Ransomware Statistics Engine (RaStE)

Bochra Jendoubi, Belhassen Essid, Tojieva Jahonoro, Amira BenElkbaier, Rishad Akhundov

Eötvös Loránd University, Budapest, Pázmány Péter stny. 1117, Hungary

Eötvös Loránd University

1 Abstract

Ransomware attacks cause a detrimental amount of financial damage for businesses and individuals. The project implements a kafka based system to extract ransomware data and analyse it in real time in addition to storing it in influxDB databases for further analysis. In addition, the extracted large volume data streams are processed with the help of Apache Spark before a general visualization through a python built in library called dash. This project contributes to the growing need for effective cybersecurity solutions. The integration with Grafana allows for interactive visualization of the data, providing actionable insights for cybersecurity teams. This approach not only enhances the ability to monitor and respond to ransomware attacks but also improves strategic decision-making by uncovering hidden attack patterns and predicting future threats.

2 Introduction

Ransomware attacks are one of the most popular and easily monetized malware options present today. Businesses and individuals suffer from these attacks more and more, in fact by 2016 the losses hit 1 billion dollars in the US alone (Brewer, R., 2016). The rapid evolution of ransomware and other cybersecurity threats makes innovative approaches for real-time detection and analysis important. These attacks have evolved over the years and continue to do so which increases in magnitude more drastically than the ways to prevent them, therefore making real-time threat forecasting crucial. Recent advancements in distributed systems and big data processing have enabled the creation of architectures that can efficiently handle large-scale, high-velocity data streams. Our project leverages Apache Kafka for data ingestion, Apache Spark for processing, and tools such as Grafana and InfluxDB for visualization and storage, to create a scalable framework for ransomware data analytics.

A well-structured and well-distributed system is helpful for further historical analysis of the attacks and extensive data analysis. For this reason, Kafka was used to ensure a reliable and scalable process distribution (Kreps et al., 2011; Sharma et al., 2018). Due to limitations of our computationsal power Spark was used to set up a smooth pipeline as Spark is well-known for its suitability in low-latency computations (Databricks, 2020; Ali et al., 2017). The decision to utilize Spark over Apache Flink was informed by its broader ecosystem and ease of implementation, aligning with recommendations from recent benchmarking studies (Chintapalli et al., 2016).

Analyzing and predicting cyberattacks requires robust methods to handle the inherent challenges of cybersecurity data, such as incompleteness, imbalance, and insignificance. As highlighted by Liu et al. (2019), effective forecasting techniques must adapt to these issues while uncovering meaningful patterns and trends. General approaches include machine learning algorithms for classification and prediction, clustering techniques to identify attack patterns, and time-series analysis to forecast future threats. These methods enable cybersecurity systems to proactively respond to emerging threats, even when faced with complex and imperfect datasets, underscoring the importance of innovative analytics in maintaining digital security. With the availability of open source tools it is possible to build a well distributed and cost-efficient system.

Our project shares a common focus with the study by Patel et al. (2023) on utilizing machine learning for ransomware detection and prediction. Both approaches emphasize the importance of analyzing large-scale, real-time data for effective cybersecurity measures. While Patel et al. provide a comprehensive review of existing machine learning models for ransomware detection, identifying challenges such as feature selection, model interpretability, and data imbalance, our project builds on these insights by implementing practical solutions. Specifically, we employed Random Forest for prediction and KMeans for clustering, addressing issues like data imbalance and dynamic attack vectors. Additionally, our project extends the scope to include time-series forecasting for predicting attack trends, whereas Patel et al. focus more on static detection models. This comparative analysis highlights the complementary nature of theoretical reviews and applied implementations in advancing ransomware analytics.

The article by IBM X-Force (2023) provides an in-depth analysis of ransomware trends, highlighting the increasing sophistication of ransomware tactics and the rise of double-extortion strategies. Their findings emphasize the critical need for proactive and scalable defenses against evolving threats. Our project aligns with these insights by focusing on predictive analytics and clustering models to identify attack patterns and forecast future ransomware trends. While the IBM X-Force study primarily analyzes real-world case studies and threat intelligence, our project complements this by leveraging tools like Apache Kafka and Spark to process high-velocity data streams for real-time predictions. Together, these approaches underscore the importance of combining practical threat analysis with advanced computational models to stay ahead of ransomware attacks.

The study by Gao et al. (2022) explores the use of advanced machine learning techniques and time-series models to enhance ransomware detection and prediction. Similar to our project, their research highlights the importance of scalable architectures for processing large datasets and addressing real-time cybersecurity challenges. While Gao et al. focus on designing hybrid models combining neural networks and statistical methods for improved accuracy, our approach emphasizes practical implementation using Random Forest for prediction and KMeans for clustering. Additionally, our project integrates real-time data pipelines with Apache Kafka and Spark, providing a streamlined architecture for both prediction and visualization. This comparison demonstrates how diverse methodologies contribute to advancing ransomware analytics and detection strategies.

3 Dataset

3.1 Description

To ensure a smooth continuation of the project steps the stream mining processes were first implemented on a static data that consists of IP addresses, City Names, Country Names and their geographical latitude and longitude before implementing the mining pipeline on an online data consumed from OTX AlienVault API. This was done to ensure the workflow is up and running. The dynamic data for stream processing is fetched through the OTX API. The dataset consumed from the AlienVault OTX API consists of data related to ransomware threats, with tables such as pulses, indicators, and IP location. These tables provide crucial information for threat intelligence, including details about specific ransomware campaigns, the indicators associated with them, and the geographical locations of the IP addresses involved. Data is preprocessed to avoid redundancy and improve connectivity speeds

3.2 Limitations

The dataset has a few limitations that affect the accuracy of model results. Missing critical values and overall quality of the data made it more difficult to produce accurate results.

4 System Architecture

Kafka based system architecture is connected to the OTX AlienVault API for collecting ransomware data. As a message broker Kafka enables a real-time transfer of high-volume data. The data that is collected and stored in kafka topics can be fetched by the next tools implemented to process analyse and visualise both real-time and the provided data

The final streaming system uses Apache Spark to process the data. This change was made to facilitate easier implementation as Spark's extensive ecosystem and integration with big data tools provide greater flexibility and support. The data is processed through messages between worker consumer and master docker spark containers in real-time. We can get insight about top target countries or source regions through real time processing with the help of spark - kafka integration. The system ensures that data is efficiently filtered, aggregated, and processed with low latency, allowing for the near-instantaneous identification of threat patterns as new data flows in.

We employed a dual batch processing approach for our ransomware analytics system: clustering and prediction. For clustering, we explored both the KMeans implementation from the sklearn library and the KMeans model from the river library, which is better suited for streaming data. For prediction, we initially trained time-series models such as ARIMA and SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous variables) to detect seasonality and trends, but the prediction accuracy was unsatisfactory. Consequently, we transitioned to using RandomForestRegressor, which provided improved predictive performance. This two-step process enables both attack pattern clustering and forecasting, delivering actionable insights for proactive cybersecurity measures.

InfluxDB, used as the storage solution for handling large volumes of time-series data is optimized for fast ingestion and retrieval of time-stamped data, making it an ideal choice for storing ransomware-related metrics. Data such as attack frequency, source regions, and target countries are stored in InfluxDB, ensuring that both real-time and historical data are available for further analysis and visualization. Therefore it was used to store the predictions and results of our clustering models.

For visualization, Dash library in python is employed to create a dashboard that displays both real-time and historical analytics on ransomware attacks. Our python script integrates with InfluxDB to visualize key metrics, such as attack trends, top source regions, and target countries. The dashboard offers interactive features, allowing users to filter and explore different timeframes, detect patterns, and monitor ransomware activity. This visualization layer enables cybersecurity teams to gain actionable insights and make informed decisions in response to emerging threats.

Overall, the system architecture integrates stream processing, batch processing, machine learning, and time-series analysis, providing a robust framework for monitoring and predicting ransomware threats. The use of Kafka, Spark, InfluxDB, and Grafana ensures that the system is scalable, efficient, and capable of delivering timely, actionable insights for cybersecurity teams.

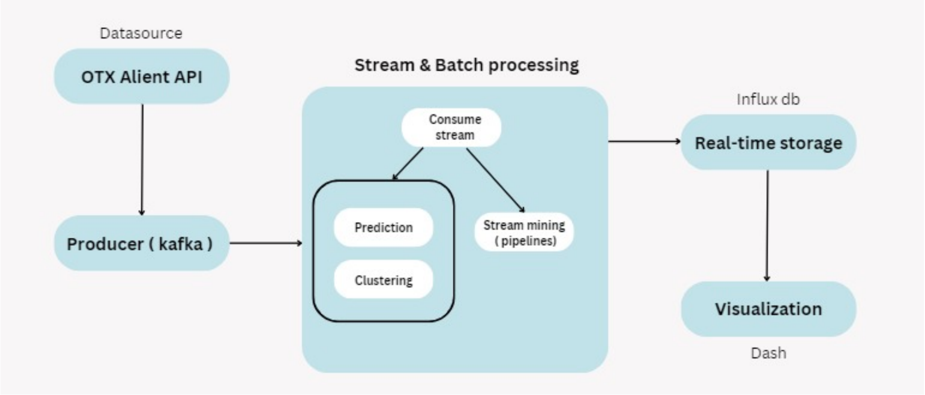

Fig. 1. Ransomware Analytics System Architecture

5 Experiments and Methodology

5.1 Data Collection and Ingestion

A custom Kafka producer script collects ransomware-related indicators (e.g., IP addresses, attack timestamps, attack types, source locations) and feeds this data into Kafka topics. The producer sends data continuously, simulating real-time attack data streams for the experiments.

The dataset is composed of structured data such as:

- IP addresses
- Countries affected
- Ransomware variants
- Timestamps
- Indicator ID
- Type of Indicator

The Kafka producer streams this data into one topic, categorized by parameters like:

- Attack type
- Affected region

The Kafka container is configured to use a high-throughput setup for efficient streaming, with a replication factor of 3 to ensure fault tolerance. The Kafka brokers are configured to listen on ports 9092 for internal communication, enabling the consumer containers (Spark) to retrieve the data.

5.2 Stream Processing

The system architecture is designed to process ransomware analytics through periodic batch processing, with distinct pipelines for forecasting, clustering, and detailed analytics. The '`process_batch(batch_df)`' function handles forecasting tasks, leveraging historical trends to predict potential threats. Simultaneously, '`process_clustering(batch_df)`' identifies attack patterns using clustering algorithms. The architecture integrates a series of analytics pipelines for in-depth insights: identifying the top 10 target countries, top 10 threat sources, top 10 active IPs, and the most common attack types. Additional pipelines detect changes in target and source countries, analyze attacks by creation day and month, and provide details on top source cities and authors. This modular design ensures comprehensive ransomware analytics, enabling actionable intelligence and proactive cybersecurity measures.

To process the streamed data, the project utilizes Apache Spark. This engine perform real-time data transformations, enrichments, and analytics. Initial experiments were conducted on Flink, however, a decision to choose spark for stream processing was made for its flexibility.

- Spark Consumer: The Spark consumer container listens to the Kafka topic, processes the data, and applies machine learning models for anomaly detection and clustering. The data is processed in micro-batches, and Spark performs aggregations like:
  - Top N target regions
  - Identifying ransomware clusters (e.g., identifying attack patterns common across multiple incidents)

Spark Experimentation:

- Clustering: K-means clustering is applied to group similar ransomware variants or attack patterns.
- Prediction: A simple regression model forecasts attack volumes based on past data.

Spark is tested under various configurations for:

- The number of workers
- Partitioning strategies
- Effect on processing time

We opted for parallel pipeline execution for efficiency and to avoid the use memory on processing each pipeline independently since we would have to process the batch repeatedly.

5.3 Data Storage and Time Series Analysis

The processed data is stored in InfluxDB, a time-series database that allows efficient querying and long-term storage of the data. The database schema is structured to accommodate different types of data, such as:

- Attack counts
- Regions affected
- Timestamps

The time-series data from Flink and Spark is written into InfluxDB, with buckets used to store attack statistics:

- The ransomware bucket stores data related to attack frequency, top regions, and countries.
- The monitoring bucket is used for tracking system metrics like processing time and throughput.

Time-Series Analysis:

- Queries are used to extract patterns such as daily attack counts, top affected regions, and ransomware variants, allowing for trend analysis.

5.4 Visualization

The processed data is visualized using Dash, which connects to InfluxDB to display real-time dashboards. These dashboards show metrics such as:

- Top 10 Target Countries: Displays the countries most frequently targeted by ransomware attacks.
- Attack Trends: Visualizes the frequency of ransomware attacks over time.
- Top 10 Source Regions: Identifies the source regions or IP addresses from which ransomware attacks are originating.

Our script also displays system performance metrics, including latency and throughput, and offers interactive drill-down capabilities for users to filter by time or region.

5.5 Model Evaluation

For the machine learning aspects of the project, including clustering and prediction models:

- Clustering: The K-means clustering algorithm is evaluated to group similar ransomware types. Performance is assessed based on the Silhouette score and cluster purity.
- Prediction: Regression models are evaluated using mean squared error (MSE) and root mean squared error (RMSE) to measure the accuracy of attack volume forecasts.

5.6 Results

Evaluation Metrics:

- Cluster Quality: Using Silhouette Score for clustering validation.
- Forecast Accuracy: Using MSE and RMSE for prediction models.

Results and Analysis: The experimental results are analyzed to compare the performance of Flink vs. Spark in terms of:

- Latency and throughput
- Scalability with increasing data
- The effectiveness of machine learning models for ransomware pattern detection

This methodology ensures that both the real-time processing and long-term trend analysis of ransomware data are effectively implemented. The experiments evaluate the robustness of the system under different load conditions, assess the accuracy of the predictions, and determine the system's scalability for use in real-world applications.

5.7 System Deployment

The deployment of the ransomware analytics system was implemented as a modular pipeline using Docker containers to ensure scalability and ease of management. Data collection begins with the AlienVault OTX API, deployed as a containerized service, which fetches cyber threat intelligence data. The data is ingested in real-time into Apache Kafka, also running in a container, with messages stored in a dedicated topic (`indicators_topic`) for stream processing. Apache Spark, hosted in its own container, processes these streams, training models for both prediction (`RandomForestClassifier`) and clustering (`KMeans`) in batch mode. The processed analytics results, including predictions and cluster metadata, are stored in an InfluxDB database container across designated buckets (`ransomware`, `prediction`, `forecast`). Finally, the Dash visualization layer, deployed as a containerized web application, provides real-time insights, including Top 10 lists and trends, enabling effective monitoring and proactive threat response. This architecture ensures a seamless flow of data while maintaining flexibility and resilience.

6 References

References

1. Brewer, R.: Ransomware attacks: detection, prevention and cure. Network Security, 2016(9), pp.5–9. Available at: https://doi.org/10.1016/S1353-4858(16)30086-1 (2016)

2. Kreps, J., Narkhede, N., Rao, J.: Kafka: A distributed messaging system. In: LinkedIn Engineering Blog, pp. 1–2. LinkedIn (2011)
3. Databricks: Real-time data pipelines with Apache Spark, https://databricks.com, last accessed 2023/10/25
4. Ali, A., Akhtar, M., Raza, B.: Performance benchmarking of stream processing frameworks. SpringerLink (2017)
5. Chintapalli, S., et al.: Benchmarking Apache Spark and Flink. Databricks (2016)
6. Zimmer, J., et al.: A review of time-series analysis for cyber security analytics: from intrusion detection to attack prediction. International Journal of Information Security **24**(1), 1–20 (2024). https://doi.org/10.1007/s10207-024-00921-0
7. Liu, Z., Hu, H., Lee, S.Y., Park, J.H., and Han, W.: Forecasting cyberattacks with incomplete, imbalanced, and insignificant data. *Security and Communication Networks*, 2019, pp. 1–14. Available at: https://doi.org/10.1155/2019/329761473.
8. Patel, A., Jain, R., and Kumar, P.: Ransomware detection using machine learning: A review, research limitations, and future directions. *Journal of Cybersecurity Research*, 2023. Available at: https://doi.org/10.1155/2023/380408103.
9. IBM X-Force: Analysis of Ransomware Trends and Tactics. *IBM Security Intelligence*, 2023. Available at: https://securityintelligence.com/x-force/analysis-of-ransomware/.
10. Gao, J., Zhang, X., and Li, H.: Ransomware detection and prediction using hybrid machine learning techniques. *Electronics*, 2022, **11**(20), 3307. Available at: https://doi.org/10.3390/electronics11203307.