Índice general

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

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.
- 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

[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].

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. Objetivo

1.4.1. 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.

1.4.2. 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.

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. Detección de Intrusos

2.1.1. Definición

Detección de intrusos es el proceso de monitorear los eventos 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].

2.1.2. Taxonomía de incidentes de seguridad

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

2.1.4. 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*[4]. 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.

2.1.5. 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.

2.1.5.1. 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.

[4] 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].

- 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.
- 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).

2.1.5.2. 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.

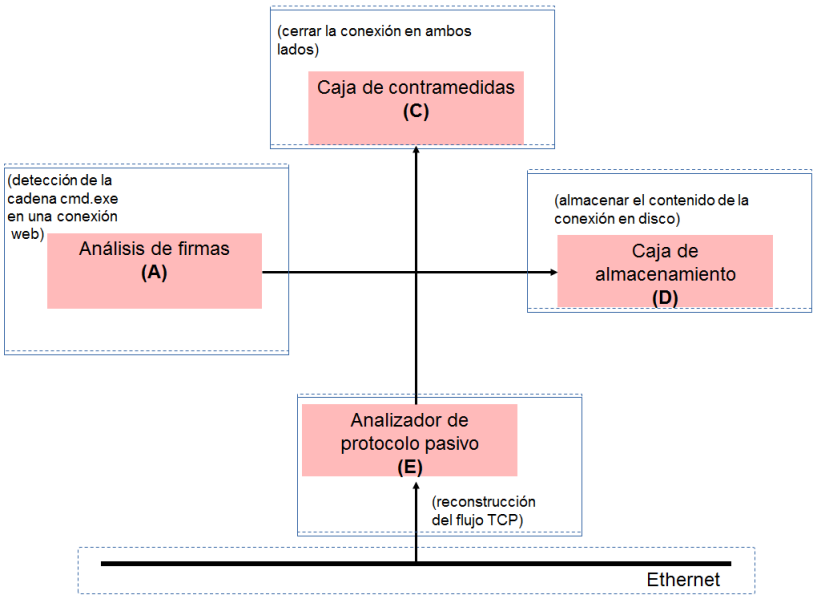


Figura 2.1: Diagrama a bloques de la arquitectura CIDF.

2.1.5.3. 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.
- **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.

2.1.5.4. 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.

2.1.5.5. Intrusion Detection Working Group

Debido a que el grupo de trabajo de ingeniería de internet (IETF[5] 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*[6]. 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

[5] 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].

[6] 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].

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)

  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.

2.1.6. 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:

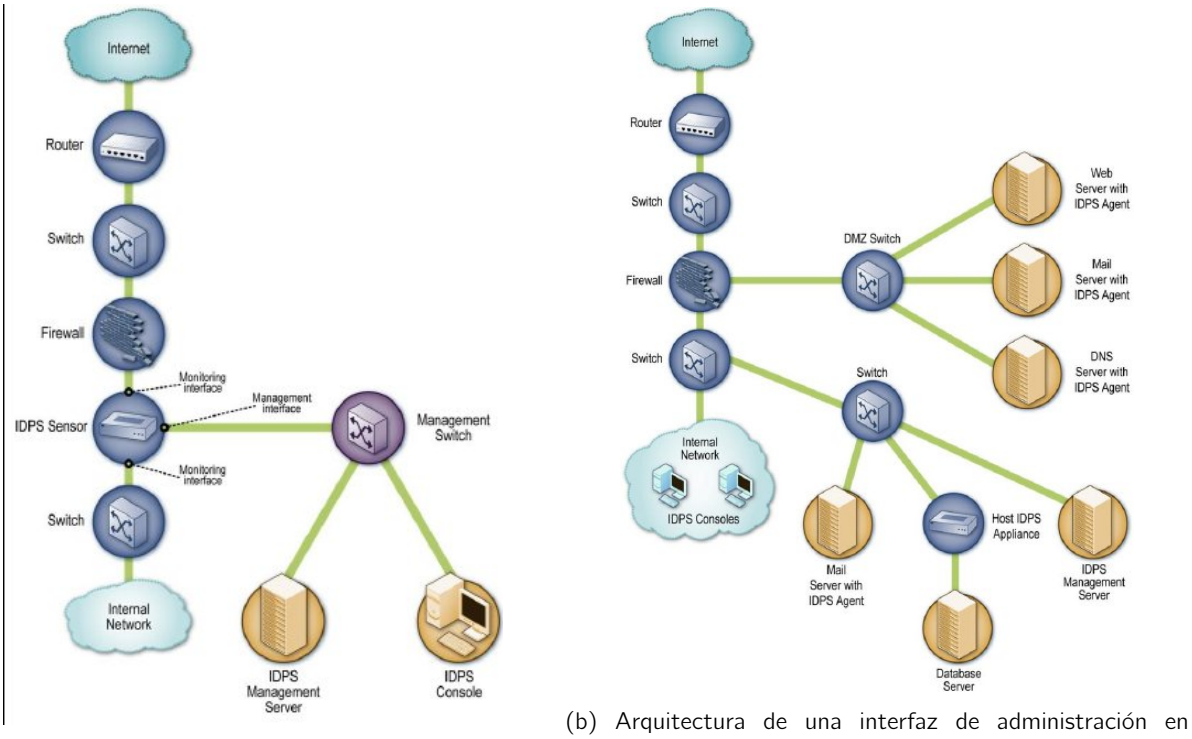
- **Sensor o Agente.** Los sensores y los agentes moniorean y analizan la actividad. El término *sensor* es mayormente usado para IDPS que monitorean redes, incluyendo las basadas en red, no guiadas y tecnologías de análisis de comportamiento de red. El término *agente* es usualmente utilizado para referirse 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 mencionar que independientemente de la respuesta a implementar elegida, las tres respuestas envían notificaciones/alertas de eventos en curso o pasados.

2.1.7. Arquitecturas de Red

Los componentes de los sistemas de detección de intrusos pueden ser conectados entre si 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 2.2a), en donde los sensores o agentes son administrados a través de la red de administración. 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 2.2b), la cual se conecta con la red de administración. 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 general al emplear esta arquitectura, son: el incremento del costo al momento de la creación de las redes y la inconveniencia para los usuarios y administradores del sistema para la administración del mismo en diferentes equipo de cómputo.



(a) Arquitectura de una red de administración.

(b) Arquitectura de una interfaz de administración en conjunto con una red de administración.

Figura 2.2: Arquitecturas de Red

14

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). Çross-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). Ünderlying 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). Çommon Intrusion Detection Framework". Gost.isi.edu. Recuperado 15 Abril 2017, de http://gost.isi.edu/cidf/

[24] Mira, J. (2017). Ïmplantación 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). Ïnternet 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/