TruthScore: Continuous, Explainable Risk Scoring for Multi-Tenant Access Control

Valerie Benedit, Marcos Cruz Bazan, Maddox Marin | Product Owner: Mrunal Gangrade | Mentor: Dr. Masoud Sadjadi

Knight Foundation School of Computing and Information Sciences, Florida International University; Miami, FL, USA

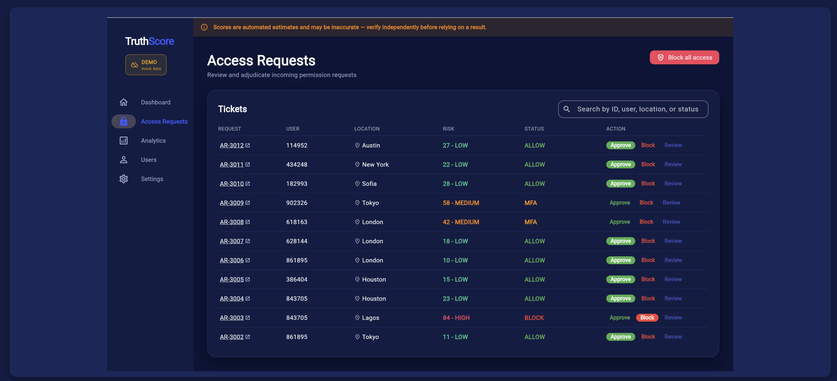
13 backlog items delivered • 5.5 sprints • 3-person team • 12 tenant-scoped tables • <200ms scoring latency • 0 cross-tenant data leaks

ABSTRACT

The majority of companies rely on users correctly inputting a password to protect their computer systems, even if the user should not have access to that system. TruthScore addresses this issue by analyzing login attempts in real time and assigning a "risk score" from 0 (normal) to 100 (suspicious) based on factors such as whether the login is coming from a recognized device or if the time and location are match the user's patterns. Based on the score, TruthScore will either automatically allow the user in or ask for extra proof of identity or block the attempt, providing an explanation in plain English, rather than an isolated number. To make this possible, TruthScore's working prototype is built from four connected pieces including a central server that makes every decision, a database that keeps each client company's data separate and private, a machine learning model that learns to distinguish normal activity for each organization, and an app that is compatible with phones, computers and browsers.

INTRODUCTION

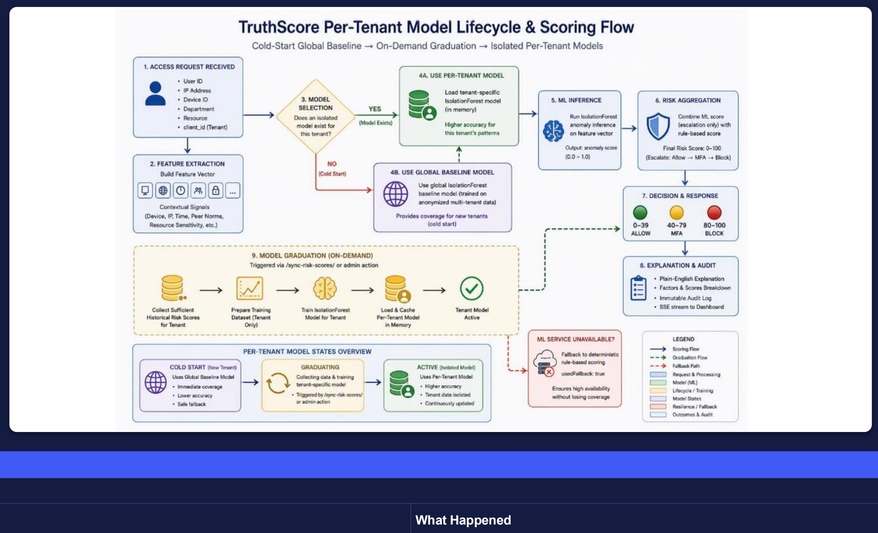
Most companies use Role-Based Access Control (RBAC) to protect its logins. This methodology assigns a role to each employee with restrictionsr on thewhat information available to them or the tasks that are appropriate for them to complete. However, RBAC is limited texclusively identifies user roles rather than the context of the logins taking place. For example, if a password gets stolen, RBAC is unable to detect whether the login is happening at 3am, in a different country or from an unrecognized device, as it only recognizes whether the password was inputted correctly. The job of security teams is to catch these unwanted logins , but it this quickly becomes an insurmountable task as thousands of logins are received per day and must be reviewed by manually be reviewed. To address this problem, the team designed TruthScore, a program that detects abnormal login behavior from three types of users: analyst, administrator, and everyday employee. The analyst role in TruthScore's framework is defined as a user in need of a quick, plain-English reason for each alert and a one-click way to approve or deny each request. In similar manner, the administrator role determines how strict the system should be the strictness of the system. Meanwhile, the everyday employee refers to those in need of logging in.



METHODOLOGY

TruthScore is built from four pieces that only talk to each other through one central "front desk," called the API. The app people use never touches the database directly. The users always go through that front desk first, which keeps every password and database connection in exactly one place.

Each login is scored two ways: simple human-written rules (like "block known-malicious addresses"), and a machine-learning model called Isolation Forest, which learns what "normal" behavior for each company prior knowledge of what "suspicious" means. TruthScore always takes whichever score is more cautious, specifically the machine learning model can only raise an alarm louder, never quiet one down.

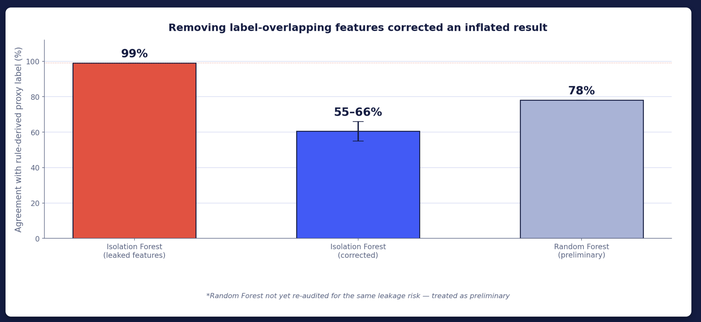


RESULTS OR PRELIMINARY RESULTS

Before trusting the model, the team tested it and checked its own work, including trying to look up another company's device on purpose, which the system correctly refused every time.

However, during the machine learning testing portion, the model's initial accuracy score looked almost perfect at 99%, until the team found it had accidentally trained the model with the answer key in disguise. Once those were removed and the test was re-run fairly, the honest accuracy dropped to about 55–66% which is a far more reasonable number.

IMPACT METRICS
Corrected Accuracy: 55–66% (an inflated 99% was caught and fixed)
Response Time: under 0.2 seconds per login
Cross-Company Data Leaks: zero, even under deliberate testing



TESTING & VERIFICATION — WHAT WE CHECKED, AND WHAT HAPPENED

Target Feature	Test Scenario	What Happened	Status
Device Isolation	Tenant A's user looked up Tenant B's device.	Correctly treated as unknown (Block/80) instead of inheriting trust.	PASSED
Cross-Tenant Write Guard	Tried to write a decision under a foreign tenant's ID.	API refused with 404 Not Found.	PASSED
ML Baseline Graduation	Scored a tenant before and after its model was trained.	Model source moved from "baseline" to "tenant-specific."	PASSED
Score Calibration	Sent a request with two known risk factors.	Score combined to 55 (MFA) instead of maxing out at 100.	PASSED
Evaluation Leakage Audit	Re-ran the accuracy test after removing leaked features.	Accuracy fell from an inflated 99% to an honest 55–66%.	PASSED — issue found & fixed

CONCLUSION OR EXPECTED OUTCOME(S)

TruthScore is a fully working system that scores logins in real time, maintains each company's data separate from any other company, keeps record of all decisions made by a human analyst in the company, and displays it live on a dashboard available to any user within the company. This is an impactful solution because stolen credentials are an increasingly common problem causing many real-word data breaches. Current security modalities have contributed to these breaches by only filtering access with proper password input. By implementing TruthScore, logins are continuously monitored and suspicious behavior can be efficiently transferred to a human analyst, providing a plain-English reasoning without interrupting other harmless logins. In essence, it acts as a security guard that never sleeps, remembers all normal behaviors and justifies flagged activity, rather than simply prohibiting it. Additionally, TruthScore evolved from a single rule-based scorer into a genuinely multi-tenant, machine-learning-backed platform. Future improvements include deepening the human-in-the-loop perspective, giving analysts richer context, clearer explanations, and more direct control over every automated decision as the system matures.

FURTHER RESEARCH IDEAS AND/OR ACKNOWLEDGMENTS

Looking ahead, the most pressing next step is making logins harder to fake: today each request is tagged with a simple ID number, and the plan is to replace that with a cryptographically signed token, so no one can pretend to belong to a company they don't. They also want to test smarter warning signs, like subtle changes in a device's fingerprint or unusually long sessions, to see whether these catch even more real threats.

The team is incredibly grateful with our Product owner, Mrunal Gangrade and sponsor JPMorgan Chase, for her exceptional mentorship and opportunity to work on such an important and interesting project. Also, a special thanks to the instructor Dr. Masoud Sadjadi, and the KFSCIS Capstone Program for hosting the summer cohort of 2026.

TRUST, FAIRNESS & NEXT STEPS

TruthScore never makes a final call alone. Every automated decision requires extra verification or blocking someone outright which is paired with a plain-English reason and logged for a human analyst to review, override, or confirm. A risk score should never be the last word on whether a real person gets into their own accounts.

The team is also up-front about two fairness risks this prototype does not solve. First, the system learns what "normal" looks like from past behavior, so an employee with a genuinely unusual but legitimate schedule such as night shifts, frequent travel, a role spanning departments that may face more friction than a colleague who simply looks "average," for reasons that have nothing to do with real risk. Second, any model trained on past security logs can inherit whatever bias already existed in those logs. Therefore, the next step would be to work on a few feature engineering tweaks that would be specific to each company during the onboarding process for the model to better learn "normal" behavior for each company.