****

**School of Computing Science and Engineering**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Code:** | | | **Project work Synopsis**  **(Phase 1 Review 1)** | | | **Academic Year:** | | |
| **Semester: 7th** | | |
| **Project Group Members Details** | | | | | | | | |
| **Group No:** | | **Group Name:** | | | | | | |
| **Sl. No** | **SRN** | **Full Name** | | **Sec** | **Mobile No.** | **Email\_ID** | | **Sign** |
| **1** | R20EF402 | CHANDAN M S | | H | 6363224034 | chandanms1312@gmail.com | |  |
| **2** | R20EF414 | MOHAMMAD FAWAZ BEEJADY | | H | 6366926710 | mohammadfawazbeejady@gmail.com | |  |
| **3** | R20EF436 | SUMAN S | | H | 9380899173 | sgrsuman90@gmail.com | |  |
| **4** | R20EF424 | ROSHAN B S | | H | 6361594870 | r20201179.roshanbs@cse.reva.edu.in | |  |
| **Task Distribution** | | | | | | | | |
| **Project Leader** | | CHANDAN M S | | | | | | |
| **Documenter Lead:** | | MOHAMMAD FAWAZ BEEJADY | | | | | | |
| **Development Lead:** | | SUMAN S | | | | | | |
| **Project Details** | | | | | | | | |
| **Project Title:** | | CLOUD-BASED INTRUSION DETECTION SYSTEM | | | | | | |
| **Type of Project** | |  | | | | | | |
| **Guide Details:** | | **Name:** SOUJANYA B K  **Designation:** ASSOCIATE PROFESSOR | | | | | **Mobile No:** 9448728458 | |
| **Place of Project Work:** | | REVA UNIVERSITY | | | | | | |
| **External Guide:** | | **Name:**  **Designation:** | | | | | **Mobile No:** | |
| **Remarks by Guide:** | | | | | | | **Guide Signature**  **Date:** | |

**Contents**

Abstract

1. Introduction
2. Literature Survey
3. Positioning
   1. Problem statement
   2. Product position statement
4. Stakeholder Descriptions
   1. User stakeholders
5. Project overview
   1. Objectives
   2. Goals
   3. Feasibility Study
   4. Alternatives
   5. Budget
   6. Key deliverables
   7. Necessary materials
   8. Methodology
   9. Modules identified.
6. Conclusions
7. References

**Abstract**

As organizations increasingly migrate their critical assets and services to cloud infrastructures, the need for effective intrusion detection systems (IDS) that can adapt to the dynamic and distributed nature of the cloud environment becomes paramount. This abstract introduces the concept of a cloud-based IDS, which harnesses the power of cloud computing to enhance threat detection and response capabilities. Unlike traditional network-based IDS solutions, cloud-based IDS leverages cloud resources, machine learning algorithms, and extensive data analysis to detect and mitigate both known and emerging threats in real-time.

1. **Introduction**

An Intrusion Detection System (IDS) is an essential component in ensuring the security of computer systems and networks. Its core function involves the continuous monitoring and analysis of system events to swiftly identify and alert about different attacks including Automated Brute Forcing on web-based login, HTTP flood attacks, SQL injections (SQLi), and Cross-Site Scripting (XSS). In the domain of web security, protocol-based intrusion detection systems are particularly indispensable. These specialized IDSs, typically deployed on web servers, are designed to exclusively scrutinize, and analyze the influx of HTTP or HTTPS requests presented in a stream-like format. Intrusion detection methods are broadly categorized into three primary approaches: *Signature/Heuristic-based Detection*, *Anomaly-based Detection*, and *Distributed/Hybrid-based Detection.* Signature-based detection relies on predefined patterns or rules to identify known malicious behavior, although its effectiveness hinges on the comprehensiveness of the signature collection, rendering it less potent against novel intrusion patterns. In contrast, anomaly-based detection leverages statistical methods or machine learning techniques to discern legitimate requests from intrusion attempts, offering more adaptability but potentially impacting system performance. Distributed/Hybrid-based detection fuses the strengths of both methods, resulting in higher identification rates and faster response times. The implementation of an IDS in a cloud environment presents unique roadblocks due to the presence of numerous data formats and massive data volumes it must process. Beyond the reliability of the detection mechanism, this demands a consideration of system performance, scalability, real-time responsiveness, and several other key factors. Current cloud-based IDS systems deal with high data volumes by deploying on networked virtual machines. In this research paper, we present an innovative solution designed to efficiently process raw data by utilizing the capabilities of Kafka and Spark Streaming. Our IDS system seamlessly integrates with these data processing tools to enhance processing capabilities and accelerate the detection of various attacks. It leverages Kafka to receive streaming data and directs it to separate Spark Streaming jobs that detect specific intrusions such as Automated Brute Forcing, HTTP flood attacks, SQLi, and Cross-Site Scripting. This research aims to enhance detection capabilities, improve response times, and efficiently process high data volumes, ultimately ensuring the security of cloud-based systems.

1. **Literature Survey:**

In 2020 Hai and Khiem [10] proposed architectures for processing IDS logs using Spark Streaming. The focus is to improve the performance of Network Intrusion Detection Systems (N-IDS) by using distributed processing and parallel computing with Apache Spark. Three distributed computing models are compared with each other, being: Distributed Snort, HBase & Impala, and HBase & Impala with Additional VMs. The computational efficiency is best when HBase and Impala are implemented with additional VMs. This inspired the usage of using a distributed system to efficiently process the great number of requests that are poured into the system.

Recently, Elmasry, Akbulut, and Zaim [11] proposed a design of integrated cloud-based intrusion detection systems (CIDS) using third-party cloud service. This design integrates the modules that are used in an IDS to be migrated to a third-party cloud environment, where the operations of monitoring,

processing, analysis, prediction, and response are accomplished on the cloud. Elmasry’s, Akbulut’s, and Zaim’s modules have been partially implemented and mapped into this paper’s proposed system. The Monitoring Module is depicted as the Kafka broker, which receives data streams from four producers depending on the HTTP request method. The Processing and Analysis Modules are mapped to the four Spark Jobs which contain algorithms to detect intrusions. The final module this paper implement is the Alert module which consists of an embedded Vonage API call for each Spark Job accompanied by the logs that are saved on a dedicated Google Cloud Storage bucket.

Debnath et al [13] have conceived a model for Real-time Log Analysis System called LogLens. The Dynamic Programming Algorithm is used to detect abnormal log sequences of an event or transaction. This algorithm goes through different parts of a log’s entry and checks important parameters against a deterministic value, string, pattern, or Wildcard. This idea is applied in this paper’s algorithm for detecting SQL injections and cross-site scripting, where collections of Wildcards are used to compare patterns or signatures that fall under each category.

1. **Positioning**

**3.1. Problem statement:**

People use web technology for wide range of purposes such as information access, entertainment, communication etc. However, these web services are implemented on REST, a common web services protocol. In REST, each HTTP request is mapped to GET, POST, PUT, DELETE which have been proven to be prone for various intrusion attacks such as Automated brute force attacks, HTTP flood attacks, SQL injections (SQLi) and Cross-site Scripting (XSS). To this end, we propose a C-IDS (Cloud-based Intrusion Detection System) using Apache Kafka and Spark Streaming. This C-IDS will offer real-time threat detection and response with scalability to handle large volumes of access requests, efficient data processing and analytics, and the ability to integrate with a wide range of data sources for comprehensive security monitoring.

**3.2. Product position statement:**

With everything in today’s world going digital, it is no doubt that while we enjoy these services seamlessly, there presents a significant risk of cybersecurity and intrusions. Therefore, it is important to have a robust security system in place which is real-time to near-real-time, provides quick response, protects your data, and have a peace of mind, where you don’t have to worry about your data security and focus on your core business objectives. Our C-IDS offers such a service where It empowers organizations to proactively identify and prevent security threats, safeguarding their critical cloud-hosted applications and data. Our product sets itself apart with its scalable, real-time threat detection, and comprehensive coverage across various cloud environments. It empowers organizations to embrace the cloud with confidence.

1. **Stakeholder Descriptions**

**4.1. User stakeholders:**

1. Security Teams: - They are the major beneficiaries as they can detect any unauthorized intrusions and respond in real-time to near-real-time.
2. Network Administrators: - They can inspect all the network traffics and view any potential security risks.
3. IT Operations: - IT teams and operational staff benefit from the intrusion system by having access to tools and data that can help improve network performance and stability.
4. Legal and Privacy Teams: Legal and privacy teams may benefit from the intrusion system's capabilities when dealing with incidents involving data breaches or privacy violations.
5. Customers and End Users: In the broader context, cloud-based intrusion systems indirectly benefit customers and end users by helping to protect their data and privacy.
6. Third-party Service Providers: If an organization relies on third-party cloud service providers to host and manage the intrusion system, these providers are beneficiaries as they contribute to the system's operation and security.
7. Vendors and Suppliers: Vendors and suppliers who provide hardware, software, or services related to the intrusion system benefit from its usage and may have ongoing partnerships with the organization.

**5. Project overview**

**5.1. Objectives:**

*1. Design and Develop a Cloud-Based IDS:* Create a scalable and efficient cloud-based IDS that can process and analyze large volumes of incoming HTTP and HTTPS requests.

*2.* *Real-Time Detection:* Ensure that the IDS provides real-time or near real-time alerts and warnings for potential intrusion attempts, minimizing response time.

*3*. *High Detection Accuracy:* Implement signature-based detection for known patterns of malicious data and anomaly-based detection using machine learning to enhance the system's detection accuracy.

*4.* *Scalability and Performance:* Design the system to handle high volumes of data while maintaining high performance, low resource consumption, and minimal downtime.

*5.* *Coverage of Intrusion Methods:* Ensure that the IDS can detect a wide range of intrusion methods, including Automated Brute Forcing, HTTP flood attacks, SQL Injections (SQLi), and Cross-Site Scripting (XSS).

*6.* *SMS Alerting:* Integrate an SMS alerting system to notify administrators or relevant personnel in real-time when an intrusion attempt is detected.

*7.* *Research and Innovation:* Explore new methods for detecting intrusion patterns, such as the approach for filtering SQL injection payloads with consideration for spaces.

**5.2. Goals:**

1. Successfully implement a cloud-based IDS using Apache Kafka and Spark Streaming.

2. Achieve a high detection rate with low false positives and false negatives.

3. Provide real-time or near real-time alerts for intrusion attempts.

4. Develop and test detection algorithms for Automated Brute Forcing, HTTP flood attacks, SQL Injections, and Cross-Site Scripting.

5. Ensure the system can handle a high volume of incoming data while maintaining performance.

6. Implement an SMS alerting system for notifying administrators.

7. Explore innovative methods for intrusion pattern detection and prevention.

**5.3.** **Feasibility Study**:

* *Tools and Technologies*: The technology stack, including Apache Kafka, Spark Streaming, and Google Cloud, is readily available and well-documented, making it suitable for the project.
* *Scalability:* The system is designed to handle high volumes of user inputs efficiently and is scalable to accommodate future growth.
* *Cost Analysis*: All the technologies used in developing the system are either free to use or very cost effective.
* *Return on Investment (ROI)*: The proposed system offers appreciable return on investment in the form of security enhancements and performance improvement.
* *System Reliability*: The proposed system reliably detects and responds to attacks without significant false positives or false negatives.

**5.4. Alternatives**:

*1.* *Use Existing Cloud-Based IDS Solutions:* Instead of developing a custom system, consider using existing cloud-based IDS solutions. This option may be quicker to implement but might not provide the level of customization and specific features required for your organization’s needs.

*2. Choose Different Data Processing Tools:* While Apache Kafka and Spark Streaming are excellent choices, other data processing tools could be considered, depending on specific project requirements and expertise.

*3. Focus on a Single Detection Method:* Rather than combining signature-based and anomaly-based detection, the project could focus on one method to simplify the system. However, this might result in lower detection accuracy.

*4. Outsource Development:* Instead of developing the system in-house, you could outsource the development to a specialized cybersecurity company. This might save time and resources but could be more costly.

**5.5. Budget:**

1. Cloud *Services*: Costs for cloud resources such as VMs, storage, servers etc.

*2. Software and Licensing*: open source/minimal cost software.

*3. SMS Alerting Service:* Costs for SMS alerting services used to send notifications.

**5.6. Key deliverables:**

The key deliverables for the proposed system are –

* **Cloud Based IDS:** The main deliverable is the implementation of a cloud-based Intrusion Detection System (IDS) capable of monitoring and evaluating REST architecture-based web services to various attacks.
* **Integration with Apache Kafka and Spark Streaming:**The proposed IDS is seamlessly integrated with Apache Kafka and Spark Streaming for efficient attack detection and response processing.
* **Attack Detecting Algorithms:**The proposed system implements robust attack detection algorithms such as HTTP flood attacks, SQLi, Cross site scripting detection.
* **Alerting System:**The proposed system implements an alerting system capable of generating SMS alerts in real-time when an intrusion attempt is detected.

**5.7. Necessary materials:**

*1.Cloud Infrastructure:* Required resources such as VMs, servers etc. from a cloud provider: Google Cloud Platform.

*2.Operating System:* Windows, Linux.

*3. Apache Kafka:* Set up Apache Kafka clusters, which consist of brokers and Zookeeper (for managing Kafka metadata).

*4. Spark Cluster:* Deploy a Spark cluster to process the streaming data. This can include Spark workers and a master node. The worker and the master node can be one.

*5.Kafka Producer:* To ingest data from network sensors or agents, you'll need a Kafka producer. This can be a component of your intrusion detection system that sends data to Kafka topics.

*6.Kafka Topics:* Organize data within Kafka topics. You may need to configure topics for different types of data, such as logs and alerts.

*7.Spark Streaming Job:* Develop Spark Streaming applications to consume and process data from Kafka topics. This job should include real-time analysis, alerts generation, and response mechanisms.

*8.Data Storage:* You might need a database or data storage solution to store processed data, logs, and historical data for analysis and reporting. Provided by cloud providers.

*9.Visualization and Dashboards:* For generating reports for analytical purposes.

*10. Alerting and Notification System:* Set up an alerting and notification system to send alerts to security personnel when suspicious activities are detected.

**5.8. Methodology:**

Here is a methodology for designing and developing a cloud-based intrusion detection system (IDS) using Apache Kafka and Spark Streaming:

*1. Requirements gathering:* The first step is to gather requirements for the IDS system. This includes understanding the types of attacks that need to be detected, the desired performance and scalability requirements, and the budget and timeline constraints.

*2. System design:* Once the requirements have been gathered, the next step is to design the IDS system. This includes selecting the appropriate components and technologies and designing the overall architecture of the system.

*3.Development:* Once the system design is complete, the next step is to develop the IDS system. This includes implementing the different modules of the system, such as the data collection and ingestion module, the data preprocessing module, the anomaly detection module, the attack classification module, and the alerting and logging module.

*4.Testing:* Once the IDS system has been developed, it is important to test it thoroughly to ensure that it is working as expected. This includes testing the system's ability to detect and classify attacks, as well as its performance and scalability.

*5.Deployment:* Once the IDS system has been tested and verified, it can be deployed in production. This may involve deploying the system on-premises, in the cloud, or a combination of both.

*6.Monitoring and maintenance:* Once the IDS system is deployed, it is important to monitor its performance and maintain it regularly. This includes monitoring the system for false positives and false negatives, and updating the system's attack patterns and machine learning models as needed.

Here are some additional considerations for designing and developing a cloud-based IDS using Apache Kafka and Spark Streaming:

* **Data collection and ingestion:** 
  + The type of network traffic data that needs to be collected and ingested.
  + The volume of network traffic data that needs to be collected and ingested.
  + The latency requirements for collecting and ingesting the data.
* **Data preprocessing:** 
  + The types of preprocessing that need to be performed on the data.
  + The volume of data that needs to be preprocessed.
  + The latency requirements for preprocessing the data.
* **Anomaly detection:** 
  + The types of anomalies that need to be detected.
  + The desired accuracy and performance of the anomaly detection algorithm.
  + The type of machine learning algorithm that will be used for anomaly detection.
* **Attack classification:** 
  + The types of attacks that need to be classified.
  + The desired accuracy and performance of the attack classification algorithm.
  + The type of machine learning algorithm that will be used for attack classification.
* **Alerting and logging:** 
  + The types of alerts that need to be generated.
  + The destinations for the alerts (e.g., email, SMS, Slack).
  + The types of logs that need to be generated.
  + The destinations for the logs (e.g., database, cloud storage).

By following these steps and considerations, you can design and develop a cloud-based IDS using Apache Kafka and Spark Streaming that is effective, scalable, and reliable.

**5.9.** **Modules identified:**

The following modules are identified in the design and development of a cloud-based intrusion detection system (IDS) using Apache Kafka and Spark Streaming:

* **Data collection and ingestion module:** This module is responsible for collecting and ingesting network traffic data into Apache Kafka. This can be done using a variety of methods, such as using a network tap, mirroring a switch port, or using a software agent.
* **Data pre-processing module:** This module is responsible for pre-processing the network traffic data before it is fed into Spark Streaming. This may involve tasks such as cleaning the data, converting it to a consistent format, and extracting relevant features.
* **Anomaly detection module:** This module is responsible for detecting anomalous traffic patterns that may indicate an attack. This can be done using a variety of machine learning algorithms, such as support vector machines (SVMs), decision trees, and random forests.
* **Attack classification module:** This module is responsible for classifying detected anomalies as specific types of attacks. This can be done using a variety of machine learning algorithms, such as SVMs, decision trees, and random forests.
* **Alerting and logging module:** This module is responsible for generating alerts and logging events when attacks are detected. This may involve sending alerts to security personnel, logging events to a database, or blocking malicious traffic.

These modules can be implemented in a variety of ways, but a typical deployment would involve using the following components:

* **Apache Kafka:** Apache Kafka is a distributed messaging system that can be used to ingest and process high-volume streams of data.
* **Spark Streaming:** Spark Streaming is a library that enables Spark to process real-time streams of data.
* **Machine learning library:** A machine learning library such as scikit-learn or TensorFlow can be used to implement the anomaly detection and attack classification modules.
* **Alerting and logging system:** An alerting and logging system such as Nagios or Splunk can be used to generate alerts and log events when attacks are detected.

The following is a high-level overview of how the different modules interact with each other:

1. The data collection and ingestion module collects and ingests network traffic data into Apache Kafka.
2. The data pre-processing module pre-processes the network traffic data before it is fed into Spark Streaming.
3. Spark Streaming reads the pre-processed data from Apache Kafka and processes it in batches.
4. The anomaly detection module detects anomalous traffic patterns in the data.
5. The attack classification module classifies detected anomalies as specific types of attacks.
6. The alerting and logging module generates alerts and logs events when attacks are detected.

**6. Conclusions:**

In conclusion, the design and development of a cloud-based IDS using Apache Kafka and Spark Streaming represents a powerful and modern approach to enhancing network security and threat detection. This combination of technologies leverages the scalability, real-time processing capabilities, and data streaming functionality required to effectively monitor and respond to potential security breaches.

**7. References:**

1. <https://ieeexplore.ieee.org/document/9836264>
2. <https://www.researchgate.net/publication/362328508_Design_and_Development_of_A_Cloud-Based_IDS_using_Apache_Kafka_and_Spark_Streaming>
3. <https://www.alibabacloud.com/blog/building-a-cloud-native-feed-streaming-system-with-apache-kafka-and-spark-on-alibaba-cloud-part-b-streaming-processing_596925>
4. [https://aws.amazon.com/blogs/big-data/real-time-stream-processing-using-apache-spark- streaming-and-apache-kafka-on-aws/](https://aws.amazon.com/blogs/big-data/real-time-stream-processing-using-apache-spark-          streaming-and-apache-kafka-on-aws/)
5. <https://www.sciencedirect.com/science/article/pii/S0965997821000582>