Team Details

a. Team name: ADROIT
b. Team leader name: Nawaz B. Sayyad
c. Problem Statement: PS-1- Enhancing Cybersecurity with Targeted Vulnerability Prevention(Business Logic Vulnerability Detection, Fraud Prevention, and Rapid Response System for Banking)

India's 1st Meta Llama Hackathon
Top 15.
AI.Meta

AWS Ideathon 3rd Ranker

GCEK IDEATHON
Special Appreciation

The Escalating Threat of Dynamic Fraud in Banking Systems

In today's banking infrastructure, security vulnerabilities such as **SQL injection**, **broken authentication**, and **unauthorized access** have become prevalent, creating significant risks. Fraudsters have adapted, using **VPNs**, **broad proxies**, and **fake email IDs**, masking their identities while exploiting weak security mechanisms. These threats are not only constant but **evolve rapidly**, making it harder for traditional systems to keep up.

Consider the following patterns:

1. **Sim Card Switching & Remote Transactions**: A fraudster hijacks a user's SIM and initiates transactions from an unexpected location, triggering security alerts after the transaction is completed, often too late to stop the fraud.
2. **Double Fraud Scams**: Scammers start by offering small returns (e.g., 4000 INR) and increase the offer drastically (e.g., 8000 INR), manipulating victims into investing larger amounts, only to scam them.
3. **Authorization Hijacking & Account Takeovers**: Attackers exploit weak authentication to gain control over bank accounts, with automated scripts attempting **thousands of login credentials per minute** through brute force methods.
4. **SQL Injection and Database Manipulation**: **Insecure databases** are targeted with **injections**, allowing attackers to steal, alter, or delete data silently.

The Impact is Alarming: Fraud is evolving faster than detection systems can respond. By the time one fraud pattern is detected, 3-4 victims may have already suffered from the same technique. 80% of fraud attacks go undetected for several hours or even days due to slow response systems, allowing attackers to escape with millions.

The Speed of Response is Key: Currently, many systems only react after fraud is identified, leading to delays in detection and prevention. This gap is critical; with the **growing complexity** of scams, a **slow response time means higher risks for both customers and the bank.**

The need for a faster, real-time response mechanism is crucial to stop the **escalating trend of fraud**. Without it, banking infrastructure will continue to be vulnerable to a growing number of sophisticated scams.

Opportunities

1.How different is it from any of the other existing ideas?

My solution is different because it provides **Rapid**, intelligent, and **adaptive defense against the ever-evolving nature of fraud** in banking systems, ensuring that financial institutions **stay ahead of fraudsters, rather than simply reacting to them.**
Moreover, it is designed to **seamlessly integrate with live banking servers** without requiring extensive changes or high additional costs, apart from the necessary hardware. This ensures a **smooth implementation** and **minimal disruption** to ongoing operations, offering banks a **cost-effective**, **real-time fraud detection system** that enhances security and protects against emerging threats.

2.How will it be able to solve the problem?

Our solution solves the problem by providing **real-time fraud detection** that monitors transactions for patterns like **unusual amounts**, **location changes**, and **SIM swaps**. As the fraud detection rate increases, the system aggregates key data such as **IP address** and **transaction history**, triggering rapid escalation.
Once fraud is detected, the system immediately **alerts authorities**, **blocks suspicious transactions**, and **auto-generates a report** with key details. **It can even call the victim, transfer them to the cyber department, and direct security teams to investigate on-site.**
This **automated, rapid response** ensures quick action, minimizing fraud impact and providing **continuous protection** with seamless integration into existing banking infrastructure, all at a **low cost**

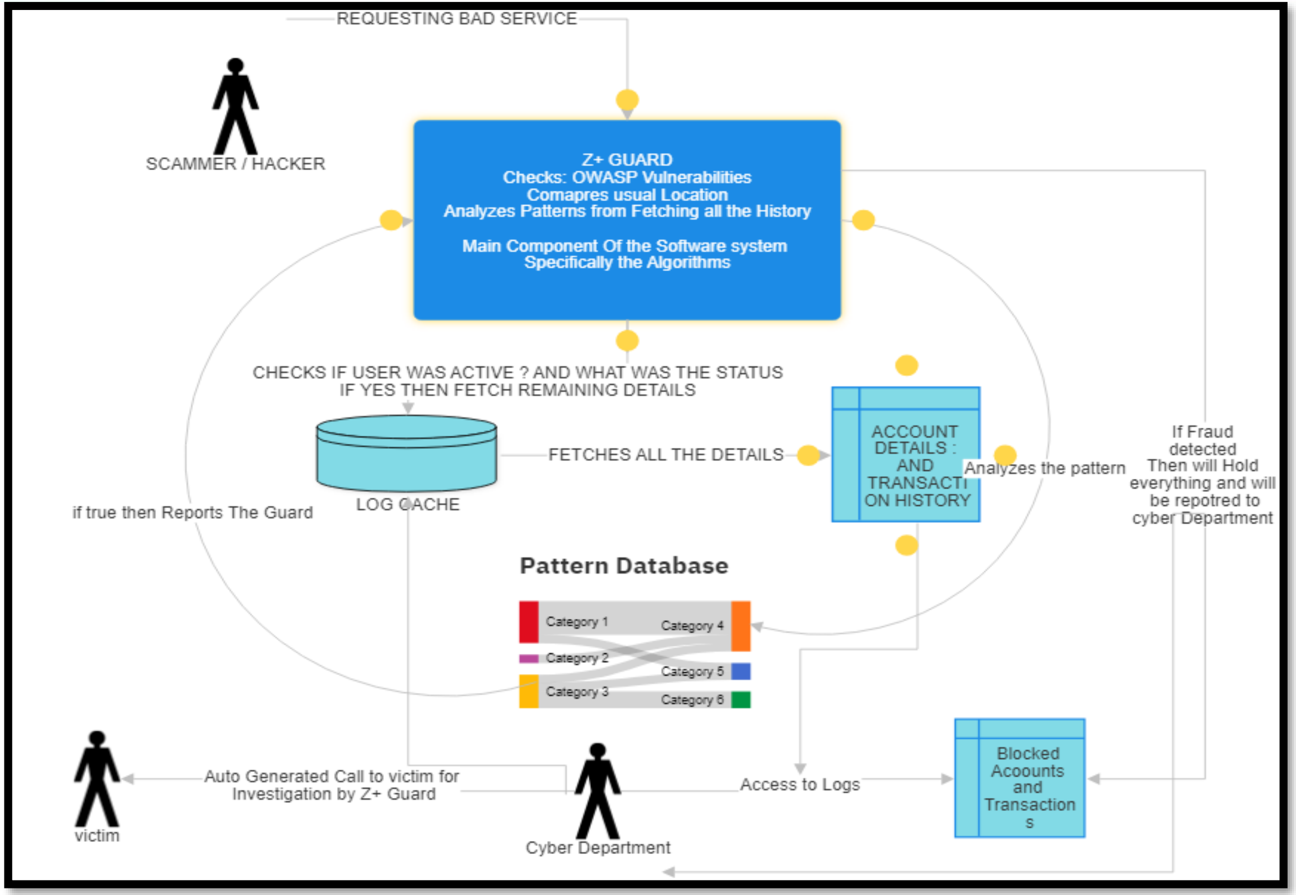
3.USP of the proposed solution

The **USP** of our solution lies in its **real-time fraud detection** and **instant automated response**. It **learns** from evolving fraud patterns, rapidly blocking transactions, alerting authorities, and triggering investigations—all without disrupting existing systems.
With **seamless integration**, low cost, and **adaptive intelligence**, it provides proactive protection, staying ahead of fraudsters.

List of features offered by the solution : Z+ Guard

1. **Real-Time Fraud Detection**: Monitors and identifies suspicious patterns (e.g., SIM swaps, location anomalies, double fraud).
2. **Automated Response**: Instantly blocks fraudulent transactions and escalates alerts to authorities.
3. **Adaptive Learning**: Continuously adapts to emerging fraud patterns, staying ahead of fraudsters.
4. **Data Aggregation**: Combines critical transaction data (e.g., IP address, history) for enhanced detection accuracy.
5. **Rapid Escalation**: Automatically triggers reporting, investigation alerts, and cybersecurity team notifications.
6. **Victim Call Automation**: Contacts victims, verifies incidents, and transfers calls to the cyber department for immediate action.
7. **Seamless Integration**: Easily integrates with existing banking infrastructure without significant costs or disruptions.
8. **Scalable Protection**: Ensures long-term, scalable defense as new fraud patterns emerge.
9. **Low-Cost Deployment**: Requires minimal extra hardware, making it cost-effective to deploy in live banking systems.

Process flow diagram



REQUESTING BAD SERVICE

SCAMMER / HACKER

Z+ GUARD
Checks: OWASP Vulnerabilities
Comapres usual Location
Analyzes Patterns from Fetching all the History

Main Component Of the Software system
Specifically the Algorithms

CHECKS IF USER WAS ACTIVE ? AND WHAT WAS THE STATUS
IF YES THEN FETCH REMAINING DETAILS

LOG CACHE

FETCHES ALL THE DETAILS

ACCOUNT DETAILS : AND TRANSACTI ON HISTORY

Analyzes the pattern

If Fraud detected
Then will Hold everything and will be repotred to cyber Department

if true then Reports The Guard

Pattern Database

victim

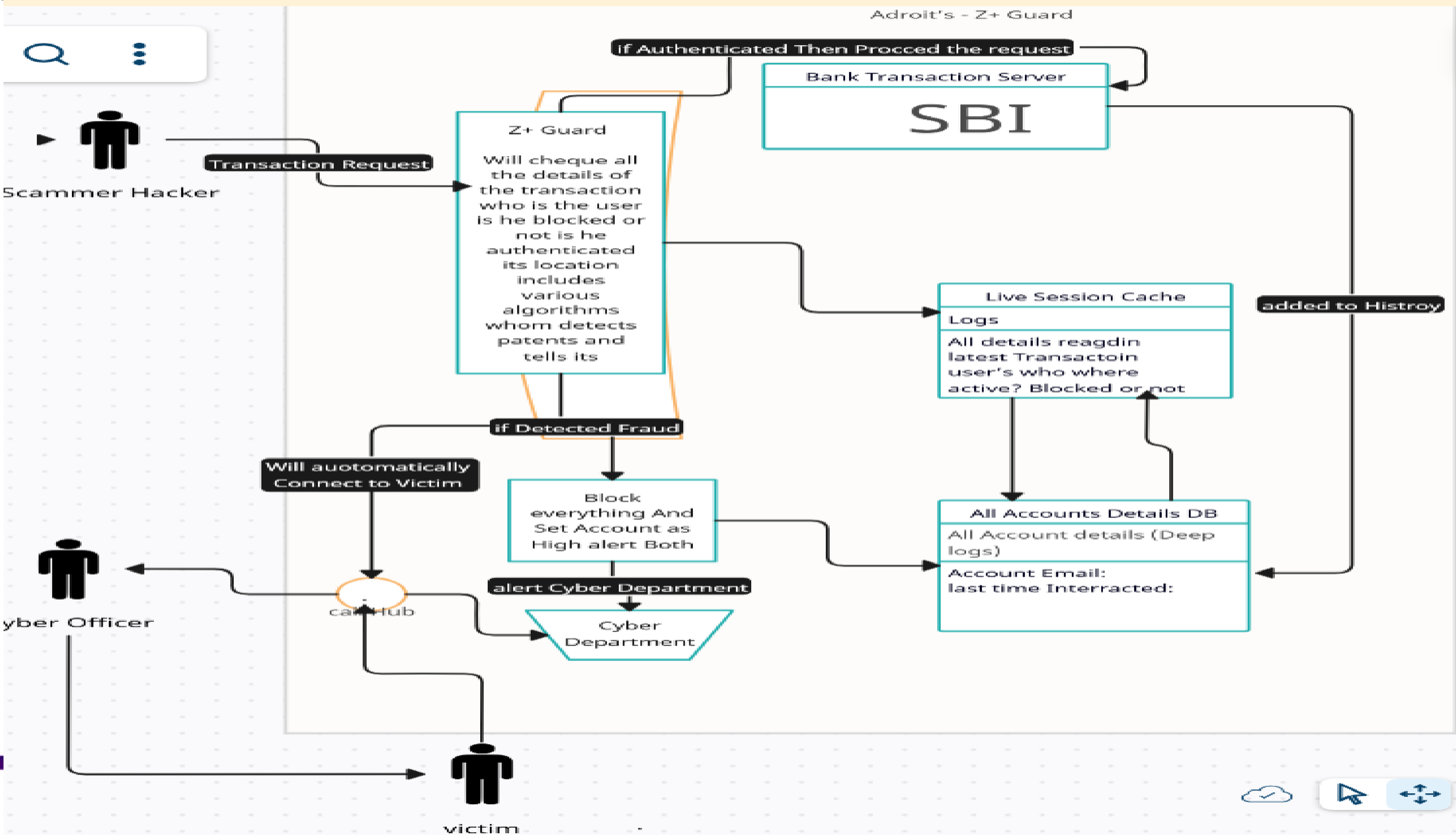
Auto Generated Call to victim for
Investigation by Z+ Guard

Cyber Department

Access to Logs

Blocked Acoounts and Transaction s

Architecture diagram of the proposed solution



Will cheque all the details of the transaction who is the user is he blocked or not is he authenticated its location includes various algorithms whom detects patents and tells its

All details reagdin latest Transactoin user's who where active? Blocked or not

Block everything And Set Account as High alert Both

Technologies to be used in the solution

Category	Tools/Technologies
Programming	Python, C#.NET (optional)
Machine Learning	Scikit-learn, PyTorch, Prophet
Caching	Redis
Databases	PostgreSQL, MongoDB, InfluxDB
Location Tools	OpenStreetMap, GeoIP2
Security	JWT, OpenSSL, OAuth2
Monitoring	Grafana
Logging	ELK Stack (Elasticsearch, etc.)

Estimated implementation cost (Only CacheMemory Can be Considered and Other Zero)

Implementation Cost Overview(Worst Case)

Using open-source technologies significantly reduces the software licensing and tool costs. The primary expenses are associated with hardware, especially high-speed RAM for caching billions of transactions and NVMe SSDs for fast storage. A high-performance server is also required to handle the processing load efficiently. Optional cloud services may incur minor costs if used for deployment or testing. Here's a detailed breakdown:

Cost Breakdown Table

Component	Estimated Cost (₹)
High-Speed RAM (for Log Cache)	50,000–1,00,000
Storage (NVMe SSDs)	60,000–80,000
Server Hardware	2,00,000–3,00,000
Cloud Services (optional)	12,000–30,000
Grand Total	**3,22,000–5,10,000**

Summary

By leveraging open-source tools like Redis for caching, PostgreSQL for databases, and ELK Stack for logging, the project avoids software licensing fees. The main cost is focused on acquiring hardware to ensure high performance and scalability. Depending on the scale of deployment, the total cost is estimated to be between **₹3,22,000 and ₹5,10,000**. If additional cloud infrastructure is needed, minimal monthly expenses can be expected.

Dear SBI Hackathon Team,

I sincerely thank you for taking the time to review my project and providing me with this incredible opportunity to present my ideas. It was truly an honor to showcase my work to such a prestigious panel.

As a student currently managing my exams, I have given my absolute best to analyze and develop this project, ensuring it aligns with real-world needs and adds value to the banking system. I am confident that this solution has the potential to make a meaningful impact and serve as a reliable tool in addressing the challenges it targets.

Your consideration and time are immensely valuable to me, and I deeply appreciate the opportunity to be part of this hackathon. I hope my project meets your expectations and earns your appreciation.

Thank you once again for your guidance, time, and support.

Warm regards,
Nawaz Sayyad - Adroit

SBI Life
Apne liye. Apno ke liye.

PRESENTS

HACK-AI-THON 2024

POWERED BY H2S HACK2SKILL

Thank You!