RNPS No. 2143
ISSN 2072-6260
Versión Digital

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

Estado del arte de las plataformas de análisis de datos en la ciberseguridad

Alexis Fernández Rivas y
Osvaldo Andrés Pérez García

RT_037 **octubre 2016**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2143
ISSN 2072-6260
Versión Digital

SERIE GRIS

REPORTE TÉCNICO
Minería de Datos

Estado del arte de las plataformas de análisis de datos en la ciberseguridad

Alexis Fernández Rivas y
Osvaldo Andrés Pérez García

RT_037 **octubre 2016**

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Tabla de contenido

1 Introducción 1
2 El análisis de datos en la ciberseguridad. Conceptos básicos para su estudio 5
2.1 El análisis de datos 8
2.2 Tipos de análisis de datos 8
2.3 Metodologías para la aplicación de la minería de datos 9
2.4 Análisis de datos desde la perspectiva de big data 10
2.5 Modelo general para el análisis de datos en ciberseguridad 10
2.6 Requerimientos para una plataforma de AD 13
2.7 Los sistemas distribuidos y los middleware 14
3 Plataformas de AD para la ciberseguridad 15
3.1 Estado del arte de las plataformas de propósito específico 15
3.2 Estado del arte de las plataformas de propósito general orientadas a la ciberseguridad 22
4 Valoración sobre plataformas de AD de propósito general basadas en la filosofía middleware 23
4.1 Plataformas DAvinCi y DIKW 26
4.1.1 Breve descripción y valoración de la plataforma DAvinCi 26
4.1.2 Breve descripción y valoración de la plataforma DIKW 27
5 Prueba de concepto del middleware utilizado en la plataforma DAvinCi 29
6 Conclusiones 34
6.1 Trabajo futuro 35
7 Referencias 36

Estado del arte de las plataformas de análisis de datos en la ciberseguridad

Alexis Fernández Rivas[1] y Osvaldo Andrés Pérez García[2]

[1]Equipo de Servicios de Infraestructura Tecnológica para la Investigación, CENATAV - DATYS, La Habana, Cuba
afernandez@cenatav.co.cu
[2]Equipo de Investigaciones de Minería de Datos, CENATAV - DATYS, La Habana, Cuba
osvaldo.perez@cenatav.co.cu

RT_037, Serie Gris, CENATAV - DATYS
Aceptado: 12 de septiembre de 2016

Resumen. El Ciberespacio actual, escenario donde se desenvuelve la Ciberseguridad, es muy complejo. Nuevos tipos de ataques como las Amenazas Persistentes Avanzadas son difíciles de detectar empleando las herramientas de seguridad tradicionales. Uno de los enfoques utilizados para fortalecer las capacidades de detección de ataques es el Análisis de Datos (AD) utilizando técnicas de la Minería de Datos y el Reconocimiento de Patrones. Vincular el AD a la Ciberseguridad requiere de plataformas basadas en middleware que garanticen aspectos tales como acceso a fuentes heterogéneas de datos, aprovechamiento eficiente de los recursos (por ejemplo, el ancho de banda), entre otros. En este trabajo se realiza un estudio del estado del arte acerca de las plataformas de AD orientadas a la Ciberseguridad. Estas son evaluadas en función de su alcance y teniendo en cuenta los requerimientos expresados por los expertos, proponiéndose como trabajo futuro el desarrollo, en una primera etapa, de varios componentes imprescindibles para la obtención de una plataforma propia para el AD en tareas de Ciberseguridad.

Palabras clave: ciberseguridad, análisis de datos, plataformas, middleware.

Abstract. The current Cyberspace, scenario where the Cyber-security unfolds, is very complex. New types of attacks such as Advanced Persistent Threats are difficult to detect using traditional security tools. The Data Analysis (DA) using Data Mining and Pattern Recognition techniques is one of the approaches used to strengthen the detection capabilities of attacks detection systems. Develop middleware-based platforms is crucial to apply the DA in Cyber-security tasks. The objectives are to ensure aspects such as access to heterogeneous data sources, efficient use of resources (e.g., bandwidth), among others. In this report a study of state of art of the DA platforms oriented to Cyber-security is performed. These platforms are evaluated in terms of its scope and taking into account the requirements expressed by experts, and proposing as future work the development, at an early stage, of several essential components to obtaining our own platform for the DA in Cyber-security tasks.

Keywords: cyber-security, data analysis, platforms, middleware.

1 Introducción

El Ciberespacio, según Lehto [1], incluye todas las infraestructuras de información que son accesibles a través de Internet sin importar las fronteras territoriales que puedan existir. Por ejemplo, el gobierno del

Reino Unido define el Ciberespacio como un dominio interactivo conformado por redes digitales que son utilizadas para almacenar, modificar y comunicar información. Esto incluye Internet y otros sistemas de información sobre los cuales se sustentan negocios, infraestructuras tecnológicas y servicios. Por otro lado, para Estados Unidos el Ciberespacio es su "sistema nervioso" (infraestructura crítica), es el sistema de control de ese país. Para ellos, el Ciberespacio está compuesto por cientos de miles de computadoras, servidores, enrutadores, conmutadores y cables de fibra óptica que les permiten a su infraestructura crítica operar. Por tanto, el sano funcionamiento del Ciberespacio es esencial para esa economía y su seguridad nacional. El Ciberespacio sobrepasa las fronteras de Internet, e incluye no solo el hardware, software, datos y sistemas de información, sino también a personas interactuando social y económicamente dentro de estas redes y en toda su infraestructura.

Teniendo lo anterior en cuenta, es lógico que cada país tenga un interés máximo en proteger el Ciberespacio de posibles ataques dirigidos a robar información, cometer fraudes, denegar servicios, deshabilitar sistemas de defensa, etc. La lista de consecuencias de un ataque cibernético puede ser tan grande como la imaginación misma. Es por eso que surge la Ciberseguridad como disciplina encargada de preservar la integridad de dicho espacio.

La Ciberseguridad está asociada, entre otras especialidades, con el control de riesgos, con el parcheo de vulnerabilidades y el mejoramiento en la recuperación de los sistemas ante la ocurrencia de errores. De forma breve, la Ciberseguridad puede ser definida como el conjunto de acciones realizadas para proteger de los ciberataques y de aminorar sus consecuencias e incluye la debida respuesta a dichos ataques [1]. La Ciberseguridad, también, está determinada por el análisis previo de las amenazas y brechas de seguridad a las cuales está o puede estar expuesta una determinada organización o institución.

Según la Unión Internacional de las Telecomunicaciones (ITU, por sus siglas en inglés) la Ciberseguridad no es un fin sino un medio para conseguir un fin [2]. Plantean además que el objetivo debería ser el de construir un mecanismo confiable que permita que la infraestructura crítica de la información pudiera trabajar fiablemente y continuar garantizando los intereses nacionales inclusive cuando ésta se encuentre bajo ataque. Por tanto, las estrategias de Ciberseguridad nacional debieran estar enfocadas a prevenir y detectar las amenazas que con mayor probabilidad puedan interrumpir el funcionamiento vital de la sociedad.

Ahora bien, el Ciberespacio es un entorno complejo compuesto por varios y disímiles elementos lo cual hace que el rango de acción de la Ciberseguridad sea muy abarcador. M. C. Libicki propuso en [3] un modelo estructural del Ciberespacio dividido en cinco capas. Este modelo permite definir los diferente roles o áreas sobre las cuales debe actuar un sistema de Ciberseguridad (ver Figura 1). Las capas son las siguientes: física, sintáctica, semántica, servicio y cognitiva.

La capa física está compuesta por todos los elementos físicos de la red de comunicaciones mientras que la capa sintáctica está formada por varios sistemas de control, gestión y por aplicaciones que facilitan la interacción entre los dispositivos conectados a la red. La capa semántica contiene la información y los conjuntos de datos en los terminales (computadoras) de los usuarios así como diferentes funciones administradas por estos. La capa de servicio es aquella donde se encuentran todos los servicios públicos y comerciales que los usuarios utilizan. Por último, la capa cognitiva describe el entorno consciente-informativo del usuario: un mundo en el cual la información está siendo interpretada y donde se crea un único significado contextual respecto a dicha información.

Los riesgos en Ciberseguridad se hacen cada vez más comunes y complejos. Nuevas herramientas y métodos están siendo empleados para lanzar ataques cada vez más elaborados y sofisticados; al mismo tiempo aumentan las vulnerabilidades y la motivación por cometer delitos cibernéticos debido al beneficio económico que estos reportan. El impacto que tienen tales hechos constituye un llamado a la creación e innovación de soluciones con el fin de mitigarlos. Tiempo atrás los atacantes estaban

constituidos por personas individuales o grupos pequeños de *hackers*[1] sin embargo, en la actualidad, estos están agrupados en organizaciones que emplean tecnologías avanzadas con el fin de comprometer la integridad de objetivos específicos mediante ataques que pueden perdurar en el tiempo y provenir de diversas fuentes. A este modo de operar se le conoce como Ataque Cibernético Dirigido. Entre los ejemplos de ataques más conocidos están Denegación de Servicios[2] (DoS, por sus siglas en inglés), Denegación Distribuida de Servicios[3] (DDoS, por sus siglas en inglés), el *spoofing*[4], y uno de los más novedosos en la actualidad es el basado en las denominadas Amenazas Persistentes Avanzadas (APT, por sus siglas en inglés). En la figura 2 se ilustran diferentes métodos de ataques según el modelo por capas del Ciberespacio antes mencionado.

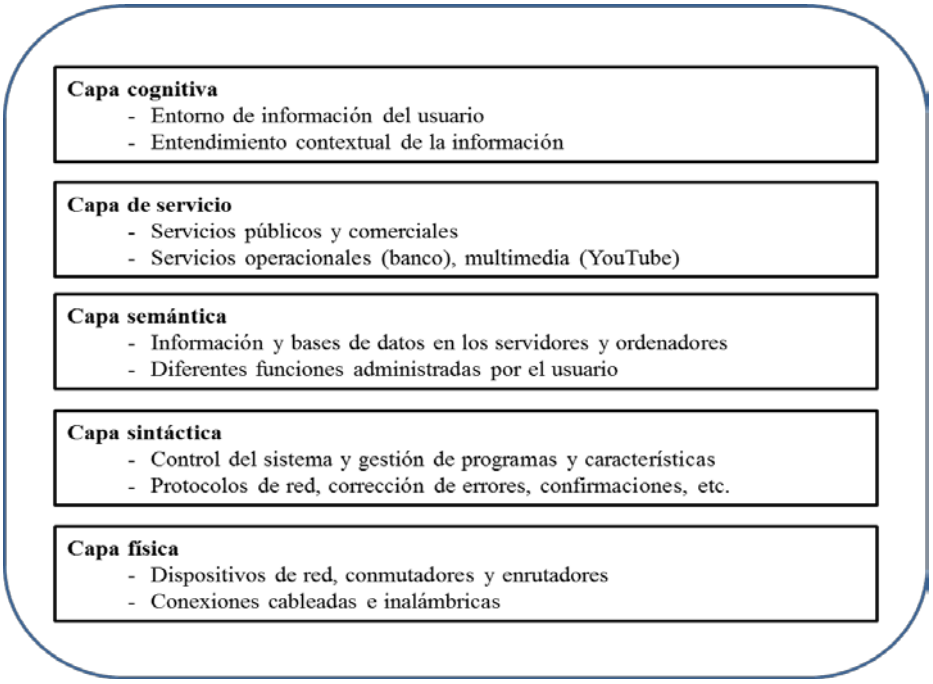


Fig. 1. Modelo estructural del Ciberespacio (Imagen tomada de [3]).

Como se aprecia en la Figura 2, la Ciberseguridad tiene un amplio rango de acción y para cada una de las capas existen técnicas y métodos para asegurar su integridad. En el presente trabajo solo nos enfocaremos en cómo proteger la capa semántica y de servicio.

El diseño de estrategias y herramientas de seguridad que permitan proteger tanto la capa semántica como la de servicio es uno de los mayores retos que tiene la Ciberseguridad. En el caso de las herramientas de seguridad su funcionamiento se basa fundamentalmente en capturar datos, procesarlos (por ejemplo, aplicando reglas) y devolver un resultado (alertas) que le permita tomar decisiones al analista [4]. Un ejemplo de herramienta de este tipo son los Sistemas de Detección de Intrusiones (IDS, por sus siglas en inglés) los cuales son capaces de capturar el tráfico de la red, analizarlo en base a reglas predefinidas por los analistas y generar alertas en caso de coincidencia [5]. Los IDSs y otras

[1] Persona con conocimientos de informática que se dedica a acceder ilegalmente a sistemas informáticos ajenos y a manipularlos.
[2] DoS: Es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos.
[3] DDoS: Cuando el ataque DoS es distribuido.
[4] Spoofing: técnicas a través de las cuales un atacante se hace pasar por una entidad distinta a través de la falsificación de los datos en una comunicación.

herramientas como los antivirus o los cortafuegos siguen siendo, hoy día, muy utilizados; sin embargo, estos no permiten correlacionar eventos que hayan sido detectados por otras herramientas de seguridad lo cual limita la posibilidad de detectar ataques que ocurren de forma distribuida. En respuesta a esta limitación surgen los denominados Gestores de Eventos y Seguridad de la Información (SIEM, por sus siglas en inglés) los cuales tienen como objetivo principal posibilitar la correlación de alertas provenientes de diferentes dispositivos de seguridad [6].

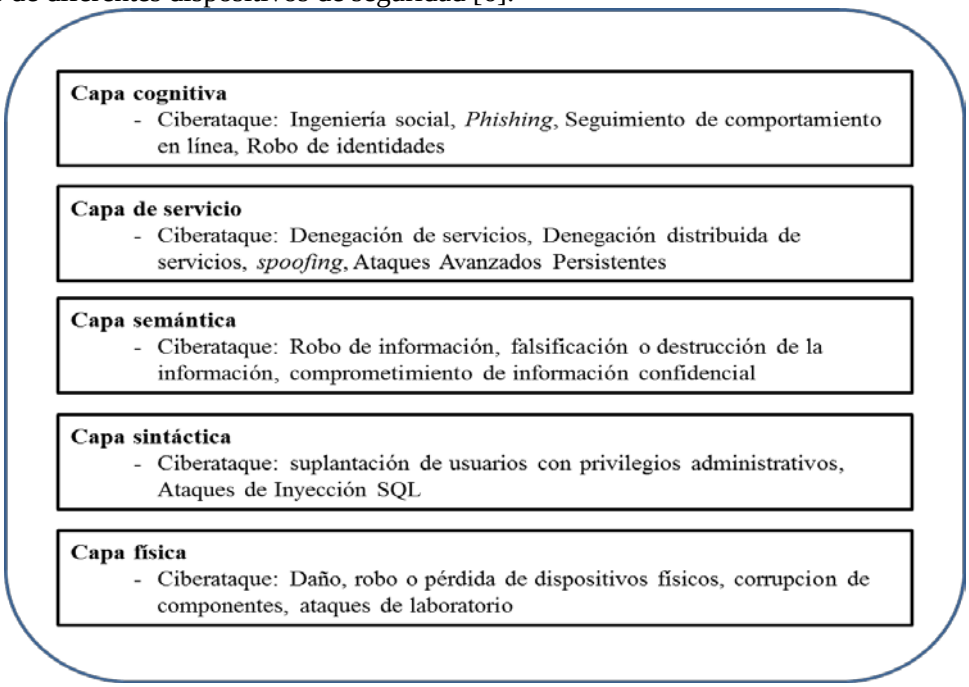


Fig. 2. Ejemplos de ciberataques en cada una de las capas del Ciberespacio (Imagen tomada de [3]).

A pesar de que los sistemas SIEM constituyen un paso evolutivo superior en el Análisis de Datos para la Ciberseguridad, en comparación con el resto de las herramientas de seguridad mencionadas, estos están restringidos en cuanto a los tipos de datos que procesan y al empleo de reglas predefinidas manualmente por los analistas [7]. En la actualidad, entre más datos puedan ser capturados y analizados por una herramienta mayor será la posibilidad de que esta tenga éxito en la detección de los ataques. Por otro lado, también se necesita que dichas herramientas sean capaces de descubrir nuevos conocimientos o patrones ocultos que les permitan generar automáticamente nuevas reglas y no depender únicamente de los ajustes del analista. El Análisis de Datos (AD) orientado a la Ciberseguridad haciendo uso de técnicas de Minería de Datos (MD) y de Reconocimiento de Patrones (RP) es uno de los enfoques que permite responder a tales requerimientos.

La unión del AD con la Ciberseguridad no es un tema nuevo. Esta tecnología ha sido muy utilizada en la detección de intrusiones [8] y en los sistemas SIEM [9]. Sin embargo, existen carencias de estudios acerca de plataformas de AD orientado a la Ciberseguridad que hayan sido desarrolladas con el propósito general de analizar tanto los datos provenientes de los dispositivos de seguridad como los provenientes de otras fuentes tales como sensores, bitácoras, y del tráfico de la red. En el presente trabajo se realiza un estudio del estado del arte acerca de las plataformas de AD orientadas a la Ciberseguridad que utilicen técnicas de MD o RP y se ofrecen valoraciones generales al respecto.

Este trabajo está estructurado de la siguiente manera: en la sección 2 se abordan los conceptos que permitirán comprender la temática del AD orientado a la Ciberseguridad, se definen los tipos de análisis que existen y se expone brevemente la filosofía básica de diseño de una plataforma de AD para la Ciberseguridad. En la sección 3 se presente el estudio del estado del arte de las plataformas de AD orientadas a la Ciberseguridad aplicando técnicas de MD mientras que en la sección 4 se realiza una

valoración acerca de las plataformas de AD de propósito general basadas en la filosofía middleware. En la sección 5 se realiza una prueba de concepto del middleware utilizado en una de las plataformas de la sección anterior. En 6 se concluye y se exponen posibles líneas de trabajo futuro.

2 El análisis de datos en la ciberseguridad. Conceptos básicos para su estudio

El desarrollo casi exponencial de las Tecnologías de la Información y las Comunicaciones (TIC) ha permitido, en los últimos años, un proceso de informatización global de la sociedad así como de la actividad económica. La mayoría de los negocios, fábricas, empresas, universidades, hospitales, entre otras entidades, están atados y se benefician de los servicios y prestaciones de las TICs, por lo que proteger todo ese andamiaje tecnológico y la información que es procesada con él, es una tarea de primer orden para esas entidades e incluso gobiernos de prácticamente todo mundo. La Ciberseguridad es la ciencia encargada de tal tarea; sin embargo, su misión se hace cada vez más compleja porque conforme aparecen nuevos servicios también surgen nuevas vulnerabilidades y brechas que son aprovechadas por los ciberatacantes. En [10] se exponen ejemplos recientes de daños ocasionados por ciberatacantes a algunas de las compañías más importantes del mundo como son Sony[5] o Adobe Systems[6]. Algunos de estos ejemplos se mencionan a continuación:

- Registros e información de tarjetas de pagos fueron robados a 77 millones de clientes de Sony en Abril de 2011.
- La infraestructura tecnológica de Adobe Systems fue comprometida en Octubre de 2013 produciéndose la pérdida de 152 millones de nombres, identificaciones de usuarios, información cifrada de tarjetas de pago y código fuente.
- Target Brands[7] perdió 110 millones de registros de usuarios, y números de tarjetas de crédito y débito debido a un ataque informático dirigido a uno de sus puntos de venta y a otros sistemas, en Diciembre de 2013.
- Anthem[8] perdió 78 millones de registros que incluían datos personales y números de seguro social de sus trabajadores producto de un ciberataque, en Febrero de 2015.

Tales ejemplos demuestran que las estrategias de seguridad actuales no son suficientes y que nuevos enfoques deben ser desarrollados. Ahora bien, esto no quiere decir que las técnicas de Ciberseguridad empleada hasta el momento deban ser descartadas, al contrario, estas deberían seguir siendo utilizadas y mejoradas ya que en determinados escenarios y para determinados tipos de ataques son efectivas. Algunas de las tácticas de detección de eventos de seguridad más utilizadas hasta nuestros días donde se emplean diferentes estrategias de AD, son las siguientes [10]:

- Análisis de bitácoras (logs, en inglés).
- Análisis de eventos generados por dispositivos de red.
- Sistema de Detección y Prevención de Intrusiones de red (IDS e IPS respectivamente, por sus siglas en inglés).
- Sistema de Detección y Prevención de Intrusiones a nivel de terminal (host, en inglés).
- Antivirus.

[5] http://www.sony.net/
[6] http://www.adobe.com/
[7] http://www.target.com/
[8] https://www.anthem.com/

- Monitor de Integridad de Ficheros (FIM, por sus siglas en ingles) y listas blancas.
- Administrador de Eventos y de Información de Seguridad.

En la Tabla 1 se muestra un resumen expuesto en [10] donde se describen brevemente cada una de las tácticas antes mencionadas y sus ventajas y limitaciones. Los sistemas SIEM serán discutidos independientemente de la tabla con el fin de abordar un poco más acerca de esta herramienta por la importancia que tiene para el AD en la Ciberseguridad.

Tabla 1. Ventajas y limitaciones de las herramientas tradicionales de seguridad.

Estrategia	Descripción	Ventajas	Limitaciones
Análisis de bitácoras	Estrategia de control primario para detectar eventos de seguridad mediante el estudio de registros [10].	Describen el comportamiento detallado de la mayoría de los dispositivos de red, terminales y aplicaciones existentes en la actualidad por lo que son fuente valiosa de información a la hora de analizar eventos.	Se generan grandes volúmenes de información difíciles de manejar sin las técnicas adecuadas. El desarrollo de reglas para el análisis consume mucho tiempo de operación.
Análisis de eventos generados por dispositivos de red	Estudio basado en los eventos generados por dispositivos (cortafuegos, proxies, enrutadores y conmutadores), de bitácoras del sistema, de eventos de formato específico desarrollado por el fabricante y eventos SNMP[9] [10].	Permite obtener una visión de lo que está sucediendo en la red y con los dispositivos que la conforman, así como saber si existen errores de configuración en estos.	Complejidad a la hora de obtener información importante para la seguridad debido al volumen de eventos que hay que procesar.
IDS e IPS de red	Mecanismo de detección para identificar comportamientos no autorizados, ilícitos y anómalos basándose en el análisis del tráfico de red. El IDS procesa y marca el tráfico sospechoso y el IPS puede bloquearlo de forma dinámica [10].	Permiten la detección de firmas de ataques bien conocidas o patrones inusuales presentes en el tráfico de red. Son adecuados para escenarios donde se requieran mecanismos avanzados de defensa.	Limitados a la detección de ataques bien conocidos. Mecanismo complejo de ajustar y tienden a producir elevados falsos positivos.
IDS e IPS de terminal	Ídem al anterior pero es basado en el comportamiento de un dispositivo o terminal determinado.	Provee una gran cantidad de información útil, especialmente cuando se trata de enfrentar malwares[10] y ataques dirigidos hacia el lado del cliente.	Ajuste complejo.
Antivirus	Herramienta que permite detectar aquellos malware cuyo objetivo sea el de alterar el funcionamiento normal de un *host* u ordenador.	Permite detectar un grupo reducido de malwares en función de las bases de firmas de estos. Mecanismo de cuarentena.	Detección de ataques basada fundamentalmente en firmas de ataques bien conocidos. Dificultades para gestionar y controlar agentes antivirus distribuidos en todos los

[9] Protocolo Simple de Gestión de Red.
[10] Código maligno o software malintencionado.

			terminales.
Monitor de integridad de ficheros y listas blancas	Mecanismos de control de seguridad basado en terminal que permite monitorear los cambios producidos en ficheros críticos y en la evaluación de reglas que especifican que acciones puede realizar un usuario en dicho terminal.	Mecanismo efectivo de control y detección. Genera eventos que pueden ser correlacionados con otros sistemas.	Constituye un desafío mantener actualizada la configuración de dicho método así como el ajuste de las políticas necesarias para garantizar su correcto funcionamiento.

Los sistemas SIEM son plataformas que permiten colectar y correlacionar información proveniente de múltiples dispositivos y aplicaciones para la seguridad permitiéndole a los analistas monitorear el entorno con una mayor profundidad y, a su vez, generar reglas de detección más eficaces. Los SIEMs están provistos de herramientas muy poderosas pero a la vez complejas de manejar, es por eso que están orientados a escenarios donde los requerimientos de seguridad son elevados y se cuenta con personal con conocimientos avanzados [7]. La tecnología SIEM proporciona las herramientas necesarias para realizar el análisis en tiempo real de las alertas de seguridad generadas por el hardware y software de red. Sin embargo, estos sistemas presentan algunas limitaciones para tratar todos los problemas actuales de la Ciberseguridad:

- Están diseñados para capturar y correlacionar eventos provenientes de dispositivos de seguridad [7].
- Para correlacionar eventos generados por múltiples dispositivos normaliza los datos y, luego los almacena. Esta normalización puede provocar la pérdida de información sensible e importante para la detección de ataques [11].

De lo antes expuesto se puede concluir que las herramientas mencionadas en la Tabla 1 permiten monitorear o detectar comportamientos anómalos, códigos malignos, entre otros eventos; mediante el procesamiento de datos originales tales como las bitácoras, paquetes IP o archivos. Cada una de ellas genera como resultado una o varias alertas que le permiten al analista tomar decisiones. Por otro lado, los SIEM ofrecen la posibilidad de interrelacionar y vincular tales alertas permitiéndole al analista obtener mayor información a lo hora de tomar decisiones.

Ahora bien, todo este proceso de monitoreo, detección, correlación, análisis y toma de decisiones se basa en el conocimiento de los expertos, que son los que deciden qué es una norma o una anomalía. Algunos enfoques descritos en la literatura logran descubrir nuevo conocimiento al profundizar en las relaciones ocultas entre las alertas y trazas de los sistemas; sin embargo, de manera general todas estas herramientas se ven privadas de la posibilidad de acceder a los datos originales provenientes de las fuentes de los sistemas que monitorean.

Es por todo esto que un enfoque de la Ciberseguridad basado en el Análisis de Datos específicamente empleando técnicas de Minería de Datos es requerido. Un enfoque de este tipo permitiría:

- Descubrir nuevos ataques partiendo del análisis de grandes cantidades de datos en bruto (*raw data*, en inglés) u originales provenientes de disimiles fuentes, incluyendo: dispositivos de seguridad, sensores proveedores de este tipo de datos, módulos de control y gestión, bitácoras, entre otros.
- Aprovechar la información que portan los tipos de datos semiestructurados y los no estructurados.
- Identificar nuevos patrones y comportamientos en el entorno que está siendo monitoreado.

- Proporcionar la infraestructura de datos necesaria para mejorar la eficacia en la detección debido a que los modelos de detección van a ser desarrollados teniendo en cuenta el análisis la mayor cantidad de información posible.

Esto aspectos permitiría incorporar a los sistemas de protección que emplean el Análisis de Datos la capacidad de prever ataques reduciendo los riesgos y costos asociados a su ocurrencia.

2.1 El análisis de datos

En [12] definen el Análisis de Datos como la ciencia encargada de examinar datos en bruto con el propósito de llegar a conclusiones respecto a estos y, por consiguiente, mejorar la toma de decisiones. E. Begoli, en [13], define el AD como una colección de métodos computacionales, estadísticos y visuales que mejoraran el entendimiento de los datos originales. Un enfoque diferente del AD es expuesto en [4] donde consideran que dicha disciplina es una etapa dentro del proceso de Descubrimiento de Conocimientos en Bases de Datos (KDD, por sus siglas en inglés) encargada de hallar patrones, reglas y, finalmente, información oculta en los datos.

De forma general, el AD ha sido aplicado en muchas compañías y organizaciones con el fin de mejorar sus métodos de Inteligencia de Negocios (BI, por sus siglas en inglés) [4] y también ha sido utilizado como forma de verificación de modelos teóricos durante estudios científicos en ramas como la salud, la biotecnología, entre otras. En el campo de la seguridad, se ha empleado, por ejemplo, como mecanismo de auditoria a la Infraestructura Tecnológica y para determinar si los sistemas desplegados protegen eficazmente los datos y operan eficientemente, de forma similar, las compañías bancarias y de tarjetas de crédito analizan las extracciones y transacciones de dinero así como los patrones de gastos con el fin de prevenir los fraudes o identificar robos.

El Análisis de Datos se divide en Análisis de Datos Exploratorio (EDA, por sus siglas en inglés) [14], donde nuevos rasgos son descubiertos en los datos, y Análisis de Datos Confirmatorio (CDA, por sus siglas en inglés) [14], donde las hipótesis son comprobadas.

2.2 Tipos de análisis de datos

En [13] definen tres tipos de Análisis de Datos: de flujos de datos, interactivo y por lotes.

- *Análisis de Flujos de Datos (SDA, por sus siglas en inglés)*: Es el proceso de extraer información y descubrir conocimiento de flujos de datos cuya generación ocurre de forma rápida e ininterrumpida. Tales flujos pueden ser los producidos por el tráfico de la red, bitácoras del sistema, llamadas telefónicas, transacciones bancarias y datos provenientes de sensores (cámaras de video, RFID[11], entre otros). Las arquitecturas o plataformas diseñadas para el SDA tienen como principal requerimiento el de garantizar altas velocidades de desempeño, es decir, los datos deben ser procesados durante períodos pequeños de tiempo (cercano al tiempo real). También deben satisfacer exigencias de análisis distribuido donde los operadores procesarían los datos en línea, es decir, sin previo almacenamiento de estos. Mantener los valores de latencias bajos sin comprometer la escalabilidad es uno de los principales desafíos en dichas arquitecturas [15].

  Stonebraker y colaboradores [16] definieron tres tipos de arquitecturas generales orientadas al SDA:

  1. Sistema de Gestión de Bases de Datos (DBMSs, por sus siglas en inglés).
  2. Motores de Reglas.
  3. Motores de Procesamiento de Flujo (SPEs, por sus siglas en inglés).

[11] Sistema de almacenamiento y recuperación de datos remoto.

- *Análisis Interactivo de Datos:* Es una forma de AD conducida interactivamente por el especialista, en donde, este explora visualmente conjuntos de datos en tiempo real. Este tipo de análisis se hace mediante herramientas de visualización y empleando consultas personalizadas o *ad hoc*. Por lo general, el analista cuenta con la ayuda de algún sistema de gestión de datos capaz de realizar búsquedas en grandes conjuntos de datos (por ejemplo, Elasticsearch[12]).
- *Análisis por lotes:* Los sistemas de análisis por lotes son las plataformas para AD más antiguas y bien establecidas que existen. El análisis por lotes se diferencia del SDA en los requerimientos de tiempo real. En este tipo de análisis los datos a analizar son capturados durante un periodo de tiempo y almacenados en BD para luego ser procesados por lotes. Es una forma eficiente de procesar grandes volúmenes de datos [23]. Como ejemplos de herramientas actuales que permiten dicho análisis son Hadoop y Spark.

2.3 Metodologías para la aplicación de la minería de datos

La aplicación de la Minería de Datos en cualquier escenario implica establecer condiciones para sistematizar el proceso de inserción y asimilación de esta especialidad. Para esto han sido definidas varias metodologías: KDD, CRISP-DM [17], SEMMA [18], entre otras. Atendiendo a la importancia que ha tenido para la aplicación de la MD y en concordancia con los propósitos del presente trabajo, se explicará la primera de ellas. No obstante, deben tenerse en cuenta las posibilidades que brindan el resto de las metodologías. Por ejemplo, en [19] se publica una encuesta que demuestra la preferencia de los especialistas por utilizar la metodología CRISP-DM en proyectos de minería de datos, debido fundamentalmente a su adaptación a los requerimientos de los escenarios actuales.

La metodología KDD incluye un conjunto de funcionalidades diseñadas para extraer nuevo conocimiento o información de conjuntos de datos (*dataset*, término en inglés) complejos y de gran tamaño. Es un campo interdisciplinario que se expande y tiene amplia aplicación conjuntamente con la Ciencias de la Computación, Estadísticas, Ciencia Cognitiva, Visualización y en otros dominios específicos como la medicina o la economía.

En [4] dividen el proceso de Descubrimiento de Nuevo Conocimiento en tres etapas (entrada de datos, Análisis de Datos y resultados de salida) y 7 operadores o funciones (adquisición, selección, preprocesamiento, transformación, técnica de AD, evaluación e interpretación) (ver Figura 3). Los primeros cuatro operadores pertenecen a la etapa de entrada y actúan sobre el flujo de datos en bruto produciendo una salida de datos preparados para ser procesados por alguna técnica de análisis. Luego de aplicada dicha técnica se obtiene información a la salida de dicha etapa, los cuales son evaluados e interpretados en la etapa final que tiene por salida el nuevo conocimiento generado.

[12] https://www.elastic.co/

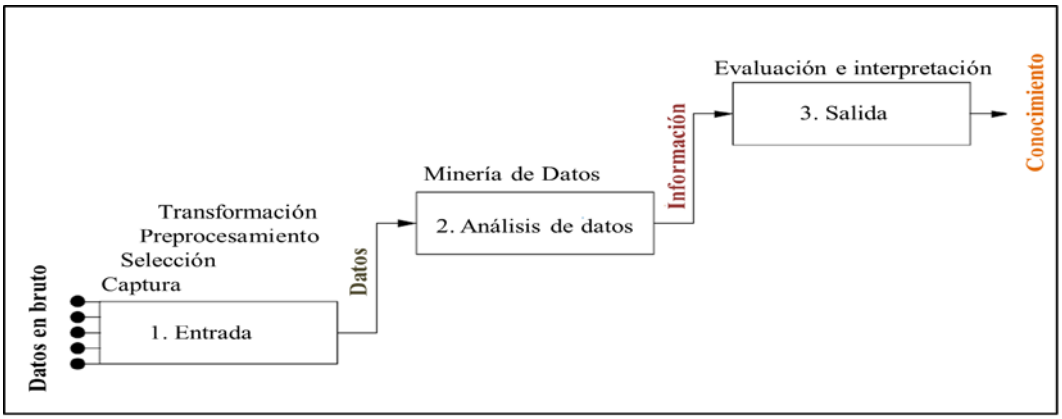


Fig. 3. Proceso de Descubrimiento de Conocimiento en Bases de Datos (Imagen tomada de [4]).

2.4 Análisis de datos desde la perspectiva de big data

Big Data hace referencia al conjunto de tecnologías y sistemas que manipulan grandes volúmenes de datos. Es también una referencia a las características de los datos que debe procesar una entidad. Define por lo tanto qué tipo de problemas enfrenta esa entidad para dar respuesta a sus necesidades a partir del AD.

Según registros reportados en [20] se estima que la cantidad de datos presentes en la web está en el orden de los zettabytes (10^{21} bytes) lo cual representa un crecimiento de 5 veces respecto al 2010. Compañías como Facebook, Google, Twitter, Amazon u Oracle almacenan diariamente decenas de Tera-Bytes lo cual complejiza apreciablemente el proceso de KDD. A estos volúmenes de datos a analizar se le adicionan requerimientos de velocidad en el análisis y manejo de la diversidad de los datos. De forma general, la disciplina tecnológica encargada del KDD en escenarios como los descritos se le denomina Análisis en Grandes Volúmenes de Datos (*Big Data Analysis* o BDA, por sus siglas en inglés).

BDA no es más que la extracción de nuevo conocimiento a partir del análisis de grandes volúmenes de datos originales. Los desafíos respecto a los datos en bruto en un escenario *Big Data* son múltiples: masividad, alta dimensionalidad, heterogeneidad, complejidad, no estructurado, inconsistencia, ruido y errores [21]. En el presente trabajo no se hará énfasis en los problemas de BDA, sin embargo, algunas soluciones diseñadas para dicho campo responden adecuadamente a determinadas características de escenarios donde debe aplicarse la Ciberseguridad. Además, debe destacarse que existen un conjunto de herramientas en BDA que han alcanzado un alto grado de madurez y están siendo utilizadas en diferentes campos científico-tecnológicos para procesar grandes volúmenes de información [4],[22], [21], [23] y [24].

2.5 Modelo general para el análisis de datos en ciberseguridad

En [10] definen la analítica orientada a la Ciberseguridad como el proceso de analizar grandes conjuntos de datos haciendo uso de tecnologías que permitan realizar análisis rápidos y precisos; correlacionar y reportar resultados con el fin de identificar eventos y patrones de interés que puedan alertar sobre un comportamiento malicioso en el entorno que está siendo monitoreado. De forma general, se puede decir que ese proceso coincide con la aplicación de la metodología KDD en el campo de la Ciberseguridad con el objetivo de descubrir nuevos patrones y comportamientos anómalos que permitan detectar ataques nuevos y complejos.

En [22] exponen lo que puede ser considerado como un esquema general o modelo guía para realizar Análisis de Eventos de Seguridad, es decir, puede aplicarse tanto a la Ciberseguridad como al

monitoreo del "mundo real". El modelo consta de las siguientes etapas (1) asimilación del dominio, (2) captura de los datos, (3) descubrimiento de conocimiento (KDD), (4) almacenamiento (salva del nuevo conocimiento) y (5) visualización.

En la etapa de asimilación del dominio se hace un estudio del contexto que se va a monitorear o del objetivo a defender, y se examina en qué espacio se encuentra este: Ciberespacio o "mundo real", para colocar los sensores apropiados (cámaras de video, micrófonos, sensores de movimiento, bitácoras, aplicaciones de captura de tráfico de red, entre otros). La captura de los datos puede describirse en tres pasos según [22]: manejo de las fuentes de datos, transmisión de los datos y preprocesamiento. El manejo de los sensores no es más que proveerle al sistema de los controladores (*drivers*, en inglés) necesarios para poder capturar los datos proveniente del hardware (por ejemplo, cámaras IP) o software (por ejemplo, *tcpdump*[13] o NetFlow[14]).

Por otro lado, la transmisión de los datos consiste en aplicar técnicas y seleccionar protocolos de transporte de datos apropiados para el escenario a enfrentar. Puede ser posible un escenario en el cual el volumen de datos en bruto generados en la etapa previa crezca de forma exponencial por lo que se requeriría de medios de transmisión y de técnicas que permitan alcanzar altos valores de ancho de banda y *throughput*[15].

El preprocesamiento en la etapa de captura se encarga de eliminar redundancia y ruidos presentes en los datos originales. Dicho preprocesamiento puede ser requerido o no por el usuario-especialista ya que al hacerlo se puede eliminar información valiosa, sin embargo esto implicaría no usar de forma eficiente los recursos de almacenamiento.

Durante la etapa de procesamiento se descubren patrones, se generan reglas y se obtiene información oculta en los datos. La mayoría de los investigadores de este campo identifican a la MD con la etapa de procesamiento [4]. En el presente trabajo tomaremos a la MD como una de las técnicas que se pueden utilizar en esta etapa de procesamiento, teniendo en cuenta que existen otras tales como la Estadística o el Aprendizaje Automatizado (*Machine Learning, término en inglés*), también de amplia aplicación.

El almacenamiento debe garantizar capacidades para guardar tanto datos en bruto como los resultados generados por el proceso KDD aplicado precisamente a esos datos. Almacenar todos los datos que se reciban por tiempo indefinido es un propósito impensable. El resultado del análisis permite reducir el espacio necesario para preservar lo que se sabe de los datos recibidos, sin embargo, también debe conservarse una cantidad de datos originales para realizar análisis "post mórtem[16]" en caso de necesidad.

[13] http://www.tcpdump.org
[14] http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html
[15] Medida de la utilización del ancho de banda del canal para transportar datos netos.
[16] Análisis que se hace tiempo después de que ocurrió el ataque. Para esto se debe contar con una base de datos con los datos en bruto capturados.

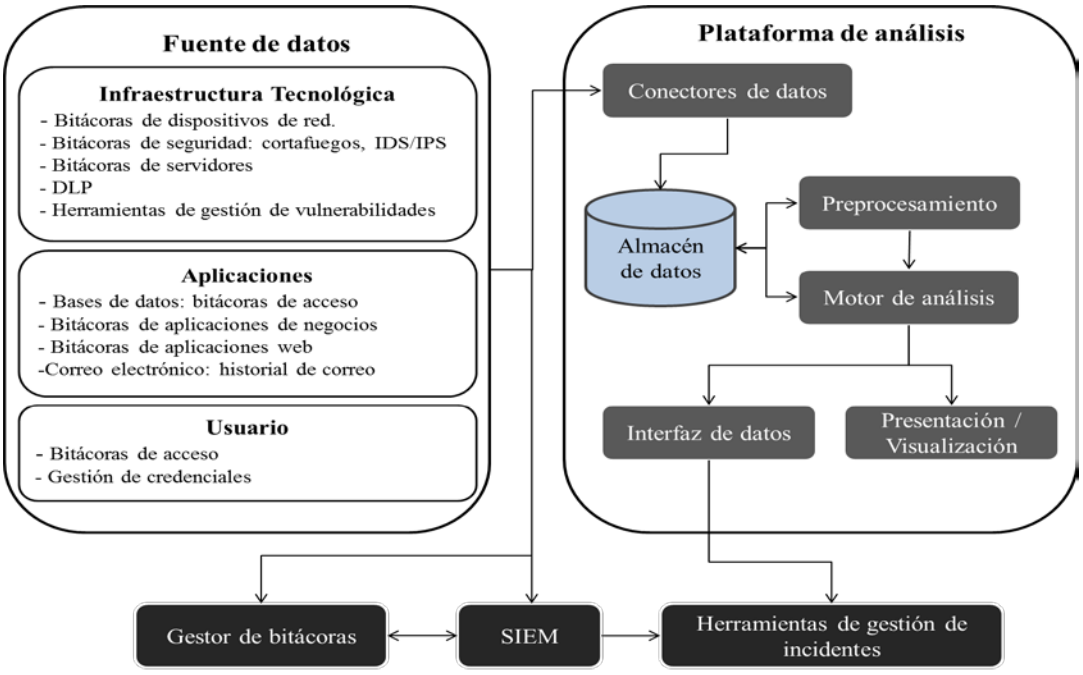


Fig. 4. Modelo de AD orientado a la Ciberseguridad (Imagen tomada de [25]).

Por último, se encuentra la etapa de visualización donde se muestran los resultados con un enfoque orientado a la seguridad, es decir, que le sirvan al analista de seguridad para tomar decisiones.

Basándose en el modelo genérico anterior, G. Pan y colaboradores [25] proponen un modelo para el AD orientado a la Ciberseguridad empleando técnicas de MD (ver Figura 4).

Se puede apreciar en la figura como dividen las fuentes de datos en tres tipos: datos provenientes de la infraestructura tecnológica, generados por aplicaciones o servicios y los producidos por actividades de los usuarios. Del lado derecho se puede apreciar la plataforma de análisis que incluye las mismas etapas del modelo de AD para la seguridad, anteriormente discutido, con la salvedad de que agregan un elemento denominado Interfaz de Datos que está orientada a permitir la integración de la plataforma de análisis con Herramientas de Gestión de Incidentes[17]. Resulta interesante la combinación de la plataforma de AD con herramientas tales como los gestores de bitácoras y los SIEMs, lo cual le brinda un valor agregado al sistema en cuanto a eficacia al aprovechar las ventajas que ofrecen dichas herramientas, así como las capacidades de estas tecnologías para la visualización orientadas a la seguridad.

Implementar un modelo como el anterior es todo un desafío ya que se requieren de mecanismos que permitan orquestar de forma armónica todas las etapas de dicho modelo.

[17] Grupo de herramientas que permiten dar seguimiento a errores, incidentes, anomalías que se generan en un sistema de información.

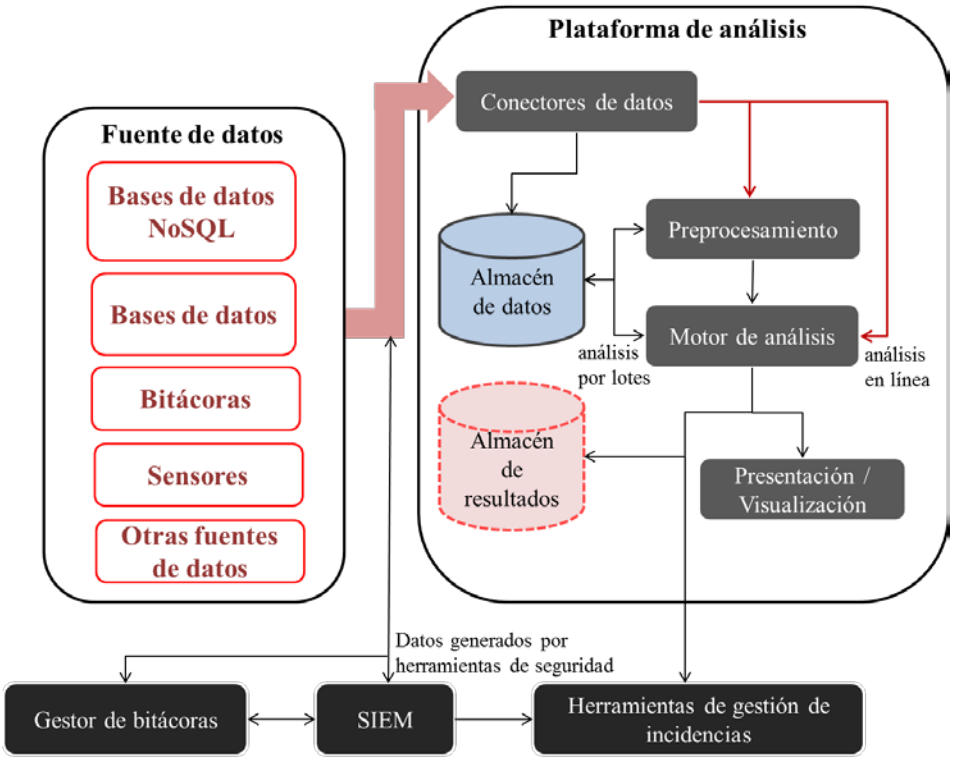


Fig. 5. Modelo de AD orientado a la Ciberseguridad que permite análisis por lotes y por flujo.

En este trabajo nos centraremos en los tipos de análisis por flujo y por lotes. Un esquema como el mostrado en la presente subsección solo permitiría realizar el análisis por lotes por lo que se propone una modificación a dicho esquema para que permita el análisis en línea (otra forma de llamar al SDA). En la Figura 5 se muestra (flechas rojas) cómo después de colectados los datos estos pueden pasar directamente a la etapa de preprocesamiento o análisis sin ser previamente almacenados.

2.6 Requerimientos para una plataforma de AD

Para implementar en la práctica el modelo propuesto por G. Pan y colaboradores en [25] se requiere de una plataforma que provea un conjunto de mecanismos que permitan orquestar todas las etapas de dicho modelo. Dicha plataforma debe de ser capaz de manejar diversas heterogeneidades, tanto de hardware como de software, las cuales estarán de manifiesto principalmente en las fuentes datos, no obstante, las estaciones de trabajo, el acceso y almacenamiento en bases de datos, los algoritmos de MD, entre otros, pueden ser elementos que aumenten el grado de dificultad asociados a dicha problemática.

Otra de las capacidades que debiera de tener dicha plataforma es la de operar de forma distribuida y permitir la modularidad de los componentes de software para adaptarse a las características y complejidades de los escenarios: grandes volúmenes de datos, alta demanda de recursos de cómputo. Esto último implica que la plataforma diseñada bajo la filosofía distribuida debe ser capaz de balancear cargas (distribución de las tareas de cómputo), usar eficientemente recursos como el ancho de banda, tener capacidad de almacenamiento distribuido, tolerancia a fallos y de garantizar escalabilidad horizontal.

2.7 Los sistemas distribuidos y los middleware

Los sistemas distribuidos (SD) permiten realizar tareas de cómputo y procesamiento a lo largo de un conjunto de computadoras independientes y que cooperan entre ellas a través de una red de comunicación, con el fin de lograr un propósito u objetivo específico. Este concepto no es solamente aplicable a las computadoras, es decir, no solo tiene una dimensión física sino también lógica, y desde ese punto de vista se puede definir un sistema distribuido como un conjunto de aplicaciones o secciones de aplicaciones (software) independientes que cooperan entre sí. En este caso, la comunicación entre dichas aplicaciones puede no depender de la red física dado que es posible que todos esos procesos residan en un mismo ordenador [26].

Un sistema distribuido se vale de una capa de gestión encargada de articular la cooperación entre todas las aplicaciones del sistema. Esta capa es conocida como *middleware* y su objetivo principal es el de abstraer a dichas aplicaciones de la infraestructura tecnológica sobre la cual son implementadas así como de otras funciones tales como la adquisición de los datos o el almacenamiento de los resultados. De forma general un middleware es una aplicación que se posiciona encima del sistema operativo y le provee a este un conjunto de herramientas y APIs para poder abstraer a este de determinadas funciones (ver Figura 6).

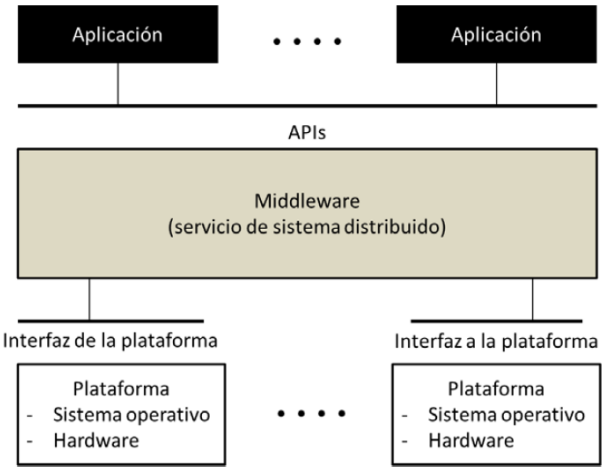


Fig. 6. Capa middleware (Imagen tomada de [27]).

Un middleware debe garantizar como mínimo 4 funcionalidades específicas para que pueda servir de instrumento de implementación de una plataforma de AD para la Ciberseguridad (estos requerimientos también se puede extender a la seguridad en sentido general). Las funcionalidades son las siguientes:

1. Capacidad para acceder a las fuentes de datos y adquisición de estos.
2. Comunicar las aplicaciones del sistema (proveer un mecanismo de comunicación).
3. Gestionar la ejecución de las tareas.
4. Almacenar datos.

En la siguiente sección expondremos algunas de las plataformas para el AD aplicando técnicas de MD orientado a la Ciberseguridad desarrolladas bajo la filosofía sistema distribuido – middleware destacando sus aspectos positivos y negativos.

3 Plataformas de AD para la ciberseguridad

Las plataformas de AD para la Ciberseguridad pueden agruparse en tres categorías (ver Figura 7): (1) de propósito experimental, (2) de propósito específico y (3) de propósito general. La primera categoría agrupa aquellas plataformas dirigidas al desarrollo y prueba de los algoritmos que posteriormente serán aplicados en el escenario real de la Ciberseguridad. Estas plataformas proveen herramientas y otras utilidades que facilitan el trabajo del desarrollador pero no permiten implementar todas las etapas del modelo presentado en la subsección 2.5. El segundo grupo contiene a las plataformas que han sido implementadas en escenarios reales de Ciberseguridad sin embargo, solo realizan un número específico y limitado de tareas dentro de la seguridad. Por último, las plataformas de propósito general tienen en común con las de propósito específico que pueden ser insertadas en escenarios reales de Ciberseguridad. La diferencia radica en que estas últimas pueden ser adecuadas para acometer cualquier tarea dentro de la Ciberseguridad. Otro criterio para el agrupamiento de las plataformas puede ser en base a los tipos de análisis que estas admiten: por lotes, por flujo o interactivo.

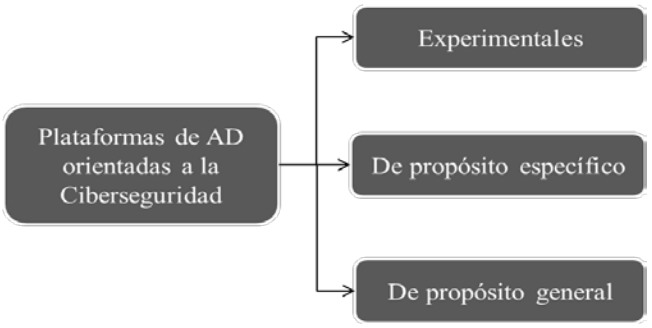


Fig. 7. Taxonomía de las plataformas de AD orientadas a la Ciberseguridad.

El presente trabajo se enfocará en las plataformas pertenecientes a las categorías (2) y (3). Algunos ejemplos de plataformas experimentales de AD son Clementine[18], Weka[19], RapidMiner[20], Knime[21] y Orange[22]. En la literatura revisada se pudo detectar la presencia de una gran cantidad de trabajos vinculados con la aplicación del AD en la Ciberseguridad pero que solo se centran en el mejoramiento de los algoritmos de MD. En [28] resumen varias de las técnicas, enfoques y *frameworks* de MD orientados a la Ciberseguridad y cuáles son los más adecuados para enfrentar determinados tipos de ciberataques.

3.1 Estado del arte de las plataformas de propósito específico

El AD aplicado a tareas específicas de Ciberseguridad con el empleo de la MD, ha sido tema recurrente en la literatura especializada, abarcando temáticas tan diversas como: Detección de Botnets[23] [29], Detección de Malware [30], Análisis de Riesgo [31], Prevención de Fuga o Pérdida de Información [32], Detección de Fraudes [33], Detección de Intrusos [11], entre otras. En el estudio realizado de la bibliografía se destacan fundamentalmente los trabajos dirigidos hacia la Detección de Intrusiones [11],

[18] http://www.ibm.com/analytics/us/en/technology/spss/spss.html
[19] http://www.cs.waikato.ac.nz/ml/weka/index.html
[20] https://rapidminer.com/
[21] http://www.knime.org/
[22] http://orange.biolab.si/
[23] Botnet: Conjunto de computadoras controladas remotamente y empleadas para atacar a otros sistemas informáticos.

encontrándose gran cantidad de artículos dedicados a plataformas cuyo propósito es implementar un Sistema de Detección de Intrusiones. En esta sección se exponen algunos trabajos relacionados con esta temática.

W. Lee y colaboradores propusieron en [34] un NIDS que emplea técnicas de MD para realizar análisis de datos en línea (por flujo). De forma general, intentaron implementar un sistema con un bajo valor de falsos positivos (alta eficacia) y que el costo computacional durante la extracción de rasgos de los datos fuera mínimo (eficiente). Desarrollaron un método genérico que permite extraer rasgos de los datos capturados que serán comparados en un *framework* que combina un modelo de detección de anomalías con otro de detección de ataques por uso inadecuado (*misuse detection*, en inglés) [35]; todo esto con el fin de mejorar la razón de detección global de ataques al sistema. Plantean los autores que la detección de intrusiones puede aportar a la prevención de ataques cuando se aplica en tiempo real, mientras la intrusión está teniendo lugar. Para lograrlo decidieron, entre otras cosas, minimizar el tiempo de cómputo de alguna de las etapas con la implementación de un algoritmo de refinamiento del proceso de cálculo de cada rasgo, lo cual logran según explican, sin comprometer la eficacia.

La arquitectura propuesta en [34] consiste en uno o varios sensores de captura de tráfico de red, uno o múltiples detectores, un almacén de datos (*data warehouse*, en inglés) y un generador de modelos adaptativos (ver Figura 8). Esta plataforma es independiente del formato de los datos capturados por los sensores y de la representación del modelo; por ejemplo, un dato generado por un sensor puede contener un número arbitrario de rasgos y cada rasgo puede ser continuo o discreto, numérico o simbólico; mientras que el modelo puede ser una red neuronal, un conjunto de reglas o probabilístico.

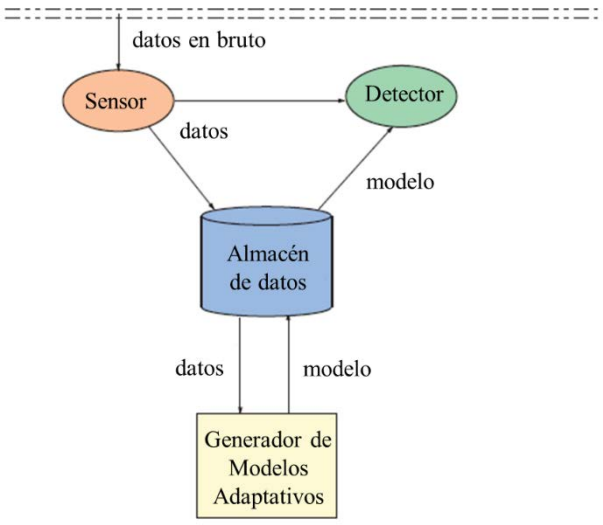


Fig. 8. IDS basado en MD propuesto en [34].

W. Lee y colaboradores utilizan el lenguaje extensible de marcas (XML, por sus siglas en inglés) como mecanismo de comunicación e intercambio de información entre componentes (middleware). XML da soporte de acceso a bases de datos y es útil cuando varias aplicaciones deben comunicarse entre sí o integrar información. XML, es un estándar para el intercambio de información estructurada entre diferentes plataformas y se puede emplear en bases de datos, editores de texto, hojas de cálculo, entre otras aplicaciones. Este lenguaje no es un protocolo de comunicación ni de transporte sino que se encapsula en el Protocolo de Transporte de Hipertexto (HTTP, por sus siglas en inglés) el cual, a su vez, es transportado por el Protocolo de Control de Transmisión (TCP, por sus siglas en inglés).

En el artículo no se ofrecen muchos detalles acerca de los sensores y de las funciones específicas que realizan. En el caso de los detectores, son secciones de código que toman los datos procesados provenientes de los sensores y mediante uno o la combinación de varios de los modelos generados se

intenta detectar el ataque. Los detectores también se encargan de guardar los resultados en el almacén de datos. El sistema puede tener múltiples detectores monitoreando el mismo objetivo, por lo que las cargas de trabajo pueden ser distribuidas (análisis de eventos en paralelo).

Otro de los aspectos que destacan en la propuesta de [34] es la posibilidad de intercambio de información con otros IDSs con el fin de colaborar y de mejorar la eficacia a partir del conocimiento de la mayor cantidad de alertas y patrones de ataques posibles. Para ello se apoyaron en las facilidades que brindan los protocolos Framework Común de Detección de Intrusiones (CIDF, por sus siglas en inglés) [36] y Formato de Intercambio de Mensajes en la Detección de Intrusiones (IDMEF, por sus siglas en inglés) [37] los cuales proveen un formato estándar y seguro para realizar dicho intercambio de información entre IDSs.

Entre las limitaciones que presenta el enfoque anterior están:

- únicamente soporta el procesamiento de flujos de datos; el almacén de datos solo le provee al detector los modelos generados para realizar la detección, es decir, los detectores no son capaces de realizar consultas al almacén de datos y obtener los datos para analizarlos por lotes; esto último le impide al usuario realizar análisis "post mórtem"[24] con el fin de poder detectar, en un futuro, ataques que ya han ocurrido.
- no tiene en cuenta la captura y procesamiento de los datos de forma distribuida, sin embargo, el procesamiento puede hacerse de forma paralela.
- el uso del lenguaje descriptivo XML para transferir datos y manejar la comunicación entre los componentes implica un consumo mayor de tiempo durante el procesamiento de dicho lenguaje debido a la alta verbosidad de la estructura de este.
- el protocolo de intercambio de información, IDMEF, está basado en XML y, por tanto, dicho protocolo acarrea las problemáticas relativas a este lenguaje [11].

Otros enfoques más recientes y similares al de [34] se pueden encontrar en [38] y [39].

En [40] proponen una solución que, a diferencia de [34], utiliza el Protocolo de Acceso Simple de Objetos (SOAP[25], por sus siglas en inglés) para establecer la comunicación entre los procesos, intercambiar datos y almacenar y recuperar información de bases de datos (middleware). SOAP es un protocolo de intercambio de mensajes que permite que aplicaciones que se ejecutan en diferentes sistemas operativos puedan comunicarse utilizando HTTP y su lenguaje descriptivo, XML.

El sistema está compuesto por varios módulos distribuidos entre varios ordenadores. Los módulos son: Motor de Minería de Datos (DME, por sus siglas en inglés), Gestor de Minería de Datos, Motor de Consultas, Motor de Abstracción, Procesador de solicitudes y un Gestor de Datos Históricos (ver Figura 9).

En el Motor de Minería de Datos residen todos los algoritmos que permiten realizar el procesamiento de los datos en bruto provenientes de los IDSs y que son almacenados en una base de datos SQL. El DME accede a los datos mediante la interfaz SOAP la cual provee una forma flexible de realizar consultas a bases SQL. El Gestor de Minería de Datos es el encargado de orquestar el funcionamiento de todo el sistema. El Motor de Consultas adecua las consultas enviadas en XML al formato SQL mientras que el Motor de Abstracción garantiza que el DME reciba los datos provenientes de la BD con el formato requerido por los algoritmos. El Procesador de solicitudes es el mecanismo encargado de distribuir la ejecución de las tareas entre más de un DME (procesamiento paralelo) y, por último, el Gestor de Datos Históricos se encarga de guardar las trazas de las invocaciones realizadas por otras aplicaciones con el fin de registrar los valores de entrada y argumentos utilizados (salva de sesiones).

[24] Análisis que se hace tiempo después de que ocurrió el ataque. Para esto se debe contar con una base de datos con los datos en bruto capturados.

[25] SOAP es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Entre las limitaciones de este sistema están:

- hace un uso ineficiente de los recursos al dedicar por cada módulo (DME, Gestor de Minería de Datos, etc.) un ordenador o servidor.
- está orientado a realizar análisis por lotes y no admite análisis de flujo. Esto se debe a que los datos provenientes de los IDSs son previamente almacenados en una base SQL para luego ser procesados. Este tipo de análisis provee tiempos de respuesta limitados y a estas demoras se le adicionan las provocadas por las interfaces SOAP durante la adecuación de los datos en bruto para que sean almacenados de forma estructurada (ver Figura 9).
- SOAP está basado en XML, por lo que dicho protocolo arrastra las limitaciones relativas a este.
- el mecanismo de intercambio de datos de SOAP es del tipo cliente-servidor [33] lo cual implica mayor consumo de recursos debido a que se necesita de un canal dedicado para responder a cada solicitud realizada e implica, además, un rígido esquema de interacción al no poderse realizar solicitudes de forma asíncrona.

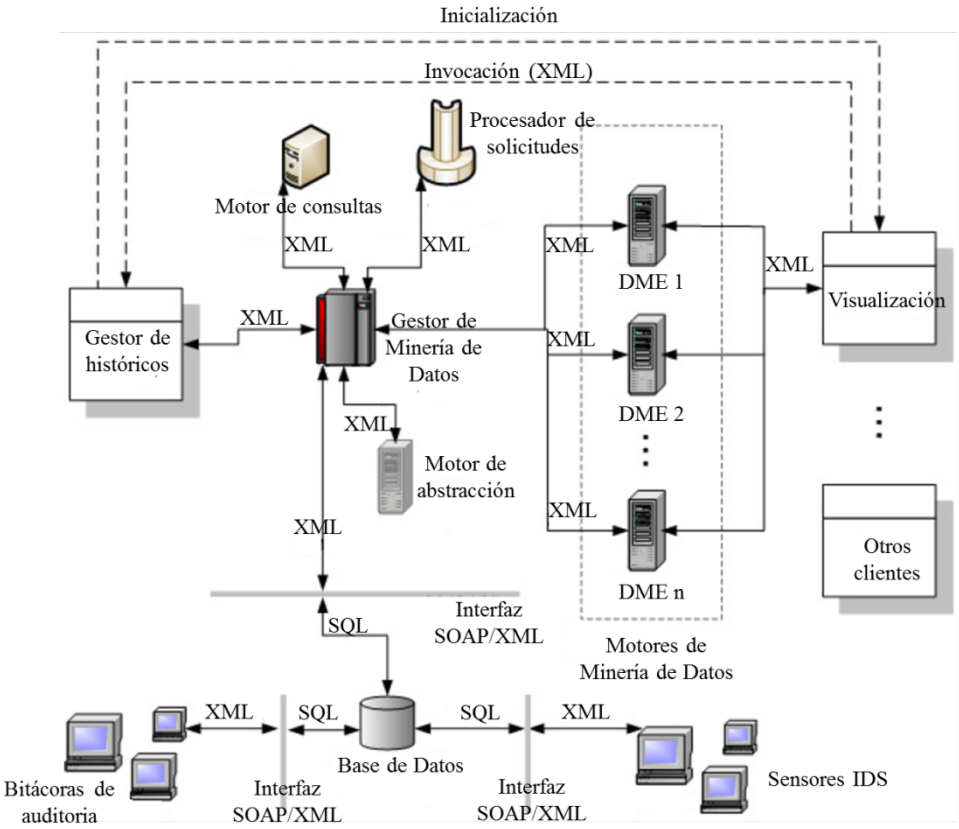


Fig. 9. Arquitectura modular de MD para un IDS (Imagen tomada de [40]).

S. Kim y colaboradores [41] proponen una plataforma que posibilita la operación distribuida de un NIDS, denominado MINDS [42]. MINDS fue desarrollado por especialistas de la Universidad de Minnesota y es un IDS basado en el tráfico de red que utiliza técnicas de MD para identificar tanto intrusiones conocidas como las de nuevo tipo. El diseño inicial de MINDS sigue una filosofía centralizada, es decir, solo analiza información recogida dentro de su red local (LAN, por sus siglas en inglés); lo que implica limitaciones a la hora de identificar ataques que ocurren de forma dispersa, por ejemplo, las APT. La idea de S. Kim y colaboradores es la de ampliar las capacidades de dicho IDS, específicamente de proveerle los mecanismos que le permitan operar de forma distribuida con el fin de

correlacionar eventos generados en diversas LAN así como compartir datos relacionados con anomalías detectadas en otros dominios. Para ello se basaron en la tecnología Grid [43] la cual provee un grupo de herramientas que permiten coordinar y compartir datos y recursos de cómputo de forma distribuida así como también garantizan la disponibilidad de estos.

El sistema general está basado en la Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés). SOA es una filosofía que permite construir aplicaciones en forma de servicios débilmente acoplados, independiente de los detalles de implementación, con interfaces bien definidas y que son accesibles desde la red (Internet por ejemplo). Los servicios Web son una forma de hacer posible una arquitectura de tipo SOA tales servicios webs se basan en el estándar SOAP-XML, aunque existen otros [44].

La Figura 10 muestra la arquitectura propuesta por S. Kim y colaboradores, la que consiste en un conjunto de servicios web basados en SOA que fueron desarrollados a partir de las herramientas que provee la tecnología Grid, en este caso GlobusToolkit 4.0 [45]. Los servicios son: (1) Servicio MINDS, el cual al ser invocado realiza la detección de manera distribuida de intrusiones, (2) Servicio de Almacenamiento, utiliza todos los recursos disponibles para almacenar y recuperar alertas de intrusiones y (3) Servicio de Seguridad para autenticación y control de acceso.

El Servicio MINDS consiste en un contenedor de interacción e interfaz de usuarios (por ejemplo, analistas, suscriptores y administradores) y varios contenedores Grid que albergan el motor MINDS. En el contenedor de interacción se procesan las solicitudes de análisis del sistema y se distribuyen entre los contenedores Grid o nodos de procesamiento. También, dicho contendor, en caso de requerirse, se encarga del particionado de los datos en bruto para que los nodos de procesamiento puedan operar en paralelo. El contenedor de interacción también se encarga de la adición o salida de los contenedores Grid así como del almacenamiento de los resultados y la adquisición de las alertas registradas apoyándose en el Servicio de Almacenamiento.

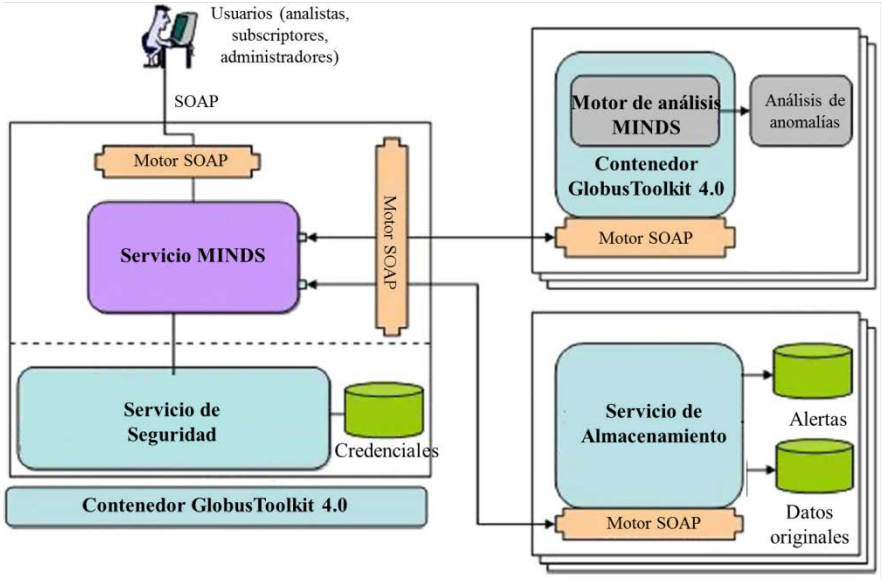


Fig. 10. Esquema de MINDS distribuido.

El Servicio de Seguridad para autenticación y control de acceso es un módulo donde están definidos los usuarios del sistema y sus privilegios dentro de este.

Esta plataforma fue probada en dos escenarios reales de forma distribuida. El primer escenario fue una red local con cuatro ordenadores interconectados a 100 Mbps. Un ordenador fue escogido como

contenedor de interacción y los tres restantes como nodos de procesamiento. El segundo escenario es una red de 20 nodos de procesamiento distribuidos geográficamente e interconectados a través de Internet por lo que el ancho de banda disponible para la comunicación fue menor. Las pruebas realizadas en el primer escenario tuvieron por intención medir las demoras del sistema relativas al establecimiento de la comunicación, al análisis de los datos, al almacenamiento de los resultados (alertas) y a la recuperación de los resultados; en la medida que la cantidad de datos aumentaba. S. Kim y colaboradores reportaron que los tiempos de establecimiento de la comunicación se mantuvieron constantes y prácticamente despreciables, sin embargo, las demoras relativas al análisis, almacenamiento y recuperación de alertas fueron incrementándose linealmente en la medida que los datos aumentaron, teniendo sus valores más elevados durante la recuperación.

La segunda prueba fue realizada en el mismo escenario y consistió en medir el nivel de impacto al realizar el análisis de forma paralela. Para lograrlo, descompusieron (particionaron) los datos capturados en un número determinado de bloques. La cantidad de bloques depende de un factor de descomposición (en el trabajo utilizaron dos factores de descomposición). Luego de particionados los datos estos fueron distribuidos entre los nodos contenedores para que fueran procesados. Los autores comprobaron que cuando el número de datos es pequeño el análisis en paralelo de los bloques ocurre mucho más rápido. Sin embargo, cuando la cantidad de datos aumentan las latencias asociadas al análisis alcanzan valores similares a las del análisis no paralelizado. Los autores consideran que este comportamiento puede deberse al algoritmo de MINDS y plantean que una mejora en la velocidad de procesamiento, cuando la cantidad de datos es elevada, puede lograrse aumentando el número de nodos de cómputo.

En el caso del escenario distribuido geográficamente los autores analizaron el comportamiento de las demoras en cuanto al análisis, almacenamiento y recuperación según se iba incrementando el número de nodos. Los resultados obtenidos mostraron que las latencias en la etapa de análisis y almacenamiento se mantienen casi constantes, sin embargo el tiempo para la recuperación de las alertas crecía continuamente. Los autores plantean que esto se debió a que la función de recuperación necesitaba verificar previamente los privilegios del usuario que solicita las alertas y, a su vez, debe enmascarar aquellos campos de la alerta a los cuales dicho usuario no tiene permiso de lectura, lo cual les resulta computacionalmente costoso.

Entre las limitaciones observadas en esta plataforma están:

- está basada en SOAP.
- solo permite realizar análisis por lotes.
- no se fundamenta el empleo del factor de descomposición.
- el particionado de los datos, a pesar de ser una técnica muy útil para aumentar la eficiencia en el análisis, puede resultar perjudicial para el análisis de eventos donde el factor tiempo es vital, como es el caso de los eventos de seguridad.

Otros ejemplos de plataformas de propósito específico se pueden encontrar en [46] y [11].

Hasta ahora las plataformas de propósito específico analizadas no están orientadas al nuevo enfoque de análisis para grandes volúmenes de información (paradigma *Big Data*). Asumir este paradigma en el AD orientado a la Ciberseguridad, implica realizar varios cambios en la forma de abordar los problemas, particularmente; desde el punto de vista tecnológico. En [13], [23] y [24] se pueden encontrar las nuevas tecnologías y enfoques asociados a dicho paradigma. Se debe destacar que *Big Data* ofrece soluciones cuando se requiere de procesar grandes volúmenes de datos (por ejemplo, > 2 Tera Bytes por día), que son heterogéneos y se necesita de rapidez para procesarlos, es decir; el empleo de esta tecnología y de sus herramientas depende de que el escenario requiera de un enfoque de este tipo.

Un ejemplo de una plataforma de propósito específico orientada a un escenario *Big Data* es la presentada en [47]. En ese trabajo proponen una arquitectura para detectar intrusiones a partir del análisis de grandes cantidades de datos en bruto sin que el sistema se sobrecargue o detenga su

operación. Los objetivos que se proponen cumplir M. D. Holtz y colaboradores son similares a los de [41]: adquirir los datos de forma distribuida, es decir, provenientes de múltiples tramos de red; procesarlos de forma paralela y escalable, correlacionar la información procesada y, finalmente, garantizar el almacenamiento y la recuperación de los resultados en bases dispersas. Todo esto con el propósito de identificar nuevos patrones que describan ataques sofisticados. Sin embargo, la diferencia entre ambas propuestas radica en los datos originales capturados.

Para realizar la captura de los datos de forma distribuida los autores propusieron colocar sensores en diferentes lugares de la red tanto en el equipamiento (enrutadores, conmutadores, entre otros) como en los servidores y en las computadoras de los usuarios. Tales sensores consisten en NIDS y HIDS que capturan el tráfico de la red y generan reportes que describen tanto el comportamiento de la red como de los hosts, para luego enviarlos a un servidor máster. Por tal motivo los datos que se procesan no son los provenientes directamente del tráfico de la red como pasaba en [41], sino que son eventos generados por los dispositivos de seguridad que sí capturan dicho tráfico (ver Figura 11). Una vez que los eventos son enviados al máster, este se encarga de distribuir entre un conjunto de nodos esclavos el almacenamiento y el procesamiento de estos. Para el procesamiento emplearon el *framework* MapReduce [48] mientras que para el almacenamiento distribuido se basaron en el Sistema de Ficheros Distribuidos de Hadoop [49] (HDFS, por sus siglas en inglés).

Para probar la viabilidad de dicha arquitectura los autores realizaron dos experimentos para analizar el desempeño durante el almacenamiento y el procesamiento. Para ambas pruebas utilizaron un nodo máster y cinco nodos esclavos. El primer experimento consistió en medir la diferencia de tiempo que le tomaba al sistema almacenar los datos una vez que son recibidos, para ello generaron ficheros que fueron variando de 1 a 250 mega bytes con un paso constante de 50 mega bytes. Los resultados mostraron que el tiempo de escritura aumenta de forma lineal en la medida que aumenta el tamaño de los ficheros (el tiempo varió de 0 a 25 segundos como máximo) lo que demuestra las posibilidades de escalabilidad en escenarios de grandes cantidades de datos. También plantearon que los tiempos pueden reducirse si se aumenta el *throughput* de la red (ejemplo GigabitEthernet[26]) o si más nodos son adicionados al sistema.

[26] Equivale a una red con ancho de banda de 1 Gbps.

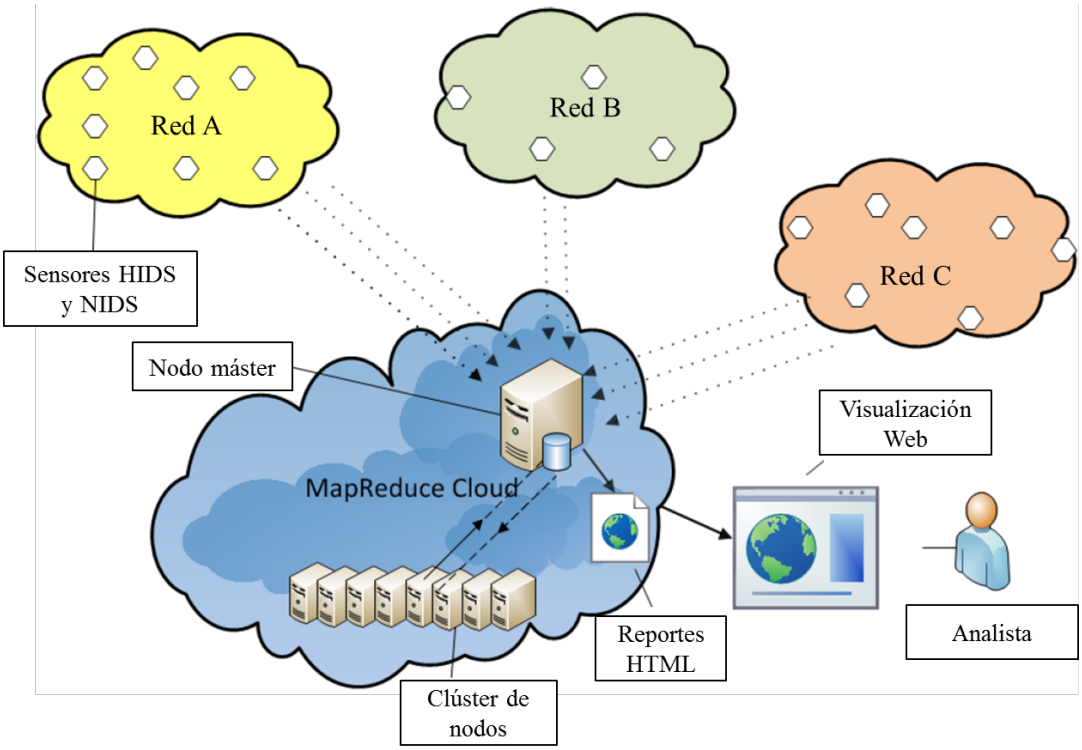


Fig. 11. Arquitectura de IDS distribuid propuesta en [47].

El segundo experimento consistió en correr un algoritmo que ordena los datos guardados durante el primer experimento mientras se medía el tiempo que demora hacer esto. Los resultados mostraron que el sistema mantiene una variación de tiempo que oscila entre 33.4 segundos y 35.6 segundos para todos los tamaños de ficheros debido fundamentalmente a las variaciones del desempeño de la red y a que cada nodo necesita un tiempo de preparación o sincronismo que también afecta la variación global del tiempo. De forma general, explica, las demoras de procesamiento no radican en la ejecución del ordenamiento en si sino en los factores tecnológicos antes mencionados.

M. D. Holtz y colaboradores no especifican los detalles relativos a los mecanismos de comunicación y solo hacen énfasis en los gestores de tareas y en los mecanismos de almacenamiento distribuido. El análisis que permite esta arquitectura es por lotes.

Otros ejemplos de plataformas de propósito específico orientadas a un escenario Big Data se pueden encontrar en [50], [21] y en [51].

3.2 Estado del arte de las plataformas de propósito general orientadas a la ciberseguridad

Las plataformas pertenecientes a este grupo son las de mayor alcance dentro del escenario de la Ciberseguridad. Este tipo de plataformas deben ser lo suficientemente flexibles como para acometer diversas tareas de seguridad, deben admitir tanto el análisis de flujo como por lotes, manejar datos diversos, escalar horizontalmente, acceder a múltiples y diversas fuentes de datos, operar con bases de datos relacionales y no relacionales, garantizar mecanismos de comunicación con calidad de servicio (QoS, por sus siglas en inglés), así como permitir procesamiento y almacenamiento paralelo y distribuido, optimizando el uso de los recursos y garantizando que se pueda alcanzar un equilibrio entre la eficiencia y la eficacia en el resultado del análisis.

Los SIEMs son plataformas de propósito general que reciben datos provenientes de diversas fuentes, monitorean las incidencias de forma continua, correlacionan eventos de seguridad y emiten alertas [52].

Sin embargo, estos sistemas están pensados para procesar datos provenientes de dispositivos de seguridad (IDSs, IPSs, cortafuegos, antivirus, entre otros), así como reportes de eventos provenientes de dispositivos de interconexión (conmutadores o enrutadores de servidores), bitácoras de Sistemas Operativos, alertas de aplicaciones, etc.

Entre las limitaciones más importantes de estos sistemas están:

- las soluciones SIEM emplean bases de datos relacionales porque trabajan con datos estructurados [11].
- la mayoría de los productos SIEM de calidad son comerciales y no admiten modificaciones de su código fuente, con la excepción de OSSIM (Open Source Security Information Management) versión en software libre, y limitada, de AlienVault [53], y desarrollada bajo licencia GPL[27].
- el funcionamiento adecuado de los SIEMs depende de que el analista monitoree sus parámetros constantemente y de que se realicen los ajustes pertinentes en la configuración.
- el diseño centralizado de los SIEMs hace que el costo en cuanto a la utilización de los recursos sea alto ya que todos los datos tienen que ser analizados en un solo lugar.

Los motivos antes expuestos califican a los SIEMs como plataformas de propósito general dentro del escenario de la Ciberseguridad, pero no adecuados para satisfacer todos sus requerimientos.

En la revisión realizada de la bibliografía disponible no se detectó la existencia de otras plataformas de propósito general que permitieran hacer AD orientado a la Ciberseguridad. No obstante algunos enfoques hechos en otras áreas del AD se corresponden con los objetivos de este estudio y permiten obtener una visión clara sobre una plataforma de propósito general que cumpla con las exigencias planteadas al inicio de este acápite. Tales enfoques tienen por base la utilización de middlewares para diseñar sus plataformas. Una valoración respecto a dichas plataformas se expone en el siguiente capítulo.

4 Valoración sobre plataformas de AD de propósito general basadas en la filosofía middleware

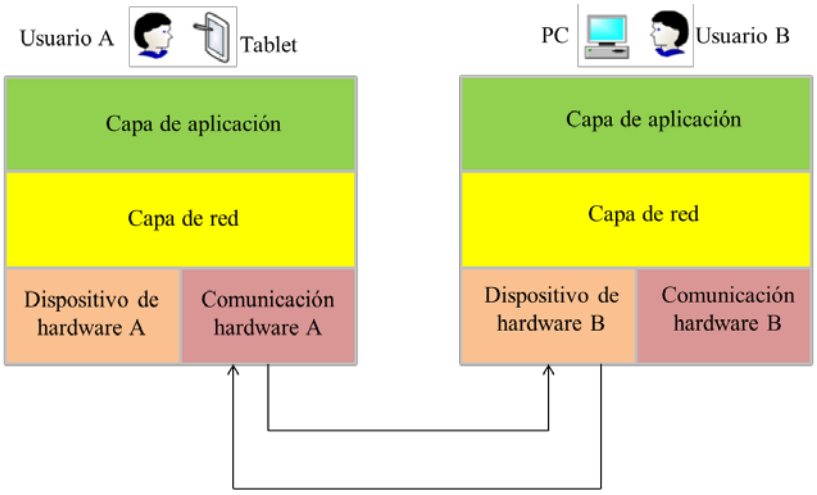
En el capítulo anterior se pudo observar como las plataformas de AD para la Ciberseguridad debían ser capaces de integrar e intercomunicar diversas fuentes de datos con múltiples aplicaciones las cuales, se encargan de ejecutar funciones tales como almacenamiento, preprocesamiento, análisis y/o visualización. Para lograr esto, las soluciones reportadas en el estado del arte, siempre se apoyaron en una capa intermedia de unificación (ej. XML, SOAP/XML o GlobusToolkit 4.0). Dicha capa permite la abstracción no solo de toda la complejidad de hardware o de las aplicaciones presente sino que también provee mecanismos que permiten estructurar de forma modular el sistema, manteniendo al usuario-especialista ajeno de tales detalles y permitiéndole desarrollar o consumir servicios con independencia de la infraestructura tecnológica del sistema. Esta capa se denomina middleware. De lo dicho hasta ahora se puede concluir que un middleware no es un producto o una solución en sí sino un medio para llegar a estas(os).

Veamos el siguiente ejemplo. Dos usuarios de diferentes países realizan una transacción de dinero a través de un servicio web provisto por un banco; dicho servicio se encarga de realizar la conversión monetaria pertinente. Uno de los usuarios utiliza un dispositivo móvil inteligente con sistema operativo (SO) Android[28] y el otro emplea un ordenador con Windows. La transacción debe fluir a través de las capas del modelo de comunicación TCP/IP en donde los datos son transformado sucesivamente

[27] La Licencia Pública General de GNU (GPL por sus siglas en inglés) es una licencia creada por la Free Software Foundation que está orientada principalmente a proteger la libre distribución, modificación y uso del software.

[28] Sistema operativo desarrollado por Google para teléfonos inteligentes.

adecuándose a los requerimientos de cada una de dichas capas. La capa de aplicación en este caso sería la encargada de interactuar con los SO de cada usuario. En la Figura 12 se muestran las capas del modelo de comunicación TCP/IP por las cuales fluye la transacción.



Usuario A

Denominación	Formato	Dato enviado
CUC	int + " CUC"	25 CUC

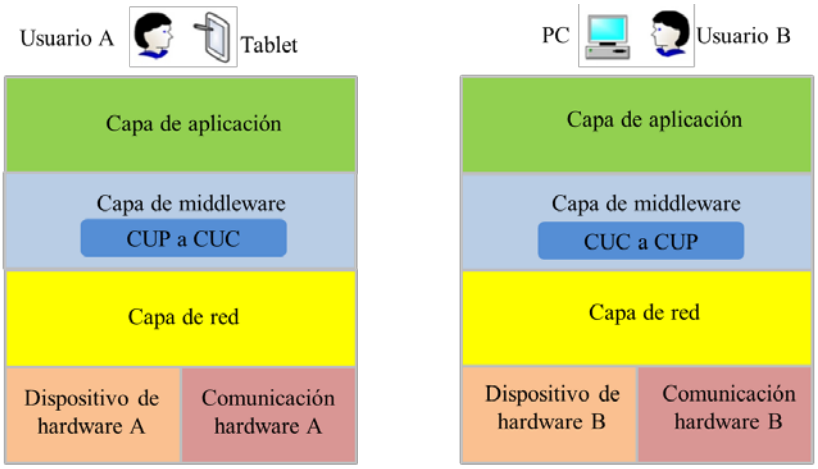
Usuario B

Denominación	Formato	Dato mostrado
CUP	double + " CUP"	25 CUC

Fig. 12. Transacción realizada sin emplear una capa middleware.

Cuando el usuario A realiza la transferencia, le pasa el monto total a transferir a su aplicación Android y esta, a su vez, envía los datos a la capa de aplicación con un formato de tipo entero concatenado con una cadena (ej. int + " CUC" = 25 CUC). Por otra lado, la aplicación Windows del usuario B espera recibir un tipo de dato doble concatenado con una cadena diferente a la de A (ej. double + " CUP" = 625 CUP) sin embargo la capa de aplicación del modelo TCP/IP por sí sola no es capaz de adaptar los datos al formato adecuado para la aplicación receptora, esta solo le presenta la información tal y como la recibe, por lo que el usuario B visualizará en su ordenador 25 CUC en vez de 625 CUP.

En la Figura 13 se muestra como la inserción de una capa middleware es capaz de realizar las conversiones y adecuaciones necesarias del formato de los datos y abstrae a la capa de aplicación de tales funciones y, por consiguiente, el usuario recibe la información esperada.



Usuario A

Denominación	Formato	Dato enviado
CUC	int + " CUC"	25 CUC

Usuario B

Denominación	Formato	Dato mostrado
CUP	double + " CUP"	625 CUP

Fig. 13. Transacción realizada empleando una capa middleware.

El ejemplo anterior demuestra de forma simple la importancia de los middleware para proveer soluciones que determinadas aplicaciones no son capaces de hacerlo.

Las funcionalidades mínimas que debe poseer un middleware orientado al desarrollo de plataformas de AD orientadas a la Ciberseguridad son: acceso a múltiples fuentes de datos, comunicación entre todas las aplicaciones y/o componentes de la plataforma, gestión de la ejecución de las tareas y almacenamiento, además también tienen que garantizar:

- Acceso a las fuentes de datos:
  1) Capacidad para manejar múltiples y diversas fuentes (múltiples y diversos sensores, formatos de datos heterogéneos y bases de datos de diferentes tipos).
  2) Capturar los datos en bruto de forma distribuida y en paralelo.
  3) Adquirir los datos en bruto haciendo un uso eficiente de los recursos (por ejemplo, ancho de banda de la red de datos) siempre que sea posible.

- Comunicación interna de la plataforma:
  1) Deben ser flexibles, es decir, permitir a los componentes realizar solicitudes de forma asíncrona.
  2) Transmisión eficiente de los datos, por ejemplo, a través de la multidifusión de mensajes.
  3) Proveer mecanismos de QoS a la hora de transmitir la información, por ejemplo, haciendo uso de colas.

- Ejecución de las tareas:
  1) Permitir la ejecución de tareas de forma secuencial o paralela.
  2) Capacidad para realizar análisis por flujo o por lotes.
  3) Escalar horizontalmente.

- Almacenamiento.
  1) Soporte para realizar almacenamiento distribuido.
  2) Independencia de las BD y de sus características.

4.1 Plataformas DAvinCi y DIKW

Del estudio de la literatura disponible se detectaron dos plataformas que se adecuan en buena medida a lo que se necesita, estas son:

- Inteligencia Colectiva con Agentes Distribuidos o DAvinCi [54] y
- DIKW [15] (significa Datos, Información, Conocimiento y Sabiduría).

Ambas plataformas provienen del campo de la robótica orientado a la integración de múltiples autómatas equipados con sensores, en una red colaborativa de robots.

4.1.1 Breve descripción y valoración de la plataforma DAvinCi

DAvinCi es una plataforma escalable que permite interconectar varios robots de exploración para que ejecuten tareas de alta demanda de recursos como son el procesamiento de imágenes, detección de objetos, planificación de recorridos, generación de mapas, entre otras, todas ellas en un clúster de alto desempeño. La necesidad de ejecutar tales tareas en procesadores (CPUs) distribuidos se debe a que las capacidades de cómputo de los autómatas es baja y solo garantizan las funciones básicas de este, además, escalar verticalmente estos dispositivos es una solución en extremo costosa. Otro de los propósitos de esta arquitectura es que los autómatas puedan compartir información entre ellos y no tengan que duplicar o repetir tareas ya realizadas.

La plataforma propuesta por R. Arumugam y colaboradores tiene por base el middleware de código abierto orientado al desarrollo de la robótica ROS [27]. Dicho middleware provee una forma estándar de comunicación e intercambio de mensajes (datos) entre hardware (robots) y secciones de programas (en el entorno ROS denominados nodos) y, entre programas. ROS permite el acceso a las fuentes de datos mediante pequeños programas llamados controladores (*drivers*, en inglés) los cuales adquieren los datos a través del protocolo HTTP. También cuenta con un repositorio de bibliotecas, aplicaciones y controladores (similar a los repositorios de las distribuciones de Linux) que es gestionado y actualizado por una amplia comunidad de desarrolladores y colaboradores lo cual le permite manejar las heterogeneidades relativas tanto al hardware como al software. Si bien los datos son capturados de la fuente de la forma cliente-servidor dentro del entorno ROS estos son intercambiados de la forma uno a muchos y muchos a muchos mediante un mecanismo de multidifusión de mensajes denominado publicador / subscriptor (o pub / sub) (ver Figura 14) [55]. Este mecanismo también provee a los nodos ROS abstracción de los tipos de datos, y la cola tipo FIFO garantiza la QoS [56] y la comunicación asíncrona.

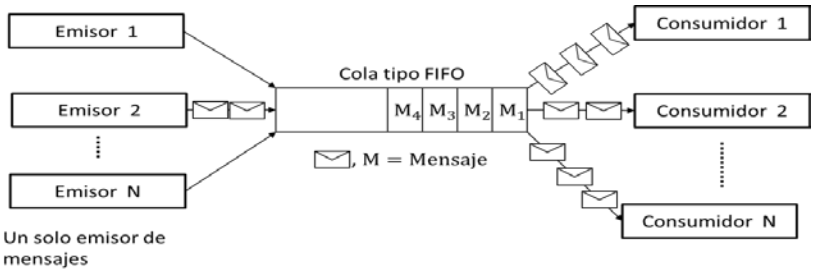


Fig. 14. Paradigma publicador / subscriptor con cola de mensajes tipo FIFO (Imagen tomada de [27]).

ROS carece de mecanismos que se encarguen de la ejecución de las tareas y de almacenar datos, es por eso que R. Arumugam y colaboradores decidieron aumentar las capacidades de ROS al combinarlo con un *framework* que permite el procesamiento paralelo y almacenamiento distribuido de grandes cantidades de datos denominado Hadoop [57].

Hadoop es una plataforma de código abierto integrada por varios componentes que permite almacenar y procesar grandes conjuntos de datos a través de un clúster de computadoras [24]. Permite escalar horizontalmente a cientos de ordenadores y es altamente tolerante a fallos. Los componentes de Hadoop utilizados en DAvinCi fueron: MapReduce para ejecutar tareas en paralelo y HDFS para almacenar los datos de forma distribuida (ver Figura 15). Existen otros componentes pero se salen del alcance del presente trabajo.

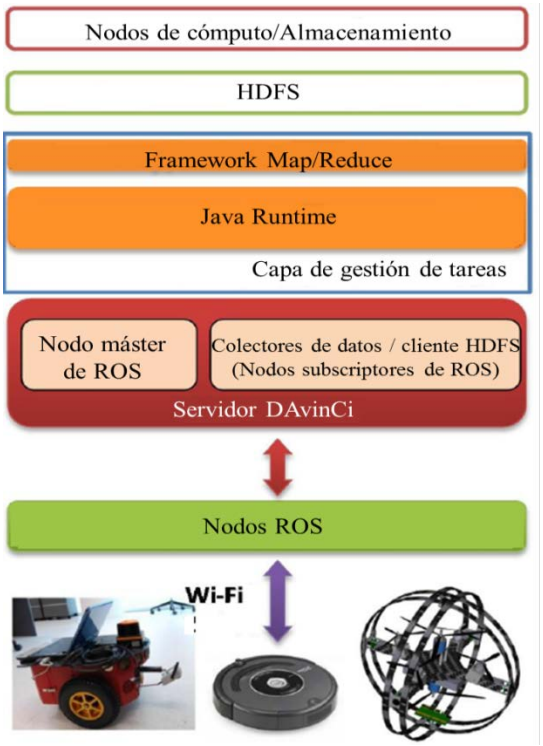


Fig. 15. Arquitectura de la plataforma DAvinCi (Imagen tomada de [54]).

La plataforma DAvinCi realiza todas las funcionalidades necesarias para ser utilizada como posible esquema de referencia para el diseño de una plataforma de propósito general, que con las modificaciones necesarias permitiría obtener una solución para su aplicación en la Ciberseguridad. Entre las limitaciones detectadas en DAvinCi están: que no llega a satisfacer todos los requerimientos asociados a cada funcionalidad. Por ejemplo, Hadoop está orientado únicamente al análisis por lotes así como ROS siempre emplea el protocolo HTTP para transportar los datos provenientes de las fuentes y esto, en el caso de tipos de datos como imágenes o señales de audio, provoca un consumo innecesario de ancho de banda.

4.1.2 Breve descripción y valoración de la plataforma DIKW

DIKW, al igual que DAvinCi, ofrece una solución escalable y distribuida que permita integrar redes de dispositivos automatizados que requieren de elevadas capacidades de procesamiento de grandes cantidades de datos generados por sensores que dichos dispositivos portan consigo.

La plataforma DIKW, a diferencia de DAvinCi, está diseñada para realizar tanto análisis de flujo como por lotes dándole prioridad al primero. Esto se debe a que los sensores generan datos tales como imágenes, video o señales de audio que requieren de análisis en tiempo real es decir, con requerimientos de bajas latencias [16]. El análisis por lotes introduce una demora sustancial durante el procesamiento ya que se requiere que los datos sean previamente almacenados.

DIKW está dividida en tres capas: (1) capa de pasarela, (2) capa de comunicación y la (3) capa de procesamiento (ver Figura 16). En la capa de pasarela ocurre el acceso al flujo de datos generado por la red de sensores o por cualquier otro dispositivo alternativo y se gestionan los controladores de los sensores. Un controlador, al igual que en DAvinCi, es un puente entre los sensores y la plataforma de procesamiento. Cada controlador posee un ID único. Un mismo controlador puede ser replicado varias veces recibiendo por cada instancia un ID diferente. Por otro lado, estos pueden tener varios canales de comunicación y cada uno con un nombre único. Los controladores y demás componentes del sistema difunden la información apoyándose en el esquema de intercambio de mensajes publicador/subscriptor al igual que en DAvinCi.

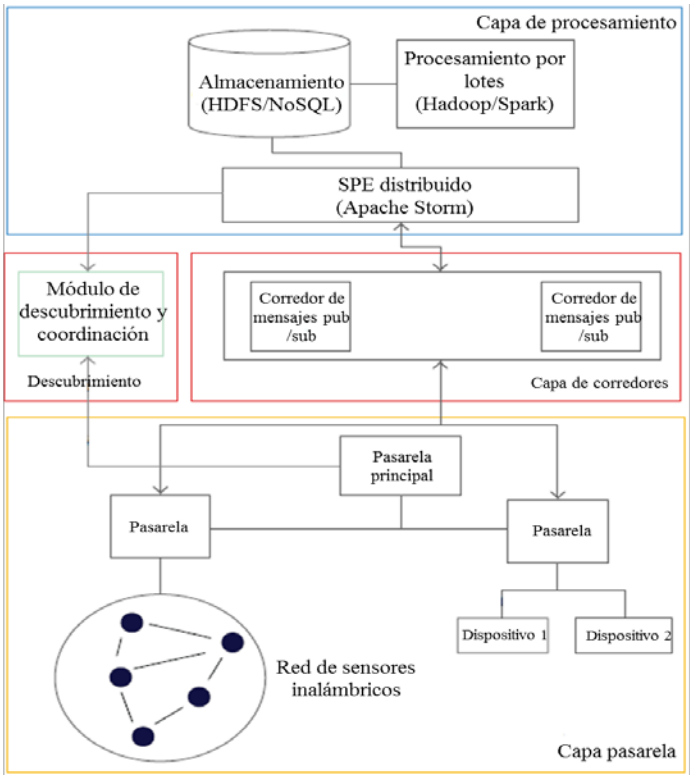


Fig. 16. Plataforma DIKW (Imagen tomada de [15]).

Las pasarelas (*gateway*, en inglés) son los contenedores donde residen los controladores. Una pasarela principal (*master gateway*, en inglés) se encarga de gestionar todos los controladores y es la que permite adicionarlos o quitarlos, activarlos o desactivarlos.

En la capa de comunicación reside un módulo de coordinación y descubrimiento y los corredores de mensajes (*message brokers*, en inglés). En el módulo de coordinación y descubrimiento se almacena la información relativa a los controladores (IDs, nombres de los canales, metadatos, entre otros) y la pasarela principal es quien se la provee. Este módulo es necesario para garantizar el correcto funcionamiento de la capa de procesamiento. Por otro lado, los corredores de mensajes son aplicaciones intermediarias encargadas de entregar el mensaje enviado por el transmisor con formato que sea capaz de interpretar el receptor. Dichas aplicaciones son capaces de realizar otras funciones tales como

balancear cargas y garantizar la entrega de mensajes [58]. En la presente arquitectura los corredores de mensajes reciben los mensajes provenientes de los controladores y los envían a la capa de procesamiento con el formato adecuado para ello. Existen diversas aplicaciones corredoras de mensajes basadas en el paradigma publicador/subscriptor como es el caso de ActiveMQ [59], RabbitMQ [60] y Kafka [61]. Cada uno de estos corredores provee diversas prestaciones, en este caso, S. Kamburugamuve y colaboradores se decantaron por utilizar ActiveMQ y Kafka; el primero se adecua mejor a las exigencias del procesamiento de flujo mientras que el segundo provee mejor escalabilidad lo cual puede ser utilizado en el procesamiento por lotes.

Finalmente, la capa de procesamiento está compuesta por un Motor de Procesamiento de Flujo (SPE, por sus siglas en inglés) llamado Apache Storm [62]. Apache Storm es un SPE que provee un modelo o arquitectura computacional para realizar procesamiento de flujos continuos de datos de forma paralela. En la presente plataforma Storm es utilizado para responder lo antes posible a las solicitudes de procesamiento hechas por los dispositivos (robots, por ejemplo) o puede hacer algún tipo de preprocesamiento de los datos y almacenarlos para que luego sean analizados por lotes [15]. S. Kamburugamuve y colaboradores no llegan a implementar ningún mecanismo de procesamiento por lotes solo sugieren que puede utilizarse HDFS o algún otro tipo de base de datos NoSQL [63] para el almacenamiento. Por último, Storm depende del módulo de coordinación y descubrimiento para realizar el procesamiento paralelo de forma correcta ya que este se encarga coordinar como las unidades de procesamiento paralelo de datos accediendo a la misma fuente de datos mientras evitan su duplicación y que el procesamiento se haga fuera de orden. También le advierten a estas unidades de cuando un canal de datos está activo o inactivo.

DIKW, al igual que DAvinCi, realiza todas las funcionalidades requeridas para ser utilizado como plataforma de propósito general y, además, permite realizar análisis por lotes y de flujo. Sin embargo, su principal debilidad radica en el acceso a las fuentes de datos ya que estos siempre son transportados sobre la red utilizando HTTP y, además, no cuentan con un repositorio de controladores que les permita manejar heterogeneidades relativas a los dispositivos. Otra desventaja de DIKW frente a DAvinCi es su poca flexibilidad. Esto se debe a que los autores diseñaron una solución a la medida de sus necesidades (permitir que los datos generados por una red de sensores sean procesados) y no proveen un entorno que permita insertar de forma sencilla y rápida nuevas aplicaciones (por ejemplo, una GUI de visualización, etc.). En cambio DAvinCi no afronta tales dificultades dado que ROS si provee una interfaz que permite insertar nuevas aplicaciones o secciones de código dentro de su entorno pub / sub. Para ello en su página oficial [64] están publicadas las APIs que permiten construir un nodo ROS.

5 Prueba de concepto del middleware utilizado en la plataforma DAvinCi

Una de las necesidades de la Ciberseguridad es la de poder capturar de forma distribuida la mayor cantidad posible de datos en bruto, con el propósito de poder correlacionar información proveniente de diversas fuentes y así, detectar, por ejemplo, las APT y otros tipos de ataques con características similares. Sin embargo, al transportar tal cantidad de información no siempre se dispone de conexiones rápidas y puede generarse sobresaturación de los canales y por consiguiente demoras en la recepción de los datos. Otro de los aspectos críticos son las fuentes de datos. Varios programas o procesos dentro de la arquitectura, debido al carácter distribuido de esta, pueden hacer solicitudes simultáneas a la misma fuente de datos. Luego de recibida cada solicitud, la fuente dará respuesta a cada una según su capacidad. Un incremento exponencial de las solicitudes puede exceder la capacidad de respuesta del emisor y, por tanto, derivar en una interrupción del envío de los datos, o lo que es lo mismo, se estaría en presencia de una denegación del servicio.

En esta sección se realizará una prueba de concepto del middleware utilizado por R. Arumugam y colaboradores en [54] (ROS) con el propósito de evaluar su desempeño en cuanto al uso eficiente tanto de las fuentes de datos como del canal de transmisión, para lo que se diseñaron tres experimentos.

El primer experimento combina un controlador de cámaras de video AXIS[29] de ROS que accede a una cámara de este tipo y un nodo de visualización que será consumidor del flujo de video capturado por dicho controlador (ver Figura 17). Por otro lado, se tiene una aplicación no basada en un middleware, por ejemplo un navegador web, que accede a la misma cámara (ver Figura 18) y el flujo de video es visualizado en una de las pestañas de dicho navegador. Tanto en el caso de ROS como del navegador el acceso a la cámara es en la forma cliente/servidor y el flujo de video es transportado a través de HTTP.

El segundo experimento es similar al primero pero, en este caso, se aumentó el número de nodos consumidores a tres y, en el caso del navegador, se abrieron tres pestañas cada una con la dirección de la misma cámara. En ambos experimentos se midió el ancho de banda utilizado durante el acceso y se monitorearon la cantidad de instancias que intentan acceder a la fuente de datos. La herramienta utilizada para dicho propósito fue *iftop* que está disponible en el repositorio del SO Linux – Ubuntu.

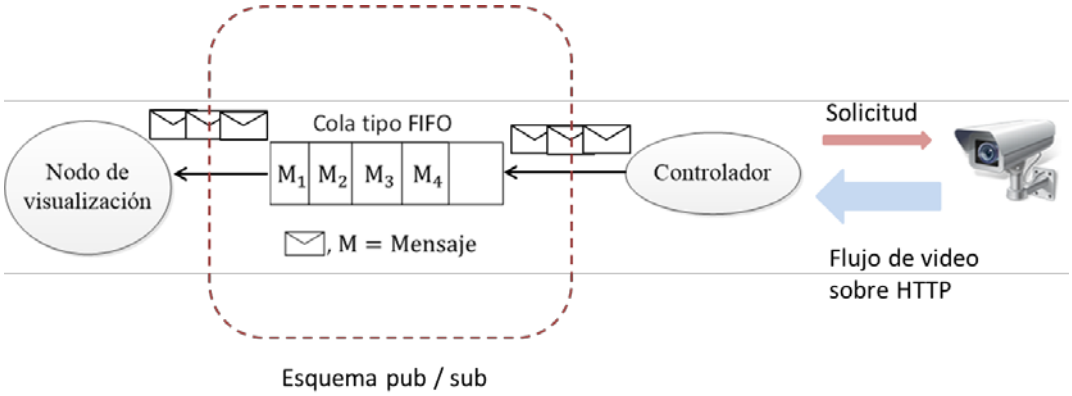
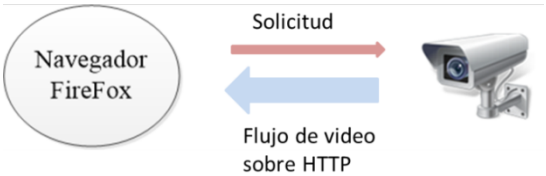


Fig. 17. Accediendo al flujo de video a través del middleware ROS.

En la Figura 19 se muestra el escenario físico del experimento. Se tiene una red LAN de dos terminales, uno es una PC Intel Core 2 Quad con sistema operativo Linux (distribución Ubuntu 14.04) y ROS en su versión Indigo Igloo. El otro terminal es una cámara IP AXIS. En la PC se está ejecutando la aplicación *iftop* sobre la interfaz eth0.

Las Figuras 20 y 21 muestran las salidas del *iftop* del primer experimento cuando se utiliza ROS y cuando se utiliza un navegador web para acceder a la cámara, respectivamente. En ambos casos el número de instancias es el mismo (una) y la utilización del canal es similar (tomando el pico como referencia). Se puede apreciar que cuando se realiza una sola solicitud a la cámara en ambos casos los resultados son similares en cuanto al número de instancias y al consumo de ancho de banda.



[29] http://www.axis.com/global/es/

Fig. 18. Accediendo al flujo de video a través de un navegador.

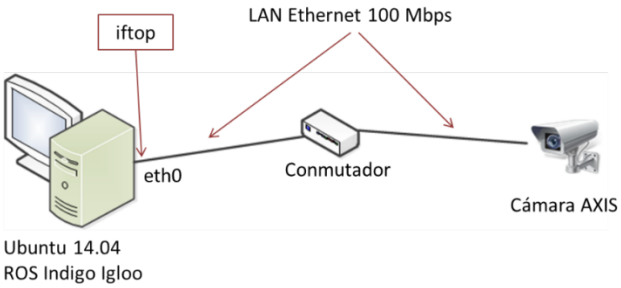


Fig. 19. Escenario físico del experimento.

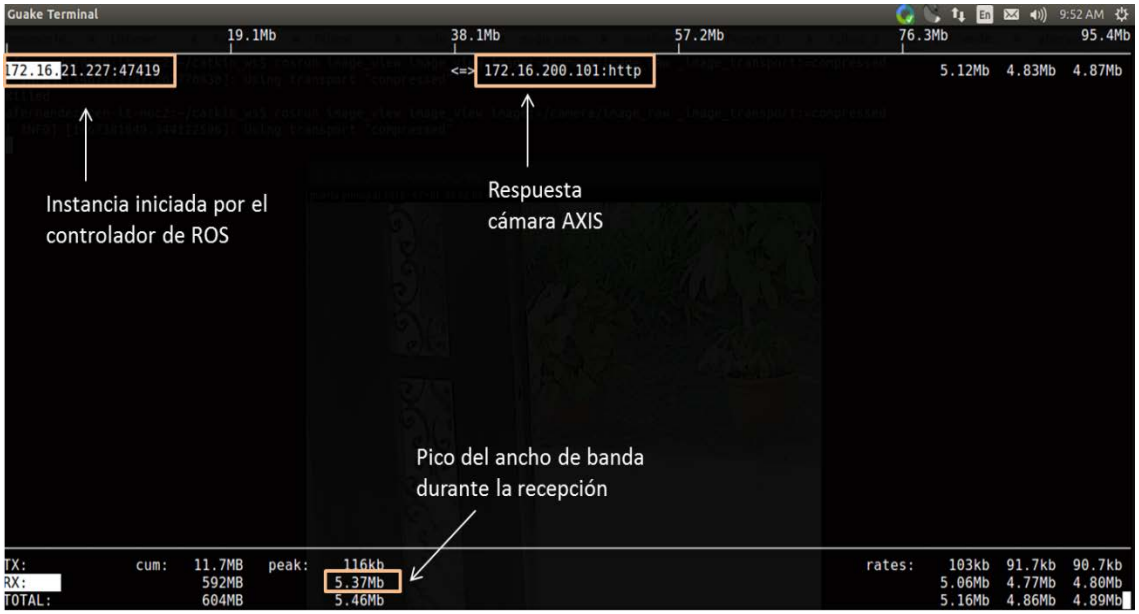


Fig. 20. Utilización de recursos cuando se visualiza el video utilizando el navegador.

Fig. 21. Utilización de recursos cuando se visualiza el video utilizando el middleware ROS.

En las Figuras 22 y 23 se muestran los resultados del segundo experimento cuando se utiliza el navegador para observar el mismo flujo de video en tres pestañas diferentes y cuando se utiliza el middleware ROS con el mismo propósito. En el primer caso, se aprecia como el navegador abre tres canales independientes que intentan acceder a la cámara y esta, a su vez, inicia tres instancias de respuesta. Este comportamiento deriva en un mayor consumo de ancho de banda. El esquema lógico de lo ocurrido se muestra en la Figura 24. Por otro lado, cuando tres nodos de visualización solicitan el flujo de video de la misma cámara, entra en acción el mecanismo de multidifusión de mensajes propio del esquema pub / sub y los datos le son entregados a cada nodo sin tener que iniciarse más de una instancia a dicha fuente, por lo que el consumo del ancho de banda se mantiene inalterado y la fuente sigue respondiendo a una sola solicitud. El esquema lógico de este último escenario se puede observar en la Figura 25. En la tabla 3 se hace un resumen de todos los experimentos.

Este experimento demuestra que el esquema pub / sub de ROS ofrece garantías de uso eficiente de los recursos de red y de las fuentes de datos.

Fig. 22. Utilización de recursos cuando se visualiza el video en tres pestañas del navegador.

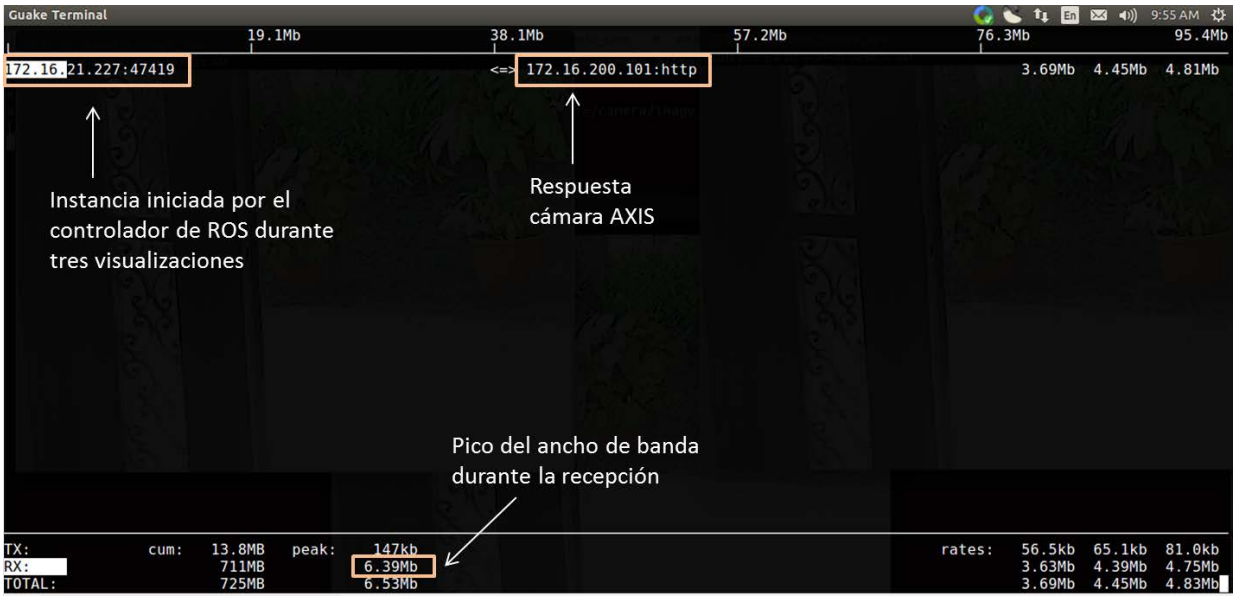


Fig. 23. Utilización de recursos cuando se visualiza el video en tres nodos ROS.

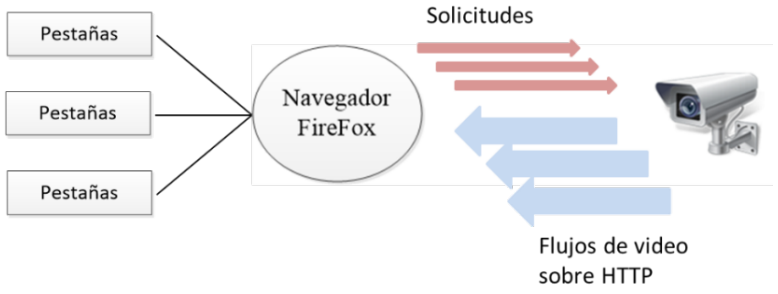


Fig. 24. Visualizando el flujo de video en tres pestanas del navegador.

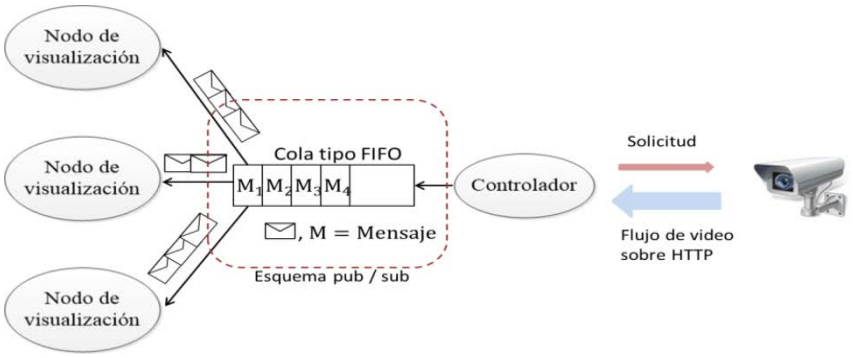


Fig. 25. Visualizando el flujo de video en tres nodos de ROS.

Tabla 2. Resumen de los resultados de los experimentos.

	Utilización del ancho de banda (Mbps)	Pico de utilización durante la recepción (Mbps)	Número de conexiones TCP / IP
Navegador (1 pestaña)	4.87	5.37	1
ROS (1 nodo de visualización)	3.85	5.27	1
Navegador (3 pestañas)	11.3	12.3	3
ROS (3 nodos de visualización)	3.69	6.39	1

6 Conclusiones

En la actualidad la cantidad de soluciones de propósito específico dentro del AD en la Ciberseguridad es mayor que las de propósito general, y dentro de las plataformas de propósito específico las orientadas a la detección de intrusiones son mayoría.

En la Tabla 2 se resumen alguna de las características de las plataformas de propósito específico reportadas en el estado del arte. Entre los aspectos negativos de las propuestas relativas a dichas plataformas están:

- alto consumo de ancho de banda provocado por la transferencia de datos utilizando únicamente el protocolo HTTP. Esto pudiera mejorarse, por ejemplo, empleando técnicas de multidifusión (en la práctica es ampliamente utilizada), lo cual funciona cuando son múltiples los destinatarios, pero no evita la latencia provocada por la ubicación del HTTP dentro de la pila de protocolos, ni tampoco por las propias características de éste, en la que resalta su orientación a conexión. Es posible también emplear un mecanismo de selección del protocolo de transporte adecuado en dependencia del tipo de datos o, también, diseñando un nuevo protocolo.
- las propuestas revisadas carecen de mecanismos de QoS en la entrega de los datos ya que ni XML ni SOAP utilizan un sistema de colas de mensajes [65]. Esto también impide que el sistema pueda realizar comunicaciones asíncronas.
- respecto a SOAP-XML debe significarse que este esquema presenta limitaciones para transportar tipos de datos no estructurados.
- no permiten trabajar con diferentes tipos de análisis de datos a la vez: por lotes y de flujo.

Tabla 3. Resumen de las características principales de las plataformas de propósito específico.

Plataformas de propósito específico	Tipo de análisis	Distri-buida	Acceso a los datos	Mecanismo de comuni-cación	Procesa-miento	Almace-namiento	Escalabi-lidad	Probada
W. Lee y colaboradores [34]	de flujo	no	distribuido	XML/HTTP	paralelo	centralizado	vertical	no
K. Xynos y colaboradores [40]	por lotes	no	no se hace referencia	SOAP/HTTP	paralelo	centralizado	vertical	no
MINDS Distribuido [41]	por lotes	sí	distribuido	SOAP/HTTP	paralelo	distribuido	horizontal	sí
M. D. Holtz y colaboradores [47]	por lotes	sí	distribuido	no especifica	paralelo	distribuido	horizontal	sí
P.G. Prathibha y colaboradores [50]	por lotes	sí	centralizado	no especifica	paralelo	distribuido	horizontal	sí

Las arquitecturas de propósito específico estudiadas no satisfacen las necesidades actuales en el ámbito de la Ciberseguridad, sin embargo esto no implica que se descarte su uso ya que su implementación no es compleja ni demanda muchos recursos.

Los esquemas con un alcance más heterogéneo como son los sistemas SIEM, a pesar de constituir un paso evolutivo importante en relación con los enfoques específicos, presentan restricciones entre las que se destaca que dentro de su generalidad (permiten enfrentar varios tópicos de seguridad) se encuentran limitados al análisis de datos generados por dispositivos de seguridad, aplicaciones, sistemas operativos, etc. Por una parte esto permite reducir las capacidades de almacenamiento necesarias para realizar análisis "post mórtem", no se guardan los datos originales, solo un "resumen" de los mismos. Pero por la otra, esto tiene un gran inconveniente, como es preciso limpiar el dato en bruto para su procesamiento y no hay posibilidades de trabajar a posteriori con ese mismo dato original, se pierde información valiosa que pudiera ser aprovechada para el descubrimiento de nuevos eventos de seguridad.

Aparte de los sistemas SIEM no se encontraron plataformas de AD de propósito general en el ámbito de la Ciberseguridad, sin embargo, existen algunas plataformas no orientadas a este escenario que ofrecen una aproximación bastante cercana a la arquitectura que se desea. Las plataformas DAvinCi y DIKW no pueden ser utilizadas para el AD en la Ciberseguridad sin sufrir cambios importantes que permitan adecuarlas a los nuevos requerimientos. A continuación se exponen algunos aspectos relacionados con ambas arquitecturas, los cuales pueden dificultar su aplicación en los escenarios de Ciberseguridad:

1) La mayoría de los algoritmos tradicionales de MD no están diseñados para trabajar en paralelo lo cual limita el uso de las herramientas Hadoop y ApacheStorm [4].
2) La principal limitación de MapReduce radica en que es ineficiente ejecutando algoritmos recursivos [24].
3) El rendimiento de Hadoop depende en buena medida de la velocidad de entrada/salida de los dispositivos de almacenamiento [24].
4) No hacen énfasis en el consumo del ancho de banda en la red.
5) Carencia de controladores que permitan acceder a fuentes de datos relativas a la Ciberseguridad (dispositivos de red, IDSs, DLPs, entre otros).
6) Hadoop y Apache Storm son herramientas orientadas a escenarios Big Data y no todos los escenarios donde debe aplicarse la Ciberseguridad tienen esa característica.

Diseñar una plataforma para el AD orientado a la Ciberseguridad basada en los enfoques de [15] y [54] implica abordar las problemáticas antes expuestas.

6.1 Trabajo futuro

Nuestro proyecto es definir y desarrollar en una primera etapa los componentes necesarios para obtener una plataforma de AD para la Ciberseguridad definida a partir de los requerimientos que se han explicado a lo largo del presente informe. Este tipo de sistema permitiría enfrentar tareas relacionadas con la detección de ataques cibernéticos, fraudes, fuga de información sensible, etc., de acuerdo con los intereses expresados por los especialistas de las áreas de aplicación de los resultados.

Una arquitectura con esas características precisa de una capa intermedia (middleware) que se encargue de abstraer a las capas superiores de las cuestiones relativas a los sensores-capturadores-generadores de datos en bruto, que sea capaz de seleccionar automáticamente el protocolo adecuado para el transporte de los datos desde su origen hasta los procesos destinatarios. También será necesario que esta plataforma pueda procesar datos en flujo o por lotes, múltiples y diversos, lo cual implica que

en alguna etapa futura deberán incluirse el almacenamiento de datos y su procesamiento como prioridad en el estudio. Debe garantizarse además que sea capaz de adecuarse a sistemas de bajos recursos y que posibilite escalar horizontalmente garantizando el trabajo de análisis en escenarios *Big Data*.

Debido a la importancia que reviste, el módulo de selección automática del protocolo de transporte debe permitir que los controladores adquieran los datos de la forma más eficiente posible. En el entorno de la Ciberseguridad se requiere de la captura de los datos de manera distribuida pero el análisis de estos no debe ocurrir aisladamente sino correlacionando toda la información disponible, por ejemplo, en un centro de datos. Transportar tal cantidad de información implica un alto consumo de recursos de red. Este problema puede enfrentarse mejorando la infraestructura de la red, seleccionando el protocolo de transporte en función del tipo de dato u optimizando los protocolos de transporte. La primera alternativa es costosa e inviable para muchas empresas u organizaciones. La segunda solución se basa en evaluar el tipo de dato a transferir para determinar que protocolo de transporte es el más adecuado para ello. La tercera consiste en mejorar los protocolos de transporte de datos tradicionales. En este sentido se han propuesto algunas técnicas como TCP Nice [66], Dr. Multicast [67] o DCCP [68]. La solución en la cual se propone trabajar combinaría los enfoques dos y tres.

En este estudio, a partir del resultado obtenido en los experimentos desarrollados, se detectó además como un problema a resolver la obtención de un esquema multimodal de control y coordinación con el propósito de posibilitar el aumento de la escalabilidad horizontal del sistema, garantizando la tolerancia a fallos.

Finalmente, en este estudio se ha podido determinar algunos de los problemas relacionados con la obtención de una plataforma para el AD en tareas de Ciberseguridad, pero es oportuno subrayar que aún quedan aspectos medulares por abordar debido a la amplitud de la temática y la diversidad de escenarios que precisan de este tipo de soluciones, por ejemplo, que tengan en cuenta el análisis y procesamiento en tiempo real de todo tipo de datos. El nivel de complejidad y las características constantemente en cambio del Ciberespacio, escenario donde se aplica la Ciberseguridad, y la importancia que este reviste para las entidades nacionales, expresada por los expertos en las diferentes áreas de aplicación ponen de relieve la necesidad de darle continuidad a esta investigación.

7 Referencias

1. Lehto, M.: Phenomena in the Cyber World. In: Cyber Security: Analytics, Technology and Automation. pp. 3–29. Springer (2015).
2. Wamala, F.: The ITU National Cybersecurity Strategy Guide. Presented at the September (2011).
3. Libicki, M.C.: Conquest in cyberspace: national security and information warfare. Cambridge University Press (2007).
4. Tsai, C.-W., Lai, C.-F., Chao, H.-C., Vasilakos, A.V.: Big data analytics: a survey. J. Big Data. 2, 1–32 (2015).
5. Siwak, G., Siebenman, T., Witt, M.: Intrusion Detection System. (2016).
6. Vacca, J.R.: Computer and Information Security Handbook. Newnes (2012).
7. Oltsik, J.: An Analytics-based Approach to Cybersecurity. (2015).
8. Lee, W., Stolfo, S.J., others: Data Mining Approaches for Intrusion Detection. In: Usenix security (1998).
9. Zope, A.R., Vidhate, A., Harale, N.: Data Mining Approach in Security Information and Event Management. Int. J. Future Comput. Commun. 2, 80 (2013).
10. Shackleford, Da.: Using Analytics to Predict Future Attacks and Breaches. SANS Institute (2016).
11. Zuech, R., Khoshgoftaar, T.M., Wald, R.: Intrusion detection and big heterogeneous data: a survey. J. Big Data. 2, 1 (2015).

12. Runkler, T.A.: Data analytics: Models and algorithms for intelligent data analysis. Springer Science & Business Media (2012).
13. Begoli, E.: A short survey on the state of the art in architectures and platforms for large scale data analysis and knowledge discovery from data. In: Proceedings of the WICSA/ECSA 2012 Companion Volume. pp. 177–183. ACM (2012).
14. Strickland, J.: Data Analytics Using Open-Source Tools. Lulu.com (2016).
15. Kamburugamuve, S., Christiansen, L., Fox, G.: A framework for real time processing of sensor data in the cloud. J. Sens. 2015, (2015).
16. Stonebraker, M., Çetintemel, U., Zdonik, S.: The 8 requirements of real-time stream processing. ACM SIGMOD Rec. 34, 42–47 (2005).
17. Al-Radaideh, Q.A., Al-Shawakfa, E.M., Al-Najjar, M.I.: Mining student data using decision trees. In: International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan (2006).
18. Azevedo, A.I.R.L., Santos, M.F.: Kdd, semma crisp-dm: a parallel overviee. IADS-DM. (2008).
19. CRISP-DM, still the top methodology for analytics, data mining, or data science projects, http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html.
20. Emani, C.K., Cullot, N., Nicolle, C.: Understandable big data: A survey. Comput. Sci. Rev. 17, 70–81 (2015).
21. Mahmood, T., Afzal, U.: Security Analytics: Big Data Analytics for cybersecurity: A review of trends, techniques and tools. In: Information assurance (ncia), 2013 2nd national conference on. pp. 129–134. IEEE (2013).
22. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mob. Netw. Appl. 19, 171–209 (2014).
23. Chen, C.P., Zhang, C.-Y.: Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Inf. Sci. 275, 314–347 (2014).
24. Singh, D., Reddy, C.K.: A survey on platforms for big data analytics. J. Big Data. 2, 1 (2014).
25. Pan, G., Seow, P.S., Chan, C., Lim, C.Y.: Analytics and Cybersecurity: The Shape of Things to Come, (2015).
26. Puder, A., Römer, K., Pilhofer, F.: Distributed systems architecture: a middleware approach. Elsevier (2006).
27. Fernández Rivas, A., Pérez García, O.A., Hernández Palanacar, J.: Estado del arte sobre aplicaciones del middleware ROS. CENATAV (2016).
28. Said, H.M., El Gohary, R., Hamdy, M., Salem, A.-B.M.: A Study on Data Mining Frameworks in Cyber Security. (2013).
29. García, S., Zunino, A., Campo, M.: Survey on network-based botnet detection methods. Secur. Commun. Netw. 7, 878–903 (2014).
30. Fan, Y., Ye, Y., Chen, L.: Malicious sequential pattern mining for automatic malware detection. Expert Syst. Appl. 52, 16–25 (2016).
31. Fenton, N., Neil, M.: Risk assessment and decision analysis with Bayesian networks. CRC Press (2012).
32. Alneyadi, S., Sithirasenan, E., Muthukkumarasamy, V.: A survey on data leakage prevention systems. J. Netw. Comput. Appl. 62, 137–152 (2016).
33. Abdallah, A., Maarof, M.A., Zainal, A.: Fraud detection system: A survey. J. Netw. Comput. Appl. 68, 90–113 (2016).
34. Lee, W., Stolfo, S.J., Chan, P.K., Eskin, E., Fan, W., Miller, M., Hershkop, S., Zhang, J.: Real time data mining-based intrusion detection. In: DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings. pp. 89–100. IEEE (2001).
35. Kim, G., Lee, S., Kim, S.: A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst. Appl. 41, 1690–1700 (2014).

36. Kahn, C., Porras, P.A., Staniford-Chen, S., Tung, B.: A Common Intrusion... In: Journal of Computer Security. Citeseer (1998).
37. Debar, H., Curry, D.A., Feinstein, B.S.: The intrusion detection message exchange format (IDMEF). (2007).
38. Rao, K.M., Babu, G.R., Nazeer, S.: Intrusion Detection Using Knowledge Discovery Method. Data Min. Knowl. Eng. 4, 226–231 (2012).
39. Haque, M.J., Magld, K.W., Hundewale, N.: An intelligent approach for Intrusion Detection based on data mining techniques. In: Multimedia Computing and Systems (ICMCS), 2012 International Conference on. pp. 12–16. IEEE (2012).
40. Xynos, K., Blyth, A.: A Modular Data Mining Architecture for Intrusion Detection Systems' Data. In: 4th European Conference on Information Warfare and Security. pp. 229–237 (2008).
41. Kim, S., Kim, J., Weissman, J.B.: A security-enabled grid system for MINDS distributed data mining. J. Grid Comput. 12, 521–542 (2014).
42. Kumar, V.: Parallel and distributed computing for cybersecurity. Distrib. Syst. Online IEEE. 6, (2005).
43. Jacob, B., Brown, M., Fukui, K., Trivedi, N., others: Introduction to grid computing. IBM Redb. (2005).
44. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S., others: Web services description language (WSDL) 1.1. Citeseer (2001).
45. Foster, I.: Globus toolkit version 4: Software for service-oriented systems. In: IFIP international conference on network and parallel computing. pp. 2–13. Springer (2005).
46. Lv, G., Xu, R., Zu, X., Deng, L.: Information Security Monitoring System Based on Data Mining. In: Proceedings of the 2009 Fifth International Conference on Information Assurance and Security - Volume 01. pp. 472–475. IEEE Computer Society, Washington, DC, USA (2009).
47. Holtz, M.D., David, B.M., de Sousa Júnior, R.T.: Building scalable distributed intrusion detection systems based on the mapreduce framework. Rev. Telecommun. 13, 22 (2011).
48. Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U.: The rise of "big data" on cloud computing: Review and open research issues. Inf. Syst. 47, 98–115 (2015).
49. Rashmi, K.V., Shah, N.B., Gu, D., Kuang, H., Borthakur, D., Ramchandran, K.: A solution to the network challenges of data recovery in erasure-coded distributed storage systems: A study on the Facebook warehouse cluster. In: Presented as part of the 5th USENIX Workshop on Hot Topics in Storage and File Systems (2013).
50. Prathibha, P.G., Dileesh, E.D.: Design of a hybrid intrusion detection system using snort and hadoop. Int. J. Comput. Appl. 73, (2013).
51. Tian, G., Wang, Z., Yin, X., Li, Z., Shi, X., Lu, Z., Zhou, C., Yu, Y., Wu, D.: TADOOP: Mining Network Traffic Anomalies with Hadoop. In: International Conference on Security and Privacy in Communication Systems. pp. 175–192. Springer (2015).
52. Aguirre, I., Alonso, S.: Improving the automation of security information management: A collaborative approach. IEEE Secur. Priv. 10, 55–59 (2012).
53. Nicolett, M., Kavanagh, K.M.: Magic quadrant for security information and event management. Gart. RAS Core Reasearch Note May 2009. (2011).
54. Arumugam, R., Enti, V.R., Bingbing, L., Xiaojun, W., Baskaran, K., Kong, F.F., Kumar, A.S., Meng, K.D., Kit, G.W.: DAvinCi: A cloud computing framework for service robots. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on. pp. 3084–3089. IEEE (2010).
55. Kambatla, K., Kollias, G., Kumar, V., Grama, A.: Trends in big data analytics. J. Parallel Distrib. Comput. 74, 2561–2573 (2014).
56. Kim, Y., Lee, K., Shroff, N.B.: An analytical framework to characterize the efficiency and delay in a mobile data offloading system. In: Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing. pp. 267–276. ACM (2014).
57. White, T.: Hadoop: The definitive guide. O'Reilly Media, Inc. (2012).

58. Tang, K., Wang, Y., Liu, H., Sheng, Y., Wang, X., Wei, Z.: Design and implementation of push notification system based on the MQTT protocol. In: International Conference on Information Science and Computer Applications (ISCA 2013). pp. 116–119 (2013).
59. Snyder, B., Bosnanac, D., Davies, R.: ActiveMQ in action. Manning (2011).
60. Videla, A., Williams, J.J.: RabbitMQ in action. Manning (2012).
61. Kreps, J., Narkhede, N., Rao, J., others: Kafka: A distributed messaging system for log processing. In: Proceedings of the NetDB. pp. 1–7 (2011).
62. Anderson, Q.: Storm real-time processing cookbook. Packt Publishing Ltd (2013).
63. Assunção, M.D., Calheiros, R.N., Bianchi, S., Netto, M.A., Buyya, R.: Big Data computing and clouds: Trends and future directions. J. Parallel Distrib. Comput. 79, 3–15 (2015).
64. ROS.org | Powering the world's robots, http://www.ros.org/.
65. Devaram, K., Andresen, D.: SOAP optimization via parameterized client-side caching. In: Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003). pp. 785–790 (2003).
66. Venkataramani, A., Kokku, R., Dahlin, M.: TCP Nice: A mechanism for background transfers. ACM SIGOPS Oper. Syst. Rev. 36, 329–343 (2002).
67. Vigfusson, Y., Abu-Libdeh, H., Balakrishnan, M., Birman, K., Burgess, R., Chockler, G., Li, H., Tock, Y.: Dr. multicast: Rx for data center communication scalability. In: Proceedings of the 5th European conference on Computer systems. pp. 349–362. ACM (2010).
68. Kohler, E., Handley, M., Floyd, S.: Designing DCCP: Congestion control without reliability. In: ACM SIGCOMM Computer Communication Review. pp. 27–38. ACM (2006).

RT_037, octubre 2016
Aprobado por el Consejo Científico CENATAV
Derechos Reservados © CENATAV 2016
Editor: Lic. Lucía González Bayona
Diseño de Portada: Di. Alejandro Pérez Abraham
RNPS No. 2143
ISSN 2072-6260
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. A No. 21406 e/214 y 216, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200
Impreso en Cuba