Índice general

1. Antecedentes **1**
- 1.1. Introducción . . . 1
  - 1.1.1. Servicios de Seguridad . . . 1
  - 1.1.2. Controles de Seguridad . . . 2
- 1.2. Descripción del problema . . . 3
- 1.3. Solución propuesta . . . 4
- 1.4. Estado del arte . . . 4
- 1.5. Justificación . . . 5

2. Marco Teórico **7**
- 2.1. Estampa de tiempo . . . 7
- 2.2. Detección de Intrusos . . . 9
  - 2.2.1. Definición . . . 9
  - 2.2.2. Taxonomías . . . 9
    - 2.2.2.1. Taxonomía por punto de detección . . . 10
    - 2.2.2.2. Taxonomía por métodos de detección . . . 11
  - 2.2.3. Modo de operación . . . 12
  - 2.2.4. Justificación de los Sistemas de Detección de Intrusos . . . 13
  - 2.2.5. Arquitectura de IDSs . . . 13
    - 2.2.5.1. Dorothy Denning . . . 13
    - 2.2.5.2. Common Intrusion Detection Framework . . . 14
    - 2.2.5.3. Common Intrusion Specification Language . . . 14
    - 2.2.5.4. Autopost de AusCERT . . . 15
    - 2.2.5.5. Intrusion Detection Working Group . . . 16
  - 2.2.6. Componentes elementales de un IDS . . . 17
  - 2.2.7. Arquitecturas de Red . . . 18
- 2.3. Cross-site Scripting (XSS) . . . 19
  - 2.3.1. Definición . . . 19
  - 2.3.2. Actores en un ataque XSS . . . 19
  - 2.3.3. Tipos de Cross-Site Scripting . . . 19
    - 2.3.3.1. XSS almacenado (AKA persistente o tipo I) . . . 19
    - 2.3.3.2. XSS reflejado (AKA no persistente o tipo II) . . . 19
    - 2.3.3.3. XSS basado en DOM (AKA tipo-0) . . . 20

2.3.4. Riesgos de los ataques XSS . . . 20
2.3.5. Prevención . . . 20
2.4. Aprendizaje Automático . . . 21
2.4.1. Definición . . . 21
2.4.2. Concepto . . . 21
2.4.3. Métodos de aprendizaje . . . 22
2.4.3.1. Árboles de Decisión . . . 22
2.4.3.2. Redes Neuronales . . . 22
2.4.3.3. Aprendizaje Bayesiano . . . 22
2.4.3.4. Algoritmos Genéticos y Programación Genética . . . 22
2.4.3.5. Basado en Instancias . . . 23
2.4.3.6. Aprendizaje Reforzado . . . 23
2.4.3.7. Aprendizaje de Múltiples Instancias . . . 23
2.4.3.8. Aprendizaje No Supervisado . . . 23

3. Especificaciones del proyecto **24**
3.1. Objetivo . . . 24
3.1.1. Objetivo general . . . 24
3.1.2. Objetivos específicos . . . 24
3.2. Metodología . . . 24
3.3. Arquitectura . . . 25

CAPÍTULO 1

Antecedentes

1.1. Introducción

La seguridad informática ha sido y actualmente es un sector en el cual, empresas importantes de gran prestigio gastan cientos de millones de dólares para protegerse al estar conectados a una red [1], tal es la preocupación que a nivel mundial se registró una inversión en la seguridad informática de 75 billones de dólares en 2015 [2], mientras que, tanto chicas como medianas empresas suelen gastan un mínimo relacionado a este tema. En la actualidad, con la era de la revolución tecnológica por la que se está pasando, las empresas se han visto obligadas a contratar nueva tecnología para su producción, publicidad y/o servicios conectada al mundo de la Internet. Las empresas al estar conectadas a la Internet, están conectadas a millones de usuarios con cientos de posibilidades de acceso a los servicios de las empresas.

Cuando los usuarios se conectan a la red de Internet, están conectados todos los usuarios de la misma simultáneamente, esto conlleva un alto riesgo de inseguridad. Para tratar de disminuir el impacto provocado por amenazas informáticas, existen programas de computadora enfocados a detectar y proteger a los usuarios de la red contra los impactos provocados por las amenazas [3].

Hoy en día, existen diversos tipos de amenazas en la red, algunas son muy conocidas como virus informáticos que son diseñados para infectar archivos, pero no sólo existen ese tipo de amenazas diseñadas cierta forma con una actuación "automatizada". También existen las amenazas humanas o conocidos como Piratas informáticos, los cuáles, por medio de diversas técnicas de vulneración, pueden infectar, tomar el control o incluso obtener información o privilegios de computadoras o servidores con el fin de aprender o poner en práctica nuevas técnicas de vulneración, vender la información obtenida en el mercado negro o inclusive realizar un daño directo a los archivos o computadora objetivo [4][5].

Con el incremento de nuevos servicios web en Internet, se han creado y desarrollado diversos tipos de ataques hacia los servidores que proporcionan estos servicios. En los últimos años han ido en incremento los ataques web, aunque los que han tenido un mayor crecimiento y un gran impacto a nivel global, son los ataques tipo Cross-Site Scripting (de ahora en adelante XSS) [6].

1.1.1. Servicios de Seguridad

Es cuya función principal es mejorar la seguridad de un sistema de información y el flujo de información que pasa a través de una organización. Los servicios de seguridad están orientados a evitar ataques de informáticos haciendo uso de distintos controles de seguridad para proveer el servicio. Cada control de seguridad está diseñado para realizar una función determinada dependiendo del servicio de seguridad que se desee otorgar.

Los servicios de seguridad se dividen en seis clasificaciones:

- Disponibilidad: Es un requerimiento destinado a asegurar que el sistema trabaja apropiadamente y el servicio no deniega a usuarios autorizados. Este servicio protege contra:
  - Intentos intencionales o accidentales de:
    - eliminación no autorizada de datos o
    - de lo contrario causar una denegación de servicio o datos.
  - Intentos para usar el sistema o datos para propósitos no autorizados.

  La disponibilidad es frecuentemente es el principal objetivo de seguridad de una organización.
- Integridad: Tiene dos facetas:
  - Integridad de datos (la propiedad de que los datos no han sido alterados en un manejo no autorizado) o
  - Integridad del sistema (la calidad que tiene un sistema al realizar la función deseada de manera intacta, libre de manipulación no autorizada).

  La integridad es comúnmente el objetivo más importante dentro de una organización después de la disponibilidad.
- Confidencialidad: Consiste en que información dentro del sistema de una organización no sea accesada por personal no autorizado. Por diversas razones, aún se pone la confidencialidad debajo de la disponibilidad e integridad en términos de importancia. Pero a pesar de esto, para algunos sistemas, de autenticación, por ejemplo, la confidencialidad es el objetivo más importante a considerar.
- No repudio: Consiste en identificar al responsable de una acción (un ataque, por ejemplo) hacia un sistema, responsabilizándolo por los actos y sin la posibilidad de que éste niegue los hechos.
- Autenticación: Es un requerimiento que consiste la identificación de un personal para que no pueda ser suplantado, y así acceder a cierta información contenida en un sistema.

1.1.2. Controles de Seguridad

Los controles de seguridad proveen un rango comprensivo de contra-medidas para organizaciones y sistemas de información. Los controles de seguridad son diseñados para ser tecnologías neutrales tal manera que se centre en las contra-medidas fundamentales necesitadas para proteger la información de la organización durante el procesamiento, almacenamiento o su transmisión [22]. La implementación de los controles de seguridad dependen al nivel de protección que se desee tener en un sistema. Las buenas prácticas de seguridad hacen mención que para tener una buena protección en un sistema u organización, se deben de emplear los controles de seguridad en conjunto, haciendo referencia que se deben de emplear varios de estos controles para así complementar brechas de seguridad que se contengan en los controles.

Mencionando algunos controles más populares empleados, tenemos los siguientes:

- Cortafuegos: Su función es delimitar el área perimetral de la red filtrando el flujo de red, tanto de entrada como de salida e incluso entre la comunicación de diferentes áreas dentro de la misma red. Se tiene tres categorías, los cortafuegos de paquete, de estado y de aplicación, donde su modo de operación varía en que el primero sólo se fija en el algunos campos del encabezado de los paquetes de red, el segundo hace un análisis más profundo del encabezado y el último hace un análisis en el *payload*[1] del paquete de red.
- Proxy: Es una entidad que funciona como intermediario entre la comunicación entre redes de una organización.
- Antivirus: Programa que tiene como finalidad detectar código malicioso dentro de los sistemas de lo cuáles se encarga de analizar.
- Detección de Intrusos: Programas que se encargan de hacer un monitoreo de las entidades de las cuáles se encarga.
- Honey Pots (por su nombre en inglés): Entidad que se encarga se ser un señuelo específicamente diseñado para atraer atacantes yo ver el comportamiento de programas maliciosos, con la finalidad de utilizarse como fuente para estudiar las nuevas formas de intrusión.

1.2. Descripción del problema

Un ataque XSS ocurre cuando un atacante es capaz de inyectar un script, normalmente JavaScript, en la salida de una aplicación web de forma que se ejecuta en el navegador del cliente. Los ataques se producen principalmente por validar incorrectamente datos de usuario, y se suelen inyectar mediante un formulario web o mediante un enlace alterado.

Existen tres tipos de ataques XSS:

- XSS persistente o directo: este tipo de ataque consiste en embeber código HTML peligroso en sitios que lo permitan por medio de etiquetas *<script>* o *<iframe>*. Es la más grave de todas ya que el código se queda implantado en la web de manera interna y es ejecutado al abrir la aplicación web.

- XSS reflejado: en este tipo de ataque el código malicioso no queda almacenado en el servidor sino que se pasa directamente a la víctima. Es la forma más habitual de XSS. El ataque se lanza desde una fuente externa como un correo aparentemente inofensivo, un mensaje de chat u otro sitio web [8].

- XSS basado en DOM: es una variable de XSS persistente y reflejado. En un ataque XSS basado en DOM, la cadena maligna no es realmente analizada por el navegador de la victima hasta que el JavaScript legítimo de la página web es ejecutado. Estos códigos son ejecutados del lado del cliente, por lo que los filtros utilizados en el servidor no funcionan para este tipo de vulnerabilidades.

A la hora de lanzar un ataque de este tipo, los atacantes pueden utilizar varios tipos de inyección de código distinto. Los más utilizados son:

- Inyección en un formulario: se trata del ataque más sencillo. Consiste en inyectar código en un formulario que después al enviarlo al servidor, será incluido en el código fuente de alguna página. Una vez insertado en el código fuente, cada vez que se cargue la página se ejecutará el código insertado en ella.

[1] Los datos esenciales que es llevada dentro de un paquete de red y otra unidad de transmisión

- Inyección por medio de elementos: en este tipo de sistema de inyección de código se utiliza cualquier elemento que viaje entre el navegador y la aplicación, como pueden ser los atributos usados en las etiquetas HTML utilizadas en el diseño de la página.
- Inyección por medio de recursos: Aparte de los elementos en la URL y los formularios, hay otras formas en la que se puede actuar como son las cabeceras HTTP. Estas cabeceras son mensajes con los que se comunican el navegador y el servidor. Aquí entran en juego las *cookies*[2] y las sesiones [9].

Los daños potenciales que pueden causar un ataque XSS, pueden afectar tanto a los servidores en donde está contenida la aplicación web o pueden provocar serios problemas para el usuario final, éstos pueden varían en el grado de impacto, pueden ir desde una molestia para el usuario hasta un compromiso completo de la cuenta del mismo. Uno de los efectos más graves de los ataques XSS implica la divulgación de cookies de sesión del usuario, lo que permite a un atacante secuestrar la sesión del usuario y tomar control total de la cuenta. Otros ataques dañinos incluyen la divulgación de los archivos de los usuarios finales, la instalación de programas dañinos para el equipo del usuario final, redirigir al usuario a otra página o sitio web con fines malicioso, o modificar la presentación de los contenidos [10]. Los ataques XSS explotan vulnerabilidades no en el navegador del usuario, sino en las aplicaciones Web de terceros a las que accede el usuario. En este tipo de ataque el navegador no puede distinguir entre el contenido que un usuario haya podido incluir en una petición Web, y el contenido inyectado a través de un ataque XSS [11].

Se han desarrollado nuevas tecnologías que utilizan diferentes técnicas para poder detectar, contrarrestar y protegerse de los ataques tipo XSS [12], algunas de esas tecnologías son aplicadas en Cortafuegos de Aplicaciones Web (WAFs, por sus siglas en inglés), los Sistemas de Detección de Intrusos (IDSs, por sus siglas en inglés) e inclusive, las mismas empresas desarrolladoras de antivirus, han integrado nuevos módulos en sus sistemas en contra de este tipo de ataques [13].

Aunque se tiene registros de los problemas causados y el incremento que ha tenido este tipo de ataque, el principal objetivo de las tecnologías que se lanzan al mercado no es completamente enfocado a este ataque. Tal hecho provoca que al realizar auditorías de las herramientas en ésta parte de vulnerabilidades, se detecten fallos en el sistema, tales como falsos positivos o falsos negativos.

1.3. Solución propuesta

La propuesta para la solución a este problema, es desarrollar un sistema del tipo detector de intrusos con el fin de ayudar a los administradores web a tener una defensa y un alertador de ataques XSS que esté sufriendo su sitio o sistema.

1.4. Estado del arte

Anteriormente se han definido brevemente varios conceptos sobre los que se hablará en el resto del documento. Estos conceptos sirven para dar una idea muy básica al lector.

En la presente sección, en la Tabla 1.1 se entra en profundidad a definir el estado del arte y las caracteríticas mas relevantes de los IDS que existen actualmente en el mercado.

Existen además otros sistemas como lo son: Kismet, SmartDefense, Symantec Network Security 7100, Cisco

[2]Una cookie es un pequeño elemento de información que un servidor Web envía al navegador al visitar ciertas páginas web y que ambos comparten cada que este navegador vuelve a visitar [7].

Nombre	Descripción	Tipo	Año	Lugar de desarrollo	Escrito en
SNORT	Se trata de un sistema basado en red que monitoriza todo un dominio de colisión y funciona detectando usos indebidos. Dispone de un lenguaje de creación de reglas en el que se pueden definir los patrones que se utilizarán a hora de monitorizar el sistema. Además, ofrece una serie de reglas y filtros ya predefinidos que se pueden ajustar durante su instalación y configuración.	Open Source	1998	Cisco Systems	C
Suricata	Es un motor de detección de amenazas de red, maduro, rápido y robusto, de código abierto y gratuito. Es capaz de detectar intrusos en tiempo real, prevención de intrusiones en línea, supervisión de seguridad de red y procesamiento offline de pcap.	Open Source	2009	OISF	C
Bro	Es un sistema de detección de intrusiones para UNIX/Linux que analiza el tráfico de red en busca de actividad sospechosa. Su característica principal es que sus reglas de detección están basados en su lenguaje nativo que supone políticas (policies) que son las encargadas de detectar, generar logs o eventos y acciones a nivel de sistema operativo.	Open Source	2005	ICSI and NCSA	C++
Security Onion	Es un sistema que consiste en varias de las tecnologías de código abierto de mencionadas anteriormente trabajando en concierto entre sí. La plataforma ofrece una completa detección de intrusos, monitoreo de seguridad de red y administración de registros combinando lo mejor de Snort, Suricata, Bro, así como otras herramientas como Sguil, Squert, Snorby, ELSA, Xplico, entre otras.	Open Source	2010		

Tabla 1.1: Comparativa de IDSs en el mercado

Secure IDS 4230.

1.5. Justificación

La gran mayoría de los sistemas desarrollados hoy en día enfocados a la detección de intrusos basados en red, no tienen un gran soporte ante los ataques XSS, de tal forma que pueden llegar a fallar teniendo falsos positivos o falsos negativos [14][15], y las herramientas que lo tienen mejor implementado son adquiridas por empresas que puedan absorber el pago debido a su costo alto.

Para intentar solucionar tanto los ataques XSS persistentes como los no persistentes se sugiere implementar un sistema de filtrado y/o análisis, aunque estas soluciones pueden ser propuestos teóricamente como una tarea fácil, llevarlo a la práctica es mucho más complicado. Aunque la mayoría de ataques XSS conocidos están escritos en JavaScript e incrustados en documentos HTML, aunque también se pueden usar otras tecnologías como Java, Flash, ActiveX, etc., para efectuar los ataques, es por ello que es muy complicado la concepción

de un proceso de filtrado y/o análisis genérico capaz de tratar el mal uso de dichos lenguajes.

La complejidad para ser detectados radica por una parte, en la utilización de *proxies*[3] de filtrado, especialmente en la parte del servidor, que introduce limitaciones importantes referentes a la escalabilidad y rendimiento de aplicaciones Web. Por otra parte, los scripts maliciosos pueden estar incrustados en los documentos intercambiados de manera ofuscada (por ejemplo codificando el código malicioso en hexadecimal o métodos de codificación avanzados) para no ser detectado ante estos filtros y analizadores [16].

Se considera este proyecto ya que será de ayuda a aquellas empresas y personas que deseen detectar ataques de tipo XSS dirigidos a las aplicaciones instaladas en sus servidores, implementando métodos de análisis de datos, como el aprendizaje máquina orientados a la seguridad informática haciendo más eficiente su funcionamiento. Y así alertar a los administradores para poder prever efectos irreversibles en el sistema o de manera más grave, una toma de control total o escalabilidad de permisos en el sistema anfitrión del servicio.

[3]Es una aplicación que "rompe" la conexión entre el cliente y el servidor [18].

CAPÍTULO 2

Marco Teórico

2.1. Estampa de tiempo

Antes de entrar a la explicación sobre los sistemas de detección de intrusos, se van a definir las etapas que surgen antes de que suceda un incidente de seguridad en donde en el proceso de una de esas etapas se encuentra el funcionamiento de los sistemas de detección de intrusos.

Basados en lo que nos hace mención en el NIST, para llegar a una etapa en donde ocurre un incidente de seguridad se tuvo que haber pasado por otras tres etapas anteriores. La estampa de tiempo consta de tres etapas: Ataque, Intrusión e Incidente. Si en dicha estampa de tiempo se logra traspasar la etapa de un incidente, ya es considerado como un evento de seguridad.

En la Figura 2.1 se pueden ver las etapas con respecto al tiempo para la generación de un evento de seguridad. Las etapas se pueden definir como:

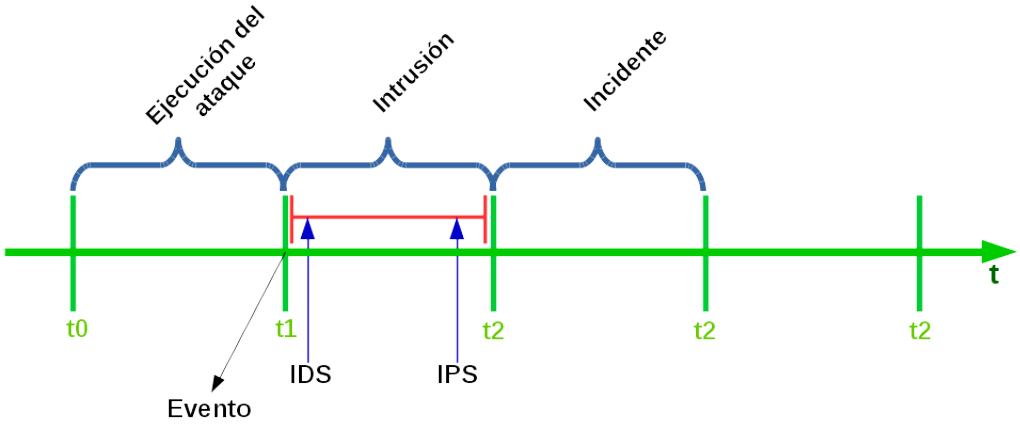


Figura 2.1: Estampa de tiempo describiendo las etapas de un evento de seguridad.

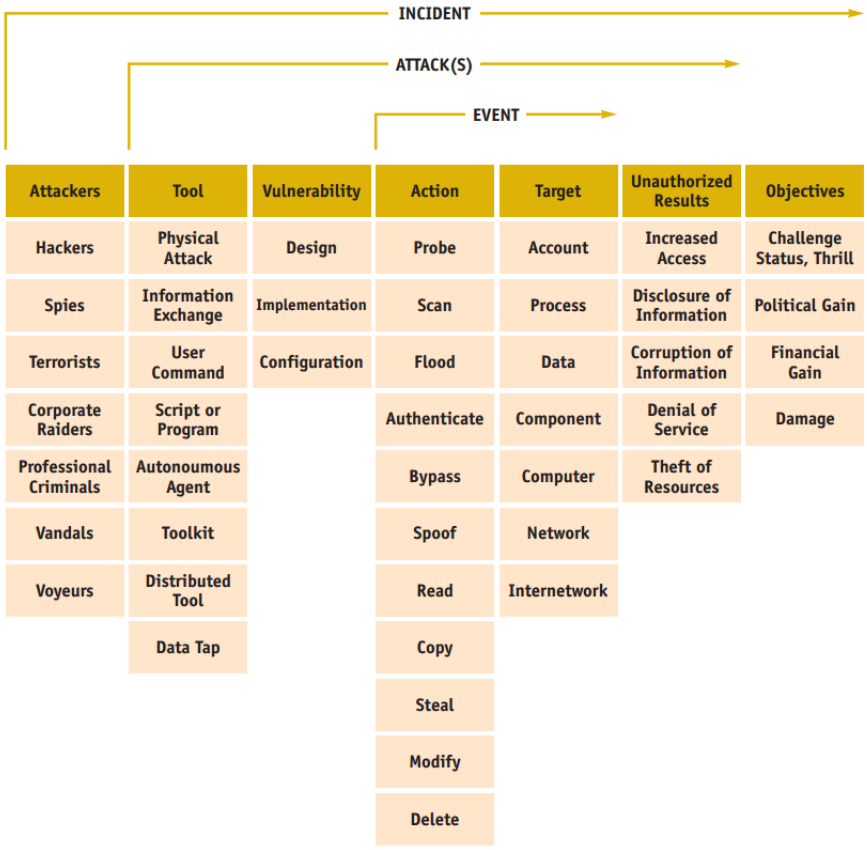


Figura 2.2: Taxonomía de incidentes de seguridad.

1. **Ataque:** Explotación de una vulnerabilidad.
2. **Evento:** Lo ocasionado por un ataque identificado, y en la mayoría con resultado, exitoso.
3. **Intrusión:** Acceso a redes no autorizadas, cuando se logra traspasar la primera linea de controles de seguridad.
4. **Incidente:** Lo posterior a un resultado no autorizado. Violación or am enaza inminente de violación de las políticas de seguridad.

Una vez conocido las etapas por las que se debe de pasar para considerar que se ha generado un incidente de seguridad, se puede proceder a ver la taxonomía de un incidente de seguridad. Es importante el conocimiento de en dónde se ubica cada etapa debido a que dependiendo de la información que se tenga y la etapa en la que se encuentre, el manejo de los términos cambia y con ello los controles de seguridad que deben entrar en acción. En la Figura 2.2 se muestra la taxonomía de los incidentes de seguridad, como se puede observar, en cada una de las etapas existe un bloque de información que se debe conocer para poder hacer uso del término. Si bien, cada etapa es acumulativa lo que implica que sea complementada con la información obtenida de la anterior.

Para que un evento pueda ser considerado así se debe tener conocimiento que una acción, en nuestro caso de estudio debe ser una acción maliciosa, fue o es ejecutada sobre un objetivo, este puede ser cualquier dispositivo conectado a una red dentro de la organización. Para hablar sobre un ataque a una organización se debe de conocer las herramientas usadas, la vulnerabilidad explotada, la acción ejecutada, el objetivo atacado y los resultados no autorizados. Y finalmente para poder considerar que ha ocurrido un incidente de seguridad, la

información que se debe conocer es: quién fue la contra-parte, la herramienta usada, la vulnerabilidad, la acción ejecutada, el objetivo atacado, los resultados no autorizados y cuáles fueron los objetivos extraídos del ataque. También en la Figura 2.2 viene incluida una taxonomía con ejemplos de las acciones que pueden ser clasificadas dentro de cada etapa de un incidente.

2.2. Detección de Intrusos

2.2.1. Definición

Detección de intrusos es el proceso de detectar un mal usos que ocurren en un sistema de cómputo o red y analizarlos por firmas o posibles incidentes que son violaciones o amenazas inminentes de violación de políticas de seguridad, políticas de uso aceptable o políticas de seguridad estándar. La prevención de intrusos es el proceso de realizar detección de intrusos e intentar detener el posible incidente detectado. Los sistemas de detección y prevención de intrusos (IDPS por sus siglas en inglés) son principalmente enfocados en identificar posibles incidentes, registrar información de ellos, intentar detenerlos y reportarlos al administrador de seguridad. La intrusión detectada puede ser efectuada desde el exterior y/o interior de una red o segmento que derive de ella. Algunas organizaciones usan los IDPSs con otros propósitos, ya sea para identificar problemas con sus políticas de seguridad, documentar las amenazas existentes o para disuadir a los individuos de violaciones de las políticas de seguridad [19].

La miríada de posibles características para describir un mal uso, ha permitido un considerable desacuerdo entre la mayoría de las definiciones básicas de un sistema de detección. Las definiciones más usadas comúnmente y tal vez una de las más familiares pero no exactas son:

- **Detección de intrusos:** Detectar acceso no autorizado a una red de computadoras.
- **Detección de un mal uso:** Detectar actividades que empareja anomalías con patrones explícitos.
- **Detección de anomalías:** Detectar cambios de comportamientos aceptables en los perfiles.
- **Falso-Positivo:** Una alarma que no es un mal uso. Los Falsos-Positivos consumen recursos y tiempo.
- **Falso-Negativo:** Una mal uso que no es detectada o alertada.

2.2.2. Taxonomías

Debido a los extensos desarrollos e implementaciones de sistemas de detección de intrusos, han surgido diversas clasificaciones de acuerdo al tipo de respuesta que ofrecen ante las intrusiones, aplicaciones, lugares en donde se sitúan, entre otros. El uso de la taxonomía depende de lo que el autor o el desarrollador del sistema de detección interprete o quiera dar a entender. En el presente trabajo se mencionarán tres taxonomías que se usan para la clasificación de los IDSs.

Unas de las taxonomías usadas para la clasificación de IDSs, son:

- Punto de detección. Esta taxonomía se basa en la ubicación física del sistema de detección dentro de la arquitectura de red.
  - Sistema de detección de intrusos basado en host (HIDS, por sus siglas en inglés)
  - Sistema de detección de intrusos basado en red (NIDS, por sus siglas en inglés)

9

- Método de detección. Esta taxonomía hace referencia a los métodos que implementa el IDS para realizar la detección de una intrusión.
  - Firmas
  - Comportamiento
  - Anomalías

2.2.2.1. Taxonomía por punto de detección

- **Basados en Host.** Son sistemas de detección que se implementan en un equipo de cómputo cuyo análisis, por parte del sistema de detección, se hace únicamente sobre los eventos que ocurren dentro del equipo de cómputo. Algunos ejemplos de cómo se caracteriza un sistema de detección de intrusos basado en host es un análisis del tráfico de red (sólo de ése equipo, tanto entrante como saliente), sistema de registro o bitácoras, procesos corriendo, acceso a archivos, modificación, entre otros. Como se mencionó, únicamente el análisis para encontrar intrusiones se va a realizar enfocado al equipo en donde fue instalado el sistema de detección, no involucrando entidades externas en su análisis. Algunas ventajas que se tiene con la implementación de ésta categoría es la granularidad de análisis, que dependiendo de la profundidad con la que se requiere hacer el análisis sobre el sistema es la que se implementará y así poder manejar distintos niveles de análisis dependiendo del equipo analizado. Mientras que unas de las grandes desventajas que se presentan con la de ésta categoría es el consumo de recursos del equipo sobre el cuál se implementa, lo que implica tener considerado espacio y procesamiento para el mismo, así como también el uso de programas que cifren las bitácoras generadas por el sistema de detección para garantizar la integridad de la información contenida en ellos.

- **Basados en Red.** Un sistema de detección de intrusos basado en red analiza el tráfico de red de un segmento de red en específico, también analiza protocolos de red, transporte y aplicación para la identificación de actividades anómalas. La implementación de los sensores de un sistema de detección de intrusos basado en red va a depender de la arquitectura de red en donde se establecerá el IDS, ya que con base en ello se puede implementar los sensores de dos maneras:

  - En línea (Inline). El sensor en línea consiste en una configuración en donde todo el tráfico de la rea a analizar pase a través de él, haciendo un simil con un cortafuegos con esa característica. De hecho, cuando se implementa un IDS con de esta manera, es muy común que se implemente en el mismo equipo de cómputo en donde se aloja el cortafuegos. El primer motivo de poner un cortafuegos y un IDS en el mismo equipo es para poder modificar las reglas del cortafuegos cuando el IDS detecte una intrusión y así evitar un incidente o detener un ataque en curso. Típicamente los IDS son puestos con o después de un cortafuegos, ya que se supone que el cortafuegos debería de ser quien detenga la mayor parte de los ataques al ser la primera línea de defensa.

  - Pasivo. La característica que tienen los sensores pasivos es que el tráfico a analizar no pasa a través de él, como lo hacen los sensores en línea, sino que su análisis se basa en una copia del tráfico de la red. Los sensores pasivos son generalmente usados para el análisis de redes en específico; por ejemplo: divisiones entre redes, segmentos clave de red, como lo puede ser el análisis a una subred que esté diseñada como zona desmilitarizada. Los sensores pasivos pueden analizar tráfico con diferentes métodos, éstos son:

    - Puerto de expansión (Spanning Port). Este método es usado por switches, es un puerto que tiene la capacidad de ver todo el tráfico de red que fluye en el switch. La desventaja que se

presenta al implementar un sensor con este método, es que si no se configura correctamente el switch, el puerto de expansión no será posible ver todo el tráfico de la red. Otra gran desventaja que se presenta es debido a la cantidad de carga que presenta al analizar el tráfico, lo que puede provocar que si la carga es mucha no sea posible el analizar todo el tráfico de la red o provocar que el puerto de expansión sufra una inhabilitación temporal.

- Toma de Red (Network Tap). Es una conexión directa entre un sensor y el dispositivo físico de red, este puede estar conectado con fibra óptica, por ejemplo. El modo Tap provee una copia del tráfico de al red al sensor. A diferencia del puerto de expansión, que están presentes en cualquier organización, las tomas de red deben ser considerados como parte de un complemento a la red.

2.2.2.2. Taxonomía por métodos de detección

- **Basado en Firmas.** Una firma es un patrón que corresponde a una amenaza conocida. La detección basada en firmas es el proceso de comparar firmas contra eventos observados para identificar un posible incidente [19]. Algunos ejemplos de firmas son:
  - En una red de servidores, una firma sería un paquete con protocolo ICMP tipo 8, que sería una petición *ping*[4] a un servidor.
  - En una transferencia de archivos por medio del protocolo FTP, la transferencia de archivos con extensión *.exe*, que es una violación de las políticas de seguridad de una organización que solo permite la transferencia de archivos de texto, por ejemplo.

  La detección basada en firmas es muy efectiva cuando se hace el análisis sobre amenazas conocidas, pero es inefectiva si no se tiene identificada la amenaza previamente o no su estructura cambia. Por lo que con ésta detección, las amenazas que actualmente usan técnicas de evasión no serían identificadas.

  La detección basada en firmas es un método de los más sencillos y simple, debido a que se basa sólo en la comparación de las unidades de actividad actuales; como: paquetes o archivos de registro con una lista de firmas utilizando operaciones de comparación de cadenas.

- **Basado en anomalías.** La detección basada en anomalías es el proceso de comparar definiciones de qué actividades son consideradas normales contra eventos observados para identificar desviaciones significantes[19]. Este tipo de detección está implementado con *perfiles* que representan los comportamientos normales de una entidad; por ejemplo: usuarios, equipos de cómputo, conexiones de red e incluso aplicaciones. La causa de que este método sea más tardado de implementar, costoso monetariamente y efectivo con respecto al método basado en firmas, es porque cada perfil se tiene que *entrenar* por un cierto tiempo, el cual depende de la organización que lo implementa. Lo que resulta que el IDS que usa el método de detección basado en anomalías no sea un sistema genérico como lo es el basado en firmas, al contrario, es un sistema que se personaliza a cada entidad. El entrenamiento de cada perfil puede ser considerado con diferentes atributos; por ejemplo: uso de procesador, uso de memoria, tiempo de actividad, intentos de inicio de sesión, etcétera.

[4] Un ping es un programa de Internet que permite la verificación existente de una dirección IP en particular dentro de una red y si esta acepta peticiones.

El beneficio que se obtiene con esta implementación es un reconocimiento de amenazas previamente desconocidas que afecten el comportamiento del sistema. Como lo que puede ocurrir cuando un sistema es infectado con *malware*[5].

Una de las notables desventajas que se presentan en esta técnica de análisis es la dificultad, en ocasiones, para determinar por qué una alerta fue generada y para validar que dicha alerta es acertada y no un falso positivo, esta desventaja se debe a la complejidad y número de eventos que pueden causar que la alerta sea generada.

- **Basado en comportamiento.** La detección basada en comportamiento o, como algunos documentos hacen referencia, Análisis de Protocolo con Estado (Stateful Protocol Analysis, por sus siglas en inglés), es el proceso de comparar perfiles predeterminados de definiciones generalmente aceptadas de actividad de un protocolo benigno para cada estado del protocolo contra eventos observados para identificar desviaciones. A diferencia del análisis basado en anomalías, que usa perfiles adaptados a los equipos de cómputo o redes en específico, el basado en comportamiento depende del desarrollo del vendedor ya que este utiliza perfiles universales que especifican cómo protocolos particulares deberían y no deberían ser usados. El "estado.[en] el análisis de comportamiento hace referencia a que el IDS tiene la capacidad de entender y rastrear el estado de los protocolos de red, transporte y aplicación que tienen una noción de "estado".

  La principal desventaja de utilizar el análisis basado en comportamiento es que este utiliza más recursos que los anteriores debido a la complejidad del análisis y que tiene que hacer un rastreo de los estados de diferentes sesiones simultáneamente. Otro gran problema es que este análisis no puede detectar ataques que no violen las características de un comportamiento general aceptable de un protocolo.

2.2.3. Modo de operación

Los IDSs están integrados por diversos módulos que trabajan en conjunto con funciones específicas la recolección de datos y el análisis de los mismos efectuados por un sistema, también la generación de alertas y una posible respuesta del tipo pasivo, activo o pro-activo. El registro de los resultados y datos que se obtiene se almacenan en bitácoras. El motor de detección de los IDSs emplea diversas formas de análisis dependiendo de su objetivo, algunas de estas formas son: estadísticos, de Inteligencia Artificial, Sistema Inmune, Machine Learning, como es este caso, entre otras formas. La operación de estos sistemas se puede contemplar en un ambiente aislado o con la interacción de otros controles de seguridad. Este último punto es muy importante tener en consideración, ya que dependiendo de dicha operación, afecta la forma en que opera el IDS y su configuración.

Los IDS pueden ser desarrolladas tanto en hardware como en software, cada uno con sus respectivas ventajas y desventajas. El desarrollo en hardware es un equipo de cómputo que debe ser implementado la arquitectura de una red, lo que implica una instalación y configuración por personas especializadas, la principal ventaja de éste desarrollo consiste en una independencia de un equipo de cómputo, sino de la robustez de los circuitos integrados y las partes que lo constituyen. El segundo desarrollo, de software, se implementa para una operación dentro de un equipo de cómputo dedicado, el cuál dependerá totalmente del sistema operativo en el equipo, implicando esto una configuración de varios componentes del equipo, así como las propias exigencias que se requieran del equipo de cómputo; memoria, almacenamiento, velocidad de procesamiento, etc.). Su ventaja

[5]Programa malicioso que su propósito es acceder a dispositivos de manera no autorizada y sin el conocimiento del usuario.

radica en que pueden ser implementados directamente sobre la aplicación o sistema a monitorear [17].

2.2.4. Justificación de los Sistemas de Detección de Intrusos

Los sistemas de detección de intrusos (IDS por sus siglas en inglés) es un control de seguridad que debe ser implementado junto con otros controles de seguridad para fortalecer yo complicar la acción de una contra-parte, como es el caso de un *cortafuegos*[6]. La implementación de estos dos controles de seguridad son comúnmente empleados ya que el trabajo del cortafuegos el filtrar el tráfico de la red con base a un análisis de filtrado de paquetes o un filtrado de estado. Así, los IDSs reciben tráfico filtrado y reconocido para su análisis de acuerdo a diversos criterios dependiendo de la taxonomía implementada (que se definirá después).

Existen hoy en día entidades que emplean IDS dentro de los cortafuegos, ya que son la primera línea de seguridad defensiva de una entidad, con el objetivo de complementar su sistema de filtrado, y así ser más eficiente y oportuno durante un ataque o intento de intrusión. Pero dicha implementación no es que sea mejor que una de forma separada entre controles, más bien radica en otros factores como la cantidad de dispositivos existentes en la entidad que lo va a implementar, la cantidad de información que va a procesar y principalmente los recursos monetarios disponibles de la entidad, haciendo mención también que al juntar estos controles, el tiempo de procesamiento de los datos dependería mucho del hardware del dispositivo, así haciendo dependiente el flujo sin retardos de la red al dispositivo. También hay que tener en consideración que si la implementación de diferentes controles de seguridad se hace en un mismo dispositivo, existe un mayor riesgo de que si el dispositivo falla o es comprometido, la entidad pueda sufrir un ataque o una intrusión.

2.2.5. Arquitectura de IDSs

Antes de describir los modelos con los que son desarrollados los IDSs, hay que hacer mención de que las técnicas utilizadas por estos sistemas para la detección, no pueden ser generalizadas en un mismo modelo, debido al intento de mejorar un mejor análisis y una mejor clasificación de los datos que este sistema recibe, ha provocado que se puedan implementar diferentes ramas científicas con este enfoque y así intentar la implementación de nuevas técnicas aplicables a los Sistemas de Detección.

2.2.5.1. Dorothy Denning

Este modelo de IDS fue propuesto por Dorothy Denning, en donde se explica mediante similitudes informáticas qué es lo que cada componente representa en un IDS. El modelo es enfocado sobre el análisis de un sólo equipo y no de una red. Se constituye por:

- Sujetos: Hace referencia a los usuarios de un proceso, sistema o equipo de cómputo.
- Objetos: Son los dispositivos periféricos, procesos de sistema, dispositivos de almacenamiento, archivos, aplicaciones de cómputo, entre otros.
- Registro de Auditoria: Es el registro obtenido de una interacción de un sujeto sobre los objetos.
- Perfiles: Es un comportamiento registrado o patrón de comportamiento que se establecen previamente de la interacción que tiene un sujeto sobre los objetos. Los perfiles son los indicadores que van a dar lugar a la identificación de un comportamiento normal o anormal dentro de un sistema.
- Registro de Anomalías: Son aquellos registros que se generan cuando el uso y las condiciones de un objeto son anormales con respecto al perfil del sujeto en cuestión. Generalmente estos son las notificaciones al momento de crearse la actividad anómala.

[6] Un cortafuegos o *firewall*, por su nombre en inglés, son dispositivos o programas que controlan el flujo del tráfico de red entre redes o computadoras que emplean diferentes posturas de seguridad[20].

- Reglas de Actividad: Es la aplicación de una política relacionada con la actividades permitidas. Cuando una condición de la regla es cumplida, se lanza una alerta que se registra en una bitácora. Los escenciales campos dentro de la bitácora son: evento, hora del evento y el perfil hallado (de la anomalía).

Este modelo se basa en sujetos, objetos y la manipulación de los mismos, en donde dicha manipulación es monitoreada y registrada con base a los perfiles establecidos, que en todo momento se está comparando con las reglas establecidas y en espera de una anomalía, que en caso de suceder, será registrada , alertada y reportada. El modelo recibió el nombre de IDES ya que en el modelo se implementó un sistema experto, como técnica de detección de intrusos (Intrusion Detection Expert System).

2.2.5.2. Common Intrusion Detection Framework

El marco común de detección de intrusos (CIDF por sus siglas en inglés), fue un intento realizado por la agencia de proyectos de investigación avanzada de defensa (DARPA por sus siglas en inglés) para desarrollar un formato de intercambio de IDS para uso de los investigadores de la DARPA. CIDF no fue considerado como un estándar que podría influenciar el mercado comercial; sólo fue un proyecto de investigación [23]. Éste modelo sugiera el uso de GIDO (General intrusion Detection Objetc) como un componente de intercambio de datos entre los módulos del IDS y la utilización del lenguaje común de especificación de intrusos o CISL(por sus siglas en inglés) para crear las reglas de detección, el cual se asimila al lenguaje LISP. Esta arquitectura se encuentra constituido por cuatro módulos o equipos:

- **(E) Generadores de Eventos**: Es una integración de sensores que siempre están es espera de que un evento suceda en el evento a sensar y generar informes.
- **(A) Analizadores de Eventos**: Este módulo es el que se encarga de recibir la información generada de los generadores de eventos y analizarla. Una vez realizado ése análisis, detectar si existe la presencia de una intrusión con forme a los criterios establecidos previamente de un comportamiento anómalo. Pueden ofrecer una prescripción y un curso de acción recomendado.
- **(D) Base de Datos**: Esta compuesto por patrones almacenados para poder determinar si se ha visto un ataque previamente por medio de correlación de datos y así determinar si se trata de un indicio de intrusión.
- **(R) Unidad de Respuesta**: Son las acciones que se toman en caso de la detección de una intrusión, en donde se puede basar en los resultados de los módulos E. A. D para la respuesta a los eventos.

La arquitectura CIDF se debería tomar como referencia para tener como referencia los componentes que debería contener un IDS entre los diferentes fabricantes, sin embargo ésta no se ha considerado, ni como estándar, dentro del desarrollo de los IDSs debido a la complejidad de su lenguaje y el uso de GIDO para el intercambio de información.

2.2.5.3. Common Intrusion Specification Language

Describe un lenguaje que es puede ser usado para diseminar el registro de eventos, análisis de resultados y directivas de contra-medidas entre la detección de intrusos y los componentes de respuesta. Es la integración de los cuatro componentes de la arquitectura CIDF. Las capacidades básicas que esta arquitectura debe de cumplir son:

- **Información de eventos en bruto:** Consiste en una auditoría de registros y tráfico de red. Esta sección se encargaría de unir el módulo E con A.

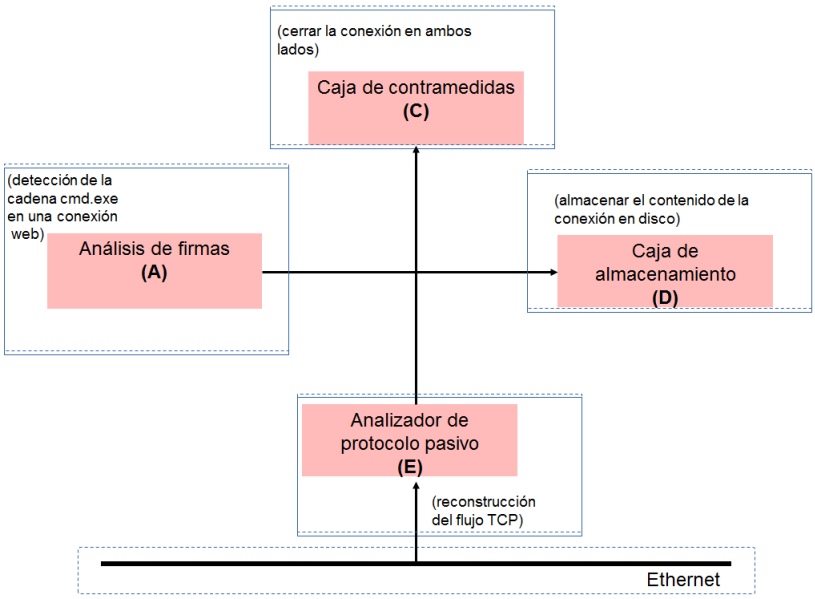


Figura 2.3: Diagrama a bloques de la arquitectura CIDF.

- **Resultados de los análisis:** Son las descripciones de las actividades anómalas y de las intrusiones detectadas en el sistema. Une el equipo A con D.
- **Prescripciones de respuesta:** Acciones realizadas para detener un ciertas actividades anómalas modificando controles de seguridad. Une los módulos A y R.

El uso de la arquitectura CISL se ha considerado bastante complicado por la implementación de la arquitectura CIDF (que ya se había considerado compleja), por lo que no llegó a ser considerada por los fabricantes de IDSs.

2.2.5.4. Autopost de AusCERT

Tras la creación de las arquitecturas CIDF y CISL, el CERT de Austraila (AusCERT) desarrolló su propio sistema con el que se trabajarían los reportes, el cuál sería sencillo y que permitiría que se analizara y se generara un informe con el lenguaje Perl. La manera en que sería construido el reporte podría ser de la siguiente:

```
Source: 216.36.45.84
Ports: tcp 111
Incident type: Network_scan
re-distribute: yes
timezone: GMT + 1300
reply: no
Time: Web 15 Mar 2000 at 14:01 (UTC)
```

Debido a la facilidad de interpretación del Autopost, se tiene una gran interoperabilidad y es fácil de construir y de analizar. Este modelo al igual que los otros no fue tomado como un estándar debido a su escasa información reportada, ya que no era suficiente para los analistas de eventos, ya que estos requería de

información detallada de los eventos para un análisis forense, por ejemplo.

2.2.5.5. Intrusion Detection Working Group

Debido a que el grupo de trabajo de ingeniería de internet (IETF[7] por sus siglas en inglés) rechazó las los enfoques de las arquitecturas CIDF y CISL, formó un equipo de trabajo llamado grupo de trabajo de detección de intrusos (IDWG por sus siglas en inglés) que como tal, no propusieron una arquitectura en específica, sino un modelo que se adaptara a las arquitecturas ya existentes. La proposición del IDWG fue:

a) El uso del lenguaje XML.

b) Para una comunicación entre los módulos de la arquitectura, le uso de un servicio de mensajería IDMEF (Intrusion Detection Message Exchange Format).

c) El uso de los protocolos IAP (Intrusion Alert Protocol) e IDXP (Intrusion Detection Exchange Protocol)

Esta propuesta aún se encuentra en evaluación, la cuál contiene cuatro borradores para ser evaluados, explicando en qué consiste las etapas que la constituyen.

- IDWG RFC 4766 (Requerimiento)

  El propósito de este es definir los formatos de información y procedimientos de intercambio para compartir información relevante entre IDSs, a de los sistemas de respuesta y a los administradores que estén en continua comunicación entre dichas partes y el IDS. Se requiere de un lenguaje que interpretar la semántica de otros IDSs en diferentes plataformas. La interacción entre el emisor y el receptor, debe tener implementadas formas de protección como un cifrado de los datos, debe pasar a través de un cortafuegos de manera transparente sin que se comprometa la seguridad del sistema de detección. Otras características deseables del comportamiento de los IDSs, es la automatización de las respuestas, donde las alertas sean compartidas entre los módulos del sistema empleando un formatos de prioridades para así, diferenciar los mensajes de intercambio generados habitualmente entre los módulos y las alertas de detección.

- IDMEF-XML RFC 4765 (Intrusion Detection Message Exchange Format - XML)

  Este borrador describe el intercambio que se debe de llevar a cabo entre los IDSs a desarrollar utilizando el lenguaje *XML*[8]. La implementación de este lenguaje para compartir datos relevantes entre los sistemas resulta más eficiente, debido a que la estructura que se usa para la lectura y escritura de dichos datos es mucho más fácil de hacerla.

- BEEP TUNNEL RFC 3620 (Block Extensible Exchange Protocol)

  Se plantea el uso de un proxy para la comunicación entre dos equipos de cómputo que se encuentren en diferentes redes. Por lo cual será empleado un túnel a través del proxy para que estos dos equipos en diferentes redes se puedan comunicar entre sí.

- BEEP IDXP RFC 4767 (Intrusion Detection Exchange Protocol)

[7] Organización internacional de normalización que tiene como objetivo hacer del Internet funcione mejor produciendo alta calidad, documentos técnicos relevantes que influencien la forma de diseñar, usar y manejar Internet para las personas [25].

[8] Extensible Markup Language is un simple, muy flexible formato de texto derivado de del SGML (ISO 8879). Originalmente diseñado para conocer los retos de la edición electrónica a gran escala [26].

Se describen las normas que deben ser empleadas entre la comunicación de las entidades de los IDSs en diferentes redes. En donde el protocolo establece la creación de un sesión de un túnel BEEP para el cifrado de la comunicación y en donde la comunicación, punto a punto, sea de una manera transparente entre el paso de los proxies y así garantizar la confidencialidad de los datos transmitidos en el túnel. Los perfiles de alertas manejados por el IDXP operan dependiendo del nivel de la prioridad de la misma, pues cada nivel de prioridad de alertas tiene su propia sesión para la transmisión de éstas entre las entidades participantes (red, host, aplicación, etc.).

Con base a lo mencionado anteriormente, se justifica el uso de este modelo. Pues la diferencia que tiene este con las arquitecturas antes descritas, se tiene que el objetivo del mismo es dar respuesta a la interoperabilidad que se debe de llevar a cabo entre los diferentes fabricantes de IDSs. Con ello, sugiriendo que la comunicación entre los componentes, y la generación de los reportes, puedan ser adaptados a las necesidades de los datos de interés requeridos del sistema que se protege. Para poder llevar a cabo los puntos mencionados, se puede hacer gracias a las características que ofrece el lenguaje XML, ya que fue diseñado como un estándar para el intercambio de información multiplataforma, ofreciendo una compatibilidad entre sistemas.

2.2.6. Componentes elementales de un IDS

Con las propuestas descritas de las arquitecturas y modelos de las buenas prácticas para el desarrollo de IDSs, y con base a las especificaciones que nos proporciona el documento NIST 800-94, se puede generar un esquema genérico de los componentes elementales que deben ser implementados en todo IDS. Dichos componentes elementales que deben estar en un IDS, son:

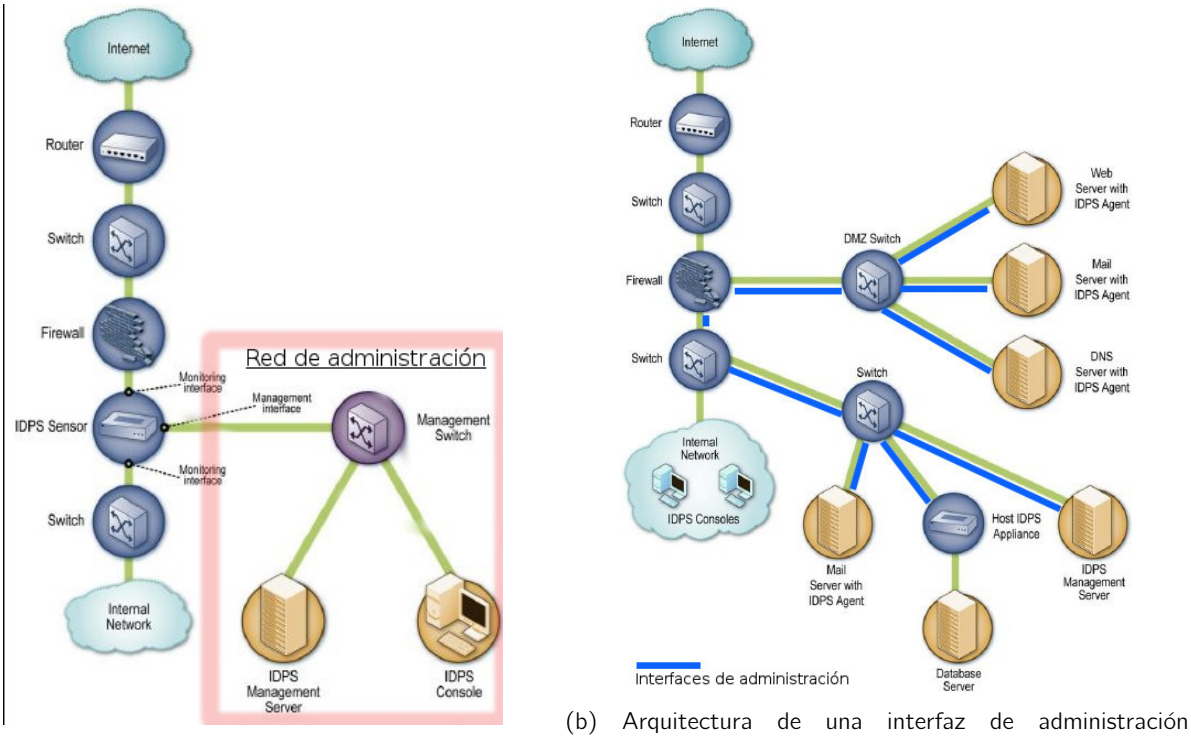
- **Sensor o Agente.** Los sensores y los agentes moniorean y analizan la actividad. El término *sensor* es mayormente usado para IDPS que monitorean redes, incluyendo las basadas en red, no guiadas y tecnologías de análisis de comportamiento de red. El término *agente* es usualmente utilizado para referirse a las tecnologías usadas para un análisis basado en host en un IDPS.

- **Servidor Administrador.** Este servidor administrador es el que recibe información de los agentes o sensores y administrarlos.Existen algunos servidores que hacen la función de análisis sobre la información enviada por los agentes o sensores e identificar eventos que por sí solos, los agentes o sensores no pueden identificarlos.

- **Servidor de Base de Datos.** Es un servidor en donde se va a almacenar toda la información registrada por los eventos o agentes, o también por el administrador. Esta información almacenada, no necesariamente será de eventos registrados, también puede ser del estado del sistema o de su comportamiento en su ejecución. Este componente es importante para los administradores ya que con la información contenida en este componente, se pueden identificar eventos no alertados o también conocidos como falso negativo para su posterior análisis o una proposición de modificaciones a los sensores o agentes.

- **Consola.** Es un programa que brinda una interfaz para el IDS para los usuarios y los administradores de éste. La consola es regularmente instalada de manera aislada en un equipo de cómputo común, como una computadora de escritorio o una computadora personal. Las consolas pueden ser usadas tanto para la administración de los agentes o sensores, como para un monitorear y analizar.

- **Respuesta.** El propósito de éste componente es proporcionar respuestas tanto Activas, Pasivas o Pro-activas. Las respuestas activas son aquellas en las que al momento de detectar una intrusión, se toman decisiones pre-configuradas en el sistema, como un bloqueo de direcciones IP, finalización de una conexión e incluso, la modificación de reglas de un control de seguridad. La pasivas son aquellas en las que se espera la intervención de un administrador para tomar las acciones necesarias sobre el evento ocurrido. Las respuestas pro-activas emplean el concepto del cómputo proactivo, es decir, la anticipación de una acción basada en lo que percibe del medio físico que se le va presentando. Cabe



(a) Arquitectura de una red de administración.

(b) Arquitectura de una interfaz de administración implementada en la red de producción.

Figura 2.4: Arquitecturas de Red

mencionar que independientemente de la respuesta a implementar elegida, las tres respuestas envían notificaciones/alertas de eventos en curso o pasados.

2.2.7. Arquitecturas de Red

Los componentes de los sistemas de detección de intrusos pueden ser conectados entre sí, a través de las redes que los implementa o a través de redes separadas estrictamente diseñadas para la comunicación entre estos componentes, así evitando una conexión desde redes no autorizadas a los componentes. A la red dedicada para la comunicación de estos componentes es conocida como *Red de administración* (ver Figura 2.4a), en donde los sensores o agentes son administrados a través de la red de administración (zona dentro del recuadro rojo). En caso de que la organización no pueda crear una red aislada para dicha comunicación, cada sensor o agente del host debe tener una red virtual aislada para su comunicación. A esta red virtual implementada, en lugar de la red física aislada, se le conoce como *interfaz de administración* (ver Figura 2.4b). Así, los agentes o sensores no pueden pasar información entre redes ni interfaces, haciendo que los servidores de administración, bases de datos y consolas estén únicamente apegadas a la red de administración. Los beneficios de realizar dicha separación de redes, es ocultar la existencia de un IDS dentro de la red de la organización a una contra parte, la protección del IDS ante ataques y asegurar una banda de ancho adecuada para su funcionamiento. Las desventajas que se generan al emplear esta arquitectura, son: incremento del costo para la creación de las redes y la inconveniencia para los usuarios y administradores del sistema para la administración del mismo, pues dicha administración se encuentra en diferentes equipos de cómputo.

18

2.3. Cross-site Scripting (XSS)

2.3.1. Definición

Los ataques *Cross-Site Scripting (XSS)* son un tipo de inyección, en la que se insertan secuencias de comandos maliciosas en sitios web benignos y de confianza. Los ataque XSS se producen cuando un atacante utiliza una aplicación web para enviar un código malicioso, generalmente en forma de una secuencia de comandos del navegador, a un usuario final diferente. Las fallas que permiten que estos ataques tengan éxito son bastante generalizadas y se producen en cualquier lugar donde un aplicación utiliza la entrada de un usuario dentro de la salida que genera sin validarla o codificarla.

Un atacante puede usar XSS para enviar una secuencua de comando malicioso a un usuario desprevenido. El navegador del usuario final no tiene forma de saber que el script no es de confianza, y ejecutará el script. Debido a que piensa que el script proviene de una fuente de confianza, el script malicioso puede acceder a cualquier cookie, token de session, u otra información confidencial que el navegador mantenga y que se utilice en el sitio. Estos scripts pueden incluso reescribir el contenido de la página HTML. [28].

2.3.2. Actores en un ataque XSS

Antes de describir detalladamente cómo funciona un ataque XSS, necesitamos definir a los actores involucrados en un ataque XSS. En general, un ataque XSS involucra a tres actores: el sitio web, la víctima y el atacante.

- El **sitio web** ofrece páginas HTML a los usuarios que las soliciten. La base de datos del sitio web es una base de datos que almacena parte de la entrada del usuario incluida en las páginas del sitio web.
- La **víctima** es un usuario normal del sitio web que solicita sus páginas usando su navegador.
- El **atacante** es un usuario malintencionado del sitio web que intenta lanzar un ataque contra la víctima mediante la explotación de una vulnerabilidad XSS en el sitio web. El servidor del atacante es un servidor web controlado por el atacante con el único propósito de robar la información delicada de la víctima.

2.3.3. Tipos de Cross-Site Scripting

[29] Primero, se identificaron dos tipos principales de XSS, XSS almacenado y XSS reflejado. En 2005, Amit Klein definió un tercer tipo de XSS, que acuñó XSS basado en DOM. Estos 3 tipos de XSS se definen de la siguiente manera:

2.3.3.1. XSS almacenado (AKA persistente o tipo I)

XSS almacenado generalmente se produce cuando la entrada del usuario se almacena en el servidor de destino, como en una base de datos, en un foro de mensajes, registro de visitantes, campo de comentarios, etc Y entonces una víctima es capaz de recuperar los datos almacenados de la aplicación web sin que la información sea segura para el navegador. Con la llegada de HTML5 y otras tecnologías de navegación, podemos ver que la carga útil del ataque se almacena permanentemente en el navegador de la víctima, como una base de datos HTML5, y nunca se envía al servidor en absoluto.

2.3.3.2. XSS reflejado (AKA no persistente o tipo II)

XSS reflejado se produce cuando la entrada del usuario es devuelta inmediatamente por una aplicación web en un mensaje de error, resultado de búsqueda o cualquier otra respuesta que incluya parte o la totalidad de la entrada proporcionada por el usuario como parte de la solicitud, sin que dichos datos se hagan seguros. Renderizar en el navegador, y sin almacenar permanentemente los datos proporcionados por el usuario. En algunos casos, los datos proporcionados por el usuario nunca pueden salir del navegador.

2.3.3.3. XSS basado en DOM (AKA tipo-0)

Como define Amit Klein, XSS basado en DOM es una forma de XSS donde el flujo entero de datos contaminados desde la fuente al destino tiene lugar en el navegador, es decir, la fuente de los datos es el DOM, el destino está también en el DOM, y el flujo de datos nunca sale del navegador.

2.3.4. Riesgos de los ataques XSS

Las consecuencias de lo que un atacante puede hacer con la capacidad de ejecutar JavaScript en una página web no pueden destacarse inmediatamente, sobre todo porque los navegadores ejecutan JavaScript en un entorno muy controlado y que JavaScript tiene acceso limitado al sistema operativo del usuario y los archivos del usuario.

Sin embargo, al considerar que JavaScript tiene acceso a casi todas las herramientas disponibles por el navegador web, es más fácil entender cómo los atacantes creativos pueden obtener con JavaScript.

- JavaScript malicioso tiene acceso a todos los mismos objetos que el resto de la página web tiene, incluyendo el acceso a las cookies. Las cookies se usan a menudo para almacenar fichas de sesión, si un atacante puede obtener una cookie de sesión de un usuario, pueden hacerse pasar por ese usuario.
- JavaScript puede leer y hacer modificaciones arbitrarias en el DOM del navegador (dentro de la página que JavaScript está ejecutando). El atacante puede insertar un formulario de inicio de sesión falso en la página utilizando la manipulación DOM, establecer el atributo de acción del formulario para orientar su propio servidor y, a continuación, engañar al usuario para que envíe información confidencial.
- JavaScript puede utilizar XMLHttpRequest para enviar peticiones HTTP con contenido arbitrario a destinos arbitrarios.
- JavaScript en los navegadores modernos puede aprovechar las APIs HTML5, como acceder a la geolocalización de un usuario, cámara web, micrófono e incluso los archivos específicos del sistema de archivos del usuario. Mientras que la mayoría de estas API requieren el opt-in del usuario, XSS junto con ingeniería social ingeniosa puede traer a un atacante un largo camino.
- El atacante puede registrar un detector de eventos de teclado mediante addEventListener y luego enviar todas las pulsaciones del usuario a su propio servidor, registrando potencialmente información confidencial como contraseñas y números de tarjetas de crédito.

2.3.5. Prevención

Recordando que un ataque XSS es un tipo de inyección de código: la entrada del usuario se interpreta erróneamente como código de programa malicioso. Para evitar este tipo de inyección de código, es necesario realizar un manejo de entrada seguro.

- **Codificación**: La mayoría de las veces, la codificación se debe realizar siempre que se incluya la entrada del usuario en una página. Escapa la entrada del usuario para que el navegador lo interprete sólo como datos, no como código.

- **Validación**: En algunos casos, la codificación tiene que ser reemplazada o complementada con validación. Filtra la entrada del usuario para que el navegador lo interprete como código sin comandos maliciosos.
- **Contexto**: El manejo seguro de las entradas debe tener en cuenta el contexto de una página en la que se inserta la entrada del usuario. El manejo seguro de las entradas debe realizarse de manera diferente dependiendo del lugar en el que se inserte la entrada del usuario en una página.
- **Manejo de entrada Entrante/Saliente**: Para evitar todos los tipos de ataques XSS, el manejo de entrada segura debe realizarse tanto en el lado del cliente como en el código del lado del servidor. El manejo seguro de las entradas puede realizarse cuando su sitio web recibe la entrada (entrada) o justo antes de que su sitio web inserte la entrada en una página (saliente).
- **Servidor/Cliente**: El manejo seguro de las entradas se puede realizar tanto en el lado del cliente como en el lado del servidor, los cuales son necesarios bajo diferentes circunstancias.
- **Política de seguridad de contenido**: La Política de seguridad de contenido proporciona una capa adicional de defensa cuando falla el manejo de entrada segura. Es un mecanismo del lado del navegador que te permite crear listas blancas de origen para los recursos del lado del cliente de tu aplicación web, p. JavaScript, CSS, imágenes, etc. CSP a través de una cabecera HTTP especial ordena al navegador que sólo ejecute o procese recursos de esas fuentes.
- **HTTPOnly cookies**: Usar cookies que únicamente sean accesibles mediante HTTP usando la bandera HTTPOnly. Cualquier cookie que se tenga no se podrá acceder mediante ningún código Javascript escrito.

2.4. Aprendizaje Automático

2.4.1. Definición

El aprendizaje automático o *machine learning*,por su nombre en inglés, es el estudio que se encarga de saber cómo realizar programas computacionales que puedan mejorar su desempeño a través de experiencias. El aprendizaje automático ha sido probado y clasificado como una buena herramienta en particulares casos, como son: (a) en donde existe poco entendimiento del dominio del problema que ocasiona un poco conocimiento por parte de los humanos para desarrollar algoritmos efectivos que lo resuelvan; (b) dominios en donde se tenga una extensa base de datos que contenga regularidades implícitas a ser descubiertas; o (c) dominios en donde los mismos programas deban estar en un cambio constante de adaptación a ciertas condiciones. El aprendizaje máquina se relaciona con el diseño y desarrollo de algoritmos que, de forma autónoma, sean capaces de adquirir e implementar conocimiento previo para mejorar la efectividad de sus tareas y que éstas sean eficientes [27].

2.4.2. Concepto

En el concepto de aprendizaje una función objetivo es representada como un conjunto de restricciones sobre atributos. La hipótesis de un espacio H consiste en el enrejado de posibles conjunciones de restricciones de atributo para un dominio de problema dado. Una estrategia de búsqueda de menor compromiso es adoptada para eliminar hipótesis en H que no son consistentes con el entrenamiento establecido D. Esto resultará en una estructura llamada el espacio de versión, el subconjunto de hipótesis que son consistentes con los datos de entrenamiento. El algoritmo, llamado la eliminación de candidatos, utiliza los operadaores de generalización y especialización para producir el espacio de version con consideración a H y D. Esto se basa en un sesgo de lenguaje (o restricción) que establece que la función objetivo está contenida en H. Esto es un método de

aprendizaje ansioso y supervisado [27].

2.4.3. Métodos de aprendizaje

2.4.3.1. Árboles de Decisión

La búsqueda en el aprendizaje de árboles de decisión suele estar guiada por una medida de ganancia de información basada en entropía que indica la cantidad de información que una prueba en un atributo produce. Los algoritmos de aprendizaje usualmente tiene una inclinación por los árboles de decisiones. El uso de éste método de aprendizaje puede provocar el ruido en los datos por una sobre alimentación en los árboles. Esto se debe a que no se puede tener un control sobre el acomodo del conocimiento previo que se adquirió durante el proceso. Se dice que este método de aprendizaje es ansioso, supervisado e inestable.

2.4.3.2. Redes Neuronales

En el aprendizaje de redes neuronales, el aprendizaje de una función objetivo se asemeja a encontrar pesos de una red de tal manera que las salidas de las redes sean las mismas que los resultados esperados como fueron especificados en los datos de entrenamiento, siempre que sea dada una estructura fija de red. Lo se considera como un vector de pesos, se define como una función objetivo. Tal consideración hace que la interpretación y lectura de la función objetivo sea difícil para el ser humano. Este método de aprendizaje aproximado es ansioso, supervisado e inestable, y además no puede acomodar conocimientos previos.

2.4.3.3. Aprendizaje Bayesiano

El aprendizaje Bayesiano ofrece un enfoque probabilístitco a la interferencia basada en suposiciones, cuya cantidad de interés es dictado por la distribución de probabilidad, y para alcanzar las óptimas decisiones o clasificaciones deben ser realizadas haciendo un razonamiento de las probabilidades junto con la observación de los datos. Este método de aprendizaje se conforma de dos grandes grupos, ambos basados en los resultados de un aprendiz: el primer grupo que produce el mayor hipótesis dado los datos de entrenamiento, el segundo grupo que produce la mayor clasificación de una nueva instancia dado los datos de entrenamiento. Una función objetivo es así explícitamente representada en el primer grupo, pero implícitamente definida en el segundo grupo. La ventaja que se tiene con éste método es la capacidad de acomodar los conocimientos previos (en forma de red Bayesiana, probabilidades previas para cada hipóstesis candidata, o a una distribución de probabilidad sobre los datos observados para una posible hipótesis). Si llega a suceder que ocurra un casi invisible para este método, dicho caso se clasifica basado en predicciones de múltiples hipótesis. Este puede aumentar proporcionalmente bien con datos largos. La ventaja al no tener problemas con el acomodo de conocimientos previos es que no tiene problemas con el ruido de los datos, lo que provocaría una mala clasificación. Una desventaja que se presenta es la dificultad de clasificación con los pequeños conjuntos de datos. Este método de aprendizaje es ansioso y supervisado y no requiere búsquedas durante el proceso de aprendizaje. También adopta una inclinación que se basa en el principio de longitud mínima de descripción.

2.4.3.4. Algoritmos Genéticos y Programación Genética

Estos métodos de clasificación han sido inspirados por la biología. Una función objetivo es representada como una cadena de *bit*[9] en algoritmos genéticos, o como programas en programación genética. El proceso de búsqueda inicia con una población basada en una hipótesis inicial. Para darle un incremento de exactitud a los siguientes miembros de la siguiente generación de población, los actuales deben ser sometidos a operaciones de intercambio (crossover) y mutación. Con cada iteración de la población, las hipótesis de la población actual

[9] Unidad mínima de información en un sistema de cómputo.

es evaluada con la consideración a una medida dada de aptitud, siendo hipótesis más apta seleccionada como un miembro de la siguiente generación. El proceso de búsqueda termina cuando los valores de aptitud de la población actual ha superado los límites de algún umbral.Los algoritmos son generacionales y de estado estable (*steady-state*).

2.4.3.5. Basado en Instancias

Este método de aprendizaje es un aprendizaje perezoso en el sentido de que se enfoca en la generalización más allá de los datos de entrenamiento que se difieren hasta que un caso no visto necesite ser clasificado. Se dice que la función objetivo no es definida, sin embargo, el aprendiz regresa un valor de una función objetivo cuando se clasifica un caso no visto. El proceso de búsqueda es basado en razonamientos estadísticos. Consiste en identificar los datos de entrenamiento que están cerca al caso no visto y con ellos producir el valor de la función objetivo basada en sus vecinos. Los algoritmos populares para este método son: *K-nearest neighbors, cas-based reasoning* y *locally weighted regression*.

2.4.3.6. Aprendizaje Reforzado

El aprendizaje reforzado es el método de aprendizaje más general. Aborda el problema de cómo aprender una secuencia de acciones llamada una estrategia de control de información de recompensa indirecta y demorada (refuerzo). Este método es ansioso y de aprendizaje no supervisado. Su búsqueda es llevada a cabo a través de episodios de entrenamiento. Dos enfoques principales que existen para el aprendizaje reforzado son: basado en modelo y libre de modelo. El algoritmo más conocido para el enfoque libre de modelo y el basado en modelo es el *Q-learning*, en donde las acciones con máximo valor Q son las preferidas.

2.4.3.7. Aprendizaje de Múltiples Instancias

El aprendizaje múltiples instancias negocia con la situación en la que cada ejemplo de entrenamiento podría tener diversas variantes de instancias. Es propuesto como una variante del aprendizaje supervisado con un conocimiento incompleto acerca de las etiquetas de los ejemplos de entrenamiento. En el lenguaje supervisado, cada instancia de entrenamiento se establece específicamente con etiquetas de valores reales o con etiquetas discretas, mientras que en el aprendizaje de múltiples instancias las etiquetas son únicamente asignadas a bolsas de instancias. En el caso binario (el más usual en la implementación de éste método), una bolsa es etiquetada como positiva si al menos una instancia de la misma en ésa bolsa es positiva, y la bolsa es etiquetada negativa si todas las instancias de ella son negativas.

2.4.3.8. Aprendizaje No Supervisado

En el aprendizaje no supervisado, el aprendiz está para analizar un conjunto de objetos que no tienen una clase de etiqueta, y discierne de categorías a la que cada objeto pertenece. Dado un conjunto de objetos, se tiene dos grupos de enfoques en un aprendizaje no supervisado: métodos de estimación de densidad que pueden ser usados para crear modelos estadísticos para capturar o explicar patrones reconocidos o estructuras interesantes detrás de la entrada, y métodos de extracción de características que pueden ser usados para recoger características estadísticas, ya sean regularidades o irregularidades, de la entrada. La desventaja que se presenta con este tipo de aprendizaje es que no tienen una cantidad o medida de éxitos debido a que se es muy difícil de establecer y validar.

CAPÍTULO 3

Especificaciones del proyecto

3.1. Objetivo

3.1.1. Objetivo general

Desarrollar un sistema del tipo detector de intrusos con el fin de ayudar a los administradores web a tener una defensa y un alertador de ataques XSS que esté sufriendo su sitio o sistema.

3.1.2. Objetivos específicos

- Generar de forma artificial los ataques.
- Detectar y alertar de ataques XSS.
- Proteger al sistema de un ataque XSS de manera básica.
- Mostrar al usuario las estadísticas e información sobre los ataques que ha sufrido el sistema portador.

3.2. Metodología

Se consideró utilizar el modelo de desarrollo evolutivo o prototipo evolutivo ya que construye una serie de grandes versiones sucesivas de un producto. Sin embargo, mientras que la aproximación incremental presupone que el conjunto completo de requerimientos es conocido al comenzar, el modelo evolutivo. En este modelo, los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. Los desarrolladores construyen una implementación parcial del sistema que recibe sólo estos requerimientos. Aplicaremos la metodología de la siguiente forma:

- Planificación: especificaremos los requerimientos que se necesitan en cada uno de los módulos para su correcto funcionamiento.
- Desarrollo: se realizará el módulo y las pruebas de funcionamiento antes de proseguir para la retroalimentación del cliente.

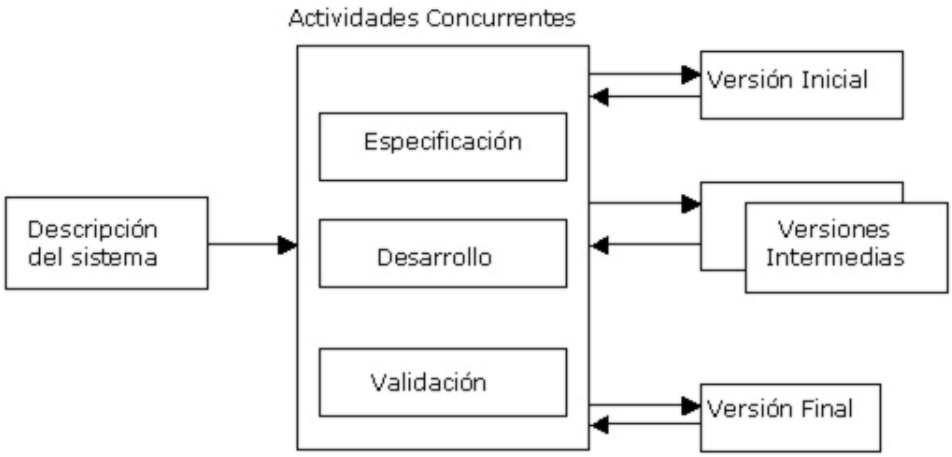


Figura 3.1: Ciclo de vida de la metodología de prototipos.

- Retroalimentación de parte del cliente: las pruebas serán realizadas por el cliente para poder tener la retroalimentación y así exponer al sistema a una prueba en el entorno que será utilizado finalmente.
- Modificaciones: si se requieren se realizaran mejoras de acuerdo a los resultados obtenidos por el cliente, de no tener alguna modificación se pasara a la siguiente fase hasta terminar el sistema.

Los módulos del proyecto así como la documentación serán realizados en el siguiente orden:

- Sensor y analizador
- Motor de interferencia
- Acciones de respuesta
- Registrador de eventos

3.3. Arquitectura

En la Figura 3.2 puede observar un esquema del proceso por el cual el sistema se sometería al estar en funcionamiento. La descripción del proceso es el que se especifica debajo de la figura.

El ataque XSS será ejecutado sobre una página web vulnerable. Al llegar el ataque al servidor, el tráfico generado por este será analizado dentro del mismo por el agente.

Al ser detectado el ataque, el agente envía los datos del ataque al servidor administrador.

El servidor administrador registra el evento en una base de datos y envía las notificaciones del ataque. La primera notificación será en la interfaz de monitoreo del sistema y la otra notificación se enviará a las cuentas configuradas de SMS y correo electrónico del administrador encargado.

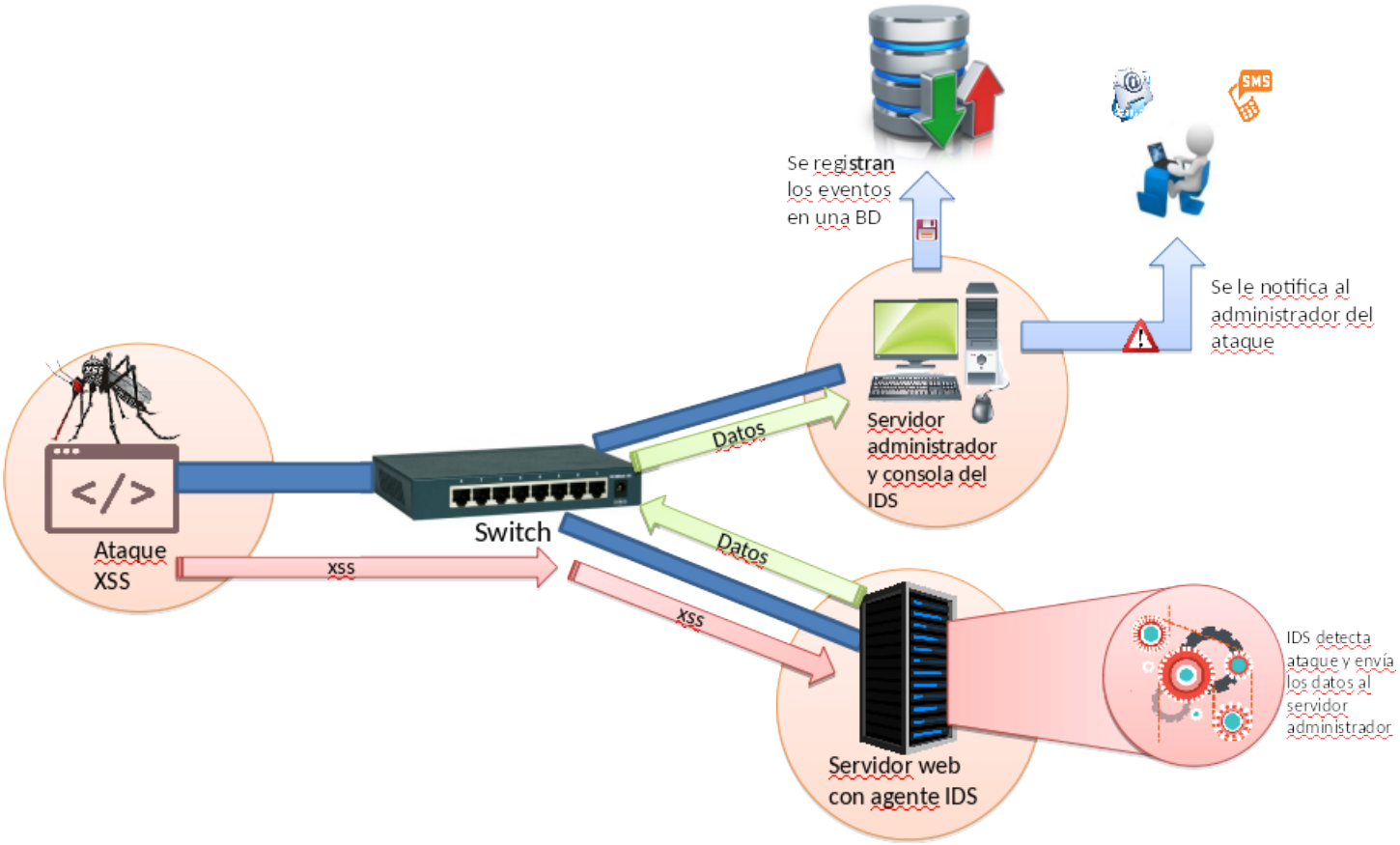


Figura 3.2: Esquema de la arquitectura a implementar.

26

El evento será almacenado en una base de datos para su posterior análisis por el administrador encargado.

El administrador recibirá las notificaciones correspondientes por parte del sistema con la información necesaria del evento.

El administrador una vez notificado, podrá ver los registros de los eventos ocurridos para tomar las medidas necesarias a la seguridad de sus aplicaciones web.

27

Bibliografía

[1] Kaplan J., Sharma S. & Weinberg A. (2011). "Meeting the cybersecurity challenge". McKinsey & Company. Recuperado 12 Marzo 2017, de http://www.mckinsey.com/business-functions/business-technology/our-insights/meeting-the-cybersecurity-challenge

[2] THE EDITORS AT CYBERSECURITY VENTURES. (2014). "The Cybersecurity Market Report covers the business of cybersecurity, including market sizing and industry forecasts, spending, notable M&A and IPO activity, and more..Çybersecurity Ventures. Recuperado 26 Septiembre 2016, de http://cybersecurityventures.com/cybersecurity-market-report/

[3] King, S. (2016). "Assessing the real risk of being online". ComputerWeekly. Recuperado 26 Septiembre 2016, de http://www.computerweekly.com/feature/Assessing-the-real-risk-of-being-online

[4] Computer Hope (2016). "Why do people hack computers?". Computerhope.com. Recuperado 26 Septiembre 2016, de http://www.computerhope.com/issues/ch001530.htm

[5] Cloudbric. (2016). "6 Reasons Why Hackers Want to Hack Your Website". Recuperado 26 Septiembre 2016, de https://www.cloudbric.com/blog/2015/10/6-reasons-why-hackers-want-to-hack-your-website/

[6] Imperva Inc. (2016). "2015 Web Application Attack Report (WAAR)". WAAR 2015. Recuperado de https://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed6.pdf

[7] Gutierrez, E. (2009). "JavaScript". 1st ed. Barcelona: Ed. ENI, p.233.

[8] Assis, R. (2016). "Primero post de la serie sobre vulnerabilidades XSS". Sucuri Español. Recuperado 19 Diciembre 2016, de https://blog.sucuri.net/espanol/2016/04/pregunte-sucuri-que-es-una-vulnerabilidad-xss.html

[9] (2016). "Qué es y cómo funciona un ataque Cross - Site Scripting". Hostalia. Recuperado 19 Diciembre 2016, de http://pressroom.hostalia.com/wp-content/themes/hostalia_pressroom/images/cross-site-scripting-wp-hostalia.pdf

[10] Ramos Pereira, K. (2016). "Cross-Site Scripting". Revistasbolivianas.org.bo. Recuperado 20 Noviembre 2016, de http://www.revistasbolivianas.org.bo/scielo.php?pid=S1997-40442013000100023&script=sci_arttext

[11] (2014). "Su navegador esta desnudo: por qué los navegadores protegidos siguen siendo vulnerables.". Panda Security. Recuperado 27 Noviembre 2016, de http://resources.pandasecurity.com/enterprise/solutions/7.%20WP%20PCIP%20ESP%20Su%20Navegador%20esta%20desnudo.pdf

[12] Greene, T. (2016). "8 cyber security technologies DHS is trying to commercialize". Network World. Recuperado 26 Septiembre 2016, de http://www.networkworld.com/article/3056624/security/8-cyber-security-technologies-dhs-is-trying-to-commercialize.html

[13] (2016). "School of Computer Science and Information Technology University of Nottingham". Firewalls, Intrusion Detection Systems and Anti-Virus Scanners (p. 57). NOTTINGHAM NG8 1BB, UK. Recuperado de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.2262&rep=rep1&type=pdf

[14] Mookhey , K. K., Nilesh, B. (2011). "Detection of SQL Injection and Cross-site Scripting Attacks — Symantec Connect". Symantec.com. Recuperado 26 Septiembre 2016, de http://www.symantec.com/connect/articles/detection-sql-injection-and-cross-site-scripting-attacks10

[15] Tim, K. (2016). "Strategies to Reduce False Positives and False Negatives in NIDS — Symantec Connect". Symantec.com. Recuperado 26 Septiembre 2016, de http://www.symantec.com/connect/articles/strategies-reduce-false-positives-and-false-negatives-nids

[16] Garcia-Alfaro, J. & Navarro-Arribas, G. (2005). "Prevención de ataques de Cross-Site Scripting en aplicaciones Web".Recuperado 25 Noviembre 2016, de http://www-public.tem-tsp.eu/~garcia_a/web/papers/recsi08-xss.pdf

[17] González Márquez, V. (2009). "Sistema de detección de intrusos basado en sistema experto (Tesis de maestría)". Centro de Investigación en Computación. México.

[18] National Institute of Standards and Technology,. (2007). "Guidelines on Securing Public Web Servers"(p. 121). Washington.

[19] National Institute of Standards and Technology,. (2007). "Guide to Intrusion Detection and Prevention Systems (IDPS)"(p. 9). Washington.

[20] National Institute of Standards and Technology,. (2009). "Guidelines on Firewalls and Firewall Policy"(p. 7). Washington.

[21] National Institute of Standards and Technology,. (2001). "Underlying Technical Models for Information Technology Security"(p. 6). Washington.

[22] National Institute of Standards and Technology,. (2014). "Summary of NIST SP 800-53 Revision 4, Security and Privacy Controls for Federal Information Systems and Organizations"(p. 6). Washington.

[23] Tung, B. (1999). "Common Intrusion Detection Framework". Gost.isi.edu. Recuperado 15 Abril 2017, de http://gost.isi.edu/cidf/

[24] Mira, J. (2017). "Implantación de un Sistema de Detección de Intrusos en la Universidad de Valencia"(1ra ed., p. 15-21). Valencia: Recuperado de http://rediris.es/cert/doc/pdf/ids-uv.pdf

[25] Anónimo. (2017). "Internet Engineering Task Force (IETF)". Ietf.org. Recuperado 16 Abril 2017, de https://www.ietf.org/

[26] Anónimo. "Extensible Markup Language (XML)". (2017). W3.org. Recuperado 21 Abril 2017, de https://www.w3.org/XML/

[27] Tsai, J., & Yu, Z. (2011). "Intrusion detection"(1st ed.). London: Imperial College Press.

[28] Anónimo. (2017). Cross-site Scripting (XSS) - OWASP. Owasp.org. Recuperado 2 Mayo 2017, de https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

[29] Types of Cross-Site Scripting - OWASP. (2017). Owasp.org. Recuperado 2 May 2017, de https://www.owasp.org/index.php/Types_of_Cross-Site_Scripting

29