APOTHEOSIS: An Efficient Approximate Similarity Search System for Digital Forensics

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)
[1,2]Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762650, e-mail: dhuici@unizar.es

Abstract

APOTHEOSIS is a similarity search system that enables fast identification of similar digital artifacts within large volumes of data. It uses approximate search techniques and approximate similarity matching algorithms for efficient detection. It was evaluated with a Microsoft Windows processes dataset, validating its functionality.

Introduction

Modern reliance on technology exposes organizations to cyberattacks, necessitating rapid and effective incident response. The large volumes and variety of digital data makes this task truly challenging.

In this work, we introduce APOTHEOSIS, a system that combines two efficient data structures (in particular, Radix Tree [1] and HNSW [2]), along with similarity digest algorithms (SDA) [3] as a basis for similarity comparison. The joint use of both technologies allows us to speed up identification within large datasets, finding similar digital artifacts in an efficient and timely manner.

To validate the system, we conduct a thorough evaluation within a large dataset composed of Windows system modules, trying to find suitable configuration values to balance precision and time. Obtained results shows that the system can perform efficiently and very quick approximate nearest neighbor search over large datasets.

Background

In this section we introduce two key areas to fully grasp the essence of our work: SDA and efficient search methods.

SDA play a main role in our approach. They allow comparison between digital artifacts that work at the byte level and use an intermediary representation. This method broadens the scope beyond exact matches, helping to identify similar but not identical objects. Our implementation focuses on two different SDA: ssdeep and TLSH, each with distinct features and suitability for various forensic tasks.

Complementing SDA, our system integrates two efficient search methods: Radix Tree and HNSW. Radix Tree [1] is a versatile data structure that optimizes space by consolidating nodes that share common prefixes, offering efficient information retrieval. On the other hand, HNSW [2] is a hierarchical graph-based approach that addresses the challenges of searching in high-dimensional data. In this structure, the lowest layer represents the entire data space, while the upper layers progressively condense this representation, allowing efficient navigation in complex data spaces, offering fast approximate nearest neighbor search.

System description

In this section, we provide a high-level overview of our system.

Figure 1 shows an overview of different use cases for APOTHEOSIS: used by any binary diffing tool (as software library) or as a service, remotely, to identify similarities in binary code or by any external forensic workflow. Our system can be applied to many forensics-related search problems (e.g., common binary code in programs, common elements in images, etc.), if some intermediate representation (e.g., a digest or a hash) can be computed for the element of interest.

As introduced above, our system uses two different data structures that work together, combined with similarity digests. When inserting a new hash, it is first inserted into the Radix Tree, traversing the tree to find the common prefixes and creating a new node at the appropriate position, if necessary. It is then inserted into the HNSW structure, which involves traversing the graph from the highest level assigned to a new node (chosen randomly) to the ground level, selecting nearest neighbors at each layer and establishing connections between them.

During the search operation, we first try to find the query in the Radix Tree to find exact matches (lookup), and when the hash is not found, we perform a KNN search on the HNSW structure by traversing the graph from the highest level to the lowest level. Once at the lowest, we loop though the nodes to find the nodes that are closes to the query.

Experiments

To perform a proper evaluation of the system we created a dataset containing similarity digests of Windows system modules extracted from different runs of various Windows versions. We then calculate the similarity hash using different SDAs (in particular, ssdeep and TLSH) and build the database. In total, we have approximately 4.2 million hashes.

We have conducted a proper evaluation of different system operations (insert, lookup and search) and modified the system configuration parameters to evaluate their impact on system performance and precission. Figure 2 shows an example of the plots that we obtained from the experimental results.

Conclusions

Efficiently handling and scrutinizing large datasets poses a significant hurdle during incident response, underscoring the critical need for fast and accurate tools to quickly detect and analyze malicious elements.

We introduced APOTHEOSIS, an extensible and versatile system that leverages the power of approximate search methods and similarity digest algorithms to facilitate similarity analysis of digital artifacts. In particular, we apply it in the context of binary data, demonstrating that the combination of this technologies provides an interesting solution for the addressed problem (rapid and efficient identification of similar digital artifacts within large binary datasets). The evaluation carried out allows us to conclude that the system is efficient and scalable for large data sets.

To improve usability and accessibility, we also provided a REST API interface for APOTHEOSIS, allowing forensic analysts to seamlessly integrate it into existing forensic workflows. To promote open science and reproducibility, we release our system under the GNU/GPLv3 license.

In the future, we aim to investigate and evaluate other K-ANNS methods and explore additional use cases to expand the scope of our system.

REFERENCIAS

[1]. Donald R. Morrison. 1968. PATRICIA–Practical Algorithm To Retrieve Information Coded in Alphanumeric. Journal of the ACM (JACM) 15 (1968), 514–5

[2]. Yu A. Malkov and D. A. Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence 42, 4 (2020), 824–836.

[3]. Frank Breitinger, Barbara Guttman, Michael McCarrin, Vassil Roussev, and Douglas White. 2014. Approximate Matching: Definition and Terminology. Techreport NIST Special Publication 800-168. National Institute of Standards and Technology GOLDBERG, K. and KEHOE, B. Cloud robotics and automation: A survey of related work. Berkeley: EECS Department, University of California Berkeley, 2013. Technical Report.

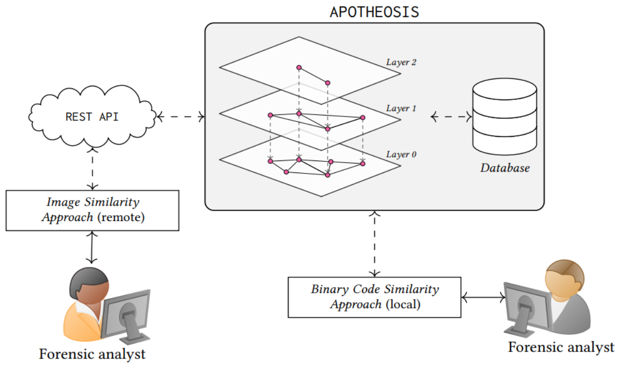

Figure 1: High-level overview of different use cases of APOTHEOSIS

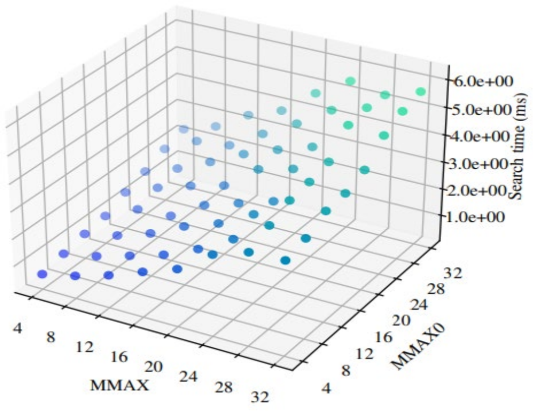

Figure 2: Approximate K-NN search operation times for specific configuration values