# Un nuevo esquema para la detección de ataques en redes inalámbricas

Jorge Vázquez, Raúl Monroy y Luis A. Trejo

Departamento de Ciencias Computacionales,
Instituto Tecnológico y de Estudios Superiores de Monterrey,
Campus Estado de México, México
{A01165550,raulm,ltrejo}@itesm.mx

**Resumen** En este trabajo presentamos una metodología para la detección de anomalías en redes inalámbricas IEEE 802.11. Señalamos las dos principales vulnerabilidades del protocolo, la de identidad y la de acceso al medio. Nuestra metodología se basa en la caracterización de secuencias de tramas de capa 2, a estas secuencias las llamamos conversaciones. Para definir un conversación realizamos una abstracción de una trama a un símbolo, mediante el uso de técnicas de mapas auto organizables, k-medias y el algoritmo C4.5. Con estos elementos construimos un compresor. El cual genera secuencias más cortas que esperamos representen eficientemente comportamiento normal. Nuestro análisis refleja un porcentaje de reducción cercano al 92 %. Finalmente, proponemos una arquitectura en la que se usarán modelos ocultos de *Markov* con el fin de obtener un modelo de comportamiento normal y que pueda determinar la similitud o diferencia entre dos conjuntos de datos.

## 1. Introducción

El uso creciente de las redes inalámbricas basadas en el protocolo IEEE 802.11 ha incrementado el problema de accesos no autorizados, provocando un uso inadecuado de los recursos que provee un sistema de red. Este tipo de intrusiones se definen como cualquier actividad maliciosa dirigida a los servicios que la red provee, y se ve como una secuencia de acciones que alteran su estado y comprometen su seguridad. Para proteger los sistemas de red de estas actividades maliciosas se han desarrollado los sistemas de detección de intrusiones o IDS, por sus siglas en inglés *Intrusion Detection Systems*. Los IDS están encargados de monitorear eventos para el descubriemiento oportuno de actividades que ponen en riesgo la integridad, disponibilidad y confidencialidad de una red inalámbrica. En este trabajo nos interesa desarrollar un detector de intrusiones que modele la interacción entre una computadora y la red, identificando desviaciones en el comportamiento normal de dicha interacción. Es de especial interés la detección de ataques de denegación de servicio o DoS, por sus siglas en inglés *Denial of Service*. Estos ataques consisten, por ejemplo, en saturar un sistema con peticiones falsas de tal manera que los servicios o recursos del sistema atacado no estén disponibles para los usuarios genuinos que lo soliciten.

## 2. El protocolo IEEE 802.11

El protocolo IEEE 802.11 se centra en la capa de control de acceso al medio o MAC, por sus siglas en inglés *Media Access Control*, del modelo OSI, específicamente en el control de enlace lógico. La manera en que una estación o STA, acrónimo del inglés *Station*, obtiene acceso a una red mediante un punto de acceso o AP, por sus siglas en inglés *Access Point*, puede verse como una máquina de estados finitos o FSM, por sus siglas en inglés *Finite State Machine*. Las tramas que son intercambiadas son de tres tipos: de control, de mantenimiento y de datos. Cada tipo de trama, a su vez, se divide en sub-tipos de tramas. Una Trama es un mensaje que se intercambia entre una STA y un AP. Cada trama contiene información que se usa para activar un proceso del protocolo. La información contenida en cada trama, depende del proceso que se ejecute. Para fines de este trabajo de investigación, nos enfocaremos en los 17 sub-tipos de tramas del protocolo IEEE 802.11. Los sub-tipos de tramas que analizaremos se muestran en el tabla 1.

**Tabla 1.** Sub-tipos de tramas del IEEE 802.11

| | |
|---|---|
| 1 | *Beacon* |
| 2 | *Probe Request* |
| 3 | *Probe Response* |
| 4 | *Authentication* |
| 5 | *Deauthentication* |
| 6 | *Association Request* |
| 7 | *Association Response* |
| 8 | *Disassociation* |
| 9 | *Reassociation Request* |
| 10 | *Reassociation Response* |
| 11 | *PS Request* |
| 12 | *PS Response* |
| 13 | *PS POLL* |
| 14 | *RTS* |
| 15 | *CTS* |
| 16 | *DATA* |
| 17 | *ACK* |

La figura 1 representa la interacción entre un AP y cualquier STA en una infraestructura de red IEEE 802.11. El estado 1 en la figura 1 representa a las estaciones que no han obtenido privilegios (no se han autenticado ni se han asociado). En cada estado de la FSM sólo ciertos sub-tipos de tramas pueden ser intercambiadas. En el contexto de una red inalámbrica, sin una infraestructura física, el atacante tiene gran flexibilidad en decidir dónde y cuándo atacar.

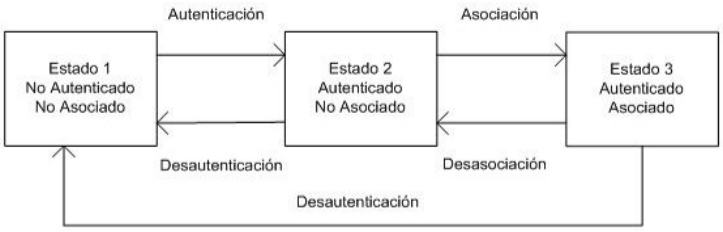


**Figura 1.** Máquina de Estados del IEEE 802.11

## 3. Problemática de seguridad del IEEE 802.11

La capa MAC del protocolo IEEE 802.11 incorpora características como la habilidad para descubrir redes, unirse y separarse de ellas, así como la coordinación para accesar al medio. Las vulnerabilidades que se presentan en el protocolo provienen de esta funcionalidad y se pueden dividir en dos categorías: de identidad y de vulnerabilidades en el control de acceso al medio.

### 3.1. Vulnerabilidades de identidad

Las vulnerabilidades de identidad surgen de la confianza implícita en la dirección MAC del emisor. Entonces un atacante puede engañar a otras estaciones y solicitar servicios de capa MAC, haciéndose pasar por un usuario válido. Por ejemplo, en un ataque de desautenticación, un adversario puede pretender ser un AP o un cliente y enviar tramas de desautenticación, y solicitar explícitamente la desautenticación uno del otro, en respuesta, el AP o el cliente saldrán del estado autenticado y rechazarán las tramas recibidas hasta que el estado de autenticación sea reestablecido. El proceso de ahorro de energía del IEEE 802.11 también presenta vulnerabilidades de identidad. Para ahorrar energía las estaciones pueden entrar en un modo de ahorro de energía o un modo dormido. Antes de entrar en este modo, la STA anuncia su intención al AP, para que el AP guarde en *buffer* las tramas de datos dirigidas a la STA y, cuando la STA despierte, recupere esas tramas. La presencia de paquetes en *buffer* es anunciada mediante paquetes difundidos periódicamente por el AP. Estos paquetes viajan en tramas de control que se denominan mapa de tráfico de información o TIM, por sus siglas en inglés *Traffic Information Map*. Si el mensaje TIM es falsificado por el atacante, entonces puede convencer a un cliente legítimo de que no hay tramas de datos pendientes para él y el cliente regresará al modo de ahorro de energía. Finalmente, el mecanismo de ahorro de energía recae en la sincronización entre el AP y la STA, esto es, que la STA sabe cuándo despertar. Mediante la falsificación de estas tramas, un atacante puede causar que un cliente salga de sincronía con el AP y fallar en la recuperación de las tramas pendientes.

### 3.2. Vulnerabilidades de acceso al medio

Las vulnerabilidades de acceso al medio surgen del mecanismo de detección de portadora virtual usado para mitigar colisiones de estaciones móviles ocultas. Cada trama de control de sub-tipo petición de envío y de sub-tipo listo para enviar o RTS CTS, respectivamente, por sus siglas en inglés *Request to Send*, *Clear to Send*, del IEEE 802.11 tienen un campo denominado duración que indica el tiempo en microsegundos que el canal es reservado. Las tramas RTS y CTS se usan para sincronizar el acceso al canal cuando una terminal oculta interfiere con las transmisiones [1]. El valor del campo duración es usado para programar el vector de asignación de red o NAV, por sus siglas en inglés *Network Alocation Vector* en cada STA. Sólo cuando el NAV de una STA alcanza el valor de cero, se le es permitido transmitir. El valor máximo de un NAV es de 32767 o cercano a los 32 milisegundos. Un atacante sólo requiere transmitir alrededor de 30 tramas por segundo para saturar el canal y evitar el acceso a la red.

## 4. Trabajo relacionado

En esta sección describiremos las heurísticas que se han propuesto para la construcción de sistemas de detección de intrusiones en redes inalámbricas.

Fanglu Go y Tzi-Cker Chiueh, en 2005 [5], propusieron un algoritmo para detectar robo de identidad mediante desviaciones en los números de secuencia de las tramas. Lo que proponen es medir la diferencia entre números de secuencia consecutivos entre la trama *i* y la trama *i-1*. La mayoría de las desviaciones fueron: 0, 1, ó 2. El 88 % de las desviaciones que ellos midieron son 1, el 3.3 % son 0, el 5.3 % son 2 y el 2.6 % son mayores a 2 debido a retransmisiones o tramas perdidas. En las pruebas realizadas, ninguna de las siete interfaces probadas transmitieron tramas fuera de orden, por lo tanto, no generaron falsos positivos. La tasa de falsos negativos fue del 0.03 %. A. Martínez en 2008 [9], propuso un método para detectar tramas de administración falsificadas en el protocolo IEEE 802.11. El método se basa en monitorear el tiempo entre llegadas de las tramas de sub-tipo *Beacon*. Si el tiempo entre llegadas estaba por debajo de un umbral, las tramas se consideraron maliciosas (falsificadas). Con este método lograron una tasa del 5 % de falsos positivos y 0 % de falsos negativos, en la detección de tramas de sub-tipo *Beacon* falsas. Yong Sheng, en 2008 [11], propuso un modelo Gaussiano mixto para describir el comportamiento de la fuerza de la señal recibida o RSS, por sus siglas en inglés *Received Signal Strength* en los dispositivos. El método se basa en que la señal recibida es difícil de falsificar. El RSS representa la potencia de transmisión menos la atenuación de la señal. La idea básica es que un dispositivo no cambia la potencia de transmisión, entonces un cambio drástico del RSS de las tramas recibidas con la misma dirección MAC fuente, sugiere un posible ataque de robo de identidad. Usan la construcción de perfiles para detectar ataques de tipo de robo de identidad. A una tasa del 3 % de falsos positivos, detectaron el 73.4 %, 89.6 % y hasta un 97.8 % de los ataques. Desventajas: los valores medidos del RSS dependen de la distancia entre emisor y receptor, por lo que, el atacante y la víctima deben estar separados una distancia mayor a 3

metros. Además asumieron que la estación válida se mantenía estática. En consecuencia, el método sólo es efectivo si los dispositivos no presentan movilidad. Qing Li en 2007 [8], propuso usar relaciones monotónicas en los números de secuencia de las tramas inalámbrica y complementan el método propuesto con un análisis de la distribución del tiempo entre llegadas de las tramas que recibe un AP. Usan la prueba de ji-cuadrada para comprobar la distribución de probabilidad de tiempos entre llegadas de las tramas entrantes a un AP bajo la suposición de que el tiempo entre llegadas de una fuente disminuye considerablemente si dos fuentes transmiten con la misma identidad. Los trabajos realizados por [9] y [11] presentaron una relación entre la cantidad de falsos positivos y la ubicación geográfica. Al cambiar la distancia física entre dispositivos, la detección se vuelve complicada. Creemos que un análisis en la coherencia de los mensajes intercambiados es una visión más general y, permitirá describir las interacciones en redes dinámicas.

## 5. Metodología

Como solución al problema de detección de ataques en redes inalámbricas, proponemos una metodología que se presenta a continuación.
La metodología está conformada por bloques de construcción, los cuales requieren la aplicación de métodos para el análisis de secuencias y clasificación de datos. La metodología de detección se sustenta en el análisis de bitácoras de tráfico del protocolo IEEE 802.11. Las bitácoras son producidas por la ejecución del protocolo al comunicarse una STA con un AP. Al cunjunto de tramas intercambiadas entre una STA y un AP lo denominaremos conversación. Una conversación es una sucesión de tramas, esta sucesión refleja los estados que transita la interacción STA-AP en un periodo de tiempo determinado.

### 5.1. Separador de sesiones

Las conversaciones se tomarán aleatoriamente mediante un selector en diferentes horas del día y en distintas fechas, dentro de periodos de tiempo de una hora. Una vez separadas las conversaciones realizaremos una traducción para representar una trama de capa 2 como un símbolo, de esta manera, una conversación la representaremos como una sucesión de símbolos, donde cada símbolo representa una trama de capa 2. Primero describiremos la metodología propuesta para representar tramas de tráfico de capa 2 por medio de símbolos. La estrucutura del separador se presenta en la figura 2. En los diagramas nosotros representamos a los procesos como círculos y a un conjunto de datos como rectángulos. El separador de conversaciones recibirá un conjunto de tramas contenidas en una ventana $W_i$ de tamaño $T$. Por cada conversación se generará un clasificador que: cada vez que reciba un número t de tramas, correspondiente a las intercambiadas entre una STA y un AP producirá una secuencia de símbolos que será analizada para obtener una caracterización de conversaciones en términos de símbolos que

pertenecen al alfabeto $\sum$.

$$\sum = \{s_1, s_2, ..., s_m\}$$

donde $m$ es la cardinalidad del alfabeto.

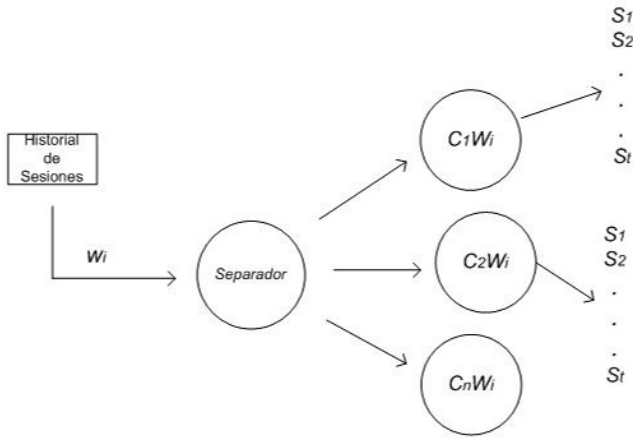


**Figura 2.** Arquitecturatura del Separador

### 5.2. Aglomerador de Sesiones

El siguiente bloque de construcción se presenta en la figura 3.

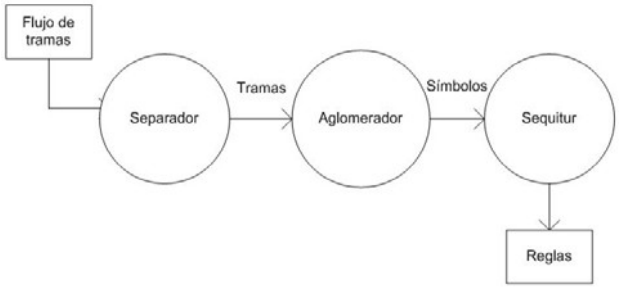


**Figura 3.** Obtención de Reglas

Los pasos que seguimos se describen a continuación. Para representar una trama de capa 2 como un símbolo, proponemos hacer una abstracción tomando en cuenta elementos mínimos, que describan correctamente una trama. Una trama típica de capa 2, tiene los siguientes campos de información:

1. Información de la trama inalámbrica.
2. Información del IEEE 802.11 (con cerca de 20 sub-campos, como la dirección MAC fuente, el identificador de red, etc.)
3. Datos cifrados (con más sub-campos.)
4. Datos en crudo.

Consideramos que no todos los campos son necesarios para caracterizar tráfico, es por ello que nos enfocaremos sólo en los campos que intervienen directamente en el cambio de estado de la FSM [7]. Este enfoque es importante porque el objetivo de los ataques DoS es precisamente, producir un cambio de estado a nombre de uno de los participantes [3]. Para ello, el atacante se basa en la construcción de tramas de capa 2, [6] con los valores de los campos modificados. Por otra parte, cuando un atacante genera tramas falsas a nombre de un participante, y logra un cambio de estado, la sucesión de tramas presentará incoherencias. A continuación enumeramos los campos que generan cambios en la FSM, ellos son:

1. *Subtype*: Sub-tipo de trama.
2. *Duration*: Duración de apartado de canal.
3. *SN*: Número de secuencia de trama.
4. *Fragmented*: Bandera de trama fragmentada.
5. *More Data*: Bandera de datos pendientes.
6. *Retry*: Bandera de reintento de envío de trama.
7. *ToDs*: Bandera que indica que la trama va dirigida hacia la red.
8. *Power Management*: Bandera que indica modo ahorro de energía.

Al analizar las tramas emitidas por el AP y una STA, toma importancia el campo: SN, ya que en condiciones normales se comporta de manera lineal en incrementos de 1 en un rango de valores de 0 a 4096, una vez que el valor llega a 4096 se reinicia desde 0. Las desviaciones en el patrón son mínimas. Para establecer el rango de variación calcularemos la diferencia entre el número de secuencia de la trama $i$ y el número de secuencia de la trama *i-1*.

$$\Delta_n = SN_i - SN_i{-_1}$$

Una vez obtenida la matriz con los campos y las diferencias en los números de secuencia entre tramas consecutivas, aplicamos mapas auto organizables o SOM, por sus siglas inglés *Self Organized Maps* para identificar patrones y agrupamientos de los campos contenidos en las tramas. De la matriz de resultados de SOM tomamos un número de grupos $K$, este número será el número de grupos que alimentamos al algoritmo K-medias con el objetivo de comprobar que los resultados arrojados por SOM son útiles para generar grupos de campos que representen una trama. Con ello agrupamos los valores de los campos para hacer un mejor razonamiento del intercambio de tramas, en lugar de usar las posibles combinaciones de campos. Una vez obtenidos los grupos de cada sub-tipo de trama inferimos reglas por medio del algoritmo C4.5 con el objetivo clasificar una trama mediante los valores de los campos contenidos en los campos

2 al 8 arriba definidos. De esta manera cada regla definió un símbolo, es decir, por cada trama clasificada correctamente de acuerdo a los valores de sus parámetros se generará un símbolo, además de un símbolo por cada grupo de tramas que el clasificador no haya clasificado correctamente. Con este conjunto de reglas construimos el alfabeto que represente cada sub-tipo de trama en condiciones de tráfico normal. Cada sub-tipo de trama estará definido por un conjunto símbolos. Con esta metodología obtuvimos la cardinalidad del alfabeto y una representación de cada trama sensible a variaciones en los valores de sus campos.

**Tabla 2.** Traducción de Tramas a Símbolos

| Tramas de capa 2 | | | | | | Símbolo |
|---|---|---|---|---|---|---|
| Tiempo | MAC Fuente | MAC Destino | Campos | | | |
| 0.365261 | 00:17:df:7d:a1:e0 | ff:ff:ff:ff:ff:ff | 0x08 0 | 2563 | 0 0 0 0 0 | 8001 |
| 2.314951 | 00:17:df:7d:a1:e0 | ff:ff:ff:ff:ff:ff | 0x08 0 | 2582 | 19 0 0 0 0 | 8001 |
| 3.082924 | 00:17:df:7d:a1:e0 | ff:ff:ff:ff:ff:ff | 0x08 0 | 2589 | 7 0 0 0 0 | 8001 |
| 4.277253 | 00:17:df:7d:a1:e0 | ff:ff:ff:ff:ff:ff | 0x08 0 | 2612 | 23 0 0 0 0 | 8004 |
| 47.054366 | 00:21:6b:27:d7:4c | 00:21:7c:98:6c:09 | 0x20 44 | 1297 | 0 0 0 1 0 | 20001 |
| 47.060497 | 00:21:7c:98:6c:09 | 00:21:6b:27:d7:4c | 0x20 44 | 3819 | 0 0 0 0 0 | 20002 |
| 47.067129 | 00:21:6b:27:d7:4c | 00:21:7c:98:6c:09 | 0x20 44 | 1298 | 0 0 0 1 0 | 20001 |
| 47.073764 | 00:21:7c:98:6c:09 | 00:21:6b:27:d7:4c | 0x20 44 | 3820 | 0 0 0 0 0 | 20002 |

En la tabla 2 mostramos un ejemplo de la traducción de tramas a símbolos. En este ejemplo se mapean cuatro tramas de sub-tipo *Beacon* ($0x08$) y cuatro tramas de sub-tipo *Data* ($0x20$) con sus parámetros, a símbolos $S_i$. De acuerdo a las reglas generadas por C4.5, podemos mapear una trama de distintos sub-tipos a símbolos en función de los parámetros de los campos que modifican la FSM. De esta manera obtuvimos 5 símbolos para representar una trama de sub-tipo *Beacon*. Las reglas generadas por C4.5 son:

1. Si el campo $Duration > 6060 -> S_1$
2. Si los campos $(Duration <= 6060) \& (\Delta_n > 1027) -> S_2$
3. Si los campos $(Duration <= 6060) \& (\Delta_n <= 1027) -> S_3$
4. Si los campos $(Duration <= 6060) \& (22 >= \Delta_n <= 1027) -> S_4$

Por otra parte, las tramas de sub-tipo *Beacon* que contengan valores fuera de estos rangos serán clasificadas como $S_5$. Otro ejemplo de este mapeo son las cuatro tramas de sub-tipo *Data*, las reglas que obtuvimos se enumeran a continuación:

1. Si el campo $(Moredata <= 0 \ \& \ tods > 0) -> s_6$
2. Si el campo $(Moredata <= 0 \ \& \ tods <= 0) \& \ duration > 31232 -> s_7$
3. Si el campo $(Moredata <= 0 \ \& \ tods <= 0) \& \ duration <= 31232 -> s_8$
4. Si el campo $(Moredata \ moredata > 0) -> s_9$

Las tramas de sub-tipo *Data* que contengan valores fuera de estos rangos serán clasificadas como $S_{10}$. Es así que al aplicar esta metodología obtuvimos un alfabeto $\sum$ con cardinalidad $m = 77$.
Para la construcción del modelo de normalidad usamos el algoritmo *Sequitur*[4], con el objetivo de detectar estructuras gramaticales repetitivas, que proporcionan poca información, y que por lo tanto pueden comprimirse y descartarse en un intento por detectar una intrusión. Nosotros redujimos sesiones identificando secuencias de tramas de capa 2 de ocurrencia frecuente, después, las reemplazamos, cada una por un nuevo meta-símbolo. Cada repetición da lugar a una regla de producción en la gramática y la sub-secuencia repetida es reemplazada por un símbolo no terminal, denominado meta-símbolo. Con esta finalidad aplicamos *Sequitur*, que es un método que infiere jerarquías de composición en la estructura de una cadena de entrada [4]. *Sequitur* detecta repetición en una cadena de entrada, y la factoriza mediante la formación de reglas en una gramática. Cuando forma reglas gramaticales, genera reglas en las que se comparten uno o más digramas, o aquellos que se utilizan sólo una vez en la producción de la cadena de entrada. Estas dos limitaciones permiten a *Sequitur* producir las reglas gramaticales más cortas que puedan generar la cadena de entrada.
Formalmente, especificamos la gramática como sigue. Sea $\sum$ el conjunto de símbolos terminales, representando tramas; sea $N = \{n_1, ..., n_k\}$, el conjunto de no terminales, conteniendo meta-símbolos; y, finalmente, sea $\delta$ el símbolo de inicio, entonces $\delta \notin N \cup \sum$. Cada regla de producción es de la forma: $n_k \rightarrow x_1 \ldots x_n$, donde $n_k \in N \cup S$ es la regla más hacia la izquierda, y donde $x_1 \ldots x_n$, con $x_i \in N \cup \sum$, es la regla más a la derecha de $N$. $\delta$, entonces, es la regla más hacia la izquierda de la regla de producción. El algoritmo recibe una sesión $B_i$ y trabaja secuencialmente leyendo los símbolos contenidos en ella en busca de sub-secuencias que se repiten dos o más veces, denotadas por $< sc_1 sc_2 ... sc_l >$. A cada sub-secuencia se le asigna un meta-símbolo $m$, que la identifica y con la que es reemplazada en $B_i$. Debido a que este algoritmo es recursivo, un meta-símbolo puede representar no sólo una sub-secuencia de tramas, sino también de meta-símbolos o una combinación de ambos. La salida de la reducción de una bitácora es una secuencia reducida $B_i^r$, y un modelo de reducción denotado por $R = \{m_1, m_2, ..., m_a\}$, tal que al remplazar recursivamente cada $m_i$ por su correspondiente regla de producción en $B_i^r$, se obtiene la secuencia original $B_i$. Este proceso se presenta en la figura 3.

## 6. Construcción del compresor

Con el conjunto reglas obtenidas por *Sequitur*, construimos un compresor. Para una compresión eficiente el compresor fué alimentado con un porcentaje del total de reglas generadas por *Sequitur*. En esta etapa implementamos el algoritmo *Boyer-Moore* [2]. Este algoritmo alinea un patrón (una cadena de símbolos $P$) con otra cadena de símbolos $Q$, donde $Q > P$ y luego comprueba si $P$ coincide con los símbolos de $Q$. Después de que la comparación es completada, $P$ se desplaza a la derecha en relación a $Q$. Este método examina si el patrón $P$ se

encunetra en $Q$ empezando de derecha a izquierda, esto normalmente le permite tener un factor significativamente más bajo que otros algoritmos de búsqueda. Para una secuencia de longitud $n$ y patrón fijo de longitud $m$, es $\frac{n}{m}$ en el mejor caso, solo uno en $m$ caracteres necesitan ser comprobados. Entre más largo es el patrón $P$ que estamos buscando, el algoritmo suele ser más rápido para encontrarlo. En función de esta idea ordenamos el conjunto de reglas generadas por *Sequitur*, el criterio a seguir fué ordenar las reglas más largas y las más frecuentes. Posteriormente tomamos un porcentaje del total de reglas obtenidas, con este sub-conjunto de reglas, se redujo cada sesión para obtener un porcentaje de reducción. Finalmente aplicamos el mismo conjunto de reglas al total de sesiones. Para la siguiente etapa de construcción del modelo, proponemos usar modelos ocultos de Markov o HMM, por sus siglas en inglés *Hidden Markov Models* [10]. El siguiente paso será comparar el modelo de una bitácora de validación HMM' contra el modelo del historial HMM canónico, figura 4, con el objetivo de encontrar posibles anomalías.

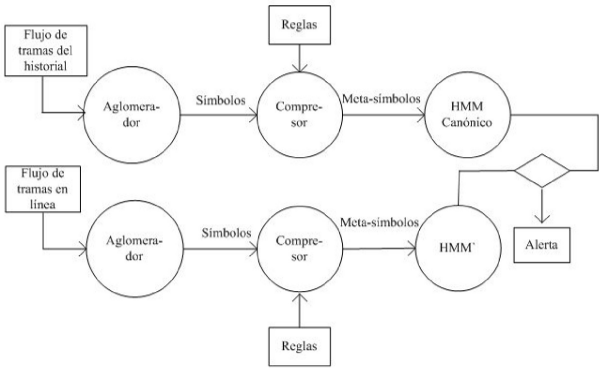


**Figura 4.** Arquitectura del Detector

## 7. Resultados

Los datos analizados son un conjunto de bitácoras que contienen una secuencia de tramas de capa 2 intercambiadas entre una STA y un AP. Seleccionamos 25 conversaciones entre diferentes parejas STA-AP del tráfico capturado en el edificio denominado *Aulas 1* del campus. Las conversaciones se tomaron aleatoriamente en diferentes horas del día y en distintas fechas, dentro de periodos de tiempo de 1 hora. Estas conversaciones representan interacciones entre STA y AP donde un usuario envía y recibe datos de la red, donde un usuario nuevo se asocia y autentica a la red del campus e interacciones ente diferentes AP's para transferir datos y mentener sincronía. El tamaño de las conversaciones fueron

de 548 tramas, la más corta y de 35259 tramas, la más larga. El total de reglas obtenidas con *Sequitur* fue de 1635. Con estos experimentos hemos logrado compactar satisfactoriamente las bitácoras, sin perder información importante. Los resultados de nuestro análisis reflejan un promedio de reducción por cada bitácora cercano al 92 %, usando solamente el 20 % de las reglas de mayor longitud y mayor frecuencia, esto se muestra en la tabla 3.

**Tabla 3.** Porcentajes de Reducción

| Número de reglas | % De reglas | % De reducción |
|---|---|---|
| 1635 | 100 | 98.2997 |
| 1380 | 80 | 97.4999 |
| 818 | 50 | 96.0227 |
| 327 | 20 | 92.0842 |

## 8. Conclusiones

Para la siguiente etapa de construcción del IDS, debemos considerar el comportamiento que presentaría un ataque, esperamos que una secuencia anómala, será difícil de comprimir, debido a patrones diferentes a los de comportamiento normal y a la persistencia del ataque. También debemos considerar que hay ataques que son efectivos con solo unas cuantas tramas, entonces, estos tipos de ataques serán detectables por la incoherencia que presentarían respecto del comportamiento normal. Es necesario considerar una arquitectura de HMM que sea capaz de diferenciar tanto secuencias que son difíciles de comprimir, como aquellas que, en longitud parecen normales pero que realmente presentan una incoherencia en el patrón con el que se haya entrenado al modelo.

## Referencias


1. Bernaschi, M., Ferreri, F., Valcamonici, L.: Access points vulnerabilities to dos attacks in 802.11 networks. Wireless Networks 14(2), 159–169 (2008)
2. Boyer, R.S., Moore, J.S.: A fast string searching algorithm. Commun. ACM 20(10), 762–772 (1977)
3. Ezeife, C.I., Ejelike, M., Aggarwal, A.K.: Wids: a sensor-based online mining wireless intrusion detection system. In: Desai, B.C. (ed.) IDEAS. ACM International Conference Proceeding Series, vol. 299, pp. 255–261. ACM (2008)
4. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of self for unix processes. In: IEEE Symposium on Security and Privacy. pp. 120–128. IEEE Computer Society (1996)
5. Guo, F., cker Chiueh, T.: Sequence number-based mac address spoof detection. In: Valdes, A., Zamboni, D. (eds.) RAID. Lecture Notes in Computer Science, vol. 3858, pp. 309–329. Springer (2005)

6. Haghani, S., Beaulieu, N.C.: Performance of s + n selection diversity receivers in correlated rician and rayleigh fading. IEEE Transactions on Wireless Communications 7(1), 146–154 (2008)
7. Könings, B., Schaub, F., Kargl, F., Dietzel, S.: Channel switch and quiet attack: New dos attacks exploiting the 802.11 standard. In: LCN. pp. 14–21. IEEE (2009)
8. Li, Q., Trappe, W.: Detecting spoofing and anomalous traffic in wireless networks via forge-resistant relationships. IEEE Transactions on Information Forensics and Security 2(4), 793–808 (2007)
9. Martínez, A., Zurutuza, U., Uribeetxeberria, R., Fernández, M., Lizarraga, J., Serna, A., Vélez, I.: Beacon frame spoofing attack detection in ieee 802.11 networks. In: ARES. pp. 520–525. IEEE Computer Society (2008)
10. Serralheiro, A.J., Ephraim, Y., Rabiner, L.R.: On nonstationary hidden markov modeling of speech signals. In: EUROSPEECH. pp. 1159–1162. ISCA (1989)
11. Sheng, Y., Tan, K., Chen, G., Kotz, D., Campbell, A.: Detecting 802.11 mac layer spoofing using received signal strength. In: INFOCOM. pp. 1768–1776. IEEE (2008)