# Uso de funciones compendio en la detección de anomalías mediante N3

R. Salazar-Hernández, J. Díaz-Verdejo, P. García-Teodoro, G. Maciá-Fernández, F. De Toro
Departamento de Teoría de Señal, Telemática y Comunicaciones. Universidad de Granada
ETSI Informática y Telecomunicación. c/ Daniel Saucedo Aranda s/n.
18071 – Granada (Granada)
Teléfono: 958 24 23 04 Fax: 958 24 08 31
E-mail: [rsalaza, jedv, pgteodor, gmacia,ftoro]@ugr.es

***Abstract***. *The Nearest Normal Neighbor (N3) is an anomaly-based intrusion detection system which has demonstrated a good performance in terms of detection capabilities when applied to the HTTP protocol. Nevertheless, N3 presents a high computational cost, as it is based in the comparison of the target HTTP payload against every payload in the normality model. The cost is proportional to the length of the payloads and to the number of elements in the model. The present paper explores the use of the hash functions as a method to reduce the computational cost of the system by decreasing the average length of the payloads. The model is, therefore, composed by fixed length hashes of each payload in the original model, and the hash of the target payload is compared against this model. The results obtained for SHA256 and SHA512 show a big decrease in computational cost with a reduced impact in system's performance.*

## 1 Introducción

Los sistemas de detección de intrusos analizan información para encontrar problemas de seguridad en las redes y equipos informáticos [1] [2].

El sistema de vecino normal más cercano (N3, *Nearest Normal Neighbor*) [3] [4], es un IDS de red basado en anomalías (A-IDS) en el que se usa una aproximación al problema por capas (protocolos), aplicándose técnicas de emparejamiento de patrones para el modelado y la detección. El sistema proporciona buenos resultados de detección manteniendo una baja tasa de falsas alarmas. Sin embargo, el algoritmo de detección presenta un elevado costo computacional, ya que se basa en el análisis de secuencias de caracteres de los protocolos analizados, utilizando para ello algoritmos de comparación de subcadenas de longitud fija. Estos algoritmos presentan una complejidad cuadrática con la longitud de las cadenas a comparar. Por otra parte, el modelo de normalidad consistirá en un conjunto suficientemente representativo de las cadenas normales, por lo que, la complejidad depende, adicionalmente, del tamaño del modelo. El coste resultante es elevado, dificultándose su implantación en entornos en explotación. En trabajos previos [5], hemos propuesto el uso de algoritmos de agrupamiento para reducir el tamaño de los modelos sin pérdidas de representatividad. Otra posible línea de actuación se basaría en la reducción de las longitudes de las cadenas a comparar. En este trabajo se propone y evalúa el uso de funciones compendio (*hash*) para reducir el tamaño de las secuencias de caracteres de los modelos y de las entradas para reducir el coste computacional.

El presente trabajo se articula de acuerdo al siguiente esquema. En el Apartado 2 se describe brevemente el sistema de detección de intrusiones N3. En el Apartado 3 se describe la aplicación de las funciones compendio para la reducción de las longitudes de las secuencias. Los conjuntos de datos utilizados se describen en el Apartado 4. En el Apartado 5 se presentan los resultados experimentales obtenidos usando las funciones compendio. Finalmente, en el Apartado 6 se muestran resultados de validación con otros conjuntos de datos y se analizan las mejoras conseguidas en el coste computacional. Por último, en el Apartado 7 se presentan las conclusiones.

## 2 El sistema detector N3

El IDS de vecino normal más cercano, N3, [3] opera en base al modelado del tráfico de red a partir de la monitorización de eventos discretos; en particular, de instancias de peticiones correspondientes a un determinado protocolo. Cada una de las instancias de tráfico, H, es procesada para obtener la carga útil (p). A continuación, éstas son analizadas por un detector que, tras su comparación con un modelo de normalidad (*M*), las clasifica como normales o anómalas.

La comparación con el modelo de normalidad se realiza a través del denominado *índice de anormalidad* de una carga útil, p, $A_s(p)$. La evaluación de dicho índice se basa en una medida de distancia, entre dos cargas útiles del protocolo, $p_1$ y $p_2$, $D(p_1,p_2)$, que es proporcional al número de subcadenas de longitud *k* dada comunes en ambas cargas útiles [3] [4]. A partir de dicha medida de distancia, el *índice de anormalidad* se obtiene como la distancia mínima entre la carga útil y cualquier elemento del modelo de normalidad, de acuerdo a

$$A_s(p) = \min_{\forall q \in M} D(p,q)$$

donde, evidentemente, el modelo de normalidad debe estar compuesto por cargas útiles normales. Si el índice de anormalidad supera un umbral

preestablecido, la carga útil será clasificada como anómala.

# 3 Uso de funciones compendio

A fin de reducir el tamaño de las cadenas en el modelo se propone en el presente trabajo el uso de funciones compendio [6]. Estas funciones se caracterizan por obtener secuencias de caracteres de longitud fija (el compendio) a partir de un mensaje o secuencia de caracteres de longitud arbitraria. De esta forma, se propone el uso de los compendios de las cargas útiles en el sistema N3, en lugar de las propias cargas útiles, tanto para la obtención del modelo de normalidad como para la evaluación de las cargas útiles a analizar.

Para los fines del presente trabajo se considerarán las funciones SHA-256 y SHA-512 [6], por lo que las longitudes de los compendios serán 256 y 512 bits, respectivamente. Así, antes de evaluar las cargas útiles o de incluirlas en el modelo, se obtendrán sus compendios, que serán los datos finalmente utilizados.

# 4 Bases de datos de tráfico

La evaluación del sistema requiere de varios conjuntos de datos que permitan establecer el modelo y obtener su rendimiento. Estos conjuntos (bases de datos de tráfico) deben contener instanciaciones del protocolo, en nuestro caso cargas útiles de peticiones HTTP. Se han recopilado dos bases de datos para realizar dos series de experimentos. La primera de ellas corresponde a parte del tráfico HTTP incluido en DARPA'99 [8], que constituye uno de los pocos referentes en la materia disponibles en la actualidad, aunque presenta serios inconvenientes y resulta un poco anticuada [9]. En particular, se ha tomado tráfico HTTP limpio, dando lugar a las bases de datos que denominaremos, respectivamente *Hume* y *Marx*. Debido a la antigüedad y bajo número de los ataques existentes, se han generado sintéticamente varios ataques HTTP, en un entorno equivalente, a partir de los ataques descritos en ArachNIDS [10], obteniéndose la base de datos denominada *Ataques*.

La segunda base de datos, denominada *UGRDB*, es una base de datos capturada en entorno real, correspondiendo a trazas del servicio HTTP proporcionado por el servidor web de la Universidad de Granada, que han sido anonimizadas. El tráfico capturado ha sido categorizado, utilizando Snort (http://www.snort.org), en función de su naturaleza maliciosa o no.

En la Tabla 1 se muestra un resumen del contenido de las bases de datos y conjuntos utilizados para el entrenamiento y evaluación del sistema. Para obtener estas particiones y el etiquetado necesario se ha seguido la metodología propuesta en [7].

**Tabla 1: Particionado de las bases de datos.**

| Base Datos | DARPA '99 | | UGRDB |
|---|---|---|---|
| | hume | marx | |
| Tráfico limpio | 12138 | 16505 | 25,000 |
| Ent. (70%) | 8508 | 11577 | 17500 |
| Eval. (30%) | 3646 | 4962 | 2500 |
| Tráfico ataques | 1500 | 1500 | 525 |

Finalmente, el modelo correspondiente para cada uno de los sistemas a evaluar será directamente el conjunto de entrenamiento obtenido (tras la obtención de sus compendios, en su caso).

# 5 Resultados experimentales

En primer lugar procedemos a evaluar el sistema N3 a partir del cálculo de los índices de anormalidad de las cargas útiles del protocolo HTTP. Este sistema constituye el *sistema de referencia*. El resultado de la experimentación se analizará mediante en curvas ROC ("Receiver Operating Curve") [11].

De acuerdo a resultados obtenidos en otras series de experimentos sobre esta base de datos [3], se ha procedido a evaluar el sistema con valores de k entre 3 y 6. Los resultados experimentales obtenidos, tanto para *Hume* como para *Marx* muestran que no existe solapamiento entre los índices de anormalidad del tráfico de ataques y del tráfico limpio, por lo que se podrá elegir un umbral de detección tal que se consiga un rendimiento correspondiente a un 100% de detección con un 0% de falsos positivos.

## 5.1 Resultados para la función compendio

La modificación propuesta consiste en el uso de las funciones compendio, en particular SHA-256 y SHA-512, para reducir la longitud de las secuencias a analizar a un valor fijo y, de esta forma, reducir el coste computacional del sistema. Por tanto, si denominamos H(p) a la función que obtiene el compendio de la carga útil p, el índice de anormalidad se calculará de acuerdo a

$$A_s(p) = \min_{\forall q \in M} D(H(p), H(q))$$

Para evaluar el sistema se han obtenido los valores de los índices de anormalidad de los compendios de las cargas útiles en las particiones de evaluación. Los resultados obtenidos para *Hume* (k=3) se muestran en la Fig. 1, si bien hay que indicar que los resultados son análogos para los restantes valores de k evaluados así como para *Marx*.

Como podemos observar en la Fig. 1, se produce un deterioro en el rendimiento del sistema en esta configuración, ya que es necesario incrementar el número de falsos positivos hasta en torno al 10% para conseguir un 100% de detección.

## 5.2 Longitud de la carga útil

Algunos trabajos descritos en la bibliografía [12] [13] muestran que existe información útil no sólo en las cadenas que conforman la carga útil del protocolo, sino también en la longitud de dicha carga útil. Resulta razonable, por tanto, evaluar si la inclusión de información sobre las longitudes de las cargas útiles mejora el rendimiento del sistema.

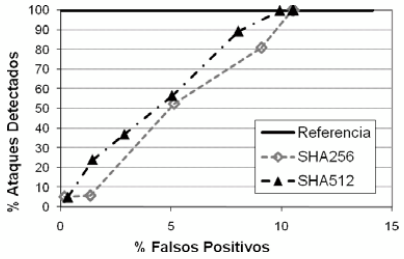


**Figura 1: Curvas ROC para *Hume* usando únicamente funciones compendio.**

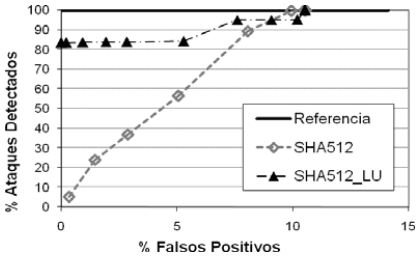


**Figura 2: Comparación de las curvas ROC en el caso de *Hume*.**

A este fin, se propone que, adicionalmente a la consideración del compendio, se limite la comparación de cada carga útil objeto de análisis a aquellas otras cargas útiles en el modelo con la misma longitud. A las cargas que no hayan sido comparadas con ninguna en el modelo, por no existir ninguna con idéntica longitud, se le asigna un índice de anormalidad arbitrariamente grande. Sin embargo, los resultados experimentales muestran la aparición de un efecto indeseable, debido a que existen cargas útiles normales cuya longitud no aparece entre las del modelo. Para evitar este problema proponemos "suavizar" el criterio usado para permitir la comparación mediante la inclusión de un umbral, Δ. Así, se comparará cada carga útil con todas aquellas del modelo cuya longitud difiera de la propia en menos del umbral considerado. Por tanto, si denominamos *L(p)* a la longitud de la carga útil *p*, el índice de anormalidad se evaluará, finalmente, de acuerdo a

$$A_s(p) = \begin{cases} \min\limits_{\substack{\forall q \in M / \\ |L(q)-L(p)|<\Delta}} D(H(p),H(q)) & si\ \exists r \in M / |L(r)-L(p)| < \Delta \\ \infty & en\ otro\ caso \end{cases}$$

Para el valor del umbral se ha seleccionado un valor Δ=10 a partir de la inspección del histograma de longitudes presentes en el modelo.

Los resultados experimentales obtenidos muestran una mejora en el comportamiento del sistema, tal como se puede observar en la Fig. 2. En esta gráfica, las curvas ROC correspondientes a la utilización conjunta del compendio y la longitud de la carga útil, cuando se considera el umbral (SHA256_LU y SHA512_LU), proporcionan un rendimiento que, en el punto de operación óptimo (100% de detección) son iguales o superiores al proporcionado únicamente por las funciones compendio. Por otra parte, cualquier punto de operación elegido mediante la asignación de un valor del umbral de detección proporcionará mejores resultados en el caso del uso de la longitud con umbral, estando siempre por encima del 80% de ataques detectados. Resultados análogos se obtienen para *Marx*.

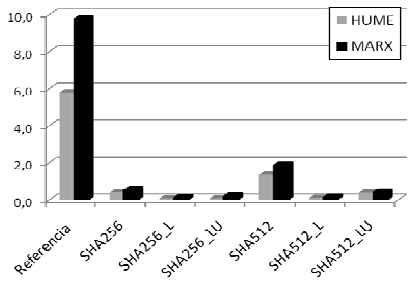


**Figura 3: Tiempos de procesamiento por carga útil analizada para las variantes evaluadas.**

# 6 Coste computacional y validación

A continuación procederemos a evaluar las mejoras conseguidas en cuanto a coste computacional asociado. Los experimentos del presente trabajo fueron realizados en un servidor, con procesador AMD Athlon 64 X2 Dual Core a 2 GHz, 512 kb de cache, con una memoria RAM de 2 GB, bajo sistema operativo Linux Red Hat 3.4.6-3 con kernel 2.6.9-42.

Los tiempos medios de procesamiento por carga útil evaluada, para las diferentes variantes analizadas, se muestran en la Fig. 3 en milisegundos. En dicho tiempo se incluye el utilizado por el procesador en evaluar la función hash sobre la carga útil objeto de clasificación y el tiempo en calcular las distancias mínimas y máximas con respecto al modelo para determinar si una petición es normal o anómala. En la Fig. 3 resulta evidente la gran reducción en el coste computacional conseguida mediante la aplicación de funciones compendio, que llega a ser de un orden de magnitud en el peor caso (uso de funciones SHA512). Por otra parte, la inclusión de la longitud de las cargas útiles introduce una reducción adicional en el coste (p.e., SHA256 frente SHA256_LU) debido a que disminuye el número de comparaciones a realizar para cada carga útil objeto de análisis.

## 6.1 Resultados de validación

Para validar la metodología propuesta se ha procedido a realizar una serie de experimentos sobre la base de datos *UGRDB*. Los resultados obtenidos tras aplicar la función compendio (SHA256) y tras considerar la longitud de las cargas útiles

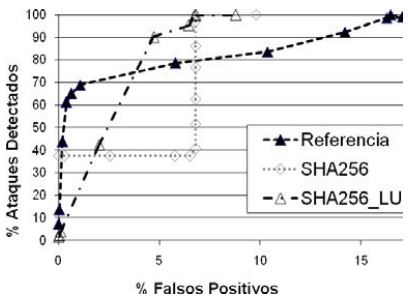


**Figura 4: Curvas ROC para la base de datos *UGRDB*.**

(SHA_256_LU) se muestran en la Fig. 4. Análogos resultados se obtienen para SHA512. Como se puede observar, los resultados de la experimentación muestran un comportamiento satisfactorio. Es más, con esta base de datos, la modificación propuesta mejora el rendimiento obtenido por el sistema N3 en su formulación original. Y, además, esta mejora se produce incluso con la aplicación exclusiva de las funciones compendio.

Este resultado nos resulta enormemente sorprendente, ya que la aplicación de la función compendio implica una pérdida de información. El objetivo inicial de la experimentación realizada era evaluar la reducción que se conseguiría en el tiempo de cómputo. Por motivos de simplicidad en la implementación, se seleccionaron las funciones SHA-n en una primera aproximación. Sin embargo, estas funciones presentan un comportamiento que no resulta acorde con la filosofía subyacente en el sistema N3: dos cargas útiles parecidas deben proporcionar una distancia reducida y, en consecuencia, un bajo índice de anormalidad. Pero la aplicación de las funciones compendio modifica esta relación. De acuerdo a las propiedades de las funciones SHA, dos cargas útiles similares deben proporcionar compendios claramente diferentes. En consecuencia, que dos compendios presenten una distancia pequeña entre ellos no implica que las cargas útiles originales fuesen parecidas. La única explicación plausible podría residir en alguna propiedad global del modelo, lo que debe ser explorado en trabajos sucesivos.

## 7 Conclusiones

En el presente trabajo se ha mostrado que, mediante la aplicación de funciones compendio, en particular SHA256 y SHA512, en el preprocesado de las cargas útiles HTTP es posible reducir la complejidad computacional del sistema IDS N3 sin degradar significativamente su rendimiento. Las modificaciones propuestas reducen considerablemente el tiempo de cómputo, llegando incluso a mejorar las capacidades de detección en algunos escenarios. Sin embargo, los resultados obtenidos muestran un comportamiento no esperado que debe ser analizado con más detalle. Por otra parte, a la vista de los resultados, cabría evaluar el comportamiento en el caso de utilizar funciones resumen con otras propiedades más adecuadas a los fines del sistema N3.

## Agradecimientos

Este trabajo ha sido parcialmente financiado por el Programa Nacional de I+D+I (2004-2007) del MEC (proyecto TSI2005-08145-C02-02, 70% fondos FEDER).

La participación de R. Salazar ha sido posible gracias al programa PROMEP y a la UAT (México).

## Referencias

[1] Info-Tech Research Group. Intrusion Detection: The Essential Buyer's Guide. London, ITRG, 2003.

[2] Kabiri P., Ghorbani A.; Research on Intrusion detection and response: A survey, International Journal on Network Security, Vol. 1, N. 2, pp. 84-102, 2005.

[3] Estévez-Tapiador, J. M., Detección de intrusiones en redes basada en anomalías mediante técnicas de modelado de protocolos, Tesis Doctoral, Universidad de Granada, 2004.

[4] Estévez-Tapiador J.M., Díaz-Verdejo J.E., García-Teodoro P., N3: A geometrical approach for network intrusion detection at the application layer, ICCSA 2004, LNCS 3043, p p. 841-850, 2004.

[5] García-Teodoro, P.; Estévez-Tapiador, J.M.; Díaz-Verdejo, J.E.; Técnicas de agrupamiento vectorial y detección geométrica de anomalías en red; Actas de las V jornadas de Ingeniería Telemática, pp. 531-538; Vigo 2005.

[6] NIST, Secure Hash Standard, FIPS PUBS 180-2. Mayo 2001 actualizado Febrero 2004. http://csrc.nist.gov

[7] M. Bermúdez-Edo, R. Salazar-Hernández, J. Díaz-Verdejo, and P. García-Teodoro. Proposals on Assessment Environments for Anomaly-Based Network Intrusion Detection Systems. CRITIS 2006, LNCS 4347, pp. 210 – 221, 2006. Springer-Verlag, 2006.

[8] Lippmann, R., Haines, J:W., Fried, D.J., Korba, J., and Das, K. Analysis and results of the 1999 DARPA off-Line Intrusion Detection Evaluation. In *Computer Networks* 34(4), pp. 579-595, 2000.

[9] McHugh, J.; The 1998 Lincoln Laboratory IDS Evaluation. A critique, In RAID 2000, LNCS 1907, pp 145-161, 2000.

[10] arachNIDS: Advanced Reference Archive of Current Heuristics for Network Intrusion Detection Systems. http://www.whitehats.com/ids

[11] Egan, J. (1975). Signal Detection Theory and ROC Analysis. Academic Press, Inc.

[12] Krügel, C., Toth, T., and Kirda, E.; Service Specific Anomaly Detection for Network Intrusion Detection, Proc. 17th ACM Symp. on Applied Computing (SAC), pp. 201-208 , 2002.

[13] J.M. Estévez Tapiador, P. García Teodoro, J.E. Díaz Verdejo; Measuring Normality in HTTP Traffic for Anomaly-Based Intrusion Detection, Computer Networks; Vol 45, pp. 175-193, 2004.