SRE Capstone

Jan 6, 2025

Real time log processing

Presented by Maxwell Benko, Alec Ippolito, Ben Kruczek

PNC
620 Liberty Ave, Pittsburgh PA
(888) 726-2265

01 High Level Summary

This project aims to accurately simulate real world server logs of web application attacks in a fast paced changing environment.

Overview

Using tools such as MongoDB, Apache kafka, mySQL, and Docker we will be analyzing data streaming logs pertaining to web application attacks. Raw data will be pulled from Apache Kafka logs and then be placed into a NoSQL database in Mongodb. The data will then be extracted and stored in a MySQL database inorder to create a more readable and useful database. This data will then be monitored and alerts will be made to notify the user if something is wrong. Everything will be containerized with docker.

Key features and capabilities

The software will be scalable such that it can handle large workloads and grow with them if attacks increase or decrease. Additionally, it must be modular such that its pieces can be monitored and updated. Finally, data will be stored efficiently and accurately such that the project is reliable.

Benefits

The advantage of being able to analyze streaming logs with the provided tools is that data will be read and manipulated very soon after it is created. This will simulate real world needs for any service that people may actually try to attack and bring down, allowing companies to avoid any loss.

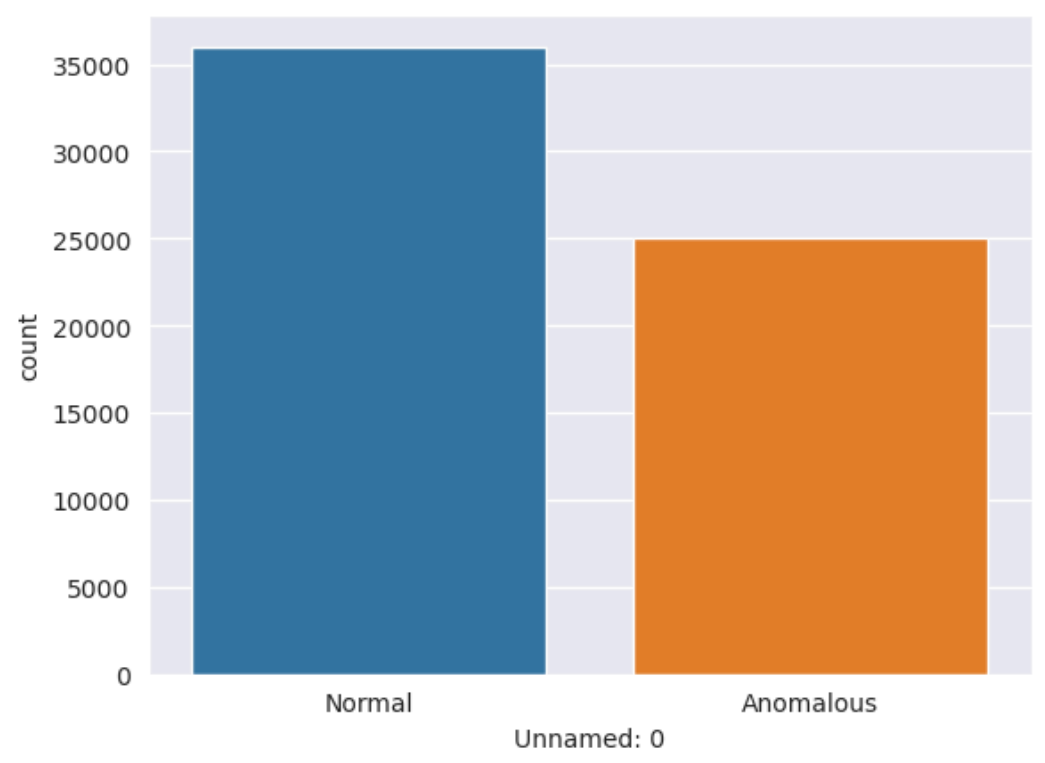
02 Dataset

CSIC 2010 web application attacks

Our data will be pulled directly from the website Kaggle using the following URL:

https://www.kaggle.com/datasets/ispangler/csic-2010-web-application-attacks

Each entry in the set of data describe whether each entry is normal or anomalous, the http method used, the user agent, pragma, cache control, accept, accept encoding, accept charset, language, and host.



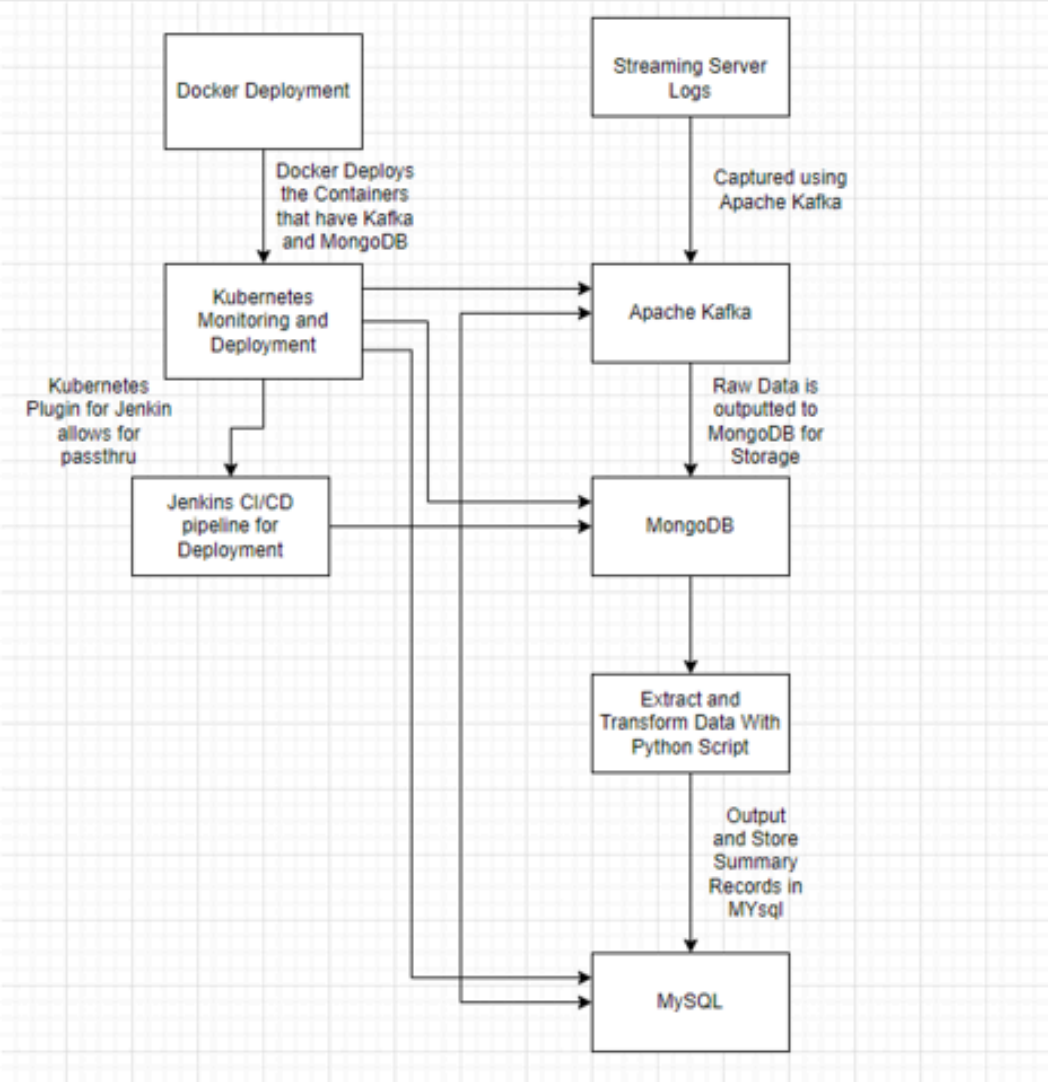
03

Project Goals

Our project has two main goals

1. To create a tool that can be used to monitor data pertaining to web application attacks such that they can be monitored and minimized.

2. To demonstrate our team's understanding of the interconnectivity and functionality of the tools and software that make up our project. This is outlined by the chart below.



04 Project Objective

Using the following list of tools, we will complete these objectives.

Tools

Linux, Python, MySQL, Site Reliability Engineering, Git, DevOps, CI/CD, Docker, MongoDB, Kubernetes, and Kafka.

Objective List

1. Create Docker and Kubernetes configuration files
2. Create instructions for environment setup
3. Create instructions for kafka and consumer setup
4. Create pymongo scripts to storing raw data in MongoDB
5. Create a Database schema for MongoDB
6. Create Python Scripts for data extraction in MongoDB
7. Create a Database schema for MySQL
8. Create scripts to add data to MySQL
9. Create a pipeline with automated deployment
10. Configure an alerting system
11. Comprehensive testing
12. Created Detailed documentation
13. Present project with slide deck

05

Project Scope

Who?

Data will be visible to all tools listed above and will only be a simulation. While the data being used is publicly available, the processed data and application will only be available to the local machine running it.

Where?

Data will be captured and streamed directly from the aforementioned Kaggle website.

When?

This project is intended to work with a data set captured in 2010; therefore, will be a simulation of live data. Data can be made to be streamed from the present assuming it is formatted correctly and a few configurations are changed.

06 Schedule

This is our4-week tentative schedule with weekly objectives.

Week 1: Planning, Environment Setup, and Log Streaming

- Planning
  - Architecture and Team Roles
  - Tools and Technology to be used
- Environment Setup
  - Install and configure Docker for Apache Kafka, MongoDB, and MySQL
  - Setup Kafka topics and test functionality
- Log Streaming:
  - Develop aKafka producer to simulate log generation
  - Create Kafka consumers to read logs and store them in MongoDB
  - Define MongoDB schema for raw logs and verify data ingestion

Week 2: Data Transformation, Summary Storage, and Initial Dockerization

- Data Transformation
  - Develop an ETL pipeline to extract logs from MongoDB and transform them
  - Create scripts for summarizing data
- Summary Storage
  - Define MySQL schema for summary records
  - Write scripts to insert transformed data into MySQL
- Dockerization
  - Write Dockerfiles for Kafka, MongoDB, MySQL and ETL pipeline
  - Test all components using Docker Compose

Week 3: Kubernetes Development, CI/CD, and Monitoring

- Kubernetes Development
  - Write Kubernetes manifests or helm charts for all services
  - Deploy Kafka, MongoDB, My SQL and ETL pipeline to a kubernetes cluster
- CI/CD Pipelines

    - Setup CI/CD pipelines for automated build, test and deployment
    - Use GitHub Actions or Jenkins for pipeline automation
- Monitoring
    - Setup Prometheus and Grafana for Service monitoring
    - Configure alerts for service health using Slack/Email notifications

Week 4: Testing, Optimization, Presentation

- Testing and Optimization
    - Perform end-to-end testing of the entire pipeline
    - Conduct performance and scalability tests/optimize for bottlenecks
- Documentation and Presentation
    - Finalize project documentation
    - Prepare a slide deck
- Demo Prep
    - Rehearse the presentation and live demo
    - Ensure all components are integrated and functioning well