenazas con mayor

riesgo en las cuales entran las vulnerabilidades de inyección por ser uno de los más comunes dentro del instituto; MEDIUM en las que se localizan las vulnerabilidades de sentencia de comandos por referencia cruzada y LOW las vulnerabilidades de APIs desprotegidas. Estas tres vulnerabilidades las podemos visualizar dentro del TOP 10 de vulnerabilidades de OWASP, las cuales son las que más afectan a la mayoría de las empresas. Para ver un ejemplo de la clasificación antes mencionada, se puede consultar la Tabla 9.

*Tabla 10. Matriz de amenazas por proyectos sin campo de ponderación*

| IdEmpresa | IdProyecto | IdUrlProyecto | A1 | A3 | A10 |
|---|---|---|---|---|---|
| E1 | P12 | UP112 | 1 | 1 | 1 |
| E1 | P23 | UP123 | 1 | 1 | 1 |
| E1 | P1 | UP11 | 1 | 1 | 0 |
| E1 | P3 | UP13 | 1 | 1 | 0 |
| E1 | P5 | UP15 | 1 | 1 | 0 |
| E1 | P7 | UP17 | 1 | 1 | 0 |
| E1 | P9 | UP19 | 1 | 1 | 0 |
| E1 | P10 | UP110 | 1 | 1 | 0 |
| E1 | P11 | UP111 | 1 | 1 | 0 |
| E1 | P16 | UP116 | 1 | 1 | 0 |
| E1 | P17 | UP117 | 1 | 1 | 0 |
| E1 | P18 | UP118 | 1 | 1 | 0 |
| E1 | P19 | UP119 | 1 | 1 | 0 |
| E1 | P21 | UP121 | 1 | 1 | 0 |
| E1 | P25 | UP125 | 1 | 1 | 0 |
| E1 | P2 | UP12 | 1 | 0 | 0 |
| E1 | P4 | UP14 | 1 | 0 | 0 |
| E1 | P26 | UP126 | 1 | 0 | 0 |
| E1 | P8 | UP18 | 0 | 1 | 0 |
| E1 | P13 | UP113 | 0 | 1 | 0 |
| E1 | P15 | UP115 | 0 | 1 | 0 |
| E1 | P24 | UP124 | 0 | 1 | 0 |
| E1 | P28 | UP128 | 0 | 1 | 0 |
| E1 | P20 | UP120 | 0 | 0 | 1 |
| E1 | P29 | UP129 | 0 | 0 | 1 |
| E1 | P27 | UP127 | 1 | 0 | 0 |

**Fuente:** Elaboración propia

### 3.3.2 Cálculo de la ponderación en la matriz de amenazas

Una vez que se hayan creado los catálogos y la matriz de amenazas hasta el proceso de verificar el tipo de amenaza y asignarle el valor correspondiente se procede a hacer el cálculo de la ponderación de acuerdo con la amenaza o amenazas encontradas, esto se logra mediante la siguiente formula:

$$Ponderación = \sum 2^2 + 2^1 + 2^0$$

Donde:
$2^2$ representa la amenaza con mayor peligrosidad que afecta a INEGI
$2^1$ representa la amenaza con un nivel medio de peligrosidad, pero que de igual forma es relevante para la institución
$2^0$ representa la amenaza con menor peligrosidad que afecta a INEGI pero que es de vital importancia detectar.

Esto se realiza con el fin de poder clasificar y obtener un valor que nos permite verificar cuales amenazas son las más importantes y cuáles no y pueda ser presentada en la regla de detección de amenazas.

Cabe mencionar que se realizarían dos tablas una por proyecto y otra por url, en la Tabla 11 podemos ver cómo queda la matriz de amenazas por proyectos y en la Tabla 12 por urls.

*Tabla 11. Matriz de amenazas por proyectos final.*

| IdEmpresa | IdProyecto | IdUrlProyecto | A1 | A3 | A10 | Ponderacion |
|---|---|---|---|---|---|---|
| E1 | P12 | UP112 | 1 | 1 | 1 | 7 |
| E1 | P23 | UP123 | 1 | 1 | 1 | 7 |
| E1 | P1 | UP11 | 1 | 1 | 0 | 6 |
| E1 | P3 | UP13 | 1 | 1 | 0 | 6 |
| E1 | P5 | UP15 | 1 | 1 | 0 | 6 |
| E1 | P7 | UP17 | 1 | 1 | 0 | 6 |
| E1 | P9 | UP19 | 1 | 1 | 0 | 6 |
| E1 | P10 | UP110 | 1 | 1 | 0 | 6 |
| E1 | P11 | UP111 | 1 | 1 | 0 | 6 |
| E1 | P16 | UP116 | 1 | 1 | 0 | 6 |
| E1 | P17 | UP117 | 1 | 1 | 0 | 6 |
| E1 | P18 | UP118 | 1 | 1 | 0 | 6 |
| E1 | P19 | UP119 | 1 | 1 | 0 | 6 |
| E1 | P21 | UP121 | 1 | 1 | 0 | 6 |
| E1 | P25 | UP125 | 1 | 1 | 0 | 6 |
| E1 | P2 | UP12 | 1 | 0 | 0 | 4 |
| E1 | P4 | UP14 | 1 | 0 | 0 | 4 |
| E1 | P26 | UP126 | 1 | 0 | 0 | 4 |
| E1 | P8 | UP18 | 0 | 1 | 0 | 2 |
| E1 | P13 | UP113 | 0 | 1 | 0 | 2 |
| E1 | P15 | UP115 | 0 | 1 | 0 | 2 |
| E1 | P24 | UP124 | 0 | 1 | 0 | 2 |
| E1 | P28 | UP128 | 0 | 1 | 0 | 2 |
| E1 | P20 | UP120 | 0 | 0 | 1 | 1 |
| E1 | P29 | UP129 | 0 | 0 | 1 | 1 |
| E1 | P27 | UP127 | 1 | 0 | 0 | 4 |

**Fuente:** Elaboración propia

*Tabla 12. Matriz de amenazas por url*

| IdEmpresa | IdProyecto | IdUrl | A1 | A3 | A10 | Ponderacion | Reprocesos |
|---|---|---|---|---|---|---|---|
| E1 | P1 | U1 | 1 | 1 | 0 | 6 | 2 |
| E1 | P1 | U2 | 1 | 0 | 0 | 4 | 1 |
| E1 | P1 | U3 | 1 | 0 | 0 | 4 | 1 |
| E1 | P1 | U4 | 0 | 1 | 0 | 2 | 1 |
| E1 | P1 | U5 | 0 | 1 | 0 | 2 | 1 |
| E1 | P1 | U73 | 0 | 1 | 0 | 2 | 1 |
| E1 | P1 | U74 | 0 | 1 | 0 | 2 | 1 |
| E1 | P1 | U75 | 0 | 1 | 0 | 2 | 1 |
| E1 | P1 | U76 | 0 | 1 | 0 | 2 | 1 |
| E1 | P1 | U77 | 0 | 1 | 0 | 2 | 1 |
| E1 | P1 | U78 | 0 | 1 | 0 | 2 | 1 |
| E1 | P1 | U79 | 0 | 1 | 0 | 2 | 1 |
| E1 | P10 | U52 | 1 | 1 | 0 | 6 | 2 |
| E1 | P10 | U53 | 1 | 1 | 0 | 6 | 2 |
| E1 | P10 | U54 | 0 | 1 | 0 | 2 | 1 |
| E1 | P10 | U55 | 0 | 1 | 0 | 2 | 1 |
| E1 | P10 | U56 | 0 | 1 | 0 | 2 | 1 |
| E1 | P11 | U57 | 1 | 1 | 0 | 6 | 2 |
| E1 | P11 | U58 | 1 | 1 | 0 | 6 | 2 |
| E1 | P11 | U59 | 0 | 1 | 0 | 2 | 1 |
| E1 | P12 | U61 | 0 | 1 | 0 | 2 | 1 |
| E1 | P12 | U62 | 0 | 1 | 0 | 2 | 1 |
| E1 | P12 | U63 | 1 | 1 | 0 | 6 | 2 |
| E1 | P12 | U64 | 1 | 0 | 0 | 4 | 1 |
| E1 | P12 | U65 | 0 | 0 | 1 | 1 | 1 |
| E1 | P12 | U66 | 1 | 0 | 1 | 5 | 2 |
| E1 | P12 | U67 | 1 | 0 | 0 | 4 | 1 |
| E1 | P12 | U68 | 0 | 1 | 0 | 2 | 1 |
| E1 | P12 | U69 | 0 | 0 | 1 | 1 | 1 |
| E1 | P12 | U70 | 0 | 0 | 1 | 1 | 1 |
| E1 | P12 | U71 | 0 | 0 | 1 | 1 | 1 |
| E1 | P12 | U72 | 0 | 0 | 1 | 1 | 1 |
| E1 | P13 | U85 | 0 | 1 | 0 | 2 | 1 |
| E1 | P13 | U86 | 0 | 1 | 0 | 2 | 1 |
| E1 | P13 | U87 | 0 | 1 | 0 | 2 | 1 |
| E1 | P13 | U88 | 0 | 1 | 0 | 2 | 1 |
| E1 | P13 | U89 | 0 | 1 | 0 | 2 | 1 |
| E1 | P13 | U90 | 0 | 1 | 0 | 2 | 1 |
| E1 | P13 | U91 | 0 | 1 | 0 | 2 | 1 |
| E1 | P13 | U92 | 0 | 1 | 0 | 2 | 1 |
| E1 | P15 | U100 | 0 | 1 | 0 | 2 | 1 |
| E1 | P15 | U101 | 0 | 1 | 0 | 2 | 1 |
| E1 | P15 | U102 | 0 | 1 | 0 | 2 | 1 |
| E1 | P15 | U103 | 0 | 1 | 0 | 2 | 1 |
| E1 | P15 | U104 | 0 | 1 | 0 | 2 | 1 |
| E1 | P15 | U98 | 0 | 1 | 0 | 2 | 1 |
| E1 | P15 | U99 | 0 | 1 | 0 | 2 | 1 |
| E1 | P16 | U105 | 1 | 0 | 0 | 4 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| E1 | P16 | U105 | 0 | 1 | 0 | 2 | 1 |
| E1 | P16 | U106 | 0 | 1 | 0 | 2 | 1 |
| E1 | P16 | U107 | 0 | 1 | 0 | 2 | 1 |
| E1 | P16 | U108 | 0 | 1 | 0 | 2 | 1 |
| E1 | P17 | U109 | 1 | 0 | 0 | 4 | 1 |
| E1 | P17 | U110 | 0 | 1 | 0 | 2 | 1 |
| E1 | P17 | U6 | 1 | 0 | 0 | 4 | 1 |
| E1 | P18 | U129 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U130 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U131 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U132 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U133 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U134 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U135 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U136 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U137 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U138 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U139 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U140 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U141 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U142 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U143 | 0 | 1 | 0 | 2 | 1 |
| E1 | P18 | U144 | 1 | 0 | 0 | 4 | 1 |
| E1 | P19 | U145 | 1 | 0 | 0 | 4 | 1 |
| E1 | P19 | U146 | 0 | 1 | 0 | 2 | 1 |
| E1 | P2 | U10 | 1 | 0 | 0 | 4 | 1 |
| E1 | P2 | U11 | 1 | 0 | 0 | 4 | 1 |
| E1 | P2 | U6 | 1 | 0 | 0 | 4 | 1 |
| E1 | P2 | U7 | 1 | 0 | 0 | 4 | 1 |
| E1 | P2 | U8 | 1 | 0 | 0 | 4 | 1 |
| E1 | P2 | U9 | 1 | 0 | 0 | 4 | 1 |
| E1 | P20 | U147 | 0 | 0 | 1 | 1 | 1 |
| E1 | P21 | U150 | 0 | 1 | 0 | 2 | 1 |
| E1 | P21 | U151 | 1 | 0 | 0 | 4 | 1 |
| E1 | P21 | U152 | 1 | 0 | 0 | 4 | 1 |
| E1 | P23 | U154 | 0 | 0 | 1 | 1 | 1 |
| E1 | P23 | U155 | 0 | 1 | 0 | 2 | 1 |
| E1 | P23 | U156 | 0 | 1 | 0 | 2 | 1 |
| E1 | P23 | U157 | 1 | 0 | 0 | 4 | 1 |
| E1 | P23 | U158 | 1 | 0 | 0 | 4 | 1 |
| E1 | P23 | U159 | 1 | 0 | 0 | 4 | 1 |
| E1 | P23 | U160 | 1 | 0 | 0 | 4 | 1 |
| E1 | P24 | U161 | 0 | 1 | 0 | 2 | 1 |
| E1 | P25 | U162 | 1 | 0 | 0 | 4 | 1 |
| E1 | P25 | U163 | 1 | 0 | 0 | 4 | 1 |
| E1 | P25 | U164 | 0 | 1 | 0 | 2 | 1 |
| E1 | P26 | U165 | 1 | 0 | 0 | 4 | 1 |
| E1 | P26 | U166 | 1 | 0 | 0 | 4 | 1 |
| E1 | P26 | U167 | 1 | 0 | 0 | 4 | 1 |
| E1 | P26 | U168 | 1 | 0 | 0 | 4 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| E1 | P26 | U169 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U170 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U171 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U172 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U173 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U174 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U175 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U176 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U177 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U178 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U179 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U180 | 1 | 0 | 0 | 4 | 1 |
| E1 | P27 | U181 | 1 | 0 | 0 | 4 | 1 |
| E1 | P28 | U182 | 0 | 1 | 0 | 2 | 1 |
| E1 | P29 | U183 | 0 | 0 | 1 | 1 | 1 |
| E1 | P3 | U12 | 1 | 0 | 0 | 4 | 1 |
| E1 | P3 | U80 | 0 | 1 | 0 | 2 | 1 |
| E1 | P3 | U81 | 0 | 1 | 0 | 2 | 1 |
| E1 | P3 | U82 | 0 | 1 | 0 | 2 | 1 |
| E1 | P3 | U83 | 0 | 1 | 0 | 2 | 1 |
| E1 | P3 | U84 | 0 | 1 | 0 | 2 | 1 |
| E1 | P4 | U13 | 1 | 0 | 0 | 4 | 1 |
| E1 | P4 | U14 | 1 | 0 | 0 | 4 | 1 |
| E1 | P5 | U15 | 1 | 1 | 0 | 6 | 2 |
| E1 | P5 | U16 | 0 | 1 | 0 | 2 | 1 |
| E1 | P5 | U17 | 0 | 1 | 0 | 2 | 1 |
| E1 | P7 | U24 | 1 | 0 | 0 | 4 | 1 |
| E1 | P7 | U25 | 0 | 1 | 0 | 2 | 1 |
| E1 | P7 | U26 | 0 | 1 | 0 | 2 | 1 |
| E1 | P8 | U27 | 0 | 1 | 0 | 2 | 1 |
| E1 | P8 | U28 | 0 | 1 | 0 | 2 | 1 |
| E1 | P8 | U29 | 0 | 1 | 0 | 2 | 1 |
| E1 | P8 | U30 | 0 | 1 | 0 | 2 | 1 |
| E1 | P8 | U31 | 0 | 1 | 0 | 2 | 1 |
| E1 | P8 | U32 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U33 | 1 | 1 | 0 | 6 | 2 |
| E1 | P9 | U34 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U35 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U36 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U37 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U38 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U39 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U40 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U41 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U42 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U43 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U44 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U45 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U46 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U47 | 0 | 1 | 0 | 2 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| E1 | P9 | U48 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U49 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U50 | 0 | 1 | 0 | 2 | 1 |
| E1 | P9 | U51 | 0 | 1 | 0 | 2 | 1 |

**Fuente:** Elaboración propia

### 3.3.3 Visualización de las reglas de detección de amenazas.

Finalmente, una vez construido el sistema, se procede a realizar la visualización de las amenazas en el cual se elegirá el proyecto y la url para poder visualizar la regla del negocio que procede, como se muestra en la Imagen 13.

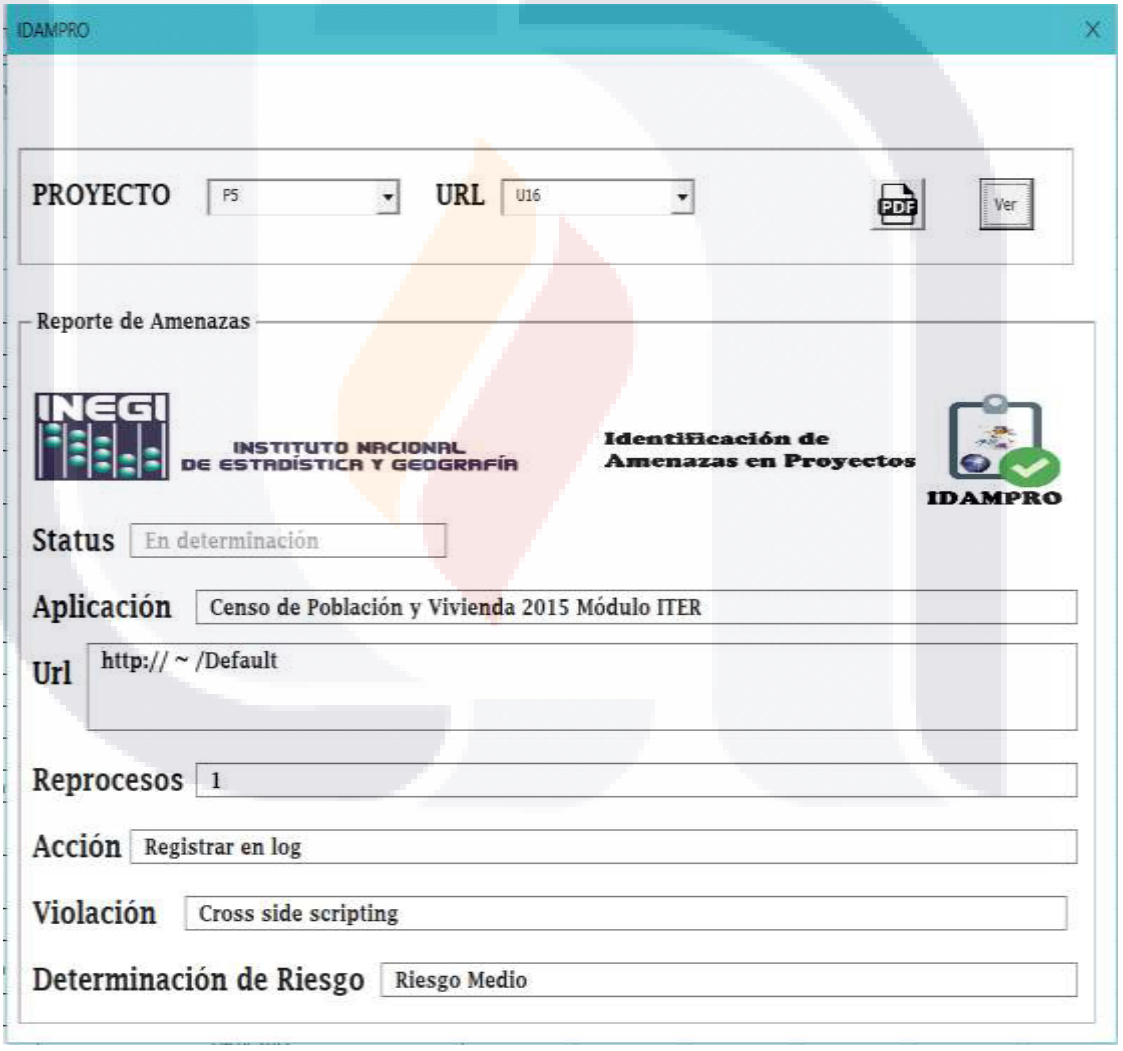


*Imagen 13. Pantalla de visualización de las reglas de detección de amenazas*

# 4. RESULTADOS.

Durante la investigación se encontraron tres tipos de resultados que ataca el problema que tiene actualmente INEGI. Los cuales contribuyen a mejorar y automatizar la obtención de reglas que permiten identificar las amenazas detectadas en los proyectos dentro del instituto. Los resultados se mencionan a continuación:

1. Metodología de determinación de reglas
2. Sistema automatizado
3. Reglas de detección de amenazas

## 4.1 METODOLOGÍA DE DETERMINACIÓN DE REGLAS.

En la Imagen 10 se puede observar con mejor detalle el proceso que se tiene que llevar para poder crear las reglas que permiten identificar amenazas en INEGI, el aplicar el proceso nos permite establecer un procedimiento para obtener las reglas que son realizadas artesanalmente por el Grupo de Ingeniería en Sistemas del instituto.

Con la metodología de la Imagen 14, se puede cubrir uno de los puntos que habla sobre procesos no definidos, ya que la metodología define a detalle cómo obtener las reglas de identificación de amenazas que INEGI necesita.

La Imagen 14 define las vulnerabilidades encontradas por el laboratorio de pentesting o laboratorio de vulnerabilidades, éstas serán el insumo de entrada o datos iniciales que permite la creación de la matriz de amenazas, que es detallada con claridad en el capítulo 3. Esta matriz tiene el compendio por reglas, que fue de gran importancia para la creación de reglas de detección de amenazas que es visualizado en forma de reporte y analizado por el grupo de Ingeniería en Sistemas del INEGI, de manera que le permita al grupo crear las reglas que alimenten al MODSECURITY.


*Imagen 14. Metodología realizada para detección de reglas*

## 4.2 SISTEMA AUTOMATIZADO.

El sistema permite ingresar un archivo matricial tipo Excel que contiene todos los datos iniciales obtenidos del laboratorio de vulnerabilidades o del laboratorio de pentesting. Esta información obtenida es el insumo que se necesita para crear las reglas que permitan identificar las amenazas dentro de los proyectos; para esto, se debe crear la matriz de amenazas mencionada en las tablas 11 y 12 respectivamente.

El sistema cuenta con dos secciones visibles al usuario final, es decir: sólo se ven dos pantallas, mientras que los cálculos y clasificación se realiza de manera interna.

El usuario visualiza la Imagen 15 al arrancar el sistema. En la caja de texto se visualiza el archivo seleccionado y el botón con los tres puntos abre un cuadro de dialogo donde aparecerá la estructura de directorios que existe en la maquina; este cuadro de dialogo permitirá sólo abrir archivos .xlsx, este cuadro se visualiza en la Imagen 16.

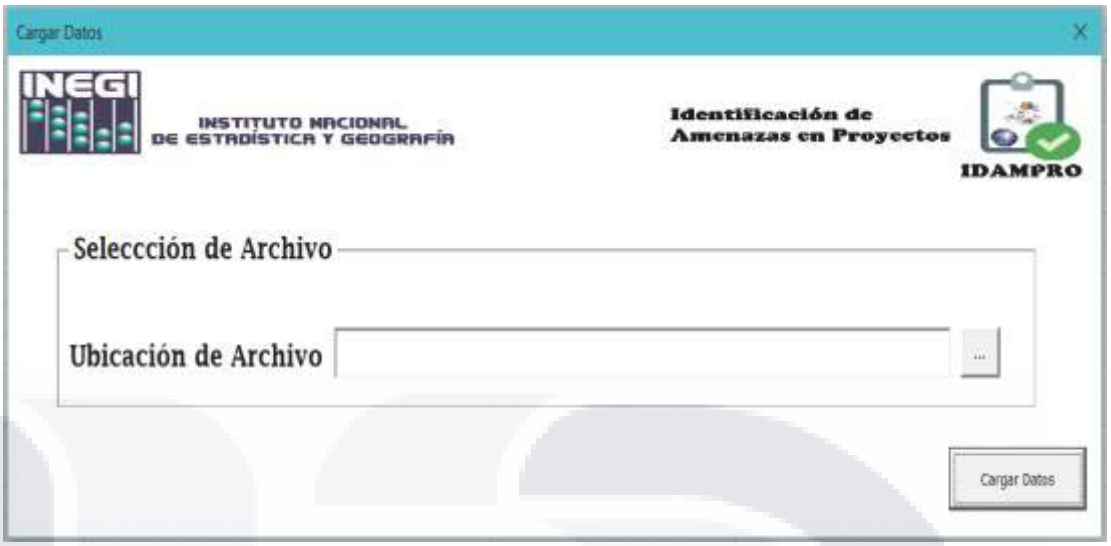


*Imagen 15. Pantalla de inicio al sistema*

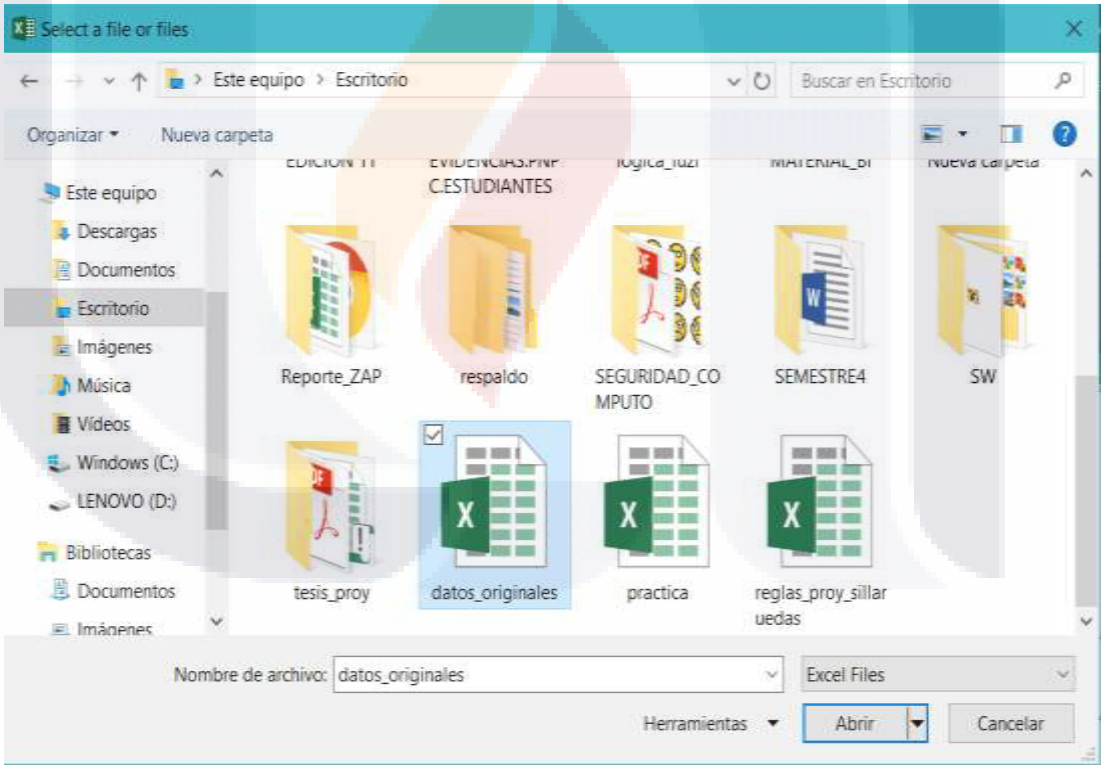


*Imagen 16. Cuadro de dialogo abrir*

Una vez seleccionado el archivo, se procede a hacer clic en el botón aceptar y de manera inmediata el sistema arroja la dirección completa del archivo en la caja de texto, como se ve en la Imagen 17.


*Imagen 17. Pantalla principal con la dirección del archivo*

Con el botón "Cargar Datos", el sistema realiza una serie de tareas de manera invisible para el usuario como lo es la creación de los catálogos, el conteo de amenazas y el cálculo de la ponderación; esto para generar la matriz de amenazas visualizadas en la Tabla 11 y Tabla 12 del capítulo 3. De esta forma, el sistema presenta la regla detectada en una determinada url y determinado proyecto.

Una vez realizado este proceso, se visualiza la pantalla donde se podrá elegir el tipo de proyecto y las urls asociadas a este proyecto para poder así identificar la regla como se visualiza en la Imagen 18.

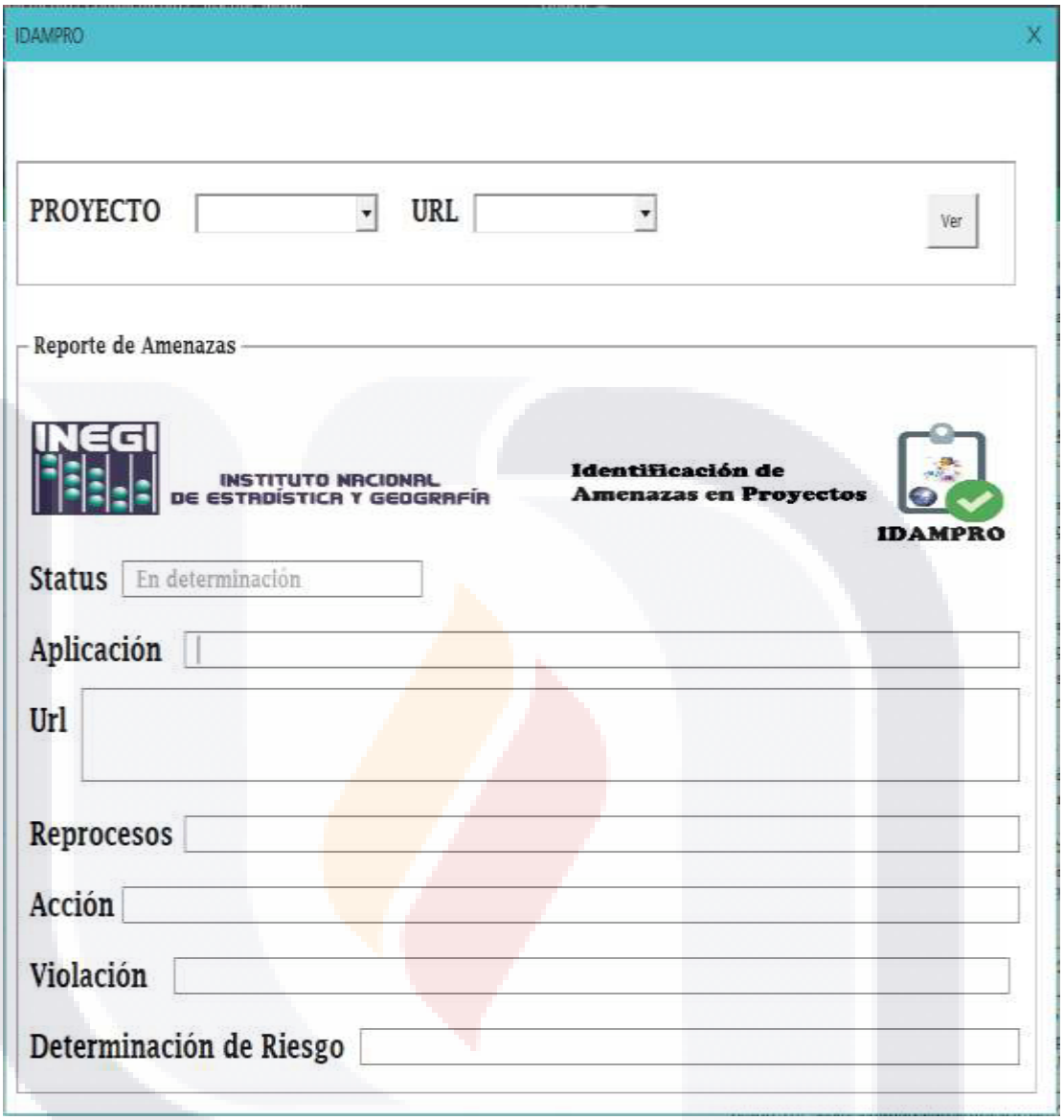


*Imagen 18. Ventana de visualización de Reportes*

En la Imagen 18 aparecen 2 cajas desplegables que tienen los identificadores de los proyectos y los identificadores de las urls con amenazas. Una vez que se selecciona el proyecto y la url, se hace clic en el botón "Ver", y esta acción llena toda la información que requiere el reporte, tal y como se muestra en la Imagen 19.
Finalmente, si el Grupo de Ingeniería de INEGI puede guardar la regla identificada haciendo clic en el botón que tiene el ícono de PDF. Esta acción provoca que se abra de manera automática el lector de PDF con él reporte.

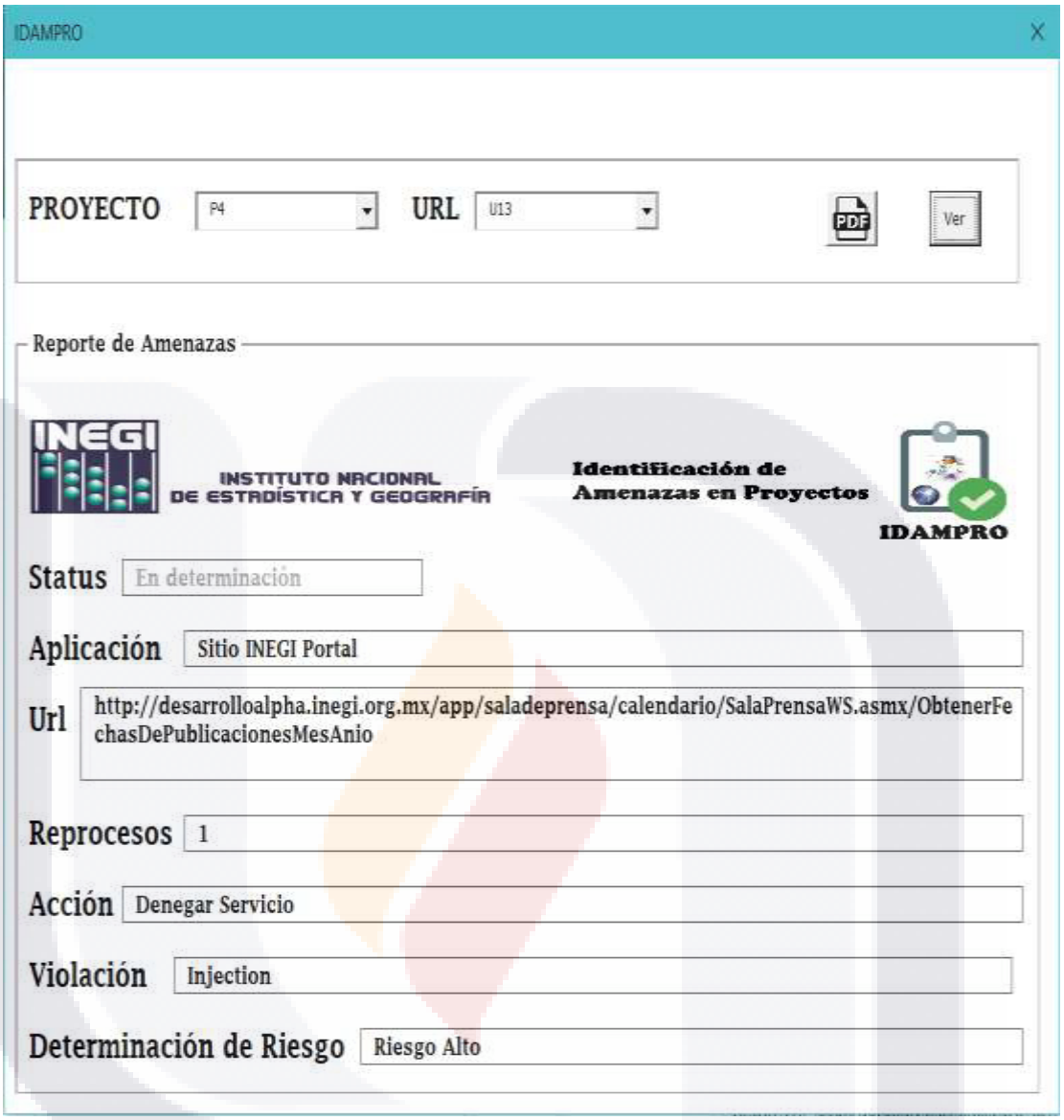


*Imagen 19. Visualización del reporte con la regla detectada*

## 4.3 REGLAS DE DETECCIÓN DE AMENAZAS

Estos resultados son de los más importantes, debido a que son las reglas que se obtienen de la información proporcionada por el laboratorio de pentesting o de vulnerabilidades y que permite adecuarlas al MODSECURITY por el encargado de crear el archivo de configuración que va a alimentar al WAF implementado en INEGI.

De acuerdo con los resultados obtenidos del WAF, se observa que hay más proyectos con Amenazas de tipo de sentencias de comandos en sitios cruzados con 99 urls

detectadas que la incluyen, así como solamente 9 urls que incluyen la amenaza de APIs desprotegidas y 53 urls que se detectaron de tipo inyección. Esto quiere decir que las amenazas más comunes serían las de tipo Inyección y las de sentencias de comandos en sitios cruzados.

Aunado a esto, se observa también que sólo 6 proyectos constan de la combinación de 2 amenazas detectados en una sola url que le corresponde al proyecto en la Tabla 13. En esta tabla, podemos visualizar este tipo de resultado notando que tres proyectos son de los más graves porque contienen en dos de sus urls más de una amenaza detectada.

*Tabla 13. Proyectos con mayor caso de amenazas detectadas*

| IdEmpresa | IdProyecto | Proyecto | IdUrl | A1 | A3 | A10 | Ponderacion | Reprocesos |
|---|---|---|---|---|---|---|---|---|
| E1 | P1 | Encuesta Intercensal Módulo Codificación 2015 | U1 | 1 | 1 | 0 | 6 | 2 |
| E1 | P10 | Encuestas Económicas Grupo SIEUE | U52 | 1 | 1 | 0 | 6 | 2 |
| E1 | P10 | | U53 | 1 | 1 | 0 | 6 | 2 |
| E1 | P11 | INEGI - ERP | U57 | 1 | 1 | 0 | 6 | 2 |
| E1 | P11 | | U58 | 1 | 1 | 0 | 6 | 2 |
| E1 | P12 | INEGI – Marco Nacional de Viviendas | U63 | 1 | 1 | 0 | 6 | 2 |
| E1 | P12 | | U66 | 1 | 0 | 1 | 5 | 2 |
| E1 | P5 | Censo de Población y Vivienda 2015 Módulo ITER | U15 | 1 | 1 | 0 | 6 | 2 |
| E1 | P9 | Sitio INEGI Servicio Profesional de Carrera | U33 | 1 | 1 | 0 | 6 | 2 |

**Fuente:** Elaboración propia

Para que sea más claro la relación que hay entre la matriz de amenazas y las reglas identificadas que se obtienen, la Imagen 20 muestra un proyecto con dos amenazas identificadas y la Imagen 21 con una amenaza identificada.


*Imagen 20. Regla identificando 2 amenazas*

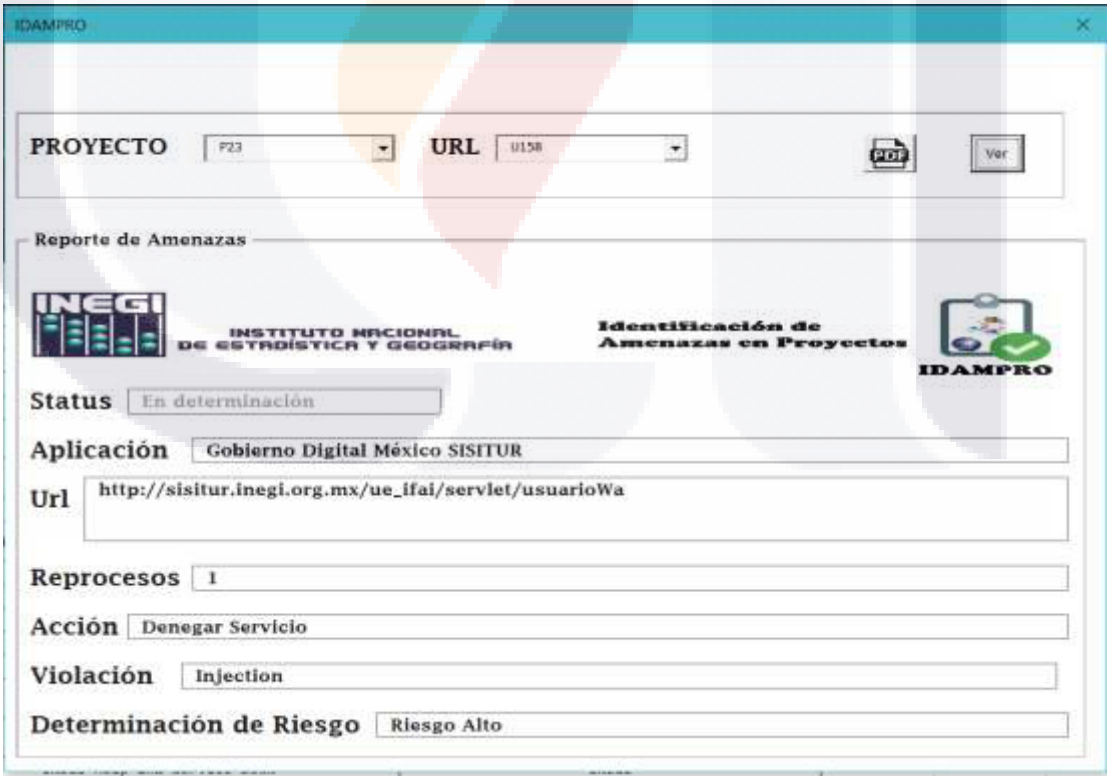


*Imagen 21. Regla con una amenaza identificada*

Cada campo representado en el reporte lanzado en el sistema se explica a detalle en la Tabla 14.

*Tabla 14. Descripción de campos utilizados en el reporte*

| Campo del reporte | Descripción |
|---|---|
| Status | Solo tiene en determinación ya que solamente se detecta y el GIS de INEGI es el que decide implementarla o no y si se implementa cambiaría a implementada. |
| Aplicación | Contiene el nombre del proyecto en el que se detecta la amenaza, este valor es obtenido del campo proyecto dependiendo del identificador seleccionado previamente. |
| Url | Contiene la página en donde se localiza la amenaza. |
| Reprocesos | Es el total de amenazas detectadas en esa página, este valor es obtenido del campo de reprocesos. |
| Acción | Es el tipo de acción que va a tomar el GIS de INEGI en caso de implementarse. |
| Violación | Es el tipo de amenaza detectada, es decir puede tener Inyección, Secuencia de comandos en sitios cruzados o APIs desprotegidas o una combinación de ellas, este valor es obtenido dependiendo si es 1 o 0 de los campos A1, A3 o A10. |
| Determinación de Riesgo | Es el tipo de peligrosidad que representa la página y este es obtenido de acuerdo con el campo de ponderación donde la ponderación arriba de 4 es catalogada como de Riesgo alto, la ponderación menor a 4 y mayor de 1 es catalogado como de Riesgo medio, los de ponderación 1 como Riesgo bajo y con 0 Sin riesgo. |

**Fuente:** Elaboración propia

Esta información reunida es la regla identificada para cada uno de los proyectos y urls que el sistema analiza de manera automática.

## 4.4 DISCUSIÓN DE RESULTADOS OBTENIDOS RESPECTO A TRABAJOS RELACIONADOS.

Los trabajos relacionados con respecto a la creación de reglas de detección de amenazas, tal y como lo indica (Torrano Giménez, 2015) en su estudio de técnicas estocásticas y máquinas de aprendizaje para detección de anomalías basadas en ataques web; nos indica que las reglas se pueden adecuar de acuerdo con el proceso sin emplear métodos sofisticados para cumplir el propósito, sin embargo, el usar algoritmos complejos o métodos estadísticos nos permitirá considerar futuras amenazas y cambiar el entorno.

## 4.5 BENEFICIOS OBTENIDOS.

Dentro de los beneficios que se obtiene es:

1. Establecer una metodología que le permite al instituto poder identificar las reglas de detección de amenazas.
2. El desarrollo de un mecanismo que podrá ser implementado por el instituto en caso de que así sea deseado.
3. Ayudar al equipo de ingeniería en sistemas en la identificación de reglas de detección de amenazas.
4. Justificar el uso de herramientas de código abierto, disminuyendo así los costos elevados de herramientas comerciales, permitiendo mediante las reglas detectadas mayor cobertura de amenazas y menos pérdida de tiempo en el análisis de las reglas.

## 4.6 PROBLEMAS ENCONTRADOS.

Durante la investigación los problemas que se detectan son:

- Datos simulados debido a la seguridad que presenta el instituto. Esto provoca que la información manejada no sea 100% verídica.
- El tiempo tomado para recabar los datos es muy largo. No se permite establecer mejoras en el diseño y detección de las reglas de detección.

## 4.7 RECOMENDACIONES PARA FUTUROS CASOS SIMILARES.

Se recomienda obtener datos 100% reales para poder así aplicar una metaheurística que permita alimentarse de las vulnerabilidades existentes en el instituto y que sea posible crear al vuelo la regla de detección de amenazas que a nivel de código el MODSECURITY o el WAF utilizado pueda comprender y mejorar la detección de amenazas a través de las reglas creadas.

# 5. CONCLUSIONES.

De acuerdo con la investigación realizada, se puede concluir que los objetivos generales y específicos planteados en el documento de tesis, se alcanzaron, ya que se logró obtener una metodología necesaria para la identificación de las reglas de detección de amenazas que el INEGI necesitaba personalizar.
Además, se logró obtener una herramienta la cual crea las reglas que necesita el WAF para ser configurado por el Grupo de Ingeniería en Sistemas de INEGI o GIS, estas reglas permitirán al grupo observar cuales son las amenazas que más le afecta a la empresa y cuales deberán de implementar en su WAF.

La identificación se logró mediante el marco de trabajo detectado que nos permite visualizar mejor el proceso, amenazas y reglas necesarias que ayudan al GIS a la implementación de las reglas en el WAF del INEGI.

Con los resultados obtenidos, se comprueba la factibilidad con respecto de las preguntas de la creación de reglas de detección de amenazas, el establecimiento de un marco de trabajo para el establecimiento de las reglas de detección de amenazas y de que fue factible la identificación de los factores de riesgos.

Podemos también concluir que la hipótesis 1 se comprobó debido al desarrollo de una herramienta la cual permite recibir de un archivo de Excel las vulnerabilidades arrojadas por el laboratorio de pentesting y así crear los catálogos y hacer los cálculos necesarios para obtener la matriz de amenazas y así visualizar el reporte de acuerdo con las reglas encontradas.

En cuanto a la hipótesis 2 sobre la identificación de factores los cuales permitirán el manejo de las reglas de detección de amenazas de INEGI; también se comprobó debido a que se encontró una relación muy importante que permite la fácil identificación de la regla gracias a la clasificación asignada acorde a la vulnerabilidad, logrando así el éxito de la hipótesis.

En cuanto a la hipótesis 3 ubicado en el apartado 1.6, también se comprobó debido a que la metodología usada permitió la creación de un marco de trabajo robusto, flexible y claro como producto de la presente investigación, la cual permitió crear un proceso muy fácil de comprender e implementar, logrando así la resolución a las preguntas planteadas en el documento alcanzando cada uno de los objetivos establecidos.

Con la creación de la herramienta y la metodología desarrollada se logra optimizar tiempos de creación de reglas de detección de amenazas que son necesarias para el WAF usado en la empresa, ya que el secreto de un buen cortafuego de aplicaciones Web es basado principalmente en las reglas de detección de amenazas. Ya que si una

regla no es bien creada el cortafuego pierde su importancia y no logra la protección total y es en vano su uso.

Finalmente se concluye que la herramienta realizada permite al GIS de INEGI ahorrar tiempos, esto se refiere al tiempo invertido para analizar y crear la regla, para así poder analizar y auditar mejor las reglas de detección creadas, debido a que es muy poco el personal capacitado para esta tarea.

Cabe mencionar que existen varios campos en los cuales se puede incursionar en investigaciones subsecuentes, sin embargo, aunque el tiempo para desarrollar este trabajo de investigación estuvo acotado, consideramos que los alcances de la misma fueron muy convenientes con resultados tangibles para el INEGI. Como ya se mencionó se cuenta con otras áreas de oportunidad que se pudieron develar con este trabajo y que son de vital importancia para la empresa.

Otra cosa que se debe de mencionar en esta investigación es la utilización de la versión Top 10 de OWASP de antes de finales de noviembre del 2017 por el tiempo escolarizado de la maestría que culminó en diciembre de 2017. Debido a su actualización a finales de noviembre de 2017, el lector interesado encontrará leves diferencias modificadas en su actualización para 2018. Por tal motivo, lo que las vulnerabilidades mencionadas en esta investigación no coincidirán con el Top 10 actual visualizados en la página web de OWASP. Estas vulnerabilidades se pueden ver más a detalle en el anexo 1 del presente documento.

Los avances y descubrimientos encontrados durante la elaboración de esta investigación ayudaron al mejoramiento de nuevas herramientas y al entendimiento de los Cortafuegos de Aplicaciones Web y la importancia que tiene su uso, aunque es necesario no sólo implementar el uso de un WAF en una empresa, sino que también es importante usar diferentes herramientas de seguridad para que una empresa este segura de cualquier amenaza, pero sobre todo que la gente esté informada de lo que es seguridad de la información, seguridad informática y las diferentes vulnerabilidades que pueden afectar a nivel mundial a cualquier empresa o persona que no tiene un conocimiento sobre el tema. El difundir una cultura de seguridad de las personas permitirá a las personas salvaguardar su información y verse menos afectada por dichas vulnerabilidades.

## 5.1 TRABAJOS FUTUROS.

Para futuras investigaciones relacionadas con el tema debido a que solo aborda una parte de la gran cantidad de información que hay relacionado con esta investigación, lo que se puede hacer es ampliar las amenazas detectadas, incluyendo todo el Top 10 de OWASP.

También se puede desarrollar mediante métodos heurísticos y estadísticos mejores herramientas que permitan no solo detectar lo que se encuentra en el top 10 de OWASP, sino que también futuras amenazas no cubiertas por este organismo, así como crear automáticamente los archivos de configuración necesarios para el MODSECURITY.

Así como automatizar todo el proceso de elaboración de la matriz de amenazas, permitiendo a los futuros investigadores emplear nuevas metodologías de aprendizaje optimizando mediante diversas herramientas de inteligencia artificial o ciencia de datos dicho proceso.

# GLOSARIO

**Amenaza.** Es un riesgo que puede afectar un objetivo en particular, puede ser un objeto, una persona, información, entre otras.

**Amenazas naturales.** Son aquellas que son causadas por desastres naturales como incendios, huracanes, inundaciones. Esto puede pasar en cualquier momento, pero hay que tomarlos en cuenta para prevenir perdidas costosas.

**Amenazas provocadas por el hombre.** Son aquellas que el hombre realiza para su beneficio. En el mundo de la computación estas amenazas buscan la obtención de la información de manera ilegal ya sea por juego, venganza o un beneficio económico.

**Aplicaciones web**. Son programas realizados para verse a través de internet o intranet por medio de navegadores como IExplorer, Chrome, entre otros.

**Arboles de decisiones.** Son técnicas que permite el análisis de decisiones secuenciales basados en resultados y probabilidades asociadas.

**Atacante.** dícese de las personas que quieren hacer mal uso de la información.

**Ataque ddos.** Ataques distribuidos de denegación de servicios utiliza varios dispositivos infectados por malware para enviar grandes cantidades de tráfico a una aplicación o sitio web.

**Ataques día cero,** Son ataques realizados contra un sistema operativo o una aplicación aprovechando las vulnerabilidades que tiene el producto.

**API.** Es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software.

**APIs desprotegidas.** Las aplicaciones actuales involucran la mayoría de las veces aplicaciones al cliente enriquecidas, así como APIs, ejemplo JavaScript en el navegador y aplicaciones móviles, que conectan a un API u otro (SOAP/XML, REST/JSON, RPC, GWT, entre otros.). Estas APIs son la mayoría de las veces desprotegidas y con muchas vulnerabilidades.

**Botnet.** Es un conjunto de dispositivos conectados a internet que permite controlar mediante un software la realización de diversas actividades vía remota.

**Cadenas de Markov.** Es una serie de eventos, en la cual la probabilidad de que ocurra un evento depende del evento inmediato anterior

**Configuración de seguridad incorrecta.** Una buena seguridad requiere tener definidos y desarrollados una configuración segura en las aplicaciones, marcos de trabajo, servidores de aplicaciones, servidores web, servidores de bases de datos, plataformas, entre otros. Las configuraciones seguras deben ser definidas, implementadas y mantenidas, automáticamente para cubrir inseguridades. Además, se debería de mantener al día el software usado.

**Cortafuegos físicos.** Son dispositivos que permiten cerrar puertos no utilizados en un servidor y los cuales pueden ser vulnerables y usados por personas no autorizadas.

**Cortafuegos genéricos**. Son programas o dispositivos usados para detener actividad no permitida en aplicaciones web usados para cualquier empresa.

**Crawler.** Es un software encargado de revisar diversas páginas web de forma automática y sistemática.

**Crowd sourcing.** Colaboración abierta de tareas o descubrimiento de nuevas tareas que son realizadas por otras personas o grupo de personas.

**Cuadrante de Gartner.** Es una gráfica hecha por la compañía estadounidense Gartner que permite identificar a las empresas cuales son las mejores dentro del mercado.

**Datos Iniciales.** Son los datos recibidos por el laboratorio de pentesting en los que se reciben el nombre de la vulnerabilidad, el del proyecto, la url afectada por esta vulnerabilidad o vulnerabilidades.

**Exposición a datos sensibles.** Muchas aplicaciones web y APIs no protegen apropiadamente los datos sensibles, como los financieros, salud y PII. Los atacantes pueden robar o modificar como hacer vulnerables los datos protegidos para llevar acabo fraudes con la tarjeta de crédito, robo de identidad u otros crímenes. Los datos sensibles requieren protección extra como la encriptación a todos los datos o a los datos enviados, así como precauciones especiales cuando se cambia de navegador.

**Falsificación de peticiones en sitios cruzados(CSRF).** Un ataque CSRF obliga al navegador de la víctima logueada a enviar una petición HTTP falsa, que incluya una cookie de la sesión de la víctima y cualquier otro de manera automática incluyendo la información de autenticación, para vulnerar la aplicación web. Cada uno de los ataques permiten al atacante forzar al navegador de la víctima generar las respuestas de la aplicación vulnerable que piensa que son respuestas legitimas de la víctima.

**Falso Positivo**. Es cuando una página web es reconocida como una amenaza cuando no lo es.

**Grupo de Ingeniería en Sistemas o GIS.** Personal encargado de la seguridad de la información y pruebas a los proyectos del instituto.

**Hacker.** Persona que entiende y conoce términos avanzados de redes de computadoras que le permite detectar las vulnerabilidades de una red para así lograr el acceso no autorizado a la información relevante de una empresa.

**Hardening robusto**. Es el proceso de asegurar un sistema reduciendo los agujeros y vulnerabilidades.

**Hardware.** Es el conjunto de dispositivos que están interconectados entre si y que conforman una computadora.

**HTTP o protocolo de transferencia de hipertexto.** Es un protocolo de comunicación para transferir información a través del internet.

**Instituto.** Se refiere al Instituto de Estadística y Geografía (INEGI).

**Insuficiente protección a los ataques.** La mayoría de las aplicaciones y APIs carecen de habilidad básica para detectar, prevenir y responder a ataques tanto manuales como automatizados. La protección contra estos ataques va desde la validación de las entradas e involucran una detección automática, logueo, respuesta y bloqueo a cada uno de los intentos de explotación. Los dueños de las aplicaciones también necesitan estar de acuerdo en el desarrollo rápido de parches para proteger contra los ataques.

**Inyección.** El flujo de la inyección, así como las inyecciones SQL, OS, XXE y LDAP ocurren cuando datos no validados son enviados para ser interpretados como parte de un comando o consulta. Esto es utilizado por el atacante para engañar al interprete y obtener de manera ilícita los datos, ejecutando comandos o ingresando de manera no intencionada a los datos sin tener la propia autorización.

**Laboratorios de Pentesting o Vulnerabilidades.** Son aplicaciones que permiten monitorear las aplicaciones web de una empresa para detectar vulnerabilidades, algunas herramientas permiten ver estas vulnerabilidades mediante reportes.

**Logging.** Indica la forma de logueo y no puede ser bloqueada.

**Matriz de Amenazas.** Es la matriz o conjunto de datos que permite visualizar las url y proyectos afectados por cada vulnerabilidad.

**Metaheurística.** Es una rama de las ciencias computacionales del área de optimización, que provee soluciones aceptables en un tiempo razonable para resolver problemas complejos y difíciles en ciencias e ingeniería.

**Métodos estocásticos**. Son métodos que utiliza al menos una variable aleatoria y funciones probabilísticas para estudiar la relación entre las variables.

**MODSECURITY.** Herramienta de código abierto que permite el monitoreo, logueo y control de acceso en tiempo de ejecución.

**OWASP (Open Web Application Security Project).** Organización Internacional sin fines de lucro que aporta conocimiento y herramientas colaborativas para mitigar riesgos en aplicaciones que transfieran información vía internet (OWASP, 2017).

**Perdida de autenticación y gestión de sesiones.** Las funciones de la aplicación relacionan la administración de la autenticación y sesión como una implementación incorrecta, permitiendo al atacante comprometer contraseñas, claves o pases de sesión, para explotar otros flujos de implementación y asumir las identidades de otros usuarios de manera temporal o permanente.

**Pérdida de control de los accesos.** Son restricciones en la cual los usuarios autenticados son autorizados para realizar ejecuciones no apropiadas. El atacante puede explotar este flujo para acceder a la funcionalidad y a los datos de forma no apropiada, como acceso a las cuentas de otros usuarios, ver archivos sensibles, modificar los datos de otros usuarios, cambiar las restricciones de acceso, entre otras.

**Protocolo.** Conjunto de normas, reglas y pautas que sirven para guiar a un cierto tipo de acciones.

**Puerto.** Son accesos en los cuales puede haber entrada y salida de información en una computadora. Dentro de las computadoras existen diversos permites con diferente tipo de tráfico como por ejemplo el puerto 80 que es donde se maneja el tráfico de entrada y salida del internet.

**Reprocesos.** Cantidad de amenazas detectadas en la url o proyecto.

**Request Body.** Importante porque permite el análisis de la petición interceptada, de manera que si cumple alguna regla procede a ejecutarla.

**Request Headers.** Intercepta la petición realizada por el cliente a partir de las cabeceras y el cuerpo para enviarse al motor de reglas de MODSECURITY. Se realiza antes de procesar la petición.

**Response Headers.** Se realiza cuando la petición es generada y ya se han analizado las cabeceras, de manera que permite inspeccionar el cuerpo de la respuesta.

**Response Body.** Se realiza el análisis del cuerpo de la respuesta de manera que el motor de reglas tome la decisión más adecuada.

**SecRequestBodyAccess.** Puede tomar los valores de on y off, dependiendo del valor le indica al módulo cuando acceder al cuerpo de las peticiones. On permite acceder a la información incrementando la memoria de la máquina. Off permite bloquear el acceso al cuerpo de peticiones.

**SecRequestBodyInMemoryLimit.** Permite indicar el tamaño en bytes que será reservado en memoria RAM para almacenar los valores del cuerpo de peticiones.

**SecRequestBodyLimit.** Permite indicar el tamaño máximo de bytes permitidos para los buffers de los cuerpos de las peticiones, en caso de subirse un archivo se le debe de indicar el tamaño del archivo a subir.

**SecRequestBodyNoFilesLimit.** Directiva similar a la anterior, pero con la diferencia de que no se toma en cuenta la existencia de aplicaciones con subida de archivos.

**SecResponseBodyAccess.** Permite activar o desactivar el procesamiento de MODSECURITY para las respuestas generadas por el servidor. Puede tomar los valores off y on.

**SecResponseBodyLimit.** Permite establecer el límite n bytes de las respuestas generadas por el servidor.

**SecResponseMimeType.** Permite indicar los tipos MIME que pueden ir en las respuestas del servidor, es decir permite indicar lo que se permite o no se permite en el módulo.

**SecRule.** Examina las cabeceras de manera que cumpla con un valor dado definido en las expresiones regulares y ejecutando una acción en caso de cumplirse.

**Seguridad informática.** Es la aplicación de normas o reglas tanto a nivel de software y hardware que evita que personas ajenas o con fines mal intencionados accedan a la información sensible que se estén manejando.

**Seguridad de la información.** Asegura que la información no sea sustraída, eliminada y accedida por personas no autorizadas.

**Sentencias de comandos en sitios cruzados (XSS).** Un XSS ocurre cuando una aplicación incluye datos no reales en una página web sin validar adecuadamente, salirse o actualizar una página web existente con datos proporcionados por el usuario usando el navegador API que pudo crear JavaScript. Permitiendo a los atacantes ejecutar scripts en el navegador de la víctima, el cual puede secuestrar la sesión del usuario, falsificar sitios web o redireccionar al usuario a sitios maliciosos.

**Servidor de Aplicaciones Web.** Máquina o computador en donde se encuentra la aplicación y permite distribuir a otras computadoras el servicio.

**Sistemas de Detección de Intrusiones o Instructions Detections Systems (IDS).** Son mecanismos que escuchan el tráfico en la red para detectar actividad anormal o sospechosa.

**Software.** Es una aplicación o conjunto de aplicaciones que permite la interacción con la computadora.

**Top 10 de amenazas.** Es un listado de las amenazas más comunes en empresas localizados en distintas partes del mundo.

**URL.** Dirección de una página web.

**Uso de componentes con vulnerabilidades conocidas**. Los componentes como librerías, marcos de trabajo y otros módulos del software, corren con los mismos privilegios en la aplicación. Si un componente es vulnerable y es explotado, entonces el ataque puede facilitar la perdida de los datos de importancia o tomar el control del servidor. Las aplicaciones y APIs usan los componentes con vulnerabilidades conocidas que pueden debilitar las defensas y habilitar varios ataques e impactos.

**Violación.** Tipo de amenaza detectada según el top 10 de vulnerabilidades.

**Vulnerabilidad.** son agujeros o espacios que los atacantes encuentran dentro de las páginas y mediante esos agujeros acceden a la información de manera ilegal, provocando a veces daños a la información sin haberse contemplado.

**WAF (Web Application Firewall).** Sistema o dispositivo que impide el acceso a peticiones desconocidas detectadas en las aplicaciones web.

**WEB 1.0.** Es un término usado para decir la primera interacción en una computadora, esto quiere decir que no permite una interacción entre usuarios servidor, solamente con el contenido del servidor y la información generalizada.

**WEB 2.0.** Es la interacción de los usuarios de forma más dinámica, que permite la construcción y transformación de la información del contenido Web.

**XML.** También llamado lenguaje de marcas extensible es un lenguaje que permite la organización y etiquetado de documentos.

**Zona desmilitarizada.** Es cuando los servidores públicos son alejados de la red interna para así asegurar que los servidores públicos no se comuniquen con los otros segmentos de la red interna.

# BIBLIOGRAFÍA

Akana. (2015). Akana Secures APIs and Web Applications from OWASP Top Ten. Business Wire (English). Recuperado a partir de http://search.ebscohost.com/login.aspx?direct=true&db=bwh&AN=bizwire.c61101815&lang=es&site=ehost-live&scope=site

Cerullo, F. E. (2010). Deploying Secure Web Applications with OWASP Resources. En C. Serrão, V. A. Díaz, & F. Cerullo (Eds.), Web Application Security (pp. 21-21). Recuperado a partir de. https://doi.org/10.1007/978-3-642-16120-9_11

De Brito, F. H., Teixeira, A. N., Teixeira, O. N., & de Oliveira, R. C. L. (2006). A Fuzzy Intelligent Controller for Genetic Algorithms' Parameters. En L. Jiao, L. Wang, X. Gao, J. Liu, & F. Wu (Eds.), Advances in Natural Computation: Second International Conference, ICNC 2006, Xi'an, China, September 24-28, 2006. Proceedings, Part I (pp. 633-642). Berlín, Heidelberg: Springer Berlín Heidelberg. Recuperado a partir de http://dx.doi.org/10.1007/11881070_87

D'Hoinne, A. H. J., & Neiva, C. (2016, julio 19). Magic Quadrant for Web Application Firewalls. Recuperado el 31 de mayo de 2017, a partir de https://www.gartner.com/doc/reprints?id=1-3BZK2PZ&ct=160720&st=sb

Díaz Méndez S. (2013, marzo 5). Firewall de Aplicación Web - Parte I | Revista. Seguridad, *16*. Recuperado a partir de https://revista.seguridad.unam.mx/node/2167

Edge, C., & O'Donnell, D. (2016). Managing the Firewall. En Enterprise Mac Security (pp. 299-321). Apress. Recuperado a partir de https://doi.org/10.1007/978-1-4842-1712-2_11

Gartner. (2017, julio 10). Magic Quadrant for Enterprise Network Firewalls. Recuperado el 22 de noviembre de 2017, a partir de https://www.gartner.com/doc/reprints?id=1-45UW8EQ&ct=170711&st=sb

Hannes Holm, & Mathias Ekstedt. (2013). Estimates on the effectiveness of Web application firewalls against targeted attacks. Information Management & Computer Security, 21(4), 250–265. Recuperado a partir de https://doi.org/10.1108/IMCS-11-2012-0064

Higuera, J. (2013, septiembre). ¿Ciberguerra o Ciber-seguridad? Tecnología Militar, pp. 2-2. Intrusion Detection and Prevention, Security Data Analytics, Personal Firewall - Privacyware. (s. f.). Recuperado 25 de noviembre de 2016, a partir de https://www.privacyware.com/intrusion_prevention.html

Hostalia. (2015, enero 28). Protege tu servidor web con el módulo MODSECURITY. Recuperado el 22 de noviembre de 2017, a partir de https://pressroom.hostalia.com/white-papers/MODSECURITY

Imperva, Inc. (2017a). SecureSphere Web Application Firewall (WAF) – Real-time protection against Web attacks. (s. f.). Recuperado 8 de noviembre de 2016, a partir de https://www.imperva.com/Products/WebApplicationFirewall-WAF

Imperva, Inc. (2017b). ThreatRadar: Web Application Threat Intelligence. (s. f.). Recuperado 25 de noviembre de 2016, a partir de https://www.imperva.com/Products/ThreatRadarSubscriptions

John Wiley & Sons, Inc. (2011) The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Second EDITION.

Magnus, M. (2009). *MODSECURITY 2.5 Securing your Apache installation and web applications* (First Edition). Olton, Birmingham, B27 6PA, UK.: Packt Publishing.

Manrique Maldonado, K. (2015, septiembre 1). La Web 2.0 y sus servicios como herramientas en el entorno educativo del siglo XXI. Revista Digital Universitaria. 16 (9). Recuperado a partir de http://www.revista.unam.mx/vol.16/num9/art76/.

Maskey, S., Jansen, B., Guster, D., & Hall, C. (2007). A Basic Firewall Configuration Strategy for the Protection of Development-related Computer Networks and Subnetworks. *Information Systems Security*, *16*(5), 281-290. Recuperado a partir de https://doi.org/10.1080/10658980701744853

Mayer, A., Wool, A., & Ziskind, E. (2005). Offline firewall analysis. *International Journal of Information Security*, *5*(3), 125-144. Recuperado a partir de https://doi.org/10.1007/s10207-005-0074-z

McMillan, J. (2009, Noviembre). IDFAQ: What is the Difference Between an IPS and a Web Application Firewall? SANS - Information Security Resources. Recuperado 8 de noviembre de 2016, a partir de https://www.sans.org/security-resources/idfaq/what-is-the-difference-between-an-ips-and-a-Web-application-firewall/1/25

MODSECURITY, Inc. (2017). MODSECURITY: Open Source Web Application Firewall. Recuperado el 21 de noviembre de 2017, a partir de https://MODSECURITY.org/

Nguyen, H. T., Torrano - Gimenez, C., Alvarez, G., Franke, K., & Petrović, S. (2013). Enhancing the effectiveness of Web Application Firewalls by generic feature

selection. *Logic Journal of IGPL*, *21*(4), 560–570. Recuperado a partir de https://doi.org/10.1093/jigpal/jzs033

Nomura, Y., & Salzetta, N. (2016). Why firewalls need not exist. *Physics Letters B*, *761*(Supplement C), 62–69. https://doi.org/10.1016/j.physletb.2016.08.003

OWASP. (2017, marzo 16). About The Open Web Application Security Project - OWASP. Recuperado el 27 de mayo de 2017, a partir de https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

Pałka, D., & Zachara, M. (2011). Learning Web Application Firewall - Benefits and Caveats. En A. M. Tjoa, G. Quirchmayr, I. You, & L. Xu (Eds.), *Availability, Reliability and Security for Business, Enterprise and Health Information Systems* (pp. 295-308). Springer Berlin Heidelberg. Recuperado a partir de https://doi.org/10.1007/978-3-642-23300-5_23

PR Newswire. (2013). SunGard Availability Services, Alert Logic Present Managed Web Application Firewall at the RSA Security Conference. *PA-SunGard-RSA*. Recuperado a partir de http://search.ebscohost.com/login.aspx?direct=true&db=bwh&AN=20130226 1000PR.NEWS.USPR.CG66185&lang=es&site=ehost-live&scope=site

Prince, B. (2013). OWASP Lists Top 10 Most Critical Web Application Risks. eWeek, 7-7.

Privacyware, Inc. (2017). Intrusion Detection and Prevention, Security Data Analytics, Personal Firewall - Privacyware. Recuperado el 21 de noviembre de 2017, a partir de https://www.privacyware.com/intrusion_prevention.html

Radware, Inc. (2017). Web Application Firewall (WAF) & Network Security Solution | AppWall. (s. f.). Recuperado 25 de noviembre de 2016, a partir de http://www.radware.com/Products/AppWall/

Ramirez Castro, Alexandra. (2017). Riesgo tecnológico y su impacto para las organizaciones parte I. *Revista .Seguridad*. Recuperado a partir de https://revista.seguridad.unam.mx/numero-14/riesgo-tecnol%C3%B3gico-y-su-impacto-para-las-organizaciones-parte-i

Torrano Giménez, C. (2015). Study of stochastic and machine learning techniques for anomaly-based Web attack detection. Recuperado a partir de http://e-archivo.uc3m.es/handle/10016/21876

Torrano-Gimenez, C., Perez-Villegas, A., & Alvarez, G. (2009). A Self-learning Anomaly-Based Web Application Firewall. En Á. Herrero, P. Gastaldo, R. Zunino, & E. Corchado (Eds.), *Computational Intelligence in Security for Information Systems* (pp. 85-92). Springer Berlin Heidelberg. Recuperado a partir de https://doi.org/10.1007/978-3-642-04091-7_11Vacca, J. R. (Ed.). (2007a). Choosing the Right Firewall. En *Practical Internet Security* (pp. 373–397). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-29844-3_30

TRUSTWAVE. (2017). *ACERCA DE MODSECURITY*. Obtenido de MODSECURITY: https://www.MODSECURITY.org/about.html

UNAM. (2017). Fundamentos Teóricos. Recuperado el 22 de noviembre de 2017, a partir de http://redyseguridad.fi-p.unam.mx/proyectos/tsi/capi/Cap1.html

Vacca, J. R. (Ed.). (2007a). Choosing the Right Firewall. En *Practical Internet Security* (pp. 373–397). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-29844-3_30

Vacca, J. R. (Ed.). (2007b). Threats and Vulnerabilities. En *Practical Internet Security* (pp. 145–175). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-29844-3_7

# ANEXOS

## ANEXO 1 A1. CAMBIOS NO REALIZADOS EN TRABAJO DE TESIS

Como se comentó en las conclusiones el trabajo de tesis se basó en una versión anterior a la que se encuentra actualmente, debido a los avances ya realizados con el trabajo.

En la Tabla 15, se ve un cuadro comparativo con respecto de las vulnerabilidades que estaban en el Top 10 de OWASP y las actuales, así como las amenazas que se hubieran elegido en caso de actualizar y con cuales se trabajó durante este trabajo.

DESVENTAJAS DE HACER CAMBIOS

1. Cambio de nomenclatura de algunas amenazas en
   - Imágenes
   - Tablas
   - Dentro del documento

2. Hacer modificaciones en la herramienta, debido a que son fijos los códigos

3. La investigación estaba hasta la etapa de resultados, por lo que se cambiarían todos los resultados.

4. Así como un análisis exhaustivo de las amenazas porque las APIs desprotegidas se descarta en la nueva versión de OWASP.

5. Se invertiría más tiempo de lo debido para la realización de la tesis, estando ya a unos meses de entregar el documento.

EXPLICACIÓN DE LAS AMENAZAS NO COINCIDENTES DE LA VERSION ACTUAL DE OWASP

**Perdida de autenticación.** Son funciones de las aplicaciones relacionada con la autenticación y manejo de sesiones que no están implementadas correctamente, permitiendo a los atacantes comprometer contraseñas, llaves o tokens de sesiones para explotar el flujo de otras implementaciones y asumir de manera temporal o permanente otros usuarios.

**Entidades externas XML.** Se refiere a procesadores de configuraciones pobres o viejas de XML evaluadas por referencias externas dentro de documentos XML. Estas entidades pueden ser usadas para revelar archivos internos usando los archivos gestores URI, archivos compartidos internos, puertos de escaneo internos, códigos de ejecución remota y ataques de denegación de servicios.

**Control de acceso roto.** Son restricciones en la que los usuarios autentificados permiten hacer y no son aplicados correctamente. Los atacantes pueden explotar estos flujos de acceso no autorizados en funcionalidad o datos, como acceder a las cuentas de otros, ver archivos sensibles, modificar datos de los usuarios, cambiar los permisos de accesos, entre otros.

**Configuración de inseguridad incorrecta.** La mala configuración es uno de los más importantes problemas. Esto es un resultado común de configuraciones predeterminadas inseguras, incompletas, configuraciones ad hoc, almacenamiento en la nube abierto, cabeceras HTTP mal configuradas y mensajes de error detallados que contienen información sensible.

**Deserialización insegura.** Este tipo de amenaza es dirigida remotamente a través de código de ejecución. Si el flujo de deserialización no resulta en el código de ejecución remota, ellos pueden usar ataques mejorados, incluyendo ataques repetidos, ataques de inyección y ataques de escalación de privilegios.

**Uso de componentes con vulnerabilidades conocidas.** Los componentes, como librerías marcos de trabajo y módulos de software, corren con los mismos privilegios como las aplicaciones. Si un componente vulnerable es explotado, como un ataque con facilidades de datos perdidos graves o en la toma de servidores. Las aplicaciones y APIs usan componentes con vulnerabilidades conocidas que pueden debilitar las defensas de las aplicaciones y habilitar varios ataques e impactos.

**Monitoreo y logueo insuficiente.** Esto se combinan con falta o inefectividad en la integración con respuestas incidentes que permiten a los atacantes fomentar los ataques al sistema, manteniendo la persistencia, avanzar y manipular a más sistemas, extraer o destruir datos.

*Tabla 15. Cuadro comparativo de versiones.*

| | Versión utilizada | | Versión actual | |
|---|---|---|---|---|
| **Código de Amenaza** | **Amenaza** | **Objeto de estudio** | **Amenaza** | **Objeto de estudio** |
| A1 | Inyección | X | Inyección (Injection) | X |
| A2 | Perdida de autenticación y gestión de sesiones | | Perdida de autenticación (Broken Authentication) | |
| A3 | Sentencias de comandos en sitios cruzados (XSS) | X | Exposición a datos sensibles (Sensitive Data Exposure) | |
| A4 | Pérdida de control de los accesos | | Entidades externas XML (XML External Entities XXE) | |
| A5 | Configuración de seguridad incorrecta | | Control de acceso roto (Broken Access Control) | |
| A6 | Exposición a datos sensibles | | Configuración de seguridad incorrecta (Security Misconfiguration) | X |
| A7 | Insuficiente protección a los ataques | | Secuencia de comandos en sitios cruzados (Cross-Site Scripting XSS) | X |
| A8 | Falsificación de peticiones en sitios cruzados(CSRF) | | Deserialización insegura (Insecure Deserialization) | |
| A9 | Uso de componentes con vulnerabilidades conocidas | | Uso de componentes con vulnerabilidades conocidas (Using Components with Known Vulnerabilities) | |
| A10 | APIs desprotegidas | X | Insuficiente registro y monitoreo (Insufficient Logging & Monitoring) | |

**Fuente:** Elaboración propia