APOTHEOSIS: Sistema eficiente para búsqueda de similitud aproximada en análisis forense digital

Daniel Huici[1], Ricardo J. Rodríguez[1], Eduardo Mena[2]

[1] Grupo de I+D en Computación Distribuida (DisCo)
[2] Sistemas de Información Distribuidos (SID)
Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, e-mail: dhuici@unizar.es

Resumen

APOTHEOSIS combina búsqueda exacta y búsqueda aproximada de similitud para análisis forense a gran escala. Integra algoritmos de resumen de similitud (SDA) con estructuras de datos avanzadas (árbol radix y HNSW), logrando alta precisión y tiempos interactivos sobre millones de artefactos digitales.

Introducción

El crecimiento de los datos y la proliferación de dispositivos heterogéneos han incrementado la complejidad de las investigaciones forenses. El hashing exacto permite identificar artefactos conocidos, pero falla cuando el contenido cambia ligeramente por recompilación, reformateo o parcheado. Los algoritmos de resumen de similitud (SDA) [2] abordan esta limitación al generar huellas cuya comparación produce una puntuación de similitud.

En la práctica, estas puntuaciones permiten detectar casi duplicados, versiones derivadas o fragmentos relacionados que ya no comparten el mismo hash criptográfico. El problema aparece al escalar: comparar una consulta contra todos los artefactos almacenados es inviable cuando el repositorio contiene millones de entradas y el analista necesita respuestas interactivas.

Este trabajo presenta APOTHEOSIS [1], un sistema extensible que combina búsqueda exacta de hashes y búsqueda aproximada escalable sobre hashes SDA para localizar duplicados cercanos o contenido derivado en escenarios forenses.

Contexto

APOTHEOSIS se apoya en SDAs como ssdeep [3] y TLSH [4]. Ambos tratan los artefactos como secuencias de bytes y generan hashes comparables, aunque su comportamiento depende del dominio analizado. Un algoritmo puede resultar más adecuado para código fuente, otro para binarios o páginas de memoria, por lo que el sistema debe permitir comparar algoritmos sin cambiar toda la infraestructura.

Para acelerar la búsqueda se combinan dos estructuras. Un árbol radix [5] almacena hashes para consultas exactas y basadas en prefijo con coste ligado a la longitud de la cadena. Un grafo HNSW [6] soporta búsqueda aproximada de vecinos cercanos, adaptando la comparación a las funciones de puntuación propias de cada SDA.

Descripción del sistema

APOTHEOSIS se organiza en dos fases. Durante la indexación, cada artefacto se procesa con un SDA. El hash resultante se inserta en un árbol radix para consultas exactas y en un grafo HNSW para búsqueda aproximada de vecinos. Los metadatos se almacenan aparte y se referencian desde los nodos del grafo para mantener una estructura compacta.

Durante la consulta, el sistema comprueba primero el árbol radix para encontrar coincidencias exactas. Si se solicita búsqueda por similitud, ejecuta HNSW para recuperar los k candidatos más cercanos o los digestos cuya puntuación supera un umbral. De este modo, el mismo índice permite responder tanto a preguntas de identificación como a búsquedas exploratorias por parecido.

Esta separación entre generación de digestos, estructuras de búsqueda y metadatos facilita añadir nuevos SDA y conservar información contextual sin cargar el grafo con datos innecesarios.

Evaluación

Caso 1: detección de plagio

El primer caso de estudio evalúa la detección de código duplicado o plagiado en 43.792 ficheros de tareas de programación [7]. Se construyeron modelos

con ssdeep y TLSH, y se compararon con un baseline MinHashLSH [8]. La Figura 1 resume la exactitud, precisión, exhaustividad y F1 para distintos umbrales de similitud.

APOTHEOSIS alcanza precisión cercana a 1,00 con ambos SDA en todos los umbrales evaluados. TLSH obtiene valores de exhaustividad y F1 superiores a ssdeep en este escenario, mientras que MinHashLSH muestra un comportamiento más irregular y capta peor las coincidencias exactas. Operativamente, esto reduce falsos positivos y prioriza candidatos para revisión manual.

Caso 2: escalabilidad con memoria Windows

El segundo caso de estudio se centra en la escalabilidad con aproximadamente 4,2 millones de páginas de memoria pertenecientes a unos 37.500 ficheros de sistema Windows. Cada página se procesa con TLSH y se indexa en APOTHEOSIS. La Figura 2 muestra el ajuste de la operación k-NN aproximada para tres configuraciones de HNSW.

Los experimentos indican que el tiempo de búsqueda crece de forma aproximadamente logarítmica con el número de elementos, en línea con la complejidad esperada de HNSW. El análisis de sensibilidad muestra que los parámetros de conectividad afectan a la exhaustividad y al uso de recursos, pero el sistema mantiene tiempos interactivos incluso con millones de entradas.

Este caso es relevante para análisis de memoria porque muchos artefactos aparecen repetidos, modificados o fragmentados entre versiones del sistema operativo. La búsqueda aproximada permite agrupar páginas relacionadas aunque no sean idénticas byte a byte, mientras que la consulta exacta sigue siendo útil para descartar elementos conocidos.

Conclusiones

APOTHEOSIS demuestra que la búsqueda exacta por hash y la búsqueda de similitud aproximada pueden integrarse en un único sistema forense. La combinación de árbol radix y HNSW adaptado proporciona alta precisión en código y escalabilidad en grandes conjuntos de memoria.

El diseño es agnóstico respecto al SDA y puede soportar otros algoritmos o tipos de artefacto siempre que pueda calcularse un digesto intermedio. Como trabajo futuro se plantea explorar otras estructuras de vecinos aproximados, automatizar el ajuste de parámetros e integrar APOTHEOSIS en flujos forenses completos.

Agradecimientos

Este trabajo ha sido financiado en parte por los proyectos PID2020-113903RB-I00 (KIT-IA), PID2023-151467OA-I00 (CRAPER), TED2021-131115A-I00 (MIMFA), Proyecto Estratégico Ciberseguridad EINA UNIZAR y Programa de Proyectos Estratégicos de Grupos de Investigación (DisCo y SID, refs. T21-23R y T42-23R), con financiación de MICIU/AEI/10.13039/501100011033, FEDER/UE, Unión Europea NextGenerationEU/PRTR, INCIBE y Gobierno de Aragón.

Figuras

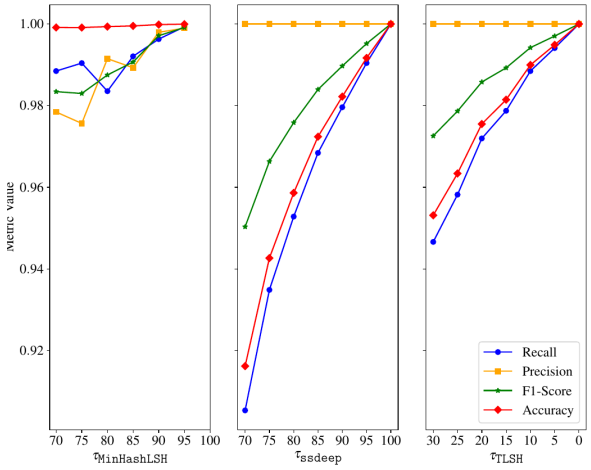

Figura 1. Métricas de rendimiento (exactitud, precisión, exhaustividad y F1) para APOTHEOSIS con ssdeep y TLSH, y para MinHashLSH, bajo distintos umbrales de similitud.

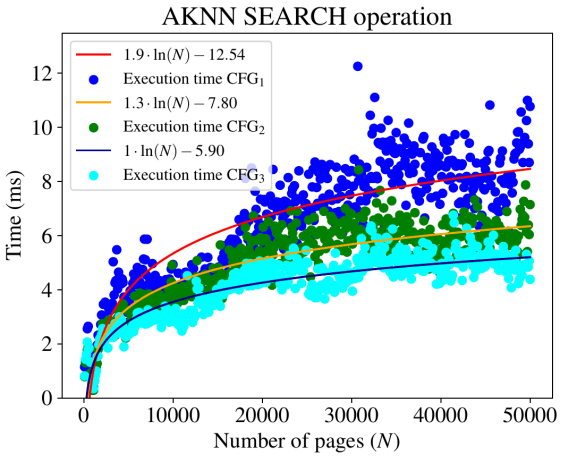

Figura 2. Ajuste de la curva de la operación k-NN aproximada para tamaños de conjunto N en [100, 50000] y tres configuraciones de APOTHEOSIS.

Referencias

[1]. HUICI, D., RODRÍGUEZ, R.J. and MENA, E. Apotheosis: An efficient approximate similarity search system. SoftwareX. 2025, 29, 102016.

[2]. BREITINGER, F., GUTTMAN, B., MCCARRIN, M., ROUSSEV, V. and WHITE, D. Approximate Matching: Definition and Terminology. Gaithersburg: National Institute of Standards and Technology, 2014. NIST Special Publication 800-168.

[3]. KORNBLUM, J. Identifying almost identical files using context triggered piecewise hashing. Digital Investigation. 2006, 3, 91-97.

[4]. OLIVER, J., CHENG, C. and CHEN, Y. TLSH - A Locality Sensitive Hash. In: 2013 Fourth Cybercrime and Trustworthy Computing Workshop. IEEE, 2013, pp. 7-13.

[5]. MORRISON, D.R. PATRICIA - Practical Algorithm To Retrieve Information Coded in Alphanumeric. Journal of the ACM (JACM). 1968, 15, 514-534.

[6]. MALKOV, Y.A. and YASHUNIN, D.A. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020, 42(4), 824-836.

[7]. LJUBOVIC, V. Programming Homework Dataset for Plagiarism Detection [online]. 2020 [accessed 28 September 2024]. Available from: https://dx.doi.org/10.21227/71fw-ss32.

[8]. BRODER, A. On the resemblance and containment of documents. In: Proceedings. Compression and Complexity of SEQUENCES 1997. IEEE, 1997, pp. 21-29.