ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 3, March 2024

Fraud Management Framework for Data Streams

[1] Niravkumar Patel

[1] University of the Cumberland, Williamsburg, Kentucky
Corresponding Author Email: [1] nirav10012@g.com

Abstract— ***Fraud prevention is an important capability to detect unlawful transactions and minimize the losses due to unlawful transactions. All companies and banks want their framework and infrastructure secure for money transfer in this activity. The challenge of controlling fraud has increased substantially, demanding fraud analysts with adequate mechanisms for defining fraud policies and system architecture for large-scale real-time streaming applications arising from multiple channels at different windows. In this paper, I describe a fraud management framework, including a rule-based financial model language for the conceptual level of modeling and validation for fraud policies, fraud prevention techniques, and implementation of fraud policies using stream SQL. The critical element in streaming is detecting fraud proactively, blocking the transactions encompassing suspicious click stream patterns. This framework is designed for multi-transaction processing and micro-transaction processing.***

I. INTRODUCTION

Financial institutions provide consumers with a diverse range of banking and money transfers into saving, checking, and investment accounts through electronic delivery channels, including ATM, card cheques and online banking technology. While this provides a highly accessible and desirable offering for consumers, numerous shortcomings in implemented systems present critical security vulnerabilities enabling malicious third parties. Fraud prevention is required in all companies with secure infrastructure applications, algorithms, techniques, tracing, reporting, fraud prevention criteria, and procedures. If the transaction is processed electronically into account, then we should have the correct mechanism to be implemented. The system presents critical security. If we use any third-party API or application for the transaction, we should authenticate properly with encryption and decryption algorithms. Fraud prevention is a technique with implemented security and protocol mechanisms to block transactions that reflect suspicious attacks on the application or transactions.

II. PROCEDURE FOR PAPER SUBMISSION

Financial institutions have now recognized that the application is isolated with secure infrastructure to delivery channels. Secure protocol algorithms will block unauthorized account activity. It will be delivering the packet data securely to the destination. We can design the secure infrastructure framework with multi-level integrated security management across all delivery channels. Fraud policies are designed for a series of complex data stream evolution functions and trigger preventive action, which can be implemented appropriately using platforms supporting the event conditions action model. It is identified that proactive systems are designed not as a replacement for data mining techniques but as complementary techniques for integrating extracted data into a real-time fraud management system. Proactive approaches advance fraud detection capabilities of reactive systems by shifting data processing from past and pre-data storage. When the transaction starts with a wired or wireless device, the security protocol has a secure protocol to track all the transactions from end to end. All the transactions to be analyzed with an authenticated user in the application. If we find some unrelated transaction, we should alert it and try to find out the suspicious activity. All the events must be traceable, and log the event a long time.

Reactive fraud management: The analytical method is data mining. Which are implemented to perform algorithmic processing and complex calculations on the stored transactional data. Fraud instances are identified against predefined fraud patterns libraries or transaction processing with unknown data sequences for future detection. If the request is coming from outside the network, we should have some secure firewall configuration before processing the request and routing it to the destination server. All the proposals need to be tested and verified from the source and authenticated as per the user credentials. If we find an unknown request, those requests will be diagnosed with how the request is coming, which is the ip address, location, country, type of the request, etc.

Proactive fraud monument: Incoming transaction data is evaluated against predefined fraud management policies. All the fraudulent data patterns before being placed. The design tries to identify the fraud transaction pattern and decline the transaction. The preventive action is triggered before transaction completion and any associated monetary value movement.

If the pattern finds fraudulent activity, then all the transactions must be held until the request is verified and authenticated as per the user credentials and accounts. Proactive fraud detection is achieved by relocating the security component to enable the real-time component of incoming data. Several techniques like two-factor

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 3, March 2024

authentication and one-time passwords may be triggered in suspicious cases while suspending accounts.

III. MATH

Multi-channel processing continues the function of incoming transactional data. It prevents additional data from being available within streaming events channels until the fraud management fragment can alert fraud over multiple incoming data channels.

IV. FINANCIAL DEFINITION USING FFML RULES

FFML provides a dedicated vocabulary of the language construct and operators to facilitate policy definitions and administration between fraud analysis and target fraud management environments. Systema, knowledge, execution, and management models were designed for structural, evaluation, and managerial language aspects, including scheduling coupling and logical operator components. Structural and primitive elements are intended with modeled data, demonstrating the challenge of defining continuous queries utilizing stored and streaming data over multiple system channels. If any atm withdrawals reach the total daily withdrawal limit, then take action to block the transaction after five days.

V. FFML POLICY PROCESSING FRAMEWORK

FFML management framework was designed to accommodate the complexity of the rules, boundaries for transactions, code standards, algorithms, and security protocols to achieve fraudulent activity and alerting. FFML management frameworks were developed for front-end applications to enable graphical and code-based Definitions of fraud management policies through the syntactical properties of the data constructed FFML language. Processing of the constructed FFML language. A policy set is to be performed to ensure that conflict and non-termination between defined policies are provided. Location of the FFML system between implemented data mining tools and target fraud management environment provided.

VI. IMPLEMENTATION OF FFML POLICIES IN STREAM SQL

This is the process to implement the FFML construct, which can be mapped and implemented using the stream SQL formalism by converting FFML policies. Stream SQL will provide an extension upon standard SQL for issuing queries over both stored and streaming data with unrestricted constraints in streaming processing expression language. This mapping can give a decomposition of defined FFML policies and implementation of the analogous stream constructs. Where store data values are required in memory table are utilized to represent external data store containing account and transactional information.

Streamsql is the decomposed component into similar queries the preceding day for which data is required, casting the return values into local variables. Streamsql date functions are exploited, enabling the deduction of the required data as simple functions on the date field associated with incoming stream data. All incoming transaction requests that exceed the allocated daily threshold for the current day and satisfy the condition on two or multiple occasions within the last four days are filtered as fraud, triggering account suspension and output to the appropriate alert system.

VII. FRAMEWORK PROCESS

The framework provides a policy definition model for mapping and implementing the complex fraud policy set in financial platforms. Core benefits for fraud analysts adopting the FFML language detailed modeling of fraud policy definitions through conceptual level construct. The application code has been developed with a security protocol to identify all the requests and tracing mechanisms in the stream SQL. All the request to be analysis before processing with correct algorithms and formula. There are several phases substantially executed simultaneously. A multi-thread application in a concurrent execution process supports the application. Policies must be validated before processing the transaction, like payment fraud detection. Transactional data processing enables real-time prevention of suspicious transactions outside the account holder's spending profile. The real-time system has an alert mechanism to send the alert and verification process to the customer to identify the genuine users. Event data related to transactions being available and examined before processing the request. Manual investigations process to be included in this phase for fraud detection process.

Stream-based visual API will be designed in real-time processing of incoming event stream data from various input sources. It implements server technology that maximizes the use of in-memory data structures and supports continuous queries over incoming event stream data capable of rule evolution. All the incoming events are to be stored in the data values. Streamsql is the underlying technology developed through academic research into stream processing. Streamsql extends standard SQL by providing additional stream-oriented constructs to support issuing continuous queries over incoming data. The system must be constructed by manually assembling the graphical data construct and functions relating to the associated stremsql component.Stremsql extends standard SQL by providing additional strem-oriented constructs to support the issuing of continuous queries over all the incoming requests. We should have an event tracing mechanism to trace all the events in the stream and filter via query language processing.

The business signatures process continuously monitors all the stream HTTP data to build a profile of each customer's online behavior, access and navigational pattern to be designed, which is used to detect fraudulent or abnormal behavior. Transactions outside of a customer's usual signature trigger further challenges to authenticate the initial user as genuine account assessment, preventing and invocation of immediate response activity. The process of the code base application to be categorized in the different process of code, authentication process, fraud detection process, alerting, event stored data, stream sql queries processing in large data volumes. In this paper, I have represented all the phases of the fraud detection process and implementation of the fraud detection application—the secure application designed with correct algorithms to process the transaction and events effectively. The output of the data from the implemented function indicates satisfaction with the specified condition. It triggers the defined if clause, which leverages the underlying logic for evolving incoming system data. Implementation of policy functionality is divided on the required external data source functions. Calculate the necessary signature figure and retrieve the account threshold against which to evaluate the derived value. The stream-based Java function is implemented as an external data mining-based function responsible for customer profiling of accounts. The distance of the requested transaction from the conventional account behavior is calculated and returned the value evaluated against the matching customer and returned value evaluated against the matching customer threshold to establish if the defined then Claus must be triggered. Related attributes with IF and Then fields are first mapped to their corresponding INPUT STREAM and OUTPUT STREAM statements. Join incoming data across account information and date values to extract the required figure from the withdrawalist data source to be added to the value of incoming withdrawal request to create value Q1.

FFML provides a complex vocabulary of the domain-specific construct and operators for the proactive financial policy sets Definition. Desired system functionality is implemented from a single modeling perspective and provides an integrated policy definition approach leveraging analysis over multiple systems. All policy sets to be implemented in a standard stream expression language algorithm process of defined FFML structure into the semantics of the target technique provide a highly efficient mapping process. Streamsql is event processing. While the event occurs in the database, the event will be pushed in the queue, and all the events will be analyzed and stored in the event tracing tool. Event streaming is tracking all the events and publishing for processing. The process of the events it's based on the application infrastructure and code base before processing the event. All the events are to be validated per the company rules and examined with secure protocols. HTTP requests are to be tracked per data request and processed via a secure API to the backed system process. The query optimization process reduces the query and performance of the transactions.

VIII. PUBLICATION PRINCIPLES

The contents of the journal are peer-reviewed and archival. The journal INTERNATIONAL JOURNAL OF ENGINEERING AND INNOVATIVE TECHNOLOGY (IJEIT) publishes scholarly articles of archival value as well as tutorial expositions and critical reviews of classical subjects and topics of current interest.

Authors should consider the following points:

1) Technical papers submitted for publication must advance the state of knowledge and must cite relevant prior work.
2) The length of a submitted paper should be commensurate with the importance, or appropriate to the complexity, of the work. For example, an obvious extension of previously published work might not be appropriate for publication or might be adequately treated in just a few pages.
3) Authors must convince both peer reviewers and the editors of the scientific and technical merit of a paper; the standards of proof are higher when extraordinary or unexpected results are reported.
4) Because replication is required for scientific progress, papers submitted for publication must provide sufficient information to allow readers to perform similar experiments or calculations and use the reported results. Although not everything need be disclosed, a paper must contain new, useable, and fully described information. For example, a specimen's chemical composition need not be reported if the main purpose of a paper is to introduce a new measurement technique. Authors should expect to be challenged by reviewers if the results are not supported by adequate data and critical details.

IX. CONCLUSION

A conclusion section is not required. Although a We can conclude from this research that if we can handle all these phases of software development for financial institutions, we can achieve high application performance. Fraud detection is a fundamental concept in banking systems. All requests need to be validated and processed correctly to the targeted systems. All events are to be stored and processed. Fraudalerting and triggering process to be followed as per rules and protocols.

REFERENCES

[1] J. Jeng, D. Flaxer, and S. Kapoor, "RuleBAM: A Rule-Based Framework for Business Activity Management," Proceedings of the 2004 IEEE International Conference on Services Computing, 2004.

[2] D. K. W. Chiu, B. W. C. Kwok, R. L. S. Wong, S. C. Cheung, and E. Kafeza, "Alert-Driven E-Service Managment," presented at Proeedings of the 37th Annual Hawaii

International Conference on System Sciences (HICSS'04), 2004.

[3] N. W. Paton and O. Diaz, "Active Database Systems," ACM Computing Surveys, vol. 31, pp. 63 - 103, 1999. E. Baralis, "Rule Analysis," in Active Rules in Database Systems, N. W. Paton, Ed.:Springer-Verlag, 1999, pp. 51 – 67