# Hacia la detección de impostores mediante grafos de navegación

J. Benito Camiña, Carlos Hernández-Gracidas y Raúl Monroy

ITESM CEM, Atizapán, Estado de México, México
{a00965049, carloshg, raulm}@itesm.mx

**Resumen** Dado que la información es un bien valioso, la detección de impostores es vital en los sistemas computacionales. El auge en el estudio de la detección de impostores se dio principalmente a partir del trabajo seminal del grupo de Schonlau, donde perfilan a los usuarios con base en los comandos que utilizan, produciendo un conjunto de datos que, por años, ha sido el estándar para comparar métodos de detección de impostores. Sin embargo, el desempeño de estos métodos no es concluyente, y, como resultado, se han investigado fuentes alternativas de información. En este artículo mostramos cómo se construyó un conjunto de datos conteniendo perfiles de usuarios a partir de sus accesos a objetos. Complementariamente, se describe una serie de ataques simulados. Finalmente, proponemos el uso de un vector de atributos como una forma de caracterizar al usuario y diferenciarlo de un impostor, proponiendo posibles experimentos sobre este conjunto de datos.

**Palabras clave:** Detección de intrusiones, conjunto de datos, grafos de navegación.

## 1. Introducción

Para el mundo actual, donde se acepta que la información es un bien extremadamente valioso, la detección a tiempo cuando nuestra computadora (sesión) está siendo ilegalmente utilizada por un intruso, conocido como impostor, es vital. El propósito de un impostor es poner en riesgo al usuario de una computadora, tan rápido como sea posible, mientras se hace pasar por el usuario para evitar ser detectado.

La detección de intrusos ha sido estudiada activamente a partir del trabajo seminal de Schonlau et al. [1], que sugiere que, para perfilar a un usuario, debe ser suficiente con modelar el historial de los comandos (sin parámetros) que introduce mientras tiene una sesión UNIX activa. Para tal propósito, Schonlau et al. desarrollaron un conjunto de datos de impostores, comúnmente conocido como SEA [2], que ha sido utilizado como un estándar *de facto* para construir, validar y comparar una gran cantidad de métodos de detección (véanse, por ejemplo, [3,4]). Sin embargo, el desempeño de métodos basados en SEA no puede decirse que sea abrumador. Así, SEA ha sido enriquecido para considerar información

adicional (en particular, argumentos de comandos); véase por ejemplo, [6]. También como resultado, la investigación en detección de impostores ha optado por el uso de fuentes de actividad alternativas en un intento por caracterizar mejor el comportamiento del usuario. Ejemplos de fuentes alternativas de información incluyen el comportamiento del usuario en un entorno gráfico [14], y el uso de aplicaciones como el empleo de un sistema de administración de documentos [8].

En este artículo, afirmamos que para caracterizar mejor el comportamiento del usuario, es necesario considerar cómo y qué recorre el usuario mientras trabaja en su sistema de archivos. Después de todo, el sistema de archivos de un usuario es una representación abstracta del usuario, al menos en cuanto al sistema de archivos se refiere. Crucial para nuestro enfoque es una estructura, que llamamos la estructura de navegación del usuario y que ya se presentó con resultados preeliminares en [16], representando la navegación de un usuario a través de su sistema de archivos. Una estructura de navegación contiene información del sistema de archivos del usuario. En particular, contiene los objetos más recientemente visitados en el sistema de archivos; también contiene información tanto acerca de cómo el usuario estructura su sistema de archivos, como de la manera en que usa y recorre tal directorio. Mostraremos que usar la estructura de navegación de un usuario obedece a un doble propósito. Primero, la estructura de navegación permite entender mejor el comportamiento de un usuario dado. Segundo, y más importante, permite construir un perfil de usuario adecuado para la detección de impostores. Aún cuando reconocemos que es necesaria la combinación de varios tipos de actividades, nuestro planteamiento pretende demostrar que la navegación del sistema de archivos es determinante para conseguir un perfilado preciso del usuario.

*Perspectiva general del artículo* El resto de este artículo está organizado de la siguiente forma. Primero, en 2 mostraremos las limitaciones, en general, de los métodos de detección de impostores. En particular, argumentaremos que, por una parte, se desempeñan bastante pobremente, aún cuando no son sujetos a acciones propias de intrusos; y que, por otra parte, son difíciles de implementar, porque la información de actividad elegida para perfilar a un usuario es, por naturaleza, poco densa. Segundo, en 3, se definen brevemente las estructuras de navegación del usuario en que basamos nuestra representación de la información del usuario, y hacemos referencia a [16], donde se definen con mayor amplitud, junto con las operaciones que toman. Tercero, en 4, bosquejaremos el conjunto de datos que hemos recolectado, el cual se divide en datos del usuario y datos de los ataques realizados. Describimos la manera en que se realizó un prefiltrado de las bitácoras con el fin de conservar únicamente la información relevante y deshacerse del ruido proveniente de los registros originales. Cuarto, en 5, proponemos una representación vectorial formada por atributos extraídos de los registros, y que consideramos pueden ser de utilidad para caracterizar al usuario. Estos vectores son obtenidos dividiendo los registros en ventanas de tamaño fijo y se proponen como una fuente de información adecuada para la creación del perfil del usuario y la detección de impostores. Quinto, en 6, se propone una serie de experimentos que pueden realizarse a partir del conjunto de datos y la repre-

sentación de los mismos que se ha propuesto. Las ideas de investigación futura tienen el propósito de sugerir algunas técnicas que permitan explotar la información recabada. Finalmente, en 7, reportaremos las conclusiones que hemos obtenido hasta el momento.

## 2. Mecanismos de detección de impostores: perspectiva general

La investigación en detección de impostores ha sido muy prolífica. Las limitaciones de espacio nos impiden dar un panorama general razonable, o una comparación justa de estos mecanismos (aún si nos limitamos a los más representativos). Deberemos confinarnos a dar una perspectiva general de unos cuantos, remitiendo al lector a [9] para un estudio detallado.

Un detector de impostores es *global* si, para determinar la presencia de un impostor, usa tanto el perfil del usuario que está siendo protegido, como el perfil de los colegas de dicho usuario. Sino, es *local*. Igualmente, un detector de impostores es *basado en temporalidad* si, para la formación del perfil del usuario, considera tanto acciones individuales como la relaciones entre ella; por ejemplo, secuencias de acciones. Sino, es *basado en frecuencias*. Dado que los métodos globales y temporales se encuentran mejor informados, usualmente son más precisos que los locales y los basados en frecuencia; sin embargo, requieren más información y esfuerzo computacional, y son inaplicables en algunos contextos.

*Gramáticas personalizadas* [4] es un método de detección global, basado en temporalidad, que perfila el comportamiento normal del usuario usando las secuencias de acciones del usuario únicas y las más repetitivas. Para identificar tales secuencias de acciones dentro del historial del usuario y respecto a los registros de auditoría de otros usuarios, las gramáticas personalizadas aplican *sequitur* [10], un método para inferir jerarquías composicionales a partir de cadenas.

*Naïve Bayes* [11,6,12] asume que las acciones de un usuario son independientes entre sí. Así, la probabilidad de que una acción $c$ haya sido originada por un usuario $u$, $Pr_u(c)$ está dada por $\frac{f_{u\,c}+\alpha}{n_u+\alpha\times K}$, donde $f_{u\,c}$ es el número de veces que el usuario $u$ empleó $c$ en el registro de acciones del usuario, $n_u$ es la longitud del registro de $u$, $K$ es el número total de acciones distintas, y donde $0 < \alpha << 1$ previene que el valor de $Pr_u(c)$ sea cero. Así, naïve Bayes es basado en frecuencias y puede ser tanto local como global, con desempeño similar [12]. Para evaluar una sesión de prueba $s$, en la que el usuario $u$ presuntamente participó, la probabilidad acumulativa de $s$ es comparada ya sea contra la probabilidad de que $s$ haya sido producido por alguien más [11] o contra un umbral definido por el usuario [12].

*Sequitur-luego-HMM* [5] es un detector de impostores local y temporal, que caracteriza el comportamiento del usuario en términos de las dependencias temporales entre secuencias de acciones más frecuentes. Este método aplica primero *sequitur* a un historial de acciones de usuario dado. Luego, la principal regla de producción del resultado de la gramática generada por *sequitur* es empleada para entrenar un modelo oculto de Markov (HMM, por sus siglas en inglés).

Consecuentemente, este modelo no solo considera dependencias temporales entre acciones, debido a *sequitur*, sino que también considera dependencias temporales entre secuencias, debido al HMM.

*Otras fuentes de información* Dado que el uso de información como la contenida en la base de datos SEA [2] para el perfilado y detección de impostores parece no ser suficiente como fuente de información, recientemente la investigación en esta área ha buscado fuentes alternativas de actividad del usuario para perfilar mejor su comportamiento. En [8] se ha experimentado con otro tipo de fuente como lo es la interacción del usuario con aplicaciones específicas (por ejemplo, el uso de un sistema de administración de documentos). En [14] se presenta trabajo con redes neuronales artificiales (ANNs) en el aprendizaje de características del usuario en un sistema gráfico. Las características aprendidas son el número de veces que una acción específica es observada, el número de veces que se presionan teclas de control, y el número de veces que se ejecutan ciertos procesos. Finalmente, en [15] los comandos del usuario se agrupan, y luego se enlazan en forma de tipos de actividades, tales como recopilación de información, búsqueda de información, comunicaciones, etc., con el objetivo de capturar la intención del usuario, probando que es fácil detectar a un impostor dado que es muy posible que éste exhiba un comportamiento de búsqueda propios de alguien que desconoce la computadora de la víctima. Aunque la idea clave detrás de este trabajo es interesante, actualmente el etiquetado de acciones requiere de intervención humana.

El desempeño de los métodos descritos (y otros) se encuentra, sin embargo, distante de ser concluyente. Esto ha sido probado en [13], donde reportan que la mayoría de estos métodos, cuando operan al 95 % de tasa de detección, producen más del 40 % de falsas alarmas. Los detectores de impostores no han sido completa o confiablemente probados. Esto es porque no han sido probados bajo condiciones críticas, donde se requiere disntinguir a un intruso que realmente intenta hacerse pasar por un usuario (ver también [13]. A partir de esto, se deduce que estos métodos tampoco han sido comparados exhaustivamente.

## 3. Estructuras de navegación

Para capturar cómo recorre un usuario su sistema de archivos, empleamos una estructura de navegación. Una estructura de navegación contiene información tanto acerca de cómo un usuario estructura su sistema de archivos, como de la manera en que la emplea y recorre. Los objetos en una estructura de navegación han sido visitados por el usuario, ya sea directamente, mediante un recorrido de directorios, o indirectamente, mediante una aplicación del usuario.

Por limitaciones de espacio, no es posible explicar de manera amplia la estructura de navegación empleada; sin embargo, ésta ya fue explicada y detallada en un trabajo previo [16], y sugerimos revisarlo para su correcto y completo entendimiento. A continuación, nos acotaremos a dar una idea general de las características de la estructura de navegación.

Una estructura de navegación está compuesta de dos grafos: un grafo de accesos y un grafo de directorios. Un *grafo de accesos* contiene un registro de los objetos del sistema de archivos que han sido visitados más recientemente, así como el orden (de hecho, un salto de ruta) que el usuario ha seguido para visitarlos. Un *grafo de directorios* es un subgrafo propio del sistema de archivos del usuario; es, de hecho, una arborescencia, con un vértice distinguido, llamado la *raíz*, y denotado por "/". A partir de esta arborescencia es posible definir el concepto de *profundidad*, que representa la cantidad de desplazamientos, partiendo de la *raíz*, que deben realizarse para alcanzar un objeto dado, tomando la profundida de la *raíz* como 0 y considerando que dicha profundidad se incrementa (o correspondientemente, decrementa) en 1 con cada desplazamiento hacia abajo (o correspondientemente, hacia arriba) de la arborescencia.

*Mantenimiento de la Estructura de Navegación* Para mantener la estructura de navegación de un usuario de un tamaño manejable, es posible y necesario aplicar un algoritmo de reemplazo que permita actualizar la estructura de navegación cuando sea necesario; por ejemplo, después de que el usuario ha movido o removido un objeto del sistema de archivos. Esto permitirá manejar adecuadamente el concepto de *objetos nuevo*, en términos de accesos o secuencias de accesos en un período de tiempo, pues después de un tiempo determinado sin utilizarse, ciertos objetos podrían ser removidos de la estructura como parte de su mantenimiento y volver a considerarse como "nuevos" si se acceden otra vez.

## 4. El conjunto de datos

Es necesario, para una correcta experimentación, contar con un conjunto de bitácoras confiable, tanto de usuarios como de ataques. El conjunto de datos se divide en datos de usuarios válidos y datos de ataques realizados, por lo que la tarea de obtención de los mismos también fue dividida en dos etapas.

### 4.1. Creación del conjunto de datos de usuarios

Para la obtención del conjunto de datos de usuarios se reclutaron voluntarios, usuarios frecuentes de (alguna versión de) el sistema Windows. Con su ayuda, y bajo previa firma de un convenio de colaboración y protección de datos, configuramos el sistema de auditoría, el cual es una herramienta ya existente en todas las versiones de Windows con las que se experimentó. Actualmente se cuenta con un total de 20 usuarios, aunque, a la fecha, se ha recabado información de ataques para 14 de ellos, faltando por simularse los ataques para los 6 usuarios restantes. El perfil de los 14 usuarios para los que se dispone de registros completos se muestra en la Tabla 1, habiéndose capturado en períodos que varían de poco más de 15 días hasta 3 meses.

Se elaboró una encuesta con el objetivo de contar con la mayor información posible de los usuarios y, así, estas características puedan ser empleadas para clasificarlos. De aquí, se observa que nuestra base cuenta con usuarios de perfiles heterogéneos (pero que pueden separarse en unas cuantas clases), en

**Tabla 1.** Perfil de los 14 usuarios para los que se cuenta con un registro completo. Se buscó que los perfiles fueran variados, obteniéndose registros de distintas versiones de Windows, de entre 17 y 67 días de captura.

| Usuario | Rol | Sistema Operativo | Fecha de inicio | Fecha final |
|---|---|---|---|---|
| 1-2 | Administrador General | Windows XP | 14/02/2012 | 24/03/2012 |
| 3 | Encargado de Sistemas | Windows XP | 14/02/2012 | 24/03/2012 |
| 4 | Contador | Windows XP | 14/02/2012 | 24/03/2012 |
| 5-9 | Secretarias de Empresas | Windows XP | 14/02/2012 | 24/03/2012 |
| 10 | Encargado de Entregas | Windows Vista | 14/02/2012 | 24/03/2012 |
| 11 | Secretaria de Universidad | Windows XP | 21/03/2012 | 13/04/2012 |
| 12 | Programadora de Sistemas | Windows 7 | 16/01/2012 | 23/03/2012 |
| 13 | Estudiante Universitaria | Windows Vista | 22/01/2012 | 23/03/2012 |
| 14 | Gerente de Ventas | Windows 7 | 16/12/2011 | 28/03/2012 |

términos de: conocimiento del sistema operativo, perfil profesional, frecuencia de uso del sistema, nivel de organización, etc. Nuestra muestra permite obtener conclusiones estadísticamente confiable respecto a los factores que influyen en una mejor detección de impostores, y, contando con la información adicional que se ha mencionado, esperamos poder obtener ciertos indicios de cómo el tipo de usuario influye en la detección de impostores (por ejemplo, una hipótesis que se pretende probar es que aquellos usuarios que son más organizados serán más fáciles de defender de una intrusión que aquellos que no lo son).

### 4.2. Creación del conjunto de datos de ataques

Para la creación y posterior simulación de ataques impostores se recurrió a una encuesta diseñada por el equipo y aplicada a casi 50 personas. En la encuesta, los participantes actuaron bajo un escenario hipotético en el que podrían hacer uso, no autorizado, de una computadora desatendida, por 5 minutos. Recopilamos el tipo de información que, como impostores, los encuestados buscarían, además del tipo de búsqueda a emplear, y la forma en que intentarían hacerse de la información. Así, diseñamos 3 tipos de ataques: básico, intermedio y avanzado (ver Tabla 2). El objetivo de estos ataques es comprometer información relevante del usuario, para lo cual, el atacante interactúa con la computadora navegando en el sistema de archivos. Cabe resaltar que cada ataque se implementó mediante un conjunto de pasos, de modo que su simulación es uniforme y controlada. Tras la simulación, recopilamos bitácoras, que pueden ser usadas para evaluar la efectividad de los mecanismos de detección de impostores. Tomando en cuenta que se realizaron tres ataques diferentes por cada usuario válido, se obtuvo un total de 42 ataques.

### 4.3. Prefiltrado de bitácoras

Dada la diversidad de los datos, por las distintas versiones de sistema operativo, los diferentes tipos de usuarios, los distintos tiempos de registro, las variaciones en la cantidad de registros, etc., y porque el sistema de auditoría genera información en demasía, los prefiltramos, obteniendo información consistente,

**Tabla 2.** Características principales de los ataques realizados a los 14 usuarios. Todos los ataques son restringidos a una duración máxima de 5 minutos.

| Ataque | Características |
|---|---|
| 1. Básico | Apertura manual de carpetas y robo de información copiando manualmente o tomando fotografías. |
| 2. Intermedio | Búsqueda de archivos especiales con el buscador de Windows y copiado de los archivos a una memoria USB. |
| 3. Avanzado | Ejecución de un script .bat en una memoria USB que se encarga de copiar automáticamente archivos a la memoria. |

sin importar su fuente, para quedarse con un listado de las acciones ocurridas en la computadora del usuario, únicamente en las carpetas seleccionadas para su monitoreo y que contuviera información exclusivamente del objeto al que se accedió, así como el momento en que fue accedido. Contar con este prefiltrado permite una creación del modelo más eficiente y reduce considerablemente el ruido proveniente de los datos recabados. Al final de este proceso, la información que es de nuestro interés es la siguiente:

$$ID\_acceso|Fecha|Hora|Tiempo\_transcurrido|Profundidad|Ruta\_acceso$$

donde $ID\_acceso$ representa el identificador del número de acceso realizado por el usuario, permitiendo llevar un orden consecutivo de los accesos. $Fecha$ contiene la fecha del acceso. $Hora$ contiene la hora en formato de 24 horas. $Tiempo\_transcurrido$ es un campo especial que contiene el tiempo que ha transcurrido (medido en segundos) desde una fecha predeterminada (en nuestro caso, desde el 01/01/2011) hasta el momento del acceso, y que nos permitirá hacer cálculos como el tiempo entre accesos de una manera sencilla. $Profundidad$ contiene la profundidad de la ruta. Y, finalmente, $Ruta\_acceso$ contiene la ruta completa del objeto en el sistema de archivos.

## 5. Creación del modelo de representación de los datos

Después de realizar el filtrado es necesaria la división de la bitácora en ventanas y la obtención de atributos (para cada una de esas ventanas) que servirán después para la detección de intrusos. Para ello, se empleará un modelo de representación basado en grafos de navegación que refleja la interacción del usuario en el sistema de archivos. Posteriormente, un conjunto de atributos es extraído a partir de la información contenida en las ventanas, para así formar vectores de atributos (estos vectores deben ser calculados tanto para los datos del usuario como para los del atacante). El modelo de representación ya fue propuesto en un trabajo previo con resultados alentadores [16], donde se empleó naïve Bayes junto con ventanas definidas en términos del número de acciones (a diferencia del presente trabajo, donde se propone todo un vector de atributos y la definición de ventanas en términos de tiempo, como se verá más adelante). Los vectores de atributos propuestos tienen la siguiente estructura:

La información contenida en estos vectores es: i) Accesos. Se obtiene el número total de accesos en la ventana ($T$) así como el número de accesos nuevos

**Tabla 3.** Atributos extraídos de los grafos de navegación. Este es el vector que describe los atributos obtenidos del grafo de navegación, que a su vez es dividido en ventanas, con los que se pretende realizar la identificación de impostores.

| Accesos | | Profundidad | | | | | | Tiempo entre accesos | | | | | | Características del grafo | | | | | Distancia en recorrido entre accesos | | | | | | Probabilidad | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T$ | $N$ | $\bar{x}$ | $\tilde{x}$ | $M_o$ | $S^2$ | $S$ | $R_e$ | $\bar{x}$ | $\tilde{x}$ | $M_o$ | $S^2$ | $S$ | $R_e$ | $\|V\|$ | $\|E_d\|$ | $\|E_u\|$ | $\frac{\|V\|}{\|E_d\|}$ | $\frac{\|V\|}{\|E_u\|}$ | $\bar{x}$ | $\tilde{x}$ | $M_o$ | $S^2$ | $S$ | $R_e$ | $NB_v$ | $NB_e$ |

($N$); ii) Profundidad. Con la profundidad a que se encuentra cada objeto accedido en la ventana, se obtienen media ($\bar{x}$), mediana($\tilde{x}$), moda ($M_o$), varianza ($S^2$), desviación estándar ($S$) y rango muestral ($R_e$) dentro de la ventana; iii) Tiempo entre accesos. Utilizando el tiempo del campo $Tiempo\_transcurrido$ se puede calcular el tiempo que transcurre entre un acceso y otro dentro de la ventana, y con estos tiempos también es posible calcular media, mediana, moda, varianza, desviación estándar y rango muestral dentro de la ventana; iv) Características del grafo. En cuanto a las características del grafo, se busca obtener el número total de vértices ($|V|$), el número total de aristas dirigidas ($|E_d|$), el número total de aristas no dirigidas ($|E_u|$), la proporción de vértices entre aristas dirigidas ($\frac{|V|}{|E_d|}$) y la proporción de vértices entre aristas no dirigidas ($\frac{|V|}{|E_u|}$); v) Distancia en recorrido entre accesos. También se calcularán distancias entre los accesos, esto basándose en la distancia que hay de un nodo a otro dentro del grafo, y se calcularán las mismas medidas estadísticas, que son: media, mediana, moda, varianza, desviación estándar y rango muestral. vi) Probabilidad. Se calcularán dos formas distintas de naïve Bayes, una basándose en la probabilidad del acceso a los vértices ($NB_v$) y otra basándose en la probabilidad de acceso a las aristas ($NB_e$).

Por otro lado, como se mencionó, se pretende perfilar al usuario y detectar a un impostor a partir de la división de la información en ventanas definidas en términos de un tiempo fijo. Así, los registros se analizarán por ventanas divididas por un tiempo $t$, donde $t$ es un tiempo (en segundos) constante durante cada análisis. Uno de los objetivos de estos experimentos es realizar pruebas usando distintos tamaños de ventana (por ejemplo, 30, 60 y 90 segundos) para encontrar cuál es más apropiada para la detección de intrusos, pues, se espera que las variaciones en la cantidad de objetos presentes en una ventana sirvan como indicios de la actividad de un impostor en el sistema.

## 6. Experimentos propuestos

A continuación describimos una serie de experimentos que se realizarán con el conjunto de datos de usuarios y atacantes con que se cuenta. Cabe mencionar que, aunque estos experimentos no se han realizado, el proceso para la realización de los mismos se encuentra ya avanzado, por lo que se espera contar con resultados preliminares en un corto plazo.

Recordando que se cuenta con una bitácora ya filtrada por cada usuario, en la etapa experimental, para generar el perfil del usuario se irán analizando las ventanas generando los vectores mencionados en la Sección 5. El proceso

de creación del perfil se realizará individualmente por cada usuario, por lo que, para $k$ usuarios se tendrán $k$ perfiles completamente independientes, que se espera capturen los patrones de comportamiento específicos a cada uno de ellos. Los datos obtenidos se emplearán para actualizar los grafos de directorios y de accesos para el usuario; de esta forma, es posible generar el perfil del usuario reflejado en un árbol, así como obtener los vectores que representen su comportamiento.

Es necesario tomar en cuenta el hecho de que es necesario contar con un período de ajuste del sistema (aquél donde los datos obtenidos deban ser considerados como poco fiables), pues, por ejemplo, ya que al inicio de los registros habrá un gran número de accesos nuevos a objetos (en realidad, desde cierto punto de vista, todos los accesos serán nuevos, puesto que no se tendrá registro de que se hayan realizado previamente). Esto lleva a la necesidad de desechar los primeros registros (calculamos que el porcentaje de datos iniciales desechados no sobrepase el 5 %), para evitar capturar ruido al perfilar al usuario. Esta información, sin embargo, será desechada parcialmente, pues, aunque no se considerarán algunos aspectos de ella para perfilar al usuario, sí es necesario actualizar los árboles que se generan con información de estos accesos. En la Figura 1 se ve una posible forma en que podrían distribuirse los datos de las bitácoras de los usuarios.


**Figura 1.** Distribución sugerida para los datos. El 5 % inicial se descartaría para la creación del perfil del usuario, aunque sí se emplearía para actualizar los grafos de directorios y accesos; el 75 % siguiente se emplearía para crear el perfil del usuario; y, finalmente, el último 20 % se emplearía para validar el modelo obtenido.

Aunque sugerimos descartar el 5 % inicial, emplear el 75 % siguiente para entrenamiento y usar el último 20 % para la validación del modelo, este esquema no se propone como algo rígido, puesto que algunos aspectos que se desean evaluar tienen relación directa con estos porcentajes y, por ende, requieren experimentar variándolos. Algunas de las preguntas que surgen a este respecto son:

- ¿Qué porcentaje de datos es fiable descartar considerándolo como ruido?
- ¿Qué porcentaje de datos es necesario utilizar para el perfilado del usuario?
- ¿Qué porcentaje de datos basta para validar correctamente el modelo?
- ¿Estos porcentajes son fijos o varían, dependiendo de las características del usuario como qué tan organizado es, cuánta información se recabó, durante qué tiempo se recabó, etc.?

Una vez contando con los vectores que representan los grafos de navegación, se propone sacar provecho de ellos mediante la aplicación de métodos inteligentes

para aprender las características del usuario. Dado que, para los ataques realizados se cuenta también con una bitácora, ésta será también preprocesada y será posible obtener una representación vectorial de estos ataques. Con esta información será posible evaluar si una nueva ventana corresponde a la actividad del usuario válido o, por el contrario, se trata de un potencial intruso. Para tal evaluación se sugieren como opciones el uso de ANNs, SVMs, K vecinos más cercanos (KNN) y regresión lineal localmente ponderada (LWLR). Un esquema general de la clasificación (incluyendo el preprocesado y la obtención de los vectores), se muestra en la Figura 2.

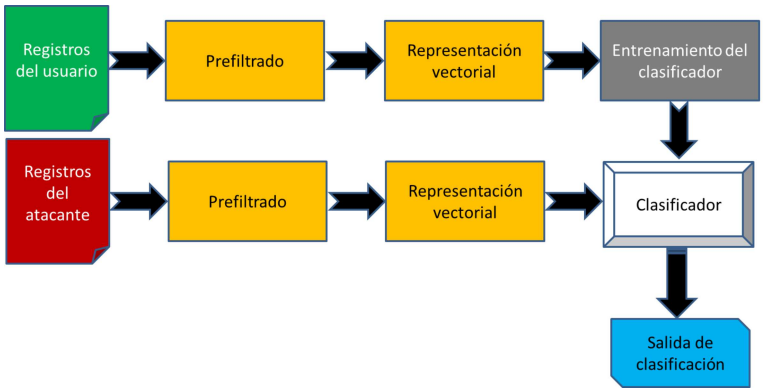


**Figura 2.** Diagrama del esquema de detección de impostores propuesto, basado grafos de navegación.

El uso de varios métodos para clasificación haría posible evaluar el desempeño de los distintos algoritmos en términos de falsos positivos (el caso cuando una ventana del usuario sea clasificada erróneamente como del atacante) y falsos negativos (el caso opuesto, cuando una ventana del atacante sea clasificada erróneamente como del usuario), principalmente, además de que permitiría sacar conclusiones acerca de qué método se adapta mejor al tipo de datos con que se trabaja. Se propone que estos clasificadores sean entrenados mediante el enfoque de una clase, es decir, proporcionándole al clasificador la información del usuario para el entrenamiento, y esperando que a partir de ella pueda detectar si una nueva entrada corresponde al mismo usuario o no, en cuyo caso se determinaría que se trata de un impostor.

## 7. Conclusiones y trabajo futuro

La estructura de navegación que se emplea en este trabajo para caracterizar al usuario respecto a su interacción con el sistema de archivos ya la hemos propuesto anteriormente [16]. En dicho trabajo se presentaron resultados preliminares que daban una idea del potencial de tal estructura.

En este trabajo, hemos ido más allá al corregir las limitaciones antes propuestas al extender los datos del usuario a 14 personas con registros de su interacción con el sistema de varios días. Los participantes del experimento son heterogéneos en términos de su perfil de usuario, lo que permitirá sacar mejores conclusiones de los experimentos que se realicen.

La metodología seguida para los ataques pretende ayudar a homogeneizar los mismos, de manera que las pruebas con los 3 ataques realizados por usuario (42 en total), sean de utilidad para evaluar diversas técnicas de detección de impostores. Se ha propuesto una representación de los grafos mediante su división en ventanas de tiempo y su posterior representación en vectores de atributos, que sean empleados para entrenar métodos de clasificación de una clase.

Con la información que se cuenta, es posible comenzar a probar los métodos de detección propuestos en este documento. Se espera que con los atributos que se extraen y haciendo uso de los grafos de accesos y de directorios se obtengan mejores resultados que en trabajos previos, principalmente en términos de porcentajes bajos de falsos positivos y falsos negativos.

A mediano plazo se planea agregar más campos a la estructura de navegación así como nuevos atributos al vector que, esperamos, puedan caracterizar mejor a un usuario. También se tiene contemplado utilizar otros tipos de clasificadores de una sola clase para encontrar cual es el más conveniente para la detección de intrusos. Finalmente, se espera contar con nuevos usuarios con perfiles distintos para poder generar conclusiones basadas en el tipo de usuario y/o sistema operativo utilizado.

**Agradecimientos.** Este proyecto fue apoyado parcialmente por CONACyT mediante una beca de doctorado y una beca de estancia posdoctoral. Agradecemos por su contribución en la extracción de los atributos de las ventanas a vectores y la obtención de los datos para la normalización de los vectores al estudiante Eduardo Murillo González.

## Referencias

[1] Schonlau, M., DuMouchel, W., Ju, W., Karr, A., Theus, M., Vardi. Y.: Computer intrusion: Detecting Masquerades. Statistical Science 16,58–74, (2001)
[2] Schonlau, M.: Masquerading user data. http://www.schonlau.net (2008)
[3] Maxion, R.A., Townsend, T.N.: Masquerade detection augmented with error analysis. IEEE Transactions on Reliability 53, 124–147 (2004)
[4] Latendresse, M.: Masquerade detection via customized grammars. In Julish, K., Kruegel, C., eds.: Proceedings of the Second International Conference on Detection of Intrusions and Malware, and Vulnerability Assesment, DIMVA 2005. Volume 3548 of Lecture Notes in Computer Science., Springer, 141–159, (2005)
[5] Posadas, R., Mex-Perera, C., Monroy, R., Nolazco-Flores, J.: Hybrid method for detecting masqueraders using session folding and hidden markov models. In: Proceedings of the 5th Mexican International Conference on Artificial Inteligence: Advances in Artificial Inteligence. Volume 4293 of Lecture Notes in Computer Science., Springer, 622–631 (2006)

[6] Maxion, R.A.: Masquerade detection using enriched command lines. In: Proceedings of the International Conference on Dependable Systems and Networks, DSN'03, San Francisco, CA, USA, IEEE Computer Society Press, 5–14 (2003)
[8] Sankaranarayanan, V., Pramanik, S., Upadhyaya, S.: Detecting masquerading users in a document management system. In: Proceedings of the IEEE International Conference on Communications, ICC'06. Volume 5., IEEE Computer Society Press, 2296–2301, (2006)
[9] Salem, M.B., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. In Stolfo, S.J., Bellovin, S.M., Hershkop, S., Keromytis, A., Sinclair, S., Smith, S.W., eds.: Insider Attack and Cyber Security: Beyond the Hacker. Advances in Information Security. Springer, 69–90, (2008)
[10] Nevill-Manning, C.G., Witten I.H.: Identifying hierarchical structure in sequences: a linear-time algorithm. Journal of Artificial Intelligence Research, 67–82, (1997)
[11] Maxion, R.A., Townsend, T.N.: Masquerade detection using truncated command lines. In: Proceedings of the International Conference on Dependable Systems & Networks, Washington, DC, IEEE Computer Society Press, 219–228, (2002)
[12] Wang, K., Stolfo, S.: One-class training for masquerade detection. In: Proceedings of the 3rd IEEE Workshop on Data Mining for Computer Security, IEEE (2003)
[13] Razo-Zapata, I., Mex-Perera, C., Monroy, R.: Masquerade attacks based on user's profile. Submited to Journal of Systems and Software (2011)
[14] Imsand, E.S., Garrett, D., Hamilton, J.A.: User identification using gui manipulation patterns and artificial neural networks. In: Computational Intelligence in Cyber Security, 130–135 (2009)
[15] Ben-Salem, M., Stolfo, S.: Modeling user search behavior for masquerade detection. In: Research Advances in Intrusion Detection, RAID'2011, Lecture Notes in Computer Science, in press. Springer (2011)
[16] Camiña, B., Monroy, R., Trejo, L.A., Sánchez, E.: Towards building a masquerade detection method based on user file system navigation. In: Proceedings of the Mexican International Conference on Artificial Intelligence, 174–186, (2011)