Software Requirement Specification (SRS) Document

Project Name: Malicious DNS Traffic Detection using Deep Learning

Version : 1.0

July 9, 2024

Author:

B.Dheeraj Chandan
Abhishek Kumar Singh
Kothapalli Mounika
Manchi.Tejasai
Ankit Kumar

Revision History

Version	Date of Release	Pages Affected	Reasons for Change	Signature
1.0	05/07/2024	All	Initial Draft	Tejasai
1.1	06/07/2024	5	Parsing the documents	Mounika
1.2	09/07/2024	All	Incorporated team review comments	Dheeraj,Ankit
2.0	10/07/2024		First release	Abhishek

Table of Contents

Title of the project

Malicious DNS Traffic Detection using Deep Learning

Introduction

Overview : This document outlines the requirements, architecture, and implementation plan for the malicious DNS traffic detection system.

Purpose : The purpose of this project is to develop a deep learning-based system to detect malicious DNS traffic. This project aims to identify and mitigate cyber threats such as phishing, malware distribution, and botnet attacks by analyzing DNS traffic. To detect malicious sites by analyzing user login timeseries data, a systematic approach is required. By collecting detailed login data, including timestamps like date and time, user IDs and IPs ,User Queries .

Doing preprocess the data by cleaning invalid or normalizing timestamps, and creating additional features such as login hour and day of the week. For more sophisticated detection, apply anomaly detection algorithms.

Scope : The scope of this project includes the collection of DNS log files, parsing these logs to populate a time series database, and applying deep learning techniques to classify the traffic as malicious or benign. The project focuses on using unsupervised learning due to the unlabeled nature of the input data.

Definitions, Acronyms, and Abbreviations

- DNS: Domain Name System
- Deep Learning: A subset of machine learning involving neural networks with many layers
- Traffic Detection: The process of identifying network traffic patterns
- Malicious Traffic: Unauthorized or harmful network traffic

Intended Audience and Reading Suggestions

This document is meant for cybersecurity professionals and data analysts tasked with monitoring and protecting user accounts from unauthorized access. It provides guidance on using time-series databases, statistical analysis, and machine learning to detect and investigate suspicious login activities, ensuring robust security practices.

Project/Product Scope

The scope of this project involves developing a comprehensive system for detecting malicious sites by analyzing user login timeseries data.

References

1. **Research Papers: -** "Deep Learning for DNS Traffic Analysis" by John Doe et al. (2020)

- "Time Series Analysis and Its Applications" by Robert H. Shumway and David S. Stoffer (2017)

2. **Documentation:** **-** Influx Db Documentation: https://docs.influxdata.com/influxdb/
- TensorFlow Documentation: https://www.tensorflow.org/guide

Overall Description

Project/Product Perspective

The system is designed to integrate seamlessly with existing network security infrastructure, utilizing a time series database and deep learning models to detect malicious DNS traffic. Rapidly evolving digital landscape, securing user accounts against unauthorized access is critical for maintaining the integrity and trustworthiness of online services. Organizations face persistent threats from malicious actors who exploit vulnerabilities to gain unauthorized access, leading to potential data breaches and financial losses. This project addresses the urgent need for a sophisticated system that leverages user login timeseries data to detect and mitigate these threats effectively.

Objectives : Enhanced Security, Data-Efficient Data Management, Driven Decision Making, Real-Time Threat Detection, Comprehensive Anomaly Detection, Automated Alerts and Responses, Continuous Improvement and Adaptation,

Product Functions

The project encompasses several key functionalities aimed at detecting and preventing unauthorized access through the analysis of user login timeseries data. These functionalities ensure a comprehensive and effective approach to cybersecurity.

It includes : - Collection and parsing of DNS log files ,
- Populating a time series database (Influx DB),
Application of deep learning models for traffic classification

User Characteristics

Users include network security analysts and IT administrators responsible for monitoring and securing network traffic.

Constraints

- Unlabeled input data requires unsupervised learning techniques.

- Real-time processing demands high-performance hardware.

Assumptions and Dependencies

- Availability of DNS log files.

- Access to necessary computational resources (CPU, RAM, GPU).

Operating Environment

Server

Server OS:

Language: InfluxDB , Python, TimeScaleDB

Database: InfluxDB , TimeScaleDB

Application Server:

Tool: InfluxDB , VSCode(Visual Studio code), TimeScaleDB

Client: Design

Operating Environment Constraints: The system must operate effectively across diverse environments, including cloud-based, on-premises, and hybrid infrastructures. Compatibility with various operating systems (such as Linux, Windows) and web server configurations is essential to ensure seamless integration into existing IT ecosystems. The solution should also accommodate different network architectures and security protocols to maintain interoperability and data integrity.

Development Environment Constraints: Development constraints primarily involve resource availability and technology compatibility. Adequate computational resources are necessary for processing and analyzing large volumes of timeseries data efficiently. Compatibility with development tools and frameworks, such as Python for machine learning implementations, database systems like TimescaleDB or InfluxDB for data storage, and visualization libraries for presenting findings, must be ensured. Furthermore, adherence to cybersecurity best practices during development, such as secure coding standards and data encryption methods, is critical to safeguard sensitive information.

Navigating these constraints requires careful planning and consideration of scalability, performance, and security aspects throughout the project lifecycle. Addressing these challenges proactively ensures the successful deployment and operation of a robust system for detecting and mitigating threats posed by malicious sites.

User Documentation

The project "Malicious Fraud Detection" aims to deliver a comprehensive software solution designed to detect and prevent fraudulent activities through sophisticated analysis of user login timeseries data. This software will be accompanied by detailed documentation to ensure ease of use and effective deployment.

User Manual and Online Help Document :

The User Manual and Online Help Document will provide comprehensive guidance on utilizing the "Malicious Fraud Detection" software. It will include: A detailed introduction to the software's functionalities and objectives.

- User Interface: Instructions on navigating the user interface, understanding different modules, and accessing key features.
- Data Input: Guidance on how to input and manage login timeseries data for analysis.
- Anomaly Detection: Steps to interpret detected anomalies, investigate flagged activities, and take appropriate actions.
- Reporting: Instructions on generating reports summarizing detected fraud instances and login activity trends.
- Troubleshooting: Common issues and troubleshooting steps for smooth operation.

Installation Manual

The Installation Manual will provide clear instructions for installing and configuring the "Malicious Fraud Detection" software environment. It will cover: System Requirements like hardware and software prerequisites needed for installation. Installation Steps: Step-by-step instructions for installing the software on different operating systems. Configuration: Guidelines for configuring database connections, setting up necessary dependencies, and ensuring system compatibility. Testing: Procedures for verifying the installation and conducting initial tests to ensure functionality.

By providing these manuals alongside the software, users will have the resources needed to effectively deploy, utilize, and maintain the "Malicious Fraud Detection" system, empowering organizations to proactively safeguard against fraudulent activities and enhance overall security measures.

External Interface Requirements

User Interfaces

- User Interface: Dashboard for monitoring and managing alerts.
- API Integration: Interfaces for data ingestion and alert management.

Hardware Interfaces

The Proposed system configuration is as follows:

Sl. No.	ITEM	Server 1	Server 2
01	Processor	Ryzen 5 /Intel core 5	
02	No., of processor	8	
03	Memory	32GB	
04	HDD capacity	500GB	
05	Network		
06	USB ports		
07	OS	Window,Linux	

System Features

- Data Visualization: Graphical representation of DNS traffic patterns.
- Alert Management: Generation and management of alerts for detected malicious traffic.

Non functional requirements

Non-functional requirements specify criteria that describe how the system should behave, rather than what it should do. For the "Malicious Fraud Detection" project, these requirements ensure that the software meets certain performance, usability, security, and maintenance standards.

Performance Requirements : DNS traffic efficiently.Scalability: System should be scalable to accommodate growing network sizes.

Usability Requirements : User Interface , Accessibility, ocumentation
Security Requirements : Data Encryption, Access Control:, Audit Trail.
Reliability Requirements : Availability, Fault Tolerance, Backup and Recovery
Maintainability Requirements : Modularity, Documentation, Version Control

Acceptance criteria

Acceptance criteria define conditions that a system must meet to be accepted by stakeholders. For the "Malicious Fraud Detection" project, the acceptance criteria might include:

1. **Functional Requirements**: Ensure all specified functionalities (e.g., login anomaly detection, real-time monitoring, reporting) work as intended.
2. **Performance**: System performance meets or exceeds defined thresholds for response time and throughput.
3. **Usability**: Users find the interface intuitive and are able to navigate and use the system effectively.
4. **Security**: All security measures (e.g., encryption, access control) are implemented correctly and tested against potential vulnerabilities.
5. **Reliability**: The system operates continuously without significant downtime or data loss.
6. **Scalability**: The system scales effectively to handle increased data volume and user load.
7. **Compliance**: Ensure the system complies with relevant regulations and standards (e.g., GDPR, HIPAA) regarding data privacy and security.

By adhering to these non-functional requirements and acceptance criteria, the "Malicious Fraud Detection" project aims to deliver a robust, secure, and user-friendly solution for detecting and preventing fraudulent activities through advanced analysis of user login timeseries data.

Deliverables

Here is a list of deliverables typically associated with the "Malicious Fraud Detection" project:

Software Application: It includes User interface for monitoring login activity and anomalies. Backend processing for data analysis and anomaly detection. Integration with time-series databases (e.g., TimescaleDB or InfluxDB).

User Manual: It provides overview of the software functionalities and objectives, User interface navigation instructions, Steps for data input, anomaly detection, and investigation , Troubleshooting tips and FAQs.

Online Help Document: It Interactive online help system offering the Contextual assistance and guidance within the software interface, Detailed explanations and tips on using specific features, Links to relevant sections of the user manual for deeper exploration.

Installation Manual: It gives us dtailed guide for System requirements (hardware and software), Step-by-step installation instructions for different environments (e.g., Windows, Linux), Configuration setup (database connections, dependencies), Initial testing procedures to ensure correct installation.

Reporting Module: There are component for generating regular reports summarizing detected fraud instances and analysis of login activity trends and anomaly patterns.

Dashboard: It consistes of real-time visualization of Login activity metrics. Detected anomalies and alerts.

Integration APIs: It is an application Programming Interfaces (APIs) for Integration with existing security systems or workflows. Data export functionalities.

Testing and Validation Documentation: It is a report on testing procedures conducted during development, Validation of software functionalities against use cases and expected outcomes.

Support and Maintenance Plan: Documentation outlining which includes support channels (e.g., help desk, email) and Maintenance schedules and procedures for updates and patches.

Project Documentation: It contains compilation of Project plan and timeline, Requirements analysis and design documentation and Any additional technical documentation produced during development.

These deliverables collectively ensure that the "Malicious Fraud Detection" project meets its objectives of providing a robust and user-friendly solution for detecting and preventing fraudulent activities through advanced analysis of user login timeseries data.