MINERÍA DE PROCESOS

MÁSTER EN CIENCIA DE DATOS E INGENIERÍA DE COMPUTADORES

Análisis de un Artículo de Investigación

Autora
Lidia Sánchez Mérida

Curso 2021 - 2022

Granada, Junio de 2022

Descripción del problema

Un **sistema de detección de intrusiones** (IDS) es capaz de monitorizar el tráfico entrante y saliente de una red de sistemas con el principal objetivo de detectar accesos no autorizados. Tras identificar una posible situación maliciosa, generalmente los IDS alertan a los administradores de redes para que, como expertos, revisen personalmente la información asociada y confirmen o descarten la alerta. No obstante, este procedimiento plantea el siguiente listado de inconvenientes:

- La cantidad de **recursos** necesaria para efectuar los mecanismos de revisión manualmente son desmesurados tanto temporal como personalmente.
- La mayoría de IDS no disponen de técnicas de detección de **conexiones lógicas o causales** entre las alertas que aparecen, lo que produce volúmenes masivos de datos en los que se incluyen posibles alertas redundantes y pérdida de conocimiento útil con el que mejorar el rendimiento del sistema.

Por lo tanto el objetivo de este artículo consiste en descubrir las distintas actividades que se realizan en **ataques multi-fase** para diseñar un IDS automatizado. Su principal cometido consiste en detectar operaciones sospechosas y facilitar la visualización de la información generada para su posterior análisis por un equipo de expertos.

Relevancia del problema

Según los autores de la publicación [1] en la última década se han implementado multitud de redes de sistemas, servicios y aplicaciones cuyo requisito indispensable reside en disponer de una conexión a Internet. Si bien puede ser beneficioso por su facilidad de acceso desde cualquier punto y dispositivo, también presenta un peligro intrínseco fundamentado en la posible explotación de las vulnerabilidades que no hayan sido parcheadas. Así la disciplina de la **ciberseguridad** continúa en auge diseñando sistemas, como los IDS, que son capaces de combatir los problemas de seguridad relacionados con el despliegue de un mayor número de **sistemas *online***.

En mi opinión esta temática debería ocupar un puesto central en todas las compañías vinculadas con proyectos *software* y/o de información puesto que en ellos se generan y almacenan datos de distinta índole con un valor notablemente alto para la propia organización, e incluso más elevado para terceras partes. Sin embargo, este aspecto suele ser uno de los que menos recursos recibe para implementar medidas de seguridad que minimicen la exposición y riesgos asociados a la intrusión de entidades externas a los sistemas *online* disponibles.

Técnicas propuestas

Con el objetivo de utilizar técnicas de descubrimiento de procesos pertenecientes al área conocida como Minería de Procesos, a continuación se definen las siguientes entidades y sus restricciones asociadas que se encuentran almacenadas en los **registros de logs** empleados como datos de entrada:

- Cada **evento** se corresponde con una actividad efectuada, siendo un ejemplo representativo la operación de identificación de usuarios.
- Un **caso** es una entidad compuesta por un listado de eventos relacionados con la misma temática. Continuando con el ejemplo anterior, su respectivo caso trata las actividades de registro de un usuario.
- Es fundamental respetar el **orden** en el que suceden los distintos eventos en cada caso puesto que los algoritmos de Minería de Procesos hacen uso de esta información con el fin de establecer sus dependencias causales.
- Finalmente se presentan dos entidades adicionales: **recurso**, al que se vincula la ejecución de las actividades, y **tiempo** que recopila la duración temporal de ejecución de cada una.

Tras establecer los criterios de calidad de los registros de logs que se pretenden emplear para producir un modelo de procesos, continuamos detallando las cuatro fases que constan en el artículo. En la primera de ellas el propósito principal consiste en **agrupar las actividades** de los mismos ataques para posteriormente investigar las relaciones entre sí. Existen diversas estrategias de agrupamiento, como ***one-to-many*** consistente en vincular la dirección IP del atacante con las direcciones destinos de las víctimas, su opuesta denominada ***many-to-one*** agrupando las direcciones IP de los diferentes delincuentes con la dirección de la entidad perjudicada, e incluso considerando aquellos casos en los que los sistemas comprometidos son empleados para cometer más ataques. Finalmente se ejecuta un **filtrado** con el fin de suprimir grupos compuestos por un único elemento, representando ataques finalizados tras la intrusión, así como clústeres que reflejan bucles de ataques.

En la segunda etapa se pretende transformar los agrupamientos resultantes en un **registro de logs** verificando las condiciones concretadas anteriormente. Para ello se procede a aplicar técnicas de **selección de características** con las que considerar únicamente los siguientes atributos:

- *Activity,* que representa la acción detectada.
- *Case,* que recopila el grupo de actividades que ocurren en el mismo período de tiempo.

- *Time,* utilizado para ordenar los eventos sucedidos a lo largo del tiempo.

La tercera fase dispone de un **algoritmo de descubrimiento** de procesos definido a partir de un conjunto de vértices, que simbolizan las actividades almacenadas en el registro de logs, individualizando la inicial y final, además de un listado de enlaces que especifica la secuencia de cada ataque detectado. Tanto los nodos como las relaciones se encuentran ponderados a partir del cálculo de la **frecuencia** de aparición de los eventos y sus conexiones, respectivamente. Su funcionamiento consiste en añadir una nueva actividad en cada iteración estableciendo sus vínculos cronológicos con los que componer la secuencia de cada ataque.

En la última etapa el objetivo consiste en determinar la complejidad del modelo de procesos resultante considerando el nivel de legibilidad en función de las competencias de sus usuarios finales: los administradores de redes. En caso de ser muy elevada, se aplica un **algoritmo de *clustering* jerárquico** capaz de generar submodelos más sencillos de analizar. Como paso previo se transforman los registros de logs en **vectores binarios de características** con los que calcular una matriz de distancia entre las distintas actividades registradas y así determinar el nivel de similitud entre sí. Así es posible agrupar aquellas alertas de mayor semejanza en clústeres que posteriormente pueden ser traducidos a modelos de procesos de menor tamaño [1].

Estado del arte

Existen multitud de estrategias diseñadas para su integración en IDS, una de las más destacadas en trabajos como [2] y [3] se denomina ***Signature-based IDS*** basada en los conjuntos de vulnerabilidades de sistemas y ataques conocidos. Los pilares de su defensa residen en **evitar la explotación** de errores previamente definidos en bases de datos públicas preparando el sistema para luchar contra los mecanismos de intrusión conocidos hasta el momento. Para ello suelen emplear métodos de **extracción de reglas** con los que modelar el conjunto de actividades característico de cada ataque, de modo que puedan ser posteriormente utilizados en la monitorización de los eventos en red.

Otra de las aproximaciones con más referencias es la conocida como ***Anomaly-based IDS***, cuyo principal cometido consiste en identificar **patrones extraños** con respecto al comportamiento estimado del tráfico de red. Pueden ser implementados de manera ***offline*** en caso de que la información no varíe en períodos de tiempo prolongados, ***online*** cuyo enfoque consiste en analizar secuencias temporales de las actividades

registradas en tiempo real, o **distribuida** si existen varios sistemas involucrados. En esta última situación, debido al considerable volumen de datos, basta con analizar una porción con la que descubrir, principalmente, aumentos elevados de recursos a partir de umbrales predefinidos [2] [3].

No obstante, gracias los relevantes progresos e investigaciones en el área de la Inteligencia Artificial, uno de los enfoques más novedosos para la resolución de este problema consiste en emplear algoritmos de **Aprendizaje Automático**, con el principal objetivo de construir modelos predictivos cuya aptitud reside en **identificar actividades maliciosas** mediante la exploración del tráfico en redes. En esta metodología se integran tres fases fundamentales, siendo la primera el preprocesamiento, la normalización y transformación del registro de actividades en conjuntos de entrenamiento y validación, para ser posteriormente utilizados en las etapas de aprendizaje y evaluación del modelo. Si bien la mayoría de las técnicas destacadas en este trabajo [4] se encuentran orientadas a la resolución de problemas de **clasificación**, como los Árboles de Decisión o los K vecinos más cercanos, también son aplicables métodos de distinta índole como los algoritmos de ***clustering***, e inclusive combinación de diversos modelos conocidos como ***ensembles***.

Adicionalmente se han planteado aproximaciones al diseño de IDS haciendo referencia a diversos algoritmos de ***Deep Learning***, siendo los más relevantes las Redes Neuronales Recurrentes por su habilidad de retener cierta información contextual que ayuda a mejorar su precisión, las Redes Neuronales Convolutivas por su buen rendimiento frente a representaciones vectoriales de los datos, y los *AutoEncoders* como algoritmo de aprendizaje no supervisado cuyo objetivo consiste en seleccionar aquella muestra más similar a la información proporcionada como entrada a partir del estudio y selección de las mejores características [4].

Comparativa entre técnicas tradicionales y las propuestas

Comenzamos examinando la primera metodología de la sección anterior denominada *Signature-Based IDS* en la que destaca una relevante desventaja con respecto a la detección **exclusiva de ataques conocidos**, lo que incrementa la probabilidad de la existencia de falsos negativos, es decir, situaciones maliciosas que comprometen el estado de la red y no son detectadas. Al **no considerar las posibles interconexiones** entre el conjunto de tareas asociadas a diferentes ataques, tampoco es capaz de reconocer operaciones sospechosas a partir de su similitud con otras ya registradas como sí lo puede efectuar el sistema propuesto gracias al análisis y agrupamiento de las actividades que realiza en las primeras fases. Finalmente cabe destacar la

desmesurada inversión de recursos necesarios en el mantenimiento y actualización constante de este IDS para la incorporación de un mayor número de vulnerabilidades y ataques conforme se den a conocer.

En relación al segundo enfoque alternativo denominado *Anomaly-based IDS* podemos intuir que el primer reto intrínseco a su metodología reside en el modelado del **comportamiento normal** del tráfico de red con el que detectar situaciones anómalas. Presumiblemente los sistemas *online* son propicios a la aparición de cambios de conceptos que requieren la adaptación del modelo a las nuevas características. En caso contrario la tasa de falsos positivos podría elevar cuantiosamente su número resultando en un sistema completamente obsoleto e inservible. Al no considerar algoritmos de descubrimiento con los que analizar las disparidades entre el procedimiento implantado y el descriptivo, este tipo de sistemas parecen ser demasiado estáticos con respecto a un ámbito tan dinámico como es la ciberseguridad.

Finalmente la comparativa entre los métodos de Aprendizaje Automático y *Deep Learning* en relación al sistema propuesto en [1] es una de las más igualadas puesto que sus algoritmos son popularmente conocidos por su considerable habilidad para adaptarse y modelar cualquier tipo de problema, lo que ha conllevado su aplicación en numerosos ámbitos de diversa naturaleza. No obstante también presentan ciertas desventajas frente a la combinación de técnicas de Minería de Procesos y Aprendizaje No Supervisado. Particularmente se encuentra la **falta de interpretabilidad** especialmente significativa en aquellas arquitecturas más complejas, lo que puede suponer una limitación dependiendo del grado de comprensión requerido por los expertos en áreas más sensibles, como la medicina. Por el contrario, en el artículo estudiado su principal objetivo consiste en generar y simplificar modelos de procesos para que sean fácilmente analizados por los administradores de redes. Otro de los aspectos a considerar se fundamenta en el crecimiento exponencial de **recursos computacionales** necesarios para construir modelos a partir de técnicas que generalmente proporcionan muy buenos resultados. Si bien el descubrimiento y modelado de ataques de intrusión conlleva cierta complejidad, la mayoría de operaciones que se describen en el artículo se llevan a cabo de forma vectorial y matricial, por lo que sus costes asociados son considerablemente menores.

Relación entre el artículo y la asignatura

El principal vínculo que presenta el trabajo [1] con respecto a la asignatura reside en la implementación y explotación de un algoritmo de **descubrimiento de procesos** con el que modelar los diferentes tipos de ataques recopilados en un fichero de logs. Sin

embargo, también integra operaciones orientadas a la **detección**, en este caso de intrusiones, con las que se pretende alertar a los expertos de posibles amenazas que comprometan la seguridad de una red de sistemas. De esta forma se pretende proporcionar una herramienta cuyo conocimiento se fundamenta en el conjunto de actividades que llevan a cabo los delincuentes para efectuar accesos no autorizados a partir del registro de tráfico real. Asimismo, se trata de un ejemplo tremendamente representativo de los beneficios producidos a raíz de la combinación de técnicas de **Minería de Procesos y Minería de Datos**, que pese a ser dos ámbitos completamente diferentes, pueden generar conocimiento de significativa utilidad para construir sistemas con un único propósito.

Tras explorar el artículo en detalle he podido comprobar que las métricas de evaluación de la calidad del IDS planteado se encuentran recopiladas en la temática de la asignatura, siendo las principales la **matriz de confusión** como medida numérica y ***simplicity***, con la que determinar el nivel de complejidad del modelo generado para proceder a su simplificación mediante algoritmos de *clustering*.

Aportaciones técnicas

Según el artículo examinado [1] su principal contribución consiste en la construcción de un **modelo de procesos** cuyo cometido sea la representación del comportamiento relativo a delincuentes que cometen accesos no autorizados en sistemas conectados en red. Para ello en la Figura 1 se encuentra definido el listado de técnicas requeridas, comenzando por un **registro de logs** en el que se recopilan trazas ejemplificantes de las actividades que ocurren durante la monitorización del tráfico de red. Al disponer de un gran volumen de información y con el objetivo de analizar las relaciones que presumiblemente existen entre los diferentes eventos que ocurren en diversos ataques, en la siguiente fase se aplican diferentes estrategias de **agrupamiento**, tal y como se ha comentado en secciones anteriores. Finalmente se consideran únicamente aquellas alertas relacionadas con los **ataques multi-fase**, que se presentan como el principal caso de estudio para su empleabilidad en la generación del modelo de procesos.

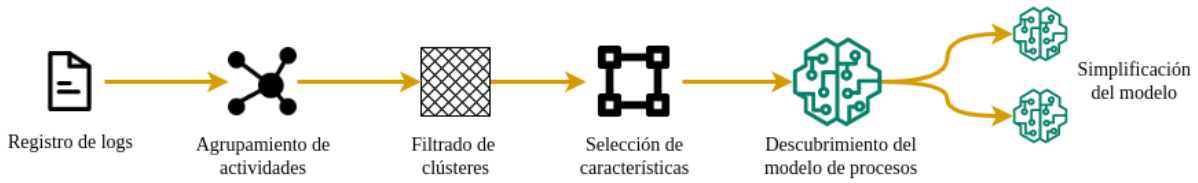

Figura 1. Etapas para definir un modelo de procesos que representa el comportamiento de los ataques de intrusión en sistemas.

Una vez se dispone de un conjunto de datos preprocesado y tras descubrir las posibles correlaciones entre eventos, la siguiente etapa simplifica la ejecución del algoritmo de descubrimiento **seleccionando aquellas características** más relevantes, como son las actividades individuales, sus agrupamientos y tiempos de aparición para establecer su orden cronológico.

La segunda aportación de este trabajo está relacionada con el **algoritmo de descubrimiento** definido brevemente en la Figura 2. Tal y como se puede apreciar, los datos de entrada se componen del **registro de logs** resultante del último paso descrito anteriormente. Posteriormente comienza a ejecutarse el algoritmo de descubrimiento estableciendo, en primer lugar, un nodo raíz como representante de la **primera actividad** del fichero. A continuación, itera sobre los diferentes agrupamientos de alertas para generar tantos nodos como eventos disponibles en el orden cronológico en el que se han registrado. Adicionalmente se calcula el **número de ocurrencias** de cada vértice y enlace con los que establecer sus pesos y no replicar la misma estructura ahorrando recursos computacionales y temporales. Para terminar cierra la jerarquía establecida añadiendo la **última actividad** recopilada en el archivo de logs.

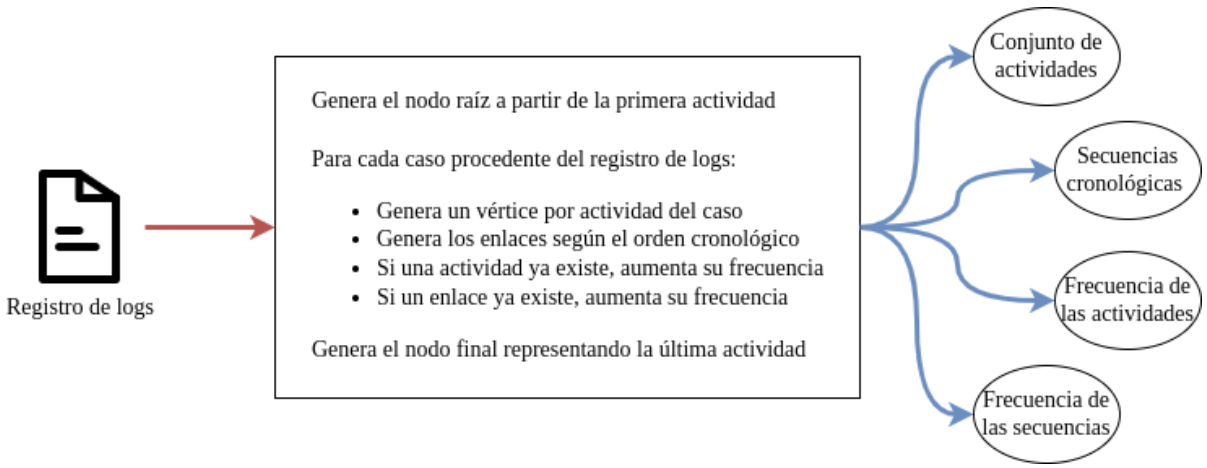

Figura 2. Esquema del algoritmo de descubrimiento para generar el modelo de procesos que represente ataques de intrusión.

Destacamos un segundo algoritmo definido para la **abreviación del modelo** de procesos producido en caso de que su complejidad sea considerablemente excesiva que no resulte útil a los expertos para su análisis del tráfico de red. Recibe como argumentos de entrada el **modelo construido** y una lista de **modelos de referencia** en la que se recopilan sus principales métricas, como el número de vértices, enlaces, el valor de *sparsity* y si se caracteriza por ser complejo en función de estos parámetros. En primer lugar transforma los grupos de eventos intrínsecos al modelo en **vectores binarios de características** con los que establecer la ejecución de las diferentes

actividades. Con esta representación numérica, a continuación calcula el grado de **similitud** entre cada vector utilizando la distancia de *Jaccard* por estar específicamente orientada al cálculo de distancias en problemas binarios. Como la mayoría de algoritmos de *clustering* jerárquico, genera un **dendrograma** con el que determina el nivel de **complejidad** del modelo de procesos mediante el número de vértices, enlaces y *sparsity* detectados. Si superan los umbrales estimados a partir de los modelos de referencia, el algoritmo continúa su descenso en profundidad por el dendrograma dividiendo los casos participantes en grupos menos densos. En caso de que la complejidad sea aún inasumible y se alcance el **último nodo**, produce diversas agrupaciones del conjunto de eventos del caso actual con el objetivo de almacenar aquellas actividades cuya duración no supere el límite de tiempo fijado. Este proceso se repite de forma **iterativa** hasta que el grado de dificultad se caracteriza por ser inferior al umbral calculado. Como salida genera los diferentes submodelos en los que ha sido simplificado el modelo de procesos entrante.

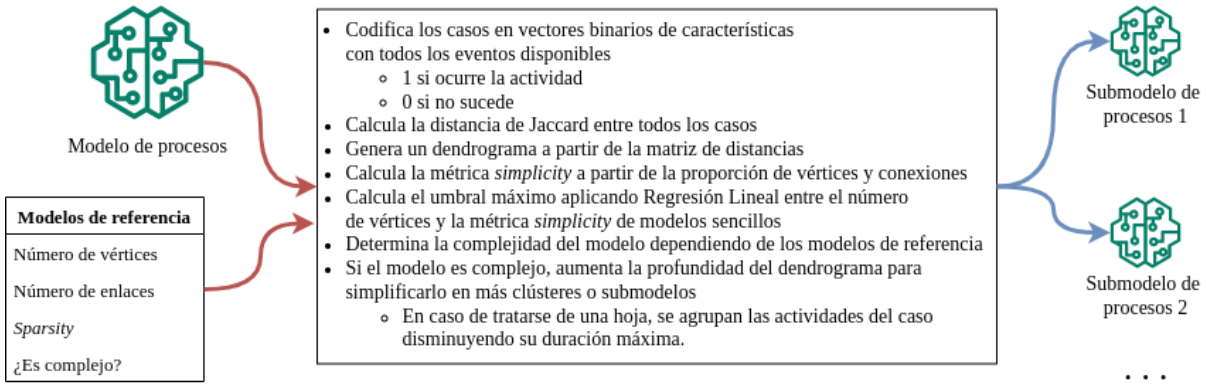

Figura 3. Esquema del algoritmo de agrupamiento que simplifica el modelo de procesos generado.

Relevancia del artículo

El conjunto de datos utilizado para realizar los diversos experimentos que demuestran el funcionamiento y las contribuciones del IDS propuesto en el artículo [1] se encuentra vinculado a un **caso real**. Se trata de un registro de logs producido por un ***Signature-Based IDS*** cuyo cometido consistía en monitorizar la red de la Universidad de Maryland durante el año 2012. Debido al volumen de datos masivo generado han

establecido un período de análisis **diario** para ayudar a los expertos a conformar una visión precisa y continua del estado de los más de cuarenta mil ordenadores participantes acerca de las amenazas que surgen diariamente.

El primer ensayo efectuado se basa en el día del año que **mayor número de alertas** se han recibido con un total de más de veintiséis millones. Sin embargo, tras aplicar las etapas de agrupamiento y filtrado detalladas en secciones anteriores **únicamente se considera el 2%** del conjunto total de eventos. Algunas de las razones explicativas que proporcionan de este suceso se fundamentan en la alta tasa de **falsos positivos** relativa al empleo de uno de los IDS más simples de los modelos existentes, e inclusive a la alta tasa de **falsos negativos**, es decir, ataques no reconocidos por el IDS que se hayan podido producir en las instalaciones de la universidad sin haber sido detectados. Por lo tanto el primer inconveniente que se vislumbra del uso de esta fuente de datos consiste en la **falta de información** que produce como consecuencia del alto porcentaje de sendas métricas. Esto implica que las intrusiones no identificadas no podrán ser analizadas para contribuir en la composición del modelo de procesos, por lo que puede conllevar un **decremento de su precisión** al reflejar los diversos procedimientos que realizan los atacantes para efectuar accesos no autorizados en redes. Tras disponer de un listado de eventos preprocesado y adaptado a los argumentos de entrada del algoritmo de descubrimiento, se ha obtenido un modelo de procesos considerablemente complejo que ha sido dividido en veinticinco grupos. Esta técnica de **simplificación** parece ser una de las contribuciones más relevantes que combina la producción de resultados automatizados con el análisis y supervisión por parte de expertos humanos. Así han podido concluir que, en este experimento, predominan las actividades de encubrimiento del ataque, la intercepción y manipulación del tráfico de red, así como el escaneo de puertos y la extracción de información confidencial. Adicionalmente, gracias a la búsqueda de modelos interpretables y al conocimiento de expertos en esta temática, se ha podido determinar la existencia de un **patrón de ataque** compuesto por cuatro fases: análisis de la red de sistemas, explotación de sus vulnerabilidades, ocultamiento de la actividad maliciosa y manutención del control remoto el máximo tiempo posible. Una vez disponen del procedimiento más popularmente utilizado para vulnerar los sistemas de esta universidad, se puede afrontar la siguiente fase de defensa en la que se deciden las medidas de seguridad más adecuadas para frustrar los ataques de intrusión perpetrados mediante la ejecución de las etapas descubiertas.

Los pilares del segundo experimento se centran en seleccionar el día del año que **mayor número de ataques** diferentes se han producido, a partir de la dirección IP fuente, con el fin de ampliar el abanico de ataques disponibles con los que precisar el

modelo de procesos. Una situación similar ocurre al intento anterior en el que se reduce drásticamente el conjunto de datos final tras el preprocesamiento y filtrado del archivo de logs. No obstante, en este caso el modelo resultante no se clasifica como complejo por lo que se puede efectuar su estudio directamente. Como conclusiones principales se determina la aparición de los **dos ataques más frecuentes** en ese período de tiempo: la colocación de una puerta trasera con la que mantener el control remoto de los sistemas vulnerados, y la extracción de la versión del número de DNS para explotar las vulnerabilidades asociadas. Con esta información los expertos son capaces de reflexionar acerca de los **métodos preventivos** más eficientes que ayuden a frustrar la aplicación de estas técnicas maliciosas y así evitar la continuación de los ataques. Además, el sistema planteado por este artículo [1] también proporciona la lista de **elementos objetivos** en los que centrar un mayor número de esfuerzos para su protección contra intrusiones.

Finalmente el último ejemplo ideado por los autores se encuentra enfocado en analizar el día con **mayor número de firmas diferentes** con el que se pretende validar el mecanismo de simplificación del modelo de procesos. Como consecuencia del enorme volumen de información generado, tras simplificar el modelo resultante se consiguen un total de once clústeres que reflejan la actuación de **tres atacantes** particulares para un total de ochenta y siete alertas generadas en el día elegido. De nuevo, gracias a su división maximizando la interpretabilidad por los expertos en redes, han podido concluir que la totalidad de los ataques perseguían el objetivo de **explotar vulnerabilidades** asociadas al protocolo de comunicaciones CGI que permite la comunicación de los sistemas con programas externos. Por lo tanto los recursos deben posicionarse en torno al conjunto de servidores que hacen referencia a esta metodología de conversación entre diferentes agentes conectados a la misma red.

Bibliografía

1. Process mining and hierarchical clustering to help intrusion alert visualization, Sean Carlistode Alvarenga, Sylvio Barbon Jr, Rodrigo Sanches Miani, Michel Cukier, Bruno Bogaz Zarpelão, 2018
2. A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions, Ankit Thakkar, Ritika Lohiya, 2022
3. Intrusion detection system: A comprehensive review, Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, Kuang-Yuan Tung, 2013

4. Network intrusion detection system: A systematic study of machine learning and deep learning approaches, Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, Farhan Ahmad, 2020