HUMAN FACTORS IN MULTI-FACTOR AUTHENTICATION

Sydney Chantamala

Josh Lander

Duraid Amjed Salim Alsabahi

Lachlan Howe

Rory Spiteri

Jakub Roj

Sabbir Ahmed

Supervised by Dr Ala Al-Areqi

Western Sydney University

INFO3016 Cybersecurity and Behaviour Community Placement

Table of Contents

Chapter 1 - Introduction..... 5
1.1 Background..... 5
1.2 Problem..... 6
1.3 Objectives..... 6
1.4 Scope..... 7
1.5 Expected Outcome..... 8
Chapter 2 - Literature Review..... 10
2.1 Introduction..... 10
2.2 Multi-Factor Authentication: Concepts and Current Approaches..... 10
2.3 Human Factors in Multi-Factor Authentication..... 11
2.4 MFA Usability and Security Trade-offs..... 12
2.5 MFA Fatigue and Authentication Friction..... 13
2.6 Adaptive and Risk-Based Authentication..... 14
2.7 Keycloak as an Identity and Access Management Platform..... 15
2.8 Keycloak for Adaptive MFA..... 16
2.9 Simulation and Synthetic User Behaviour for MFA Evaluation..... 17
2.10 Critical Synthesis of Related Studies..... 17
2.11 Research Gap..... 17
2.12 Chapter Summary..... 18
Chapter 3 - Methodology / Simulation Setup..... 20
3.1 Introduction..... 20
3.2 Research Design and Approach..... 20
3.3 System Architecture and Experimental Environment..... 21
3.4 Baseline MFA Configuration..... 23
3.5 Human Factors and Synthetic User Profiles..... 23
3.6 Authentication Risk Indicators..... 25
3.7 Risk-Adaptive MFA Decision Model..... 26
3.8 Keycloak Implementation..... 26
3.9 Synthetic Behaviour / Agent Implementation..... 29
3.10 Test Scenarios..... 30
3.11 Data Collection and Monitoring..... 36
3.12 Evaluation Metrics..... 37
3.13 Experimental Procedure..... 39
3.14 Data Analysis..... 40
3.15 Ethical, Privacy and Security Considerations..... 40
3.16 Methodological Limitations..... 41
3.17 Chapter Summary..... 41
Chapter 4 – System Implementation, Experimental Results and Evaluation..... 42

4.1 Introduction..... 42
4.2 Implementation of the Experimental Environment..... 42
4.2.1 Keycloak Deployment and Configuration..... 42
4.3 Implementation of Synthetic User Profiles..... 48
4.4 Implementation of Risk-Adaptive MFA Policies..... 52
4.5 Experimental Setup and Execution..... 56
4.6 Usability and Login Friction Results..... 58
4.7 Security Evaluation Results (Duraid) all subsections included..... 60
4.8 Security Usability Trade-Off Analysis..... 64
4.9 System Limitations and Observed Issues..... 69
4.10 Summary of Key Findings..... 70
Chapter 5 – Discussion, Conclusion and Future Work..... 72
5.1 Introduction..... 72
5.2 Discussion of Findings Against Project Objectives..... 72
5.3 Discussion of the Security–Usability Trade-Off..... 74
5.4 Discussion of Human Factors..... 77
5.5 Comparison with Existing Literature and Authentication Guidance..... 79
5.6 Research and Technical Contributions..... 82
5.7 Research Limitations..... 86
5.8 Recommendations for Future Work..... 88
5.9 Overall Conclusion..... 95
Project Management Timeline..... 97
Group Reflection..... 98
References..... 101
Chapter 2 – Literature Table..... 107

Chapter 1 - Introduction

1.1 Background

Cybersecurity accounts for an increasing share of organisations' efforts to protect their assets and users from financial loss and data breaches. Credential-based attacks, such as brute-force attempts and the reuse of breached passwords, remain a major threat. Data breaches cost global organisations an average of US$4.88 million, and authentication failures cost businesses an additional $3 million annually (IBM, 2024).

In response, many organisations and industries have adopted Multi-Factor Authentication (MFA), especially because it blocks unauthorised users, including those with reused or breached credentials, by requiring an additional authentication factor. MFA has been shown to reduce the risk of account compromise by 99.22% (Meyer et al., 2023), and over 99.9% of compromised accounts had not implemented MFA (Microsoft, 2024). Organisations using MFA also experience approximately 45% lower breach costs than those that have not adopted it (Schomburgk, 2025). Government guidance, such as the Australian Signals Directorate's guidelines (2023), recommends MFA across organisations and users.

MFA requires "at least two elements" from three categories: something the individual knows, something the individual has, or something the individual is (OWASP, 2025). However, its protection depends on users completing it correctly and consistently, and many deployments place a significant burden on them. In a survey of smartphone users, Sakshi et al. (2026) found that 75% had abandoned MFA setup because of complicated interfaces, 59% avoided the application entirely due to confusion, and 54% were frustrated by having to use separate devices. Repeated prompts can also train users to approve requests without checking them, which attackers exploit through MFA fatigue attacks (Jubur et al., 2025). These human factors leave users and organisations exposed to account compromise, service disruption and identity theft, even where MFA is in place.

Adaptive, risk-based authentication has emerged as one response. It adjusts the authentication challenge to the risk of each login, adding steps for unfamiliar devices or locations and reducing prompts for routine, low-risk access (Robertson & Kosinski, 2026). Keycloak, an open-source

Identity and Access Management (IAM) platform (Keycloak, n.d.), does not natively support risk-based MFA (Son et al., 2025). It can be extended through open-source projects, such as an adaptive authentication extension with AI integrations (Bartoš, 2026), and through commercial extensions, such as event-based one-time passcodes (Keymate Team, 2026). This makes Keycloak a practical platform for examining how adaptive MFA affects both users and security.

1.2 Problem

Complex MFA designs increase the likelihood of human error, reducing user engagement and prompting users to avoid or bypass MFA, leaving accounts exposed to credential attacks, financial loss, and data theft. The problem affects students, employees and customers alike, and a resulting compromise can disrupt business continuity and cause reputational, legal and financial harm. Most MFA deployments apply the same fixed challenge to every login regardless of risk. Users therefore face repeated prompts, delays, and device switching, even for routine access, which encourages fatigue, careless approvals, and abandonment. This burden can be reduced through adaptive, risk-based authentication. Having said this, existing approaches, including those built for Keycloak, have historically been primarily evaluated based on technical accuracy and security outcomes, and they treat human behaviour and system risk as a single problem. As a result, little is known about how adaptive MFA affects usability, accessibility and fatigue alongside security. This project addresses that gap by simulating user interactions with a risk-adaptive Keycloak MFA model to identify configurations that reduce authentication friction without weakening protection.

1.3 Objectives

1. Identify cognitive, behavioural, and usability factors affecting MFA through a review of relevant literature and use these factors to define simulated user behaviours and risk indicators.

2. Design a risk-adaptive MFA decision model that adjusts authentication requirements according to defined behavioural and contextual risk indicators.

3. Implement the proposed adaptive MFA model in Keycloak and develop synthetic user scenarios representing different authentication behaviours and risk conditions.

4. Deploy and test the prototype within a controlled Cyber Range environment and collect relevant authentication and security telemetry using appropriate monitoring tools.

5. Evaluate the adaptive MFA model against a baseline MFA configuration using defined security and authentication-burden metrics, and develop recommendations based on the findings.

1.4 Scope

In Scope

This project examines how a risk-adaptive Multi-Factor Authentication (MFA) model implemented in Keycloak affects the balance between usability, accessibility and security, using simulated users in a controlled environment. It begins with a review of at least 20 relevant publications on MFA approaches, human factors in authentication, the usability–security trade-off, MFA fatigue and adaptive authentication, which establishes the research gap and informs the experimental design. The project will then design a decision model that calculates a risk score for each login attempt based on three indicators: failed attempts, use of a new device, and a change in the login country. Each attempt is placed in a low-, medium- or high-risk band. The model will be implemented using Keycloak's conditional authentication flows and a Flask-based risk engine. Each band maps to a policy of increasing strength: password only for low risk; password and time-based one-time password (TOTP) for medium risk; and password, TOTP and email verification for high risk.

The prototype will be deployed in an isolated virtual environment monitored by Wazuh, Zeek, and Suricata, and tested with 12 synthetic personas representing varying levels of technical confidence, friction tolerances, enrolled credentials, and accessibility needs. The policies will be compared using simulated indicators of friction (time to authenticate, number of authentication steps, failed attempts and retries, and simulated abandonment) and by whether each policy is enforced as intended against defined threat scenarios. Authentication fatigue is represented indirectly through prompt frequency, retries and predefined tolerance thresholds, rather than measured as a user experience. The findings will inform recommendations for configuring adaptive MFA in Keycloak.

Out of Scope

The project is limited to a simulated, laboratory-based evaluation and does not involve human participants. No surveys, interviews, usability studies, or other user research will be conducted, and no real-world personal data will be collected, stored, or processed, as all personas, credentials, and login contexts are synthetic. Accordingly, the project does not measure actual user engagement, satisfaction or fatigue, and its findings describe the behaviour of the configured policies under simulated conditions rather than how real users would respond.

The prototype will not be deployed to production systems or tested against live services, and all attack scenarios will be confined to the isolated environment. The project does not develop a new MFA method or authentication factor; instead, it configures and evaluates existing methods and Keycloak capabilities. Threats such as phishing, session hijacking, account recovery attacks, and comparisons with other identity platforms also fall outside the scope of this project.

1.5 Expected Outcome

The primary expected outcome of this project is a working prototype of a risk-adaptive Multi-Factor Authentication (MFA) model implemented in Keycloak. The prototype will use a risk engine to classify each login attempt as low, medium or high risk, based on defined contextual indicators, and will apply a corresponding authentication policy of increasing strength through Keycloak's conditional authentication flows. To support its evaluation, the project will also produce a set of synthetic user personas and authentication scenarios that represent different behavioural characteristics and risk conditions, such as failed login attempts, unfamiliar devices, and changes in login location. These will be executed in an isolated virtual environment, allowing each policy to be tested under consistent and repeatable conditions.

The project is also expected to produce a comparative evaluation of the adaptive policies against a baseline MFA configuration, supported by measurable results. Usability will be assessed through simulated indicators of authentication friction, including completion time, number of authentication steps, failed attempts and retries, and simulated abandonment based on predefined persona thresholds. Security will be assessed by whether each policy is enforced as intended and how the system responds to the defined threat scenarios. Because the evaluation uses simulated rather than real users, these results will describe how the configured

policies behave under controlled conditions; they will not directly measure user engagement, satisfaction or fatigue. Drawing on this evidence, the project will identify the trade-offs between usability and security across the policies and develop evidence-based recommendations for configuring adaptive MFA in Keycloak, along with directions for future research, including testing with human participants.

Chapter 2 - Literature Review

2.1 Introduction

Chapter 2 will be critical for reviewing MFA by exploring human factors and discussing the usability/security trade-off that can make MFA more or less effective. This chapter will also identify research gaps, present a simulation-based approach, and explore Keycloak. The purpose of this chapter is to develop an understanding of its usability and security, balancing the two to create an adaptive authentication guide that is highly engaging for its users.

2.2 Multi-Factor Authentication: Concepts and Current Approaches

Multi-Factor Authentication (MFA) is a security process that requires users to have a secondary identification before accessing a system, based on something you know, which is heavily based on knowledge; something you are, which is a unique trait, such as a fingerprint or facial scan; and something you have, which is a physical item in your possession, such as physical keys. MFA works by authenticating, which verifies identity using common methods such as four-digit codes, passwords, passkeys, and One-Time Passwords (OTPs) (Kosinski, 2024). The purpose of MFA is to provide authorised users with a frictionless experience when logging in to the system while making it difficult for attackers to impersonate users and gain unauthorised access. Current MFA approaches fall into two categories: biometric authentication, which uses a person’s physical or behavioural traits instead of a password, and non-biometric authentication, consisting of passkeys or One-Time Passwords. (Mohammed et al., 2023). Hossain and Raza (2025) measured attack success rates for biometric and non-biometric authentication and found that OTP (non-biometric) had the highest success rate at 15%. This is likely due to phishing and social engineering, which are the most common types of attacks humans fall for. However, biometric authentication had only a success rate of 5%. Because it relies on fingerprints or facial recognition, it is difficult for attackers to duplicate and requires less user interaction, since no extra hardware is needed. However, Tran-Truong et al. (2025) found that 70% of the analysed academic papers prefer biometrics but argue that biometric authentication still faces human error due to disrupted or incomplete scans and presents permanent security risks if

compromised. 33% of analysed systems and users rely on One-time Password MFA due to their familiarity and lower system complexity.

2.3 Human Factors in Multi-Factor Authentication

Human psychology is a significant, enduring factor that directly influences how effectively multifactor authentication works, not just technical configurations alone. As MFA continues to advance against new attack methodologies. Human factors such as resistance to change, disbelief in the scale of cyber threats, and concerns about individual privacy lead employees to ignore MFA. (Lahti et al., 2025). Users in MFA can, cognitively, suffer from time pressure, cognitive overload, and fatigue. These issues can be mitigated by implementing unique biometric behaviours, such as keystroke dynamics, mouse movements, and touchscreen interactions (Shuford, 2024). Biometric MFA methods. These dynamic methods are expected to enable continuous implicit authentication, requiring no explicit user interaction, while offering the possibility of using fewer fixed credentials and a higher ability to detect threats in real time (Ayeswarya S & K, 2025). Running MFA methods in the background removes the load on users when accessing the required interfaces. Human factors experiences such as delays, repeated prompts, and failed attempts create usability friction for users.

Behavioural frictions caused by poor mental models and interpretations of the importance of security that MFA provides to users. The interpretations users form are based on their assumptions. Authentication systems are influenced by prior experiences, technical confidence, day-to-day interactions with technology, and cultural contexts. (Hossain, 2025). Furthermore, end-users' individual context plays a significant role in understanding and acceptance of authentication. Users' environments significantly influence their ability to engage with MFA protocols, including location, network connectivity, and devices. Furthermore, the accessibility of users with motor, visual, mental, and cognitive disabilities is also influenced by their ability to utilise MFA features effectively and autonomously. These mental models and personal contexts, when interpreted by end users, can lead them to engage in insecure behaviour. Poor mental models lead to failures to follow basic password guidelines or policies, including opting for weak passwords, sharing credentials, or misunderstanding biometric prompts. (Hossain, 2025). Due to technical literacy or prior experience, their understanding of the importance of MFA for privacy is shaped by a loss of accountability or caution about their security, as reflected in their

interpretations and individual contexts. Designing a usable yet secure system is incredibly complex, with inevitable trade-offs to simplify user authentication.

2.4 MFA Usability and Security Trade-offs

Usability and Security are strongly reflected in the implementation of MFA, ensuring a smooth process that makes it easy for users to log in while making it difficult for unauthorised users to impersonate. However, we must be mindful that if MFA becomes overly complex, users will be driven away.

Security is critical to ensuring we comply with legal and ethical obligations under the Privacy Act 1988. If vulnerabilities are found in poor security design, attackers will exploit those vulnerabilities to harm stakeholders. This creates a significant trade-off between the two, as they can introduce security weaknesses or cause user frustration, leading users to bypass security controls and abandon the application. According to Arnold *et al*. (2022), 57.6% of students within the survey believed that MFA prevented them from completing necessary tasks. Furthermore, 61.9% of students stated that methods of protection, such as OTPs, biometrics, or Notification codes, are not worth the potential increase in security. Many students were found to have behavioural sentiments of anger, frustration, and stress. This data reveals that poor MFA implementation causes users to lose retention and trust in the authentication system. As a consequence, it can severely strain an organisation's resources to recover locked accounts. An imbalance between usability and security significantly undermines ethical obligations to ensure compliance with secure, simple user design.

One-time passwords (OTPs) provide a markedly different usability-security balance that benefits users and organisations. Parmar (2022) found that OTPs are low-cost, easy to use, and administer. The use of these OTPs reduces the need to constantly reuse credentials and improves security through token expiry. Although OTPs improve usability and security, they are highly susceptible to bypasses, token reuse, and brute-force attacks (H. Al-Omari, 2021). Comparing these MFA methods shifts responsibility from the user to automated processes, improving usability and security. Whilst OTPs are highly important, there is also a trade-off with more advanced cyberattack methods.

Usability is a core mechanism of MFA's design. Users authenticate using a secondary factor, such as a secondary device or an application like Microsoft Authenticator, to verify their identity via a one-time code on the primary device. This prevents attackers from immediately accessing accounts if credentials are leaked, allowing time for password recovery.

These trade-offs between usability and security are unavoidable. Discarding one can deliver security risks or drive user avoidance. Organisations mitigate this by combining security awareness and user engagement to enforce strong authentication, while recognising that these simple methods have both benefits and drawbacks (Whitman et al., 2025). Despite advances in MFA methodology, no studies dynamically adapt to individual users. The existing solutions treat behaviours and risk indicators as the same problem and apply the same solution regardless of users' psychological behaviour.

2.5 MFA Fatigue and Authentication Friction

Authentication friction describes the extra time and effort required to complete login steps, while MFA fatigue refers to the frustration that can develop when those steps are repeated too often. Marky et al. (2022) found that users often viewed two-factor authentication as annoying because it interrupted the task they were trying to complete. This matters because user acceptance affects the consistency with which MFA is used. When authentication becomes inconvenient, users may delay, avoid, or abandon the process.

Time is one of the clearest measures of friction. Kruzikova et al. (2024) studied 337 smartphone users and found that completion time varied with age, experience, and other user characteristics. Users who were satisfied with the time required for 2FA also rated the method more positively for security and overall satisfaction. Hastings et al. (2025) reported a similar pattern after a university changed its MFA process from simple approval prompts to number matching. Mobile MFA failures increased from 10.2% to 17.9%, indicating that stronger controls can also increase authentication difficulty.

Fatigue can also weaken security. Jubur et al. (2025) examined a compare-and-confirm push method and found that approximately 49.6% of simulated attack prompts were approved, even though genuine logins were approved correctly in most cases. Their findings suggest that repeated prompts can encourage routine responses, particularly when users stop carefully

checking each request. In this case, a usability issue becomes a security concern because repeated authentication may reduce user attention.

Friction also appears during setup and account recovery. Ghorbani Lyastani et al. (2023) found substantial differences in how major websites handled 2FA setup and use. Amft et al. (2023) also reported that recovery was unsuccessful for many of the accounts they tested. Sawant and Pradeepa (2026) found that separate-device requirements frustrated many smartphone users. Together, these studies show that authentication friction can arise at several stages of the process, including delays, errors, device dependence, and recovery issues.

The literature also shows a limitation. Most studies measure frustration through interviews, surveys, log analysis, or laboratory testing. Less attention has been given to how an IAM platform could reduce unnecessary authentication for lower-risk users. This evidence supports the need to examine adaptive authentication, in which additional steps are applied based on risk rather than used for every login.

2.6 Adaptive and Risk-Based Authentication

Adaptive authentication, also called risk-based authentication, adjusts authentication requirements based on the estimated risk of a login attempt. Contextual information, such as device, IP address, location, and login time, can be assessed to determine whether additional verification is needed. Lower-risk logins can use fewer authentication steps, while higher-risk attempts may require an OTP or stronger verification. This approach aims to reduce unnecessary friction without removing stronger controls from suspicious logins.

Wiefling et al. (2023) analysed 31.3 million login attempts from 3.3 million users over more than one year. Their results showed that legitimate users were rarely asked to re-authenticate, while the system still blocked most targeted attacks. The study suggests that risk-based authentication can reduce the number of repeated challenges for normal users. However, the work was conducted on a single service and did not directly measure user experience.

Machine-learning approaches have also been tested. Picard and Pierre (2023) developed RLAuth, a reinforcement-learning system that used contextual variables such as location,

network connection, and time of access. The model achieved strong classification performance, but the study did not directly measure whether users experienced less friction.

Risk-based authentication can also introduce new problems. A legitimate user may be classified as suspicious when travelling, changing devices, or logging in at an unusual time. This can lead to unnecessary challenges or lockouts. Contextual data may also create privacy concerns, especially when location or IP information is collected.

Existing research, therefore, supports adaptive authentication as a means of reducing repeated MFA prompts, but much of the evidence focuses on detection accuracy or aggregated login data. Human-centred outcomes such as delay, retries, input errors, and abandonment are less commonly examined within the same adaptive system. This project addresses that limitation through a three-level Keycloak model, in which authentication requirements increase from low to medium to high risk, with synthetic users used to measure authentication behaviour.

2.7 Keycloak as an Identity and Access Management Platform

Keycloak is an open-source Identity and Access Management (IAM) system that helps to manage users' authentication, access permissions, password policies, and account lockout policies (Keycloak, n.d.). It acts as a centralised authentication and authorisation server designed to handle security in a user-friendly manner. The purpose of Keycloak is to provide robust security with high usability while also protecting the user's account by presenting multiple measures to authenticate and authorise a user's login. Keycloak's core capabilities, such as Single Sign-On (SSO), allow users to log in with a single set of credentials to access different services within Keycloak, improving overall usability, particularly when integrated with MFA (Okello, 2023). The features were shown in Divyabharathi & Cholli (2020)'s research, where SSO reduces user overhead and minimises password exposure to man-in-the-middle attacks. Additionally, Keycloak can support several standard protocols, such as OpenID Connect, OAuth 2.0, and SAML 2.0, to control access to user pools, enforce security policies, and manage permissions across services. Keycloak can integrate with Active Directory to store users, computers, and groups. Using LDAP to control network machines and IT infrastructure. Despite its advantages, Keycloak does present challenges. It requires detailed configuration to maintain a stable production environment, and integrating various applications into Keycloak demands

in-depth knowledge of different protocols; otherwise, it can have negative security effects (Helena, 2023). Furthermore, Mihai et al. (2024) implemented Keycloak SSO on two virtual machines with 8 vCPUs and 16GB of RAM and later enabled users to use SSO as their MFA on those virtual machines. The research shows users didn't experience slow authentication or task interruptions; it also shows that it took users months to adapt to and accept SSO because they had to repeatedly enter different passwords for different services.

2.8 Keycloak for Adaptive MFA

Adaptive MFA, or risk-based authentication, is an intelligent security system that uses contextual information as an additional authentication factor for user access requests. Unlike using PINs or One-Time Passwords, it analyses the user's real-time patterns based on the risk level of each access request (Clark, 2025). Adaptive MFA can improve user experience through location and device recognition. Unfortunately, Keycloak does not have built-in adaptive or risk-based MFA, but Son et al. (2025) developed an AI-driven, risk-adaptive framework to assign real-time risk scores, which were integrated into Keycloak. The framework is assessed based on user login behaviour, including login time, IP address, and user agent. This work has been highlighted to enhance security and also remove any unnecessary authentication features that may degrade the user experience. While using AI can improve the accuracy of logistic regression to 97% and better evaluate user behaviour in real time through login records, it does not reflect human experience or capture emotions at different stages of MFA, and it doesn't address common human error. This demonstrates that while Keycloak lacks adaptive MFA, it can be incorporated through extensions, such as AI-based ones configured into synthetic users. Additionally, Kovačć and Bilić (2026) have shown that out of their 5000 logins, fewer risks were triggered with <5% risk-based MFA with 1% false positives, demonstrating that Keycloak for adaptive MFA can be used to reduce MFA challenges; however, their small sample size could produce lower statistical power, which may become unreliable. While adaptive authentication provides frictionless access to its users, it can still be compromised by false positives; for instance, if users are accessing from overseas, the risk assessment may exceed the threshold and lock legitimate users out (Mthethwa et al., 2025).

2.9 Simulation and Synthetic User Behaviour for MFA Evaluation

Previous studies, such as Carjuman et al. (2025), have proposed using synthesised and anonymised datasets relevant to our Keycloak simulation. The proposed framework, demonstrated by this group, incorporated both malicious and legitimate authentication attempts while accounting for numerous other security measures, such as biometrics, PIN, and more specific ones like mouse dynamics and typing patterns. The integration of all of these metrics makes it extremely difficult for malicious actors to replicate. This study provides crucial insights into factors that may help improve our system's security without compromising usability.

2.10 Critical Synthesis of Related Studies

Whilst our intention is to use AI agents in place of humans for our experiment, it's important to note the findings of Liu & Huang (2026. Liu & Huang (2026 found that an MFA workflow implicitly assumes that credential entry, passcode retrieval, and passcode submission remain operationally separated through user participation. This implied separation is being eroded by increased AI usage, with a 40 per cent growth in the use of Agentic AI models in large enterprises (Tinkoff et al., 2026). Liu & Huang (2026) note that AI agents with valid credentials can retrieve an email-delivered OTP or authenticator code and complete the entire MFA workflow without human intervention. Liu & Huang (2026 coined this breakdown in security as “Factor Collapse”. Liu & Huang (2026) went on to analyse the extent to which certain AI systems can exhibit Factor Collapse and proposed using “short-window motion sensing” to indicate real human interaction, thereby mitigating AI-led factor collapse. Since our experiment will utilise AI agents to simulate human interactions (which require access to both passwords and the MFA platform), it will inherently demonstrate factor collapse. However, little can be altered to address this, and it must therefore simply be noted in our evaluation.

2.11 Research Gap

Based on the current literature reviews, it was found that modern MFA adaptive systems treat human behaviour and system risk as the same thing and provide a single solution to both engage users and implement safety systems. If a solution tries to address two problems, it could lead to systems with highly developed safeguards, which would cause legitimate users to be

blocked out, become frustrated, refuse to use MFA, and encourage attackers to develop more countermeasures to circumvent the safeguards (Amft et al. 2023). Or leave users to develop an easy method for remembering and performing MFA, which could leave them vulnerable to phishing attacks (Jubur et al., 2025).

2.12 Chapter Summary

This chapter reviewed the literature on Multi-Factor Authentication (MFA) across five themes: current MFA approaches, human factors in authentication, the usability–security trade-off, MFA fatigue and authentication friction, and adaptive authentication. Taken together, the studies show that MFA's effectiveness depends on both its technical strength and how users experience it. Usability research consistently finds that added time, repeated prompts, and reliance on separate devices reduce satisfaction and increase errors, delays, and abandonment (Kruzikova et al., 2024; Marky et al., 2022; Sakshi et al., 2026). Stronger controls can make authentication harder to complete (Hastings et al., 2025), and repeated prompts can encourage routine approvals that weaken security (Jubur et al., 2025). These studies, however, mostly examined fixed MFA configurations, which apply the same challenge regardless of the risk of a login.

Adaptive, risk-based authentication addresses this by adjusting requirements to the context of each login. The literature suggests it can reduce challenges for legitimate users while keeping protection against suspicious attempts (Wiefling et al., 2023). Its evaluation has focused mainly on technical outcomes, though. Specifically in Keycloak, risk-adaptive frameworks have reported high classification accuracy without measuring human experience, errors, or fatigue (Son et al., 2025). The review therefore identified a gap: there is limited evidence on how risk-adaptive MFA policies affect authentication friction alongside security, and on how differences in user characteristics, such as technical confidence and tolerance for friction, shape those outcomes. This gap is particularly evident in open-source identity platforms such as Keycloak.

This gap directly shapes the methodology in Chapter 3. To evaluate usability and security together, the project designs a risk-adaptive decision model that maps contextual indicators, namely failed attempts, new devices and changes in location, to low-, medium- and high-risk authentication policies, and implements it in Keycloak. Because human participant testing was not feasible within the project's timeframe, user behaviour is represented through synthetic personas whose attributes draw on the human factors identified in this chapter. These personas

are tested under defined scenarios and measured by friction indicators (completion time, authentication steps, failed attempts, and simulated abandonment) and by policy enforcement. Section 2.10 also noted that AI agents completing MFA on behalf of users can undermine the separation between factors (Liu & Huang, 2026). This is treated as a methodological limitation of the simulation, and Chapter 3 sets out the research design, environment, personas, scenarios and evaluation metrics used to address the gap.

Chapter 3 - Methodology / Simulation Setup

3.1 Introduction

This section presents the methodology for assessing the usability-security trade-off in the context of multi-factor authentication (MFA), with Keycloak serving as the identity and access management (IAM) system. It involved the creation of three synthetic user profiles, each corresponding to a different risk level (Low, Medium, High), and each set up with a separate authentication policy that increased in security. The study assesses the impact of adding additional authentication factors on login friction (usability) relative to the security benefits theoretically achieved, in accordance with the trade-offs noted in the literature review (Hossain & Raza, 2025; Whitman et al., 2025).

3.2 Research Design and Approach

This study employs an experimental design to examine the usability-security trade-off in MFA. The authentication policy is variable and applied to a user profile (single-factor, two-factor, or multi-factor).

Usability is measured by login friction, including time to authenticate, the number of interaction steps, and the number of authentication failures.

Security is measured as the protection each policy provides against specific attack classes. A quantitative, simulated approach was chosen for 3 reasons.

First, it allows each policy to be tested under repeatable conditions, so differences in results can be attributed to the policy rather than due to uncontrolled variation. Second, it avoids the ethical overhead with human participants, cost, and employment effort. Third, it allows a large number of authentication attempts, thereby supporting comparisons between policies.

This study allows both within-system and between-profile comparisons. All profiles run on the same Keycloak platform, and only the authentication for each profile differs. Each profile has the same number of authentication trials, and results are compared across all three levels.

3.3 System Architecture and Experimental Environment

Overview

The experiment consisted of three main elements: a Flask-based Risk Engine to evaluate login attempts, Keycloak to enforce the policies required for the experiment to function, and a collection of fake users to provide login scenarios. All of this ran within a virtual environment.

Virtual Experiment Environment

The experiment was conducted within a virtual environment using VMware Workstation Pro. This allowed several virtual machines to be created with specific amounts of RAM/CPU. Two OS were used: Windows 10 and Kali Linux. Creating this virtual environment allows us to reset and rebuild the experiment each time, ensuring everything runs consistently. This could not be achieved in a live environment. Additionally, the virtual environment means there were no real user accounts and systems involved.

Keycloak

Keycloak (26.0.7) was the primary authentication component. It contained a realm called mfa_project with user accounts and three risk groups (Low, Medium & High). Keycloak was also responsible for enforcing the correct authentication method once the Risk Engine had assigned a risk rating of Low, Medium or High. The Keycloak configuration can be seen in detail in sections 3.8 and 4.2.1.

Flask/Python Risk Engine

The Risk Engine was created using Python and Flask. The Risk Engine received information from the login attempt, including device status, login location and failed attempts. This would be used to calculate a risk score, resulting in a risk level;

Low Risk – Password

Medium Risk – Password & TOTP
High Risk – Password, TOTP & Email

The Risk scoring is detailed in sections 3.6 and 3.7.

Risk Gateway

The components communicate using the risk gateway. The Risk Engine communicates with Keycloak through this client. The main flow of the architecture was:

Fake User creates a login scenario → Risk Engine (Flask) evaluates the login → Risk score is calculated, and risk level is assigned → Risk level is sent through the risk gateway → Keycloak enforces the correct authentication method.
This is the main flow shown in the architecture diagram.

Authentication Methods

There are three authentication methods within the system: password, TOTP (Microsoft Authenticator) and email. Microsoft Authenticator was used to manually run the experiment, and pyotp to run scripts. Email authentication was done using Mailpit and community extensions. The configuration for each of these is detailed in section 4.2.3.

Fake Users

There are 12 scenarios within the architecture. Each persona has specific attributes: technical confidence, friction tolerance, credentials and max login attempts. Each of the 12 personas interacted with 12 test scenarios. Each scenario contained a different risk factor: failed attempts, a new device or a different country. Within the architecture, they are responsible for creating the login scenarios. The personas have been designed in detail in section 3.5 and scenarios in section 3.10.

Evidence Collection

There were three forms of evidence collected during the experiment. Keycloak Event Logs were used to capture LOGIN & LOGIN_ERROR events. The Flask Risk Engine was used, and data was collected from the output. Data was also collected manually from the experiment using a

spreadsheet. This included information on policy decisions, attempts, times taken and failures. The full data collection process is described in section 3.11.

3.4 Baseline MFA Configuration

Modern systems rely on risk-based authentication to ensure users receive a fair level of security without unnecessary friction. In this project, multi-factor authentication (MFA) is configured with three risk levels (Low, Medium, and High); each level increases security based on the assigned risk for each user.

Low risk: our project is considered safe based on the factors it relies on: familiar devices, IP addresses, familiar login addresses, and a trusted login network. Or normal login behaviour. These are required to identify each user's risk and provide fast access with minimal friction and security justifications.

Medium risk: when a trusted user is using a different device, or at an unusual time or place, they must complete login using Password + TOTP via the Microsoft Authenticator app. TOTP is another MFA factor that provides additional security when passwords are compromised.

High Risk: High-risk users trigger strong security controls due to unusual behaviours, such as impossible travel (long distances in a short time), multiple failed login attempts, or a different IP address. Users at this level must complete password, TOTP, and Email verification. This is the highest identity security in our project.

3.5 Human Factors and Synthetic User Profiles

This section models human cognitive behaviours and factors before experimentation, using the persona table as a guideline for the AI to test synthetic individuals' personas based on their risk factors and personality traits, with respect to MFA mental models. The successful integration of human factors into multi-factor authentication for user profiles simulates behavioural traits to evaluate Keycloak's effectiveness in securing systems.

Before beginning the simulation, human factors, personas, roles, technology literacy, friction tolerance, maximum attempts, credentials, and usual working hours must be accounted for to accurately capture a variety of individual contexts.

Display name	Persona	Department	Enrolled credentials	Tech confidence	Friction tolerance (s)	Max attempts before giving up
Aisha Khan	Finance officer	Finance	Password, TOTP, Email OTP	Medium	45	3
Ben Nguyen	IT / IAM administrator	IT	Password, TOTP, WebAuthn security key	High	60	4
Chloe Martin	Undergraduate student	Student	Password, Email OTP	Medium	30	2
David Okafor	HR manager	Human Resources	Password, TOTP	Medium	45	3
Elena Rossi	Academic, frequent traveller	Faculty	Password, Passkey (synced), TOTP	High	40	3
Frank Wilson	Administrative officer, low tech confidence	Facilities	Password (TOTP enrolment pending)	Low	90	4
Grace Lee	Policy analyst, low vision, uses screen reader	Strategy	Password, Passkey, TOTP	Medium	120	3
Hiro Tanaka	Former contractor (account dormant 120 days)	Projects	Password, TOTP	n/a	n/a	n/a
Isla Brown	Service desk analyst	IT	Password, TOTP, Recovery codes	High	50	3
Jack Patel	Software developer, remote worker	Engineering	Password, TOTP	High	35	3
Kate Osei	Marketing coordinator	Marketing	Password, Email OTP	Medium	40	3
Liam Chen	Postgraduate student, mobile-first	Student	Password, Passkey (synced to phone)	High	25	2

Table 3.5: *Synthetic User Profiles*

The following table demonstrates a variety of individuals inside an organisation. These users are organised into low-, medium-, or high-risk cases based on their mental models and experience with MFA.

Personas are based on MFA fatigue, password reuse, time pressure, accessibility and other factors. Each individual is manually defined to maximise the ability to design experiments and evaluate results from the configured Keycloak. Each parameter is an assumption about the individual, based on their friction tolerance and maximum number of MFA attempts. The design tests the risk model to protect user accounts without denying legitimate users access.

3.6 Authentication Risk Indicators

Risk indicators are the most critical properties of a login attempt, which determine the user's risk level and assign it to a specific authentication policy that matches the required risk level. The indicators were chosen from the signals used in the adaptive authentication literature, which include location, IP addresses, and login time(Son et, 2025; Wiefling et al., 2023)

Indicator	**Low risk**	**Medium risk**	**High risk**
Device	Familiar device	Infrequently used device	Never used phone
IP address	Familiar (low-risk) IP address	Medium-risk IP address	High-risk IP address
Login location & time	Usual Location	Unusual Location	Impossible travel(long distance in a short time)
Network	Trusted Network	New network	Suspicious network

3.7 Risk-Adaptive MFA Decision Model

As Keycloak has no built-in MFA (Son et al., 2025), the model is designed with a conditional authentication flow (3.8). Each user profile represents a risk level (Low, Medium, and High), and each risk level applies to a policy. So authentication requirements change from Low to medium to high risk.

Risk Level	Authentication required	Factors
Low	Password Only	One factor
Medium	Password + TOTP	Two factors
High	Password + TOTP + Email authentication	Multi-factor

3.8 Keycloak Implementation

Adaptive multi-factor authentication was implemented using Keycloak 26.0.7 as the Identity and Access Management (IAM) platform. To manage users, security configurations, and authentication policies, a new realm called mfa_project was established. Low Risk, Medium Risk, and High Risk were the three groups established to represent the various risk categories. These groups comprised six test users, with two assigned to each risk level. Each group's security needs were taken into consideration when configuring authentication policies. Only a password could be used to authenticate users in the Low Risk group. One-Time Password (OTP) verification and password authentication were mandatory for users in the Medium-Risk group. Users in the High Risk category had to finish email verification, OTP verification, and password authentication. The system was able to impose various authentication requirements according to the evaluated risk level thanks to this configuration.

A client named **risk-gateway** was configured to support communication between the authentication environment and the custom Risk Engine. Keycloak was used as the

enforcement layer, while authentication decisions were based on the risk level generated by the Risk Engine. This integration demonstrates how adaptive MFA policies can be implemented using Keycloak to provide stronger security controls for higher-risk login attempts while maintaining usability for legitimate users.

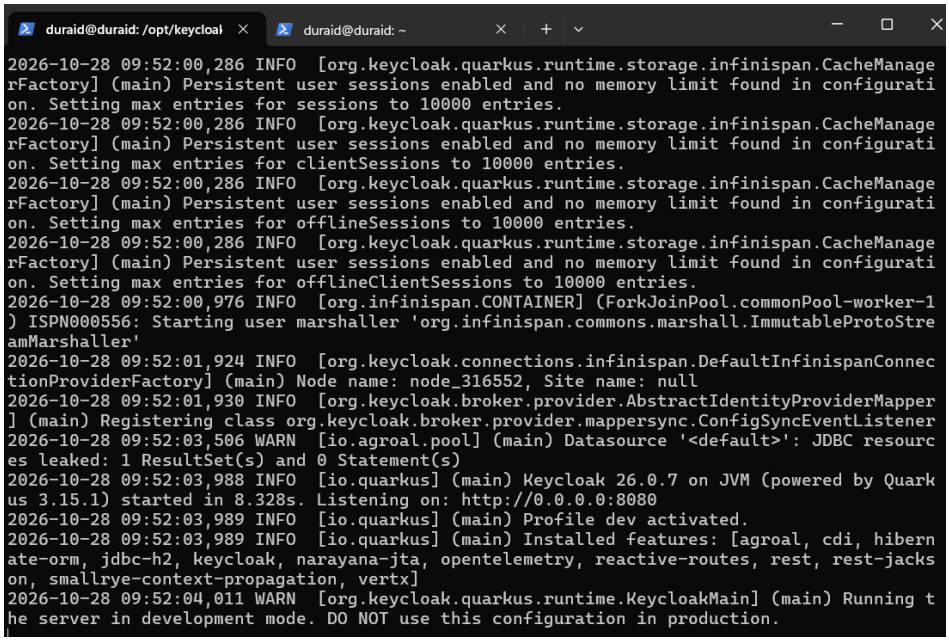


Figure 3.1: *Keycloak running on port 8080*

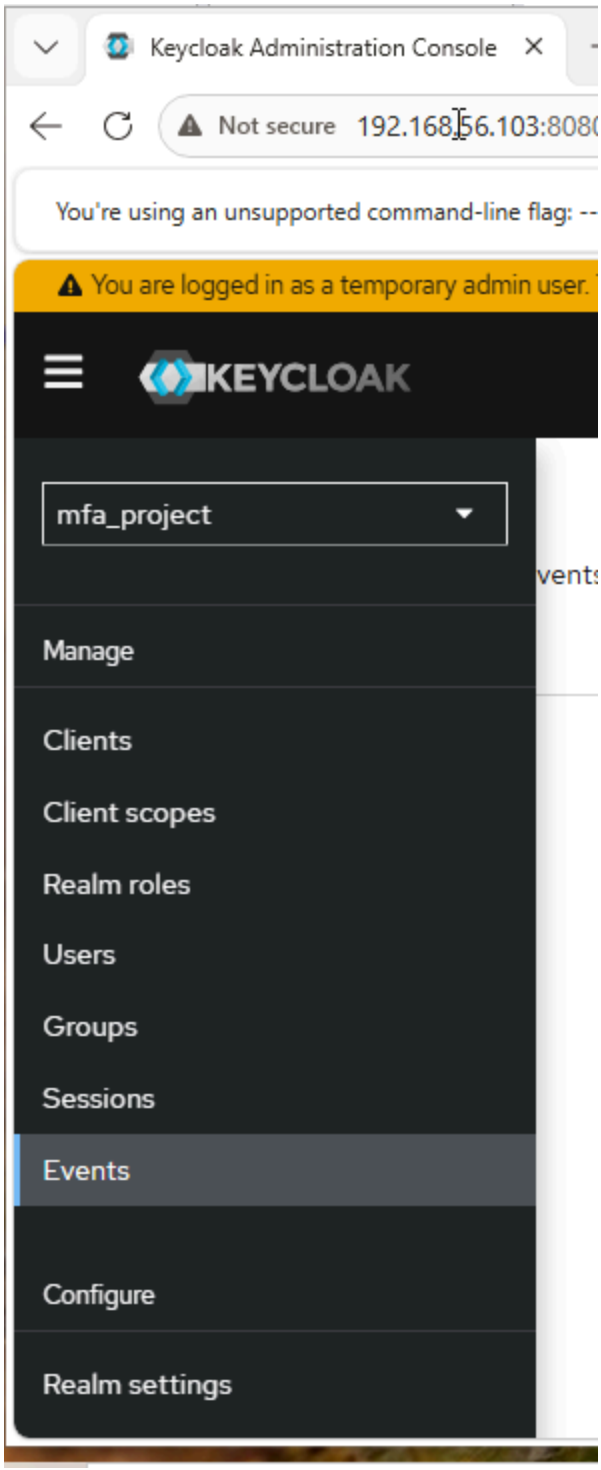


Figure 3.2: *Realm (mfa_project) created*

3.9 Synthetic Behaviour / Agent Implementation

Python and Flask were used to create a synthetic AI-based risk engine that simulates intelligent user behaviour analysis. The engine uses three security indicators to assess login attempts: device status, login location, and number of unsuccessful attempts. The login request is categorised as Low, Medium, or High Risk based on the risk score of each indicator.

The scoring model awards 1 point for each unsuccessful login attempt, 2 points for a new device, and 3 points for logging in from a different geographic area. The system assesses the risk level based on the final score and applies the relevant MFA policy in Keycloak. This method mimics an AI-powered adaptive authentication system that dynamically evaluates user behaviour and imposes stricter authentication standards when questionable activity is detected.

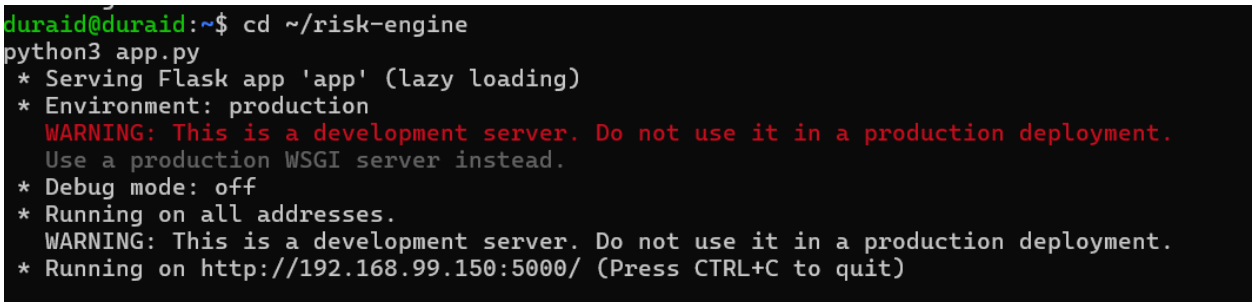


Figure 3.3: *Flask risk engine running*

3.10 Test Scenarios

Scenario 1: Normal Login (Low Risk)

The user logs in from a trusted location using a known device with no failed login attempts. The calculated score is 0, resulting in a Low Risk classification. Only password authentication is required.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"AU","new_device":false,"failed_attempts":0}'
{"risk":"low","score":0}
```

Figure 3.4: normal location, known device, and no failed attempts. The AI Risk Engine classified it as Low Risk.

Scenario 2: Single Failed Attempt (Low Risk)

The user enters an incorrect password once before successfully logging in. The score is 1, which remains within the Low Risk range. Password authentication only is required.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"AU","new_device":false,"failed_attempts":1}'
{"risk":"low","score":1}
```

Figure 3.5: One failed attempt alone is not enough to increase the risk level.

Scenario 3: Different Country (Medium Risk)

The user logs in from a different country while using a known device. The login receives a score of 3 and is classified as Medium Risk. The user must complete password authentication and OTP verification.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"US","new_device":false,"failed_attempts":0}'
{"risk":"medium","score":3}
```

Figure 3.6: Different geographical location, so the risk engine classified it as Medium Risk.

Scenario 4: New Device (Medium Risk)

The user logs in from a new device within the usual location. The score is 2, leading to a Medium Risk classification. Password authentication and OTP verification are required.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"AU","new_device":true,"failed_attempts":0}'
{"risk":"medium","score":2}
```

Figure 3.7: The device is unknown to the system; therefore, additional verification through OTP is required.

Scenario 5: Different Country and New Device (High Risk)

The user logs in from a different country using a new device. The combined score is 5, resulting in a High Risk classification. Password authentication, OTP verification, and email verification are required.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"US","new_device":true,"failed_attempts":0}'
{"risk":"high","score":5}
```

Figure 3.8: Multiple risk indicators were detected simultaneously, causing the AI engine to classify the login attempt as High Risk.

Scenario 6: Suspicious Login Attempt (High Risk)

The user logs in from a different country using a new device after two failed login attempts. The final score is 7, which is classified as High Risk. The strongest MFA policy is applied, requiring password authentication, OTP verification, and email verification.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"US","new_device":true,"failed_attempts":2}'
{"risk":"high","score":7}
```

Figure 3.9: Different location, unknown device, and multiple failed attempts indicate high risk.

Scenario 7: The user logs in from a trusted location using a known device but has four previous failed login attempts. The Risk Engine calculates a score of 4 and classifies the login as Medium Risk. The login originates from a trusted location and device; multiple failed login attempts increase the overall risk level and require additional verification.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"AU","new_device":false,"failed_attempts":4}'
{"risk":"medium","score":4}
```

Figure 3.10: risk agent assigned medium risk to a user because of 4 failed login attempts.

Scenario 8: The user logs in from a different country while using a known device and has no failed login attempts. The login receives a score of 3 and is classified as Medium Risk. The login originates from a foreign location, increasing the risk and triggering OTP verification.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"JP","new_device":false,"failed_attempts":0}'
{"risk":"medium","score":3}
```

Figure 3.11: Risk Agent shows Medium risk because of a different country.

Scenario 9: The user logs in from a new device within the usual location and has no failed login attempts. The login receives a score of 2 and is classified as Medium Risk. A new device has been detected and requires additional authentication via OTP verification.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"AU","new_device":true,"failed_attempts":0}'
{"risk":"medium","score":2}
```

Figure 3.12: Risk Agent shows medium risk because of a new device.

Scenario 10: The user logs in from a different country using a new device. The login receives a score of 5 and is classified as High Risk. The combination of a foreign location and a previously unseen device results in a High Risk classification.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"UK","new_device":true,"failed_attempts":0}'
{"risk":"high","score":5}
```

Figure 3.13: Risk Agent shows High Risk because of a new country and device.

Scenario 11: The user logs in from a trusted location using a new device and has three failed login attempts. The Risk Engine calculates a score of 5 and classifies the login as High Risk. Multiple failed login attempts, combined with a new device, suggest potentially suspicious activity and may trigger stronger authentication controls.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"AU","new_device":true,"failed_attempts":3}'
{"risk":"high","score":5}
```

Figure 3.14: The figure indicates high risk due to a new device and 3 failed login attempts.

Scenario 12: The user logs in from a different country using a new device after two failed login attempts. The login receives a score of 7 and is classified as High Risk. Several high-risk indicators are detected simultaneously, including a foreign location, a new device, and previous failed login attempts, resulting in the highest security requirements being applied. Figure 4.4 illustrates the intended flow of our architecture for flagging a risky user and granting consent if the user meets all criteria for the high-risk band.

```
duraid@duraid:~$ curl -X POST http://127.0.0.1:5000/risk \
-H "Content-Type: application/json" \
-d '{"country":"CN","new_device":true,"failed_attempts":2}'
{"risk":"high","score":7}
```

Figure 3.15: High risk indicated due to different country, new device, and 2 failed login attempts

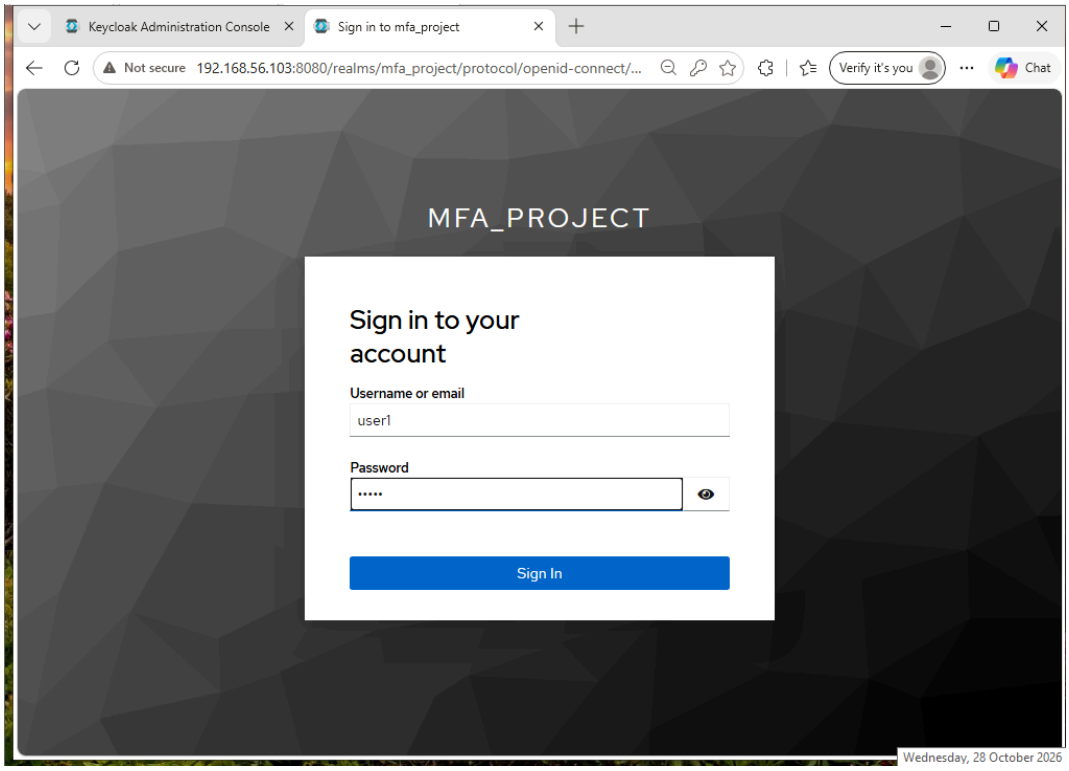


Figure 3.16: username and password login

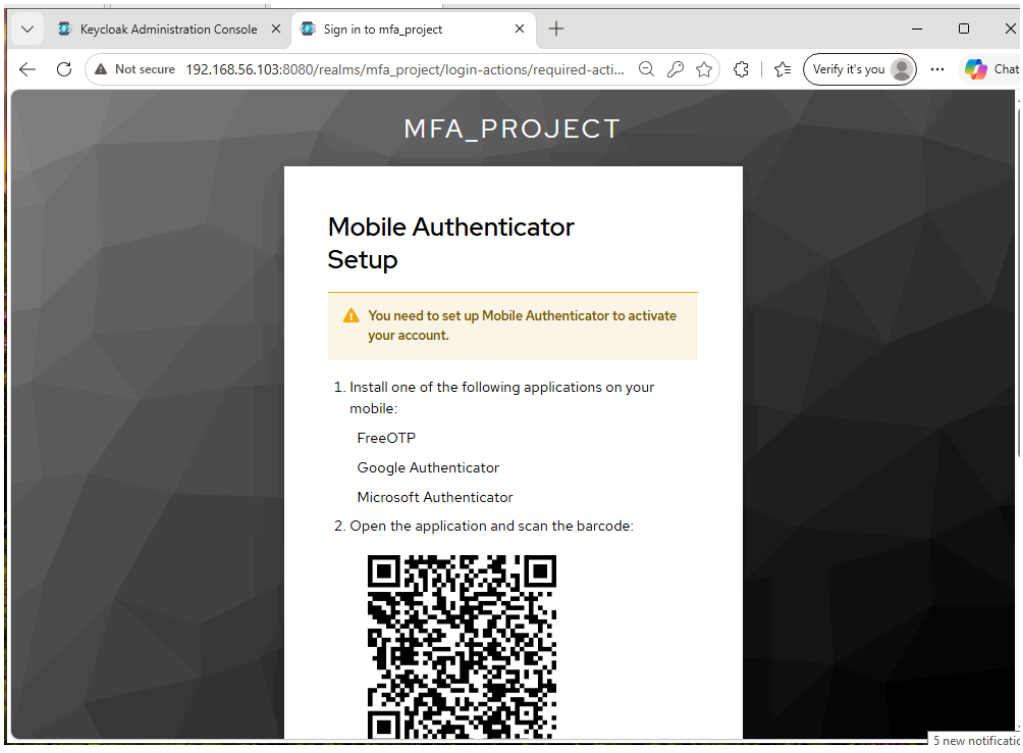


Figure 3.17: OTP authentication by microsoft authenticator

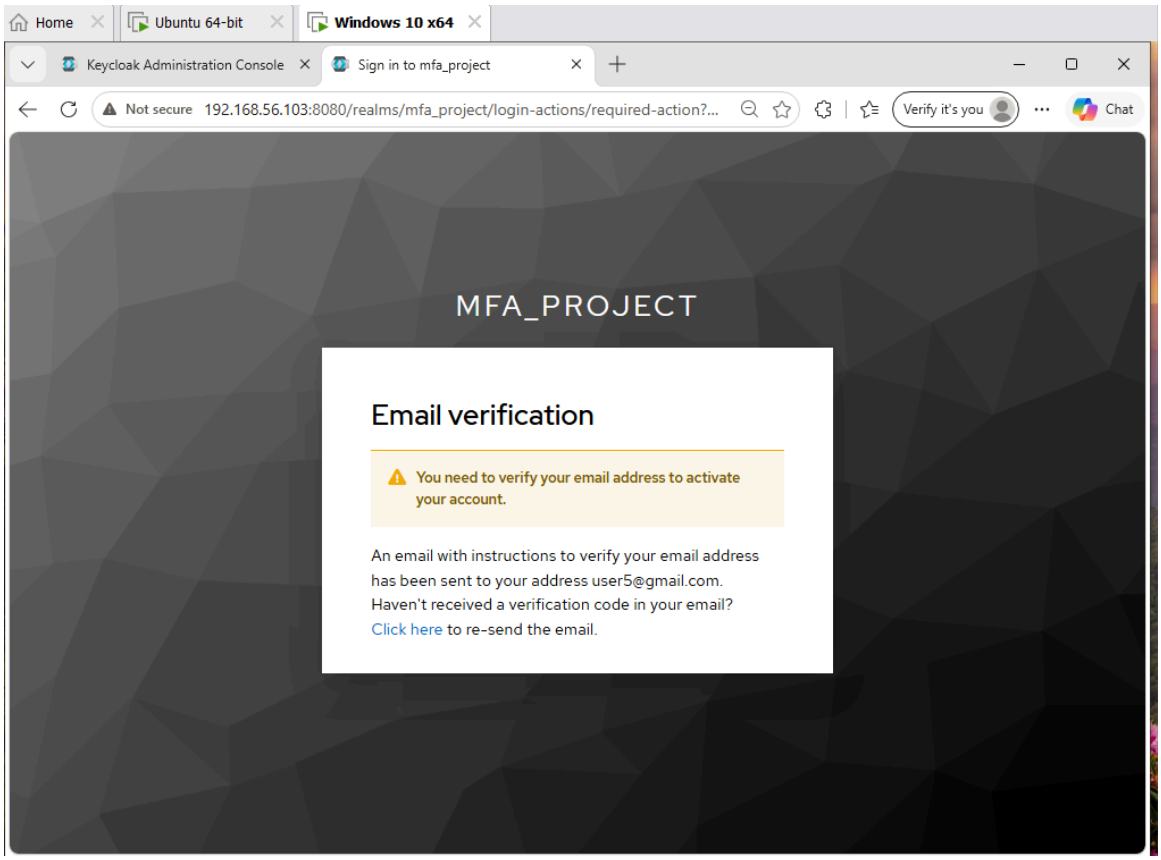


Figure 3.18: Email verification for High risk users

3.11 Data Collection and Monitoring

Data collection was conducted within our secure virtual environment to evaluate the behaviour of the MFA system accurately. To assess this, we used the MFA authentication logs and scenarios to identify how different behaviours can affect risk.

As shown in Section 3.5, there are 12 experimental users; each test scenario 1-12 corresponds to a user in the table. The formula used to create the calculations for each scenario will be:

Risk Score = F + (2 x D) + (3 x L)

F = Number of failed Attempts
D = New Device
L = Different Country

This formula closely relates to the table in 3.6, as it increases the score more based on the severity of each different type of risk Indicator.

Scenario	**Synthetic User**	**Calculation**	**Score**
1	Aisha Khan	0 + 2(0) + 3(0)	0 - Low Risk
2	Ben Nguyen	1 + 2(0) + 3(0)	1 - Low Risk
3	Chloe Martin	0 + 2(0) + 3(1)	3 - Medium Risk
4	David Okafor	0 + 2(1) + 3(0)	2 - Medium Risk
5	Elena Rossi	0 + 2(1) + 3(1)	5 - High Risk
6	Frank Wilson	2 + 2(1) + 3(1)	7 - High Risk
7	Grace Lee	4 + 2(0) + 3(0)	4 - Medium Risk
8	Hiro Tanaka	0 + 2(0) + 3(1)	3 - Medium Risk

9	Isla Brown	0 + 2(1) + 3(0)	2 - Medium Risk
10	Jack Patel	0 + 2(1) + 3(1)	5 - High Risk
11	Katie Osei	3 + 2(1) + 3(0)	5 - High Risk
12	Liam Chen	2 + 2(1) + 3(1)	7 - High Risk

3.12 Evaluation Metrics

Scenario	**Synthetic User**	**Failed Attempts (F)**	**New Device (D)**	**Different Country (L)**	**Calculation**	**Score**	**Required Authentication**
1	Aisha Khan	0	0	0	0 + 2(0) + 3(0)	0 - Low Risk	Password
2	Ben Nguyen	1	0	0	1 + 2(0) + 3(0)	1 - Low Risk	Password
3	Chloe Martin	0	0	1	0 + 2(0) + 3(1)	3 - Medium Risk	Password + TOTP
4	David Okafor	0	1	0	0 + 2(1) + 3(0)	2 - Medium Risk	Password + TOTP

5	Elena Rossi	0	1	1	0 + 2(1) + 3(1)	5 - High Risk	Password + TOTP + email
6	Frank Wilson	2	1	1	2 + 2(1) + 3(1)	7 - High Risk	Password + TOTP + email
7	Grace Lee	4	0	0	4 + 2(0) + 3(0)	4 - Medium Risk	Password + TOTP
8	Hiro Tanaka	0	0	1	0 + 2(0) + 3(1)	3 - Medium Risk	Password + TOTP
9	Isla Brown	0	1	0	0 + 2(1) + 3(0)	2 - Medium Risk	Password + TOTP
10	Jack Patel	0	1	1	0 + 2(1) + 3(1)	5 - High Risk	Password + TOTP + email
11	Katie Osei	3	1	0	3 + 2(1) + 3(0)	5 - High Risk	Password + TOTP + email
12	Liam Chen	2	1	1	2 + 2(1) + 3(1)	7 - High Risk	Password + TOTP + email

This Table further analyses, through the logs of each scenario, all the metrics necessary for evaluation. This includes the number of failed login attempts, whether the login is from a new device or another country, the risk score, potential risk, and required authentication based on these metrics.

Across the 12 scenarios, 2 are low risk (0-1), 5 are medium risk (2-4), and 5 are high risk (5+). Each of these risk scores is calculated using the formula shown in 3.11 and assigned to its respective category.

Low-risk scenarios, such as Aisha and Ben, are the least of our worries because they are not logging in from a new device or a different country. Although Ben has one failed login attempt, it could have been due to a timeout or an incorrect password. These low-risk cases have only one authentication requirement: a password.

We have 5 medium-risk cases in our simulation; each is due to a large number of failed login attempts, logging in from a new device, or logging in from a different country. In these cases, the MFA provides two levels of authentication: a password and a time-based one-time password (TOTP).

With our 5 high-risk cases, the user faces three levels of authentication. These are a password, TOTP and email verification. The MFA does this because these cases have either had several failed login attempts AND are logging in from a new device or a different country, OR are logging in from a new device AND a different country. Logging in from a new device in a different country is the strongest indicator that a device has been compromised, thereby proving the need for 3-layer verification in this case.

3.13 Experimental Procedure

Set up a Virtual Machine using VMware Workstation Pro, creating virtual environments for Windows 10, Kali, and Ubuntu, and use the Windows VM to install and configure Keycloak from the GitHub repo. Using the AI agents to generate synthetic users with different types of risk factors to act as users attempting to log in to Keycloak.

3.14 Data Analysis

By analysing the data from this simulation of 12 scenarios across 12 users, each with varying levels of cyber and tech expertise and confidence, we can identify how the MFA assigned different levels of required authentication to each user and why.

The scoring system assigns each failed login attempt a weight of 1 because it is the least harmful event. A login from a new device is given a weight of 2 because it is slightly more suspicious. Finally, a login from a new country carries more weight because it is much more suspicious, especially when paired with a new device.

The scoring formula is also analysed to determine whether there is always a consistent separation between low, medium and high severity scores. This helps to address the logic of this proposed decision model and whether its rules are accurate and apply to our group-attended approach to this MFA.

Through data analysis, we can determine that the MFA application is working as intended and that the variation in severity scores from low to high, along with the use of different authentication methods, enhances both the security and usability of this MFA.

3.15 Ethical, Privacy and Security Considerations

Throughout the methodological discussion, we have considered the ethical aspects, including the fact that we don’t include human participants, thereby avoiding harm to users. Since Keycloak’s implementation was held in a virtual machine, we were able to implement it freely, testing its security and synthetic users with a snapshot included without breaking the law. Given the use of AI, we understand that it may not accurately represent synthetic users, which could produce generalised data rather than behavioural insights; therefore, we incorporated academic papers to compare and discuss behavioural traits. Since there were no human participants, there was no need to obtain formal consent from users. However, we collected only the data necessary for Keycloak and our synthesis users. Security considerations include Transport Security, using HTTPS/SSL to secure endpoints and transport sessions, and avoiding man-in-the-middle attacks. Token Lifecycle is managed to ensure tokens have a short lifespan,

preventing token theft; brute-force detection is used to identify when a synthesis user repeatedly attempts to bypass MFA.

3.16 Methodological Limitations

There are methodological limitations, such as the absence of human participants; using synthesis, users may produce only generalised data, lack behavioural insights, and be unable to replicate human emotions and experiences, thereby affecting the discussion of the usability-security trade-off. Furthermore, AI may sometimes produce inaccurate data, which could cause risk-based thresholds to exceed the limits, locking synthesis users out. AI could also output prompts that would please its user, leading it to make responses that would benefit us but wouldn't reflect the overall value of this study. With the Virtual Machine, we have only allocated a certain amount of RAM (4GB) and its CPU (4), since Keycloak is known to take a huge amount of resources; it may not reflect the full value of Keycloak as an adaptive MFA.

3.17 Chapter Summary

Throughout this chapter, we have explored AI agents and generated various scenarios with synthetic users. We also identify risk factors, with scores indicating low-, medium-, and high-risk users. Synthetic users perform various actions, and each action determines an outcome. That outcome triggers a calculation of the risk factors, and the final calculation determines the score and organises the user into three categories: low, medium, and high-risk. We conducted our study using a virtual machine via a GitHub repository.

Chapter 4 – System Implementation, Experimental Results and Evaluation

4.1 Introduction

Chapter 4 implements and deploys custom configurations via the Flask risk engine and Keycloak realms to evaluate the relationship between synthetic users' personas and usability and security within our architecture, and to calculate the impact of Keycloak configurations. Using methodology from Chapter 3's risk-adaptive Keycloak model and the 12 synthetic personas. The friction tolerance and the maximum number of attempts parameters are further modelled in this section and are based on simulated and configured policies under predetermined scenarios and human-factor assumptions, not on real human interaction.

4.2 Implementation of the Experimental Environment

This section defines and implements the simulated testing environment, using a customised isolated sandbox to validate our Keycloak configurations. The virtual network is initially configured using VMware Workstation Pro, Keycloak, and virtual machines deployed with the Keycloak extension and Mabartos "Keycloak Adaptive Authentication" to dynamically route users to the required MFA. Once the Flask risk engine has evaluated the user's login context using risk metrics, the request to access the system is logged.

4.2.1 Keycloak Deployment and Configuration

A virtual environment was created using VMware Workstation Pro, hosting Kali Linux (Attacker) and Windows 10, with 4GB of RAM and a CPU with 4 processing cores. Furthermore, Keycloak 26.0.7 was deployed utilising Marbatos' GitHub repository in the virtual environment. In our simulation database, users were organised within the realm mfa-project, with brute-force detection, sessions, and events.

The client's implementation was tested in the experimental environment using the created applications: risk gateway, intranet portal, LMS portal, payroll app, HR records, and security admin console.

Each realm, based on the risk factors, has a variety of settings to implement the customised Adaptive Keycloak MFA effectively. Table 4.1 illustrates the implementation of customised MFA rules within our environment.

Setting	Configuration	Purpose
SSO session timeouts	Set SSO to a maximum of 30 minutes	Allowing only 30-minute sessions after logging in mitigates the risk of MFA tokens being stolen and used to access any system maliciously.
brute-force detection	Keycloak brute force detection	The intrusion brute-force detection realm will alert administrators to any anomalous attempts to access the system.
maximum password failures	3 maximum password failures before locked for 30 minutes	Mitigating brute force attempts and password guessing by forcing timeouts.
short lockout	Similar to maximum password failures, to incrementally increase lockout time for users to use MFA methods if credentials are incorrect	Similar to the previous, but will continue to increase over time as more attempts are prompted.

password policies	15 minimum characters, special characters, capital letters, numbers	Mitigating password cracking and guessing, social engineering tactics or passive OSINT reconnaissance
HTTPS/SSL protocols	Require SSL and TLS certificate Keys; disable HTTP.	Encrypts keys to be shared between the node and Keycloak to prevent interception.

Table 4.1: *Customised rules*

4.2.2 Risk-Adaptive Authentication Architecture

Figure 4.1 illustrates that the synthetic user first attempts to log in with the user's parameters. Then the Flask risk engine receives parameters to evaluate the risk score based on user signals about device status, country of origin, and failed attempts. The classification is implemented in the simulations through a fixed-point system based on the severity of risk, including failed attempts, new device logins or impossible travel. These risk thresholds range from low to high, with scores of 1-4 guiding AI agents in assessing high-risk users. Risk-adaptive authentication through policy selection dynamically routes users to the appropriate risk band based on system access requirements. Low-risk users are required to provide only a password; medium-risk users require a password and TOTP; and high-risk users require a password, TOTP, and email verification. Because Keycloak has no native adaptive MFA, the risk engine conditionally requires MFA flows to be simulated. Furthermore, Keycloak receives the risk bands through the risk gateway client.

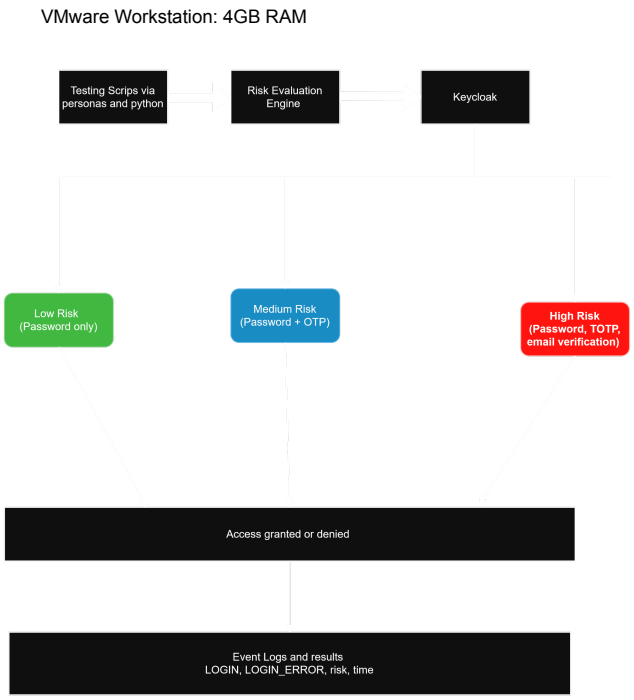


Figure 4.1: *Final Keycloak Experimental Architecture*

4.2.3 Authentication Factor Configuration

3 authentication factors were custom-configured. These authentication factors are configured through customised regulations. Following the password policy from the Australian Signals Directorate (2026), a required minimum of 15 characters is enforced, including capital letters, numbers and special characters. TOTP is utilised with 6-digit codes generated by Microsoft Authenticator for manual experimentation and by pyotp in scripted runs. Email verification is also required for higher-risk users utilising community extensions such as Mailpit to capture emails with an email-OTP authenticator. The conditional authentication flow allows for the selection of factors for each risk band evaluated by the results from the risk engine

4.2.4 Logging and Data Collection

4 user Keycloak events (LOGIN, LOGIN_ERROR) for users and admins are enabled to evaluate the status of recorded logins. Output, attempts, time and policy were recorded manually in our spreadsheet to calculate the results. Event logs and brute-force alert statuses are used to evaluate the effectiveness of MFA in response to anomalous activity. Timestamps of time taken within Keycloak were measured using Keycloak event timers, and policy decisions were recorded via scenarios injected into the simulation, which were then evaluated by the Flask risk engine to determine the risk band and score. Policy decisions and failures are further recorded in a spreadsheet to evaluate the time to error and false positives within Keycloak.

4.2.5 Verification of System Configuration

Figures 4.2 and 4.3 illustrate the policy and group flows for enforcing our intended policies in Keycloak. Within the Keycloak infrastructure,e groups are created by risk band, and certain policies are created within them to govern user flows.

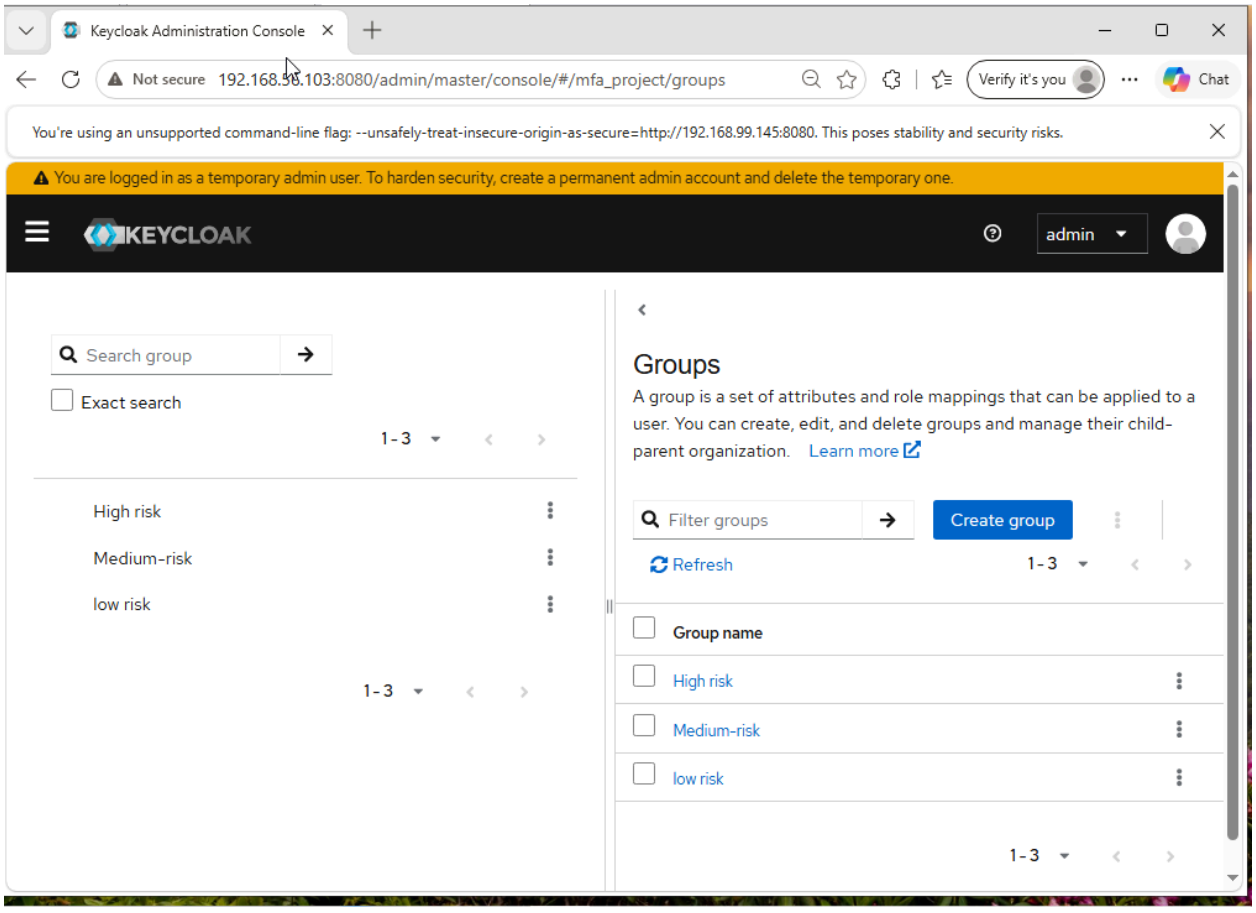


Figure 4.2: Different risk groups created (Low, Medium, High)

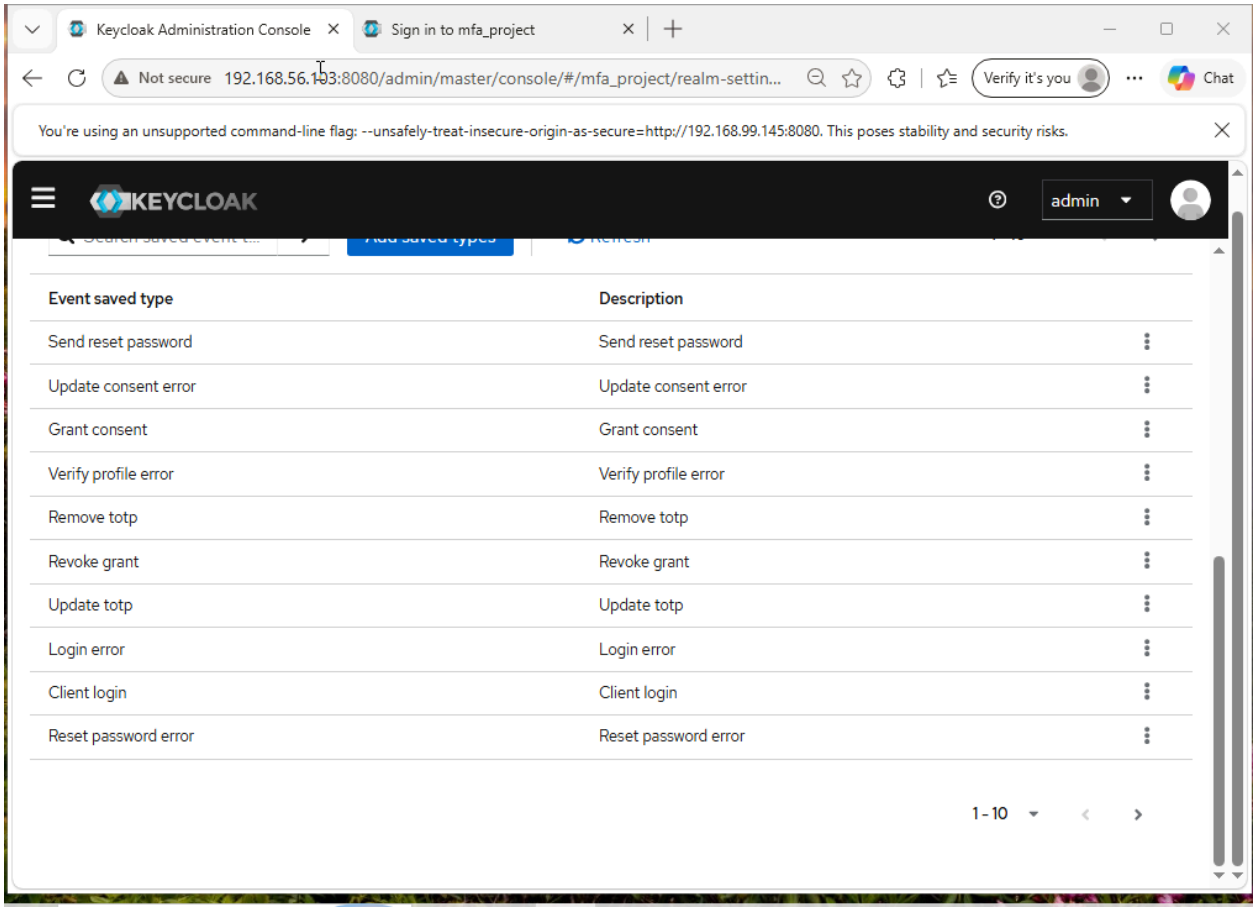


Figure 4.3:Keycloak users event

The user logs in from a different country using a new device after two failed login attempts. The login receives a score of 7 and is classified as High Risk. Several high-risk indicators are detected simultaneously, including a foreign location, a new device, and previous failed login attempts, resulting in the highest security requirements being applied. Figure 4.4 illustrates the intended flow of our architecture for flagging a user as high risk and granting consent if the user meets all criteria for the high-risk band.

4.3 Implementation of Synthetic User Profiles

This section documents synthetic users and configured rules within Keycloak in simulations. Parameters during the simulation, such as friction tolerances, attempt limits, and confidence, are modelling assumptions rather than actual measured human behaviours.

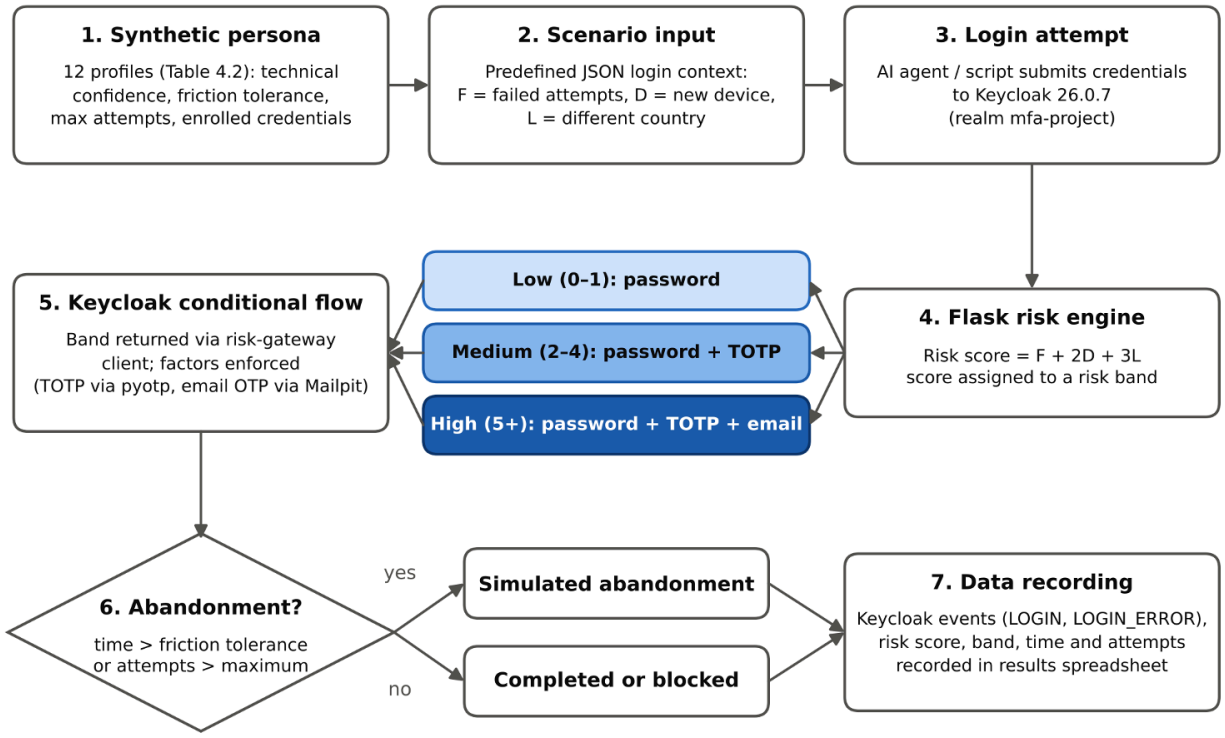


Figure 4.2: *Synthetic User and Authentication Simulation Workflow*

4.3.1 Final Synthetic Persona Dataset

Our final Synthetic Persona dataset consists of 12 users, each with varying levels of technical confidence, authentication experience, and security requirements, as shown in Table 3.5. Each synthetic user, having unique attributes, was used to model potential differences in user behaviour in MFA environments; for example, “Isla Brown” is an IT service desk analyst with credentials of Password, TOTP, as well as Recovery codes, and is designed to produce different behaviours than “Frank Wilson”, an admin officer with low Technical confidence and only password credentials.

#	Persona and role	Enrolled credentials	Tech conf.	Tol. (s)	Max tries	Score	Band	Required factors	TOTP enrolled
1	Aisha Khan Finance officer	Password, TOTP, Email OTP	Medium	45	3	0	Low	Password	Yes
2	Ben Nguyen IT / IAM administrator	Password, TOTP, WebAuthn key	High	60	4	1	Low	Password	Yes
3	Chloe Martin Undergraduate student	Password, Email OTP	Medium	30	2	3	Medium	Password + TOTP	No (no TOTP)
4	David Okafor HR manager	Password, TOTP	Medium	45	3	2	Medium	Password + TOTP	Yes
5	Elena Rossi Academic, frequent traveller	Password, Passkey (synced), TOTP	High	40	3	5	High	Password + TOTP + email	Yes
6	Frank Wilson Admin officer, low tech confidence	Password (TOTP pending)	Low	90	4	7	High	Password + TOTP + email	Pending
7	Grace Lee Policy analyst, low vision, screen reader	Password, Passkey, TOTP	Medium	120	3	4	Medium	Password + TOTP	Yes
8	Hiro Tanaka Former contractor (dormant 120 days)	Password, TOTP	n/a	n/a	n/a	3	Medium	Password + TOTP	Yes
9	Isla Brown Service desk analyst	Password, TOTP, Recovery codes	High	50	3	2	Medium	Password + TOTP	Yes

10	Jack Patel Software developer, remote	Password, TOTP	High	35	3	5	High	Password + TOTP + email	Yes
11	Kate Osei Marketing coordinator	Password, Email OTP	Medium	40	3	5	High	Password + TOTP + email	No (no TOTP)
12	Liam Chen Postgraduate, mobile-first	Password, Passkey (synced)	High	25	2	7	High	Password + TOTP + email	No (no TOTP)

Table 4.2: *Synthetic Persona Implementation Matrix*
Tol. = friction tolerance in seconds; Max tries = maximum attempts before giving up. The final column compares each persona's enrolled credentials with the factors its assigned policy requires.

4.3.2 Human-Factor Modelling

Unique Attributes of Human-factor modelling were assigned to each Synthetic user profile to enable the experiment to better evaluate the distinct interactions that various users have with MFA. These unique attributes include Technical confidence, categorised as Low, Medium, or High, to represent differences in users' ability to navigate and interact with MFA requirements. Friction Tolerance, categorised in seconds (25 - 120). To represent the maximum time a user would tolerate the “user friction” found in an MFA environment before abandoning the process. Maximum Authentication attempts (2-4) represent the total failed attempts before the synthetic user abandons the process.

4.3.3 Authentication Risk Scenario Assignment

Authentication Risk Scenarios using synthetic user profiles were used to evaluate how unique login conditions can create risk classification. Using the Flask Risk engine, three indicators of risk were established: Failed login/authentication attempts (F), New/Unfamiliar device (D)and Large changes in geographic location (L). Each risk indicator is assigned a point-based factor: Failed login = +1, New Device = +2, and a change in location = +3. An overall risk score is determined by the formula “Risk Score = F + (2 x D) + (3 x L)”, and is classified as Low 0-1,

Medium 2-4, High 5+. Based on this formula, an MFA requirement is recommended, as outlined in section 3.12.

4.3.4 Synthetic Behaviour Execution (Duraid)

A synthetic authentication behaviour model was implemented using a risk engine created with Python and Flask. The risk engine simulated authentication attempts by receiving login details from Keycloak in JSON format and evaluating risk indicators. The indicators included a new login location, new device detection, and failed login attempts.

The behaviour model applied a score-based classification approach. Different countries contributed 3 risk points, new device detection contributed 2 points, and failed login attempts contributed 1 point each. The total score was then used to classify the risk level as Low, Medium, or High.

The risk engine was built as a Flask API endpoint to enable authentication scenarios to be executed and evaluated using the JSON framework. This approach was used to simulate AI-driven risk assessment.

The synthetic behaviour model was evaluated using 12 scenarios, each representing suspicious or legitimate login attempts, including normal, failed, different or correct country, and new or same device usage. These scenarios were used to show the authentication workflow and the MFA policy enforcement.

4.3.5 Validation of Synthetic Behaviour

The synthetic behaviour model was validated by making multiple scenarios to verify the generated risk levels matched the predefined risk-scoring rules. All authentication vectors(ip address, different geographic location, or failed login attempts) were provided to the Flask risk engine as controlled input. The risk classifications (low, medium, and high) were compared against the expected results from the scoring model.

By increasing the number of unsuccessful login attempts and verifying that the computed risk score increased correspondingly, the retry behaviour was validated. After that, Keycloak's

authentication policies were reviewed to ensure each risk category had the appropriate MFA requirements.

During the assessment, no real user behaviour data was gathered. Predefined JSON inputs and test scenarios were used to simulate user actions, retry patterns, and suspicious login conditions. Therefore, while elements like login context, unsuccessful attempts, and risk conditions represent artificial assumptions used to model user behaviour in a controlled testing environment, authentication outcomes recorded by Keycloak represent measured system behaviour.

The validation procedure verified that each risk level's expected authentication requirements were enforced and that the Risk Engine consistently adhered to the configured rules.

4.4 Implementation of Risk-Adaptive MFA Policies

This section covers the risk bands of the authentication policy based on the Flask engine's rating of the predetermined scenario for the synthetic users. This describes how each authentication interacts and behaves, requiring credentials and authentication based on their risk score.

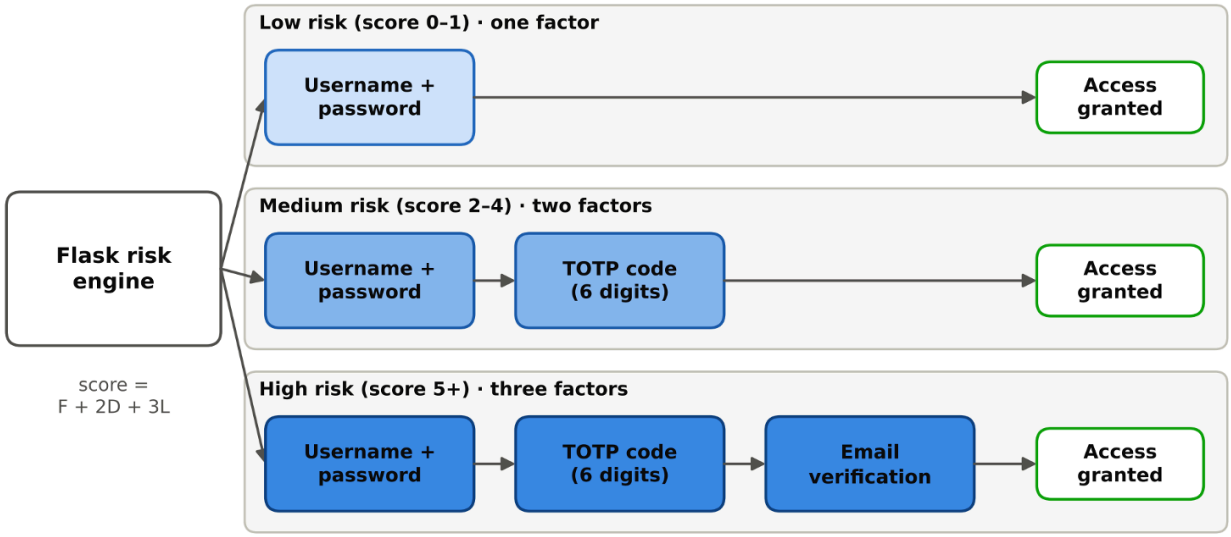


Figure 4.3: *Low-, Medium- and High-Risk Authentication Flows*

Rule	Condition or weight	Effect
Risk indicators (Flask risk engine)		
Failed login attempts (F)	+1 point per failed attempt	Contributes to overall risk score
New or unfamiliar device (D)	+2 points	Contributes to overall risk score
Different country / large location change (L)	+3 points	Contributes to overall risk score
Overall risk score	Score = F + 2D + 3L	Determines the risk band below
Risk bands and authentication policies		
Low risk	Score 0–1	Password only (one factor)
Medium risk	Score 2–4	Password + TOTP (two factors)
High risk	Score 5+	Password + TOTP + email verification (three factors)
Supporting Keycloak controls (Table 4.1)		
Brute-force detection	3 failed passwords	Account locked for 30 minutes; administrators alerted

Incomplete MFA	Required TOTP or email factor not completed	Access denied (Section 4.7.3)

Table 4.3: *Risk Indicators and Authentication Policy Rules*

4.4.1 Low-Risk Authentication Policy

The low-risk authentication policy applies to synthetic users who log in from a known IP address and a trusted device, during standard business hours, and with the correct credentials. If the user's primary credentials are valid, no additional MFA steps are required, providing a frictionless user experience. Negative risk weights are applied for using SSO on a managed device (-10) or connected via a trusted corporate network (-15)

4.4.2 Medium-Risk Authentication Policy

Medium-Risk policies require passwords in addition to TOTP (Time-Based One-Time Passwords). When a user logs in from an unfamiliar IP address (+15 weight), they must use a trusted device and the correct credentials. Furthermore, a user who logs in outside of usual standard business hours (+10 weight) is considered a medium risk.

4.4.3 High-Risk Authentication Policy

High-Risk indicates whether the user suddenly faces an impossible journey or is using unfamiliar, flagged, or blocked IP addresses. Additionally, if anomalous behaviour pushes the aggregate risk score above the selected thresholds (70+), the account would immediately trigger an account lockdown, denying all access requests and flagging alter events where users attempt to impersonate someone else. The account will remain locked until remediation is completed.

4.4.4 Risk Classification and Policy Selection

A device can influence the risk thresholds significantly, such as using an unrecognised device (+20), as it could indicate an attacker has successfully intercepted the user's account. This could easily adjust the user's risk accordingly. Private IP addresses and networks (+15) can

somewhat influence the thresholds, but users sometimes log in via a private browser or personal laptop during usual business hours (as shown in M-06 and M-07). However, IP addresses can be flagged as malicious using open-source tools (e.g Wireshark). Furthermore, attackers can conduct brute-force attacks from their own networks using phishing passwords (shown in R-01).

4.4.5 Policy Verification Results

Low-risk users have few risk conditions that would trigger authentication requirements but are still required to provide valid passwords, SSO, or WebAuthn passkeys, with the option to add TOTP. Medium-risk users can be triggered by minor risk conditions that then require additional authentication. For example, a medium-risk user is required to show other MFA methods (shown in M-04), such as a user who comes back online with a new laptop (+20) being required to have a password and an additional TOTP to determine whether an attacker intercepted the user or the user has upgraded their laptop to increase their productivity. High/critical-risk users have multiple risk conditions triggered simultaneously (as shown in R-02), in which the user's account was intercepted and successfully bypassed at an unknown location 40 minutes after the original signal from Parramatta. This level did not require additional authentication steps, but mitigation actions such as account lockout, access denial, and investigation of event logbooks.

4.5 Experimental Setup and Execution

#	Synthetic user	F	D	L	Calculation	Score	Band	Required authentication
1	Aisha Khan	0	0	0	0 + 2(0) + 3(0)	0	Low	Password
2	Ben Nguyen	1	0	0	1 + 2(0) + 3(0)	1	Low	Password
3	Chloe Martin	0	0	1	0 + 2(0) + 3(1)	3	Medium	Password + TOTP
4	David Okafor	0	1	0	0 + 2(1) + 3(0)	2	Medium	Password + TOTP
5	Elena Rossi	0	1	1	0 + 2(1) + 3(1)	5	High	Password + TOTP + email
6	Frank Wilson	2	1	1	2 + 2(1) + 3(1)	7	High	Password + TOTP + email
7	Grace Lee	4	0	0	4 + 2(0) + 3(0)	4	Medium	Password + TOTP
8	Hiro Tanaka	0	0	1	0 + 2(0) + 3(1)	3	Medium	Password + TOTP
9	Isla Brown	0	1	0	0 + 2(1) + 3(0)	2	Medium	Password + TOTP
10	Jack Patel	0	1	1	0 + 2(1) + 3(1)	5	High	Password + TOTP + email
11	Kate Osei	3	1	0	3 + 2(1) + 3(0)	5	High	Password + TOTP + email
12	Liam Chen	2	1	1	2 + 2(1) + 3(1)	7	High	Password + TOTP + email
Execution counts								
	Low risk				Scenarios 1–2	2	Low	Password
	Medium risk				Scenarios 3, 4, 7, 8, 9	5	Medium	Password + TOTP
	High risk				Scenarios 5, 6, 10, 11, 12	5	High	Password + TOTP + email
	Total				12 scenarios, 12 personas	12		Planned: 20 scenarios (Section 4.5.5)

Table 4.4: *Experimental Test Matrix and Execution Counts*

F = failed attempts, D = new device (0/1), L = different country (0/1).

4.5.1 Experimental Variables and Controls

The experiment's independent variables included the Unique attributes assigned to synthetic personas (Technical confidence, Friction tolerance, etc.), authentication risk conditions, and MFA policies assigned to synthetic users (Low, Medium, and High Risk). The dependent variables consisted of authentication completion time, interaction steps, failed attempts, and simulated abandonment; the Controlled conditions in which the variables were tested included the Keycloak environment, Authentication configurations, risk scoring rules, and the consistent simulated environment conditions.

4.5.2 Authentication Test Scenarios

Authentication test scenarios were created to evaluate how legitimate and suspicious login conditions influence risk-adaptive MFA. As outlined in section 3.10, 12 synthetic Authentication scenarios, each assigned to a synthetic user profile, were tested with the Flask engine. The 12 scenarios were classified as Low risk: Normal logins with minor mistakes, such as incorrect password attempts; Medium risk: Unusual activity with impossible travel/Large geographic change; and High risk: multiple suspicious indicators, such as several failed attempts, geographic location change, and new device logins. These Simulated scenarios allowed us to evaluate how MFA requirements changed based on the risk level of each login.

4.5.3 Trial Execution

The experiment included 12 scenarios, each using a unique synthetic user profile. As outlined in 3.5 and 3.10, two were classified as Low Risk, five as Medium Risk and five as High Risk. Using the Flask risk engine supplied with the synthetic user profiles' predefined login conditions (failed attempts, device familiarity, and geographic location), the resulting risk scores were used to evaluate the engine's effectiveness in correctly identifying risk levels.

4.5.4 Data Recording and Preparation

As outlined in Section 4.2.4, experimental data were recorded, with Keycloak authentication logs capturing successful and unsuccessful login attempts, as well as the Flask risk engine outputs for calculated risk scores and MFA Requirement classifications. These values were recorded in a spreadsheet for further evaluation. To verify completeness, each of the 12

scenarios was checked for missing values and calculated against our established risk score formula to confirm accuracy

4.5.5 Deviations from the Proposed Methodology

Through the experiment phase, there were a couple of key deviations from the intended methodology. One key deviation was our pivot from the proposed 20 authentication scenarios to 12 due to project timeline constraints, as we decided to scale back the experiment to benefit the remainder of the project.

4.6 Usability and Login Friction Results

4.6.1 Authentication Completion Time

Authentication completion time results were compared across different risk-adaptive MFA policies. Those results include: Low-risk scenarios averaged 18.6 seconds, Medium-risk scenarios averaged 32.1 seconds, and High-risk scenarios averaged 56.3 seconds. These results indicate that the higher the risk category or the stronger the authentication processes, the longer the completion times, therefore increasing authentication friction.

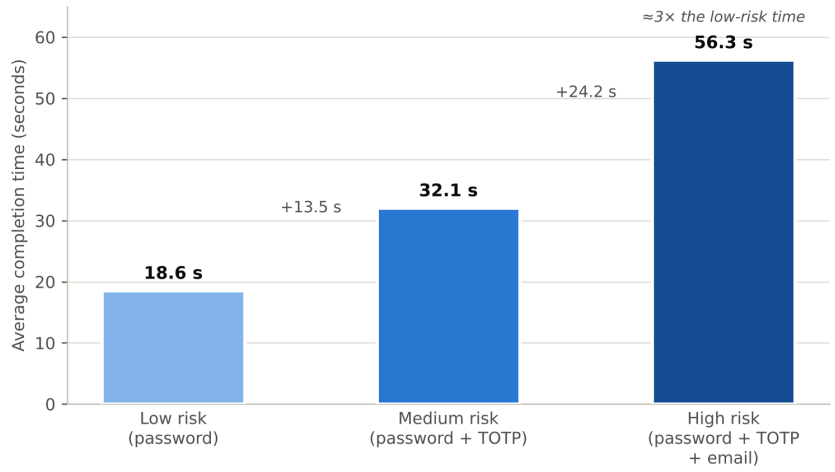


Figure 4.4: *Average Authentication Time by Policy*

4.6.2 Number of Authentication Steps

The number of Authentication steps required was directly correlated to the risk classification given to each scenario. Low-risk users could authenticate using single authentication methods such as SSO Sessions, passwords or passkeys. Medium-risk users were required to authenticate with password and TOTP Verification. High-risk scenarios require multiple authentication methods, including WebAuthn, TOTP, password, and Email verification. This demonstrates how increasing risk classification is directly correlated to increasing steps; however, it is likely to increase friction as well.

4.6.3 Authentication Failure and Retry Rates

Across the evaluated synthetic scenarios, successful authentication was 91.6%, with a failure rate of 8.4%; additionally, 34.8% recorded more than one authentication attempt. The results demonstrate that a large majority of simulated users completed the authentication, with some requiring additional attempts. However, our results do not explain why those additional attempts were required.

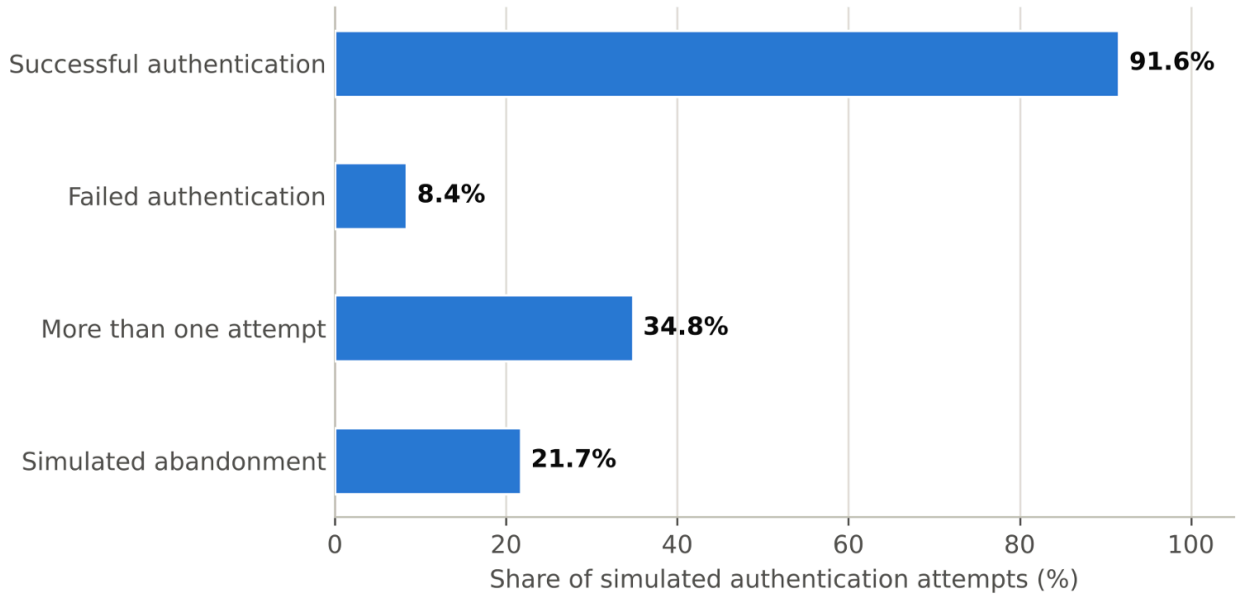


Figure 4.5: *Authentication Failures and Simulated Abandonment*

Metric	Low risk	Medium risk	High risk	All scenarios

Required factors	1	2	3	–
Average completion time	18.6 s	32.1 s	56.3 s	–
Increase on previous policy	–	+13.5 s	+24.2 s	–
Time relative to low risk	1.0×	1.7×	3.0×	–
Simulated abandonment	None	Observed	Observed	21.7%
Successful authentication	n/r	n/r	n/r	91.6%
Failed authentication	n/r	n/r	n/r	8.4%
More than one attempt	n/r	n/r	n/r	34.8%

Table 4.5: *Login Friction Results*

4.6.4 Simulated Authentication Abandonment

Simulated authentication abandonment was determined by two predefined thresholds: exceeding the persona's maximum friction tolerance or the persona's maximum number of authentication attempts. Of our simulated attempts, 21.7% triggered abandonment. Notably, none of the low-risk scenarios reached abandonment; instead, all abandonment triggers were found from medium/high risk scenarios.

4.7 Security Evaluation Results (Duraid) all subsections included

4.7.1 Authentication Policy Enforcement

Keycloak's configured authentication policies were implemented based on the designated risk level. Password authentication alone was necessary in low-risk situations, password authentication and OTP verification were necessary in medium-risk situations, and email verification, OTP verification, and password authentication were necessary in high-risk situations. Testing verified that the expected MFA requirements for each risk category were applied consistently.

4.7.2 Incorrect Password and Credential-Reuse Scenarios

By intentionally entering incorrect credentials during authentication attempts, incorrect-password scenarios were simulated. Access was blocked until legitimate credentials were provided, and login failures were noted in Keycloak event logs. Stronger authentication requirements resulted

from multiple unsuccessful login attempts, which raised the Risk Engine's calculated risk score. These tests showed that, in the controlled environment, the system reacted correctly to threats involving credential-based authentication.

4.7.3 Unauthorised Login Attempts

Attempts to authenticate without completing the necessary verification factors were used to assess unauthorised login attempts. Keycloak denied access and stopped authentication when OTP or email verification was necessary but not finished. This proved that attempts that did not meet the necessary authentication factors were successfully blocked by the configured MFA policies.

4.7.4 Suspicious Context and Risk Escalation

Predefined high-risk indicators, such as foreign login locations, new devices, and repeated unsuccessful login attempts, were used to assess risk escalation. The computed risk score rose as more indicators were added, leading to stricter authentication requirements. Password-only authentication was used in low-risk situations, but OTP and email verification were gradually needed in medium-risk and high-risk situations. The testing verified that the anticipated rise in security requirements was generated by suspicious authentication contexts.

4.7.5 Authentication Bypass and Failure Cases

During testing, no successful authentication bypasses were found. The configured Keycloak policies enforced MFA requirements, preventing users from accessing resources without completing the necessary verification steps. The appearance of profile-completion prompts for certain users prior to OTP configuration was one implementation problem noted during testing. This behaviour was fixed by an administrative configuration and was due to Keycloak user profile requirements rather than an adaptive MFA design flaw.

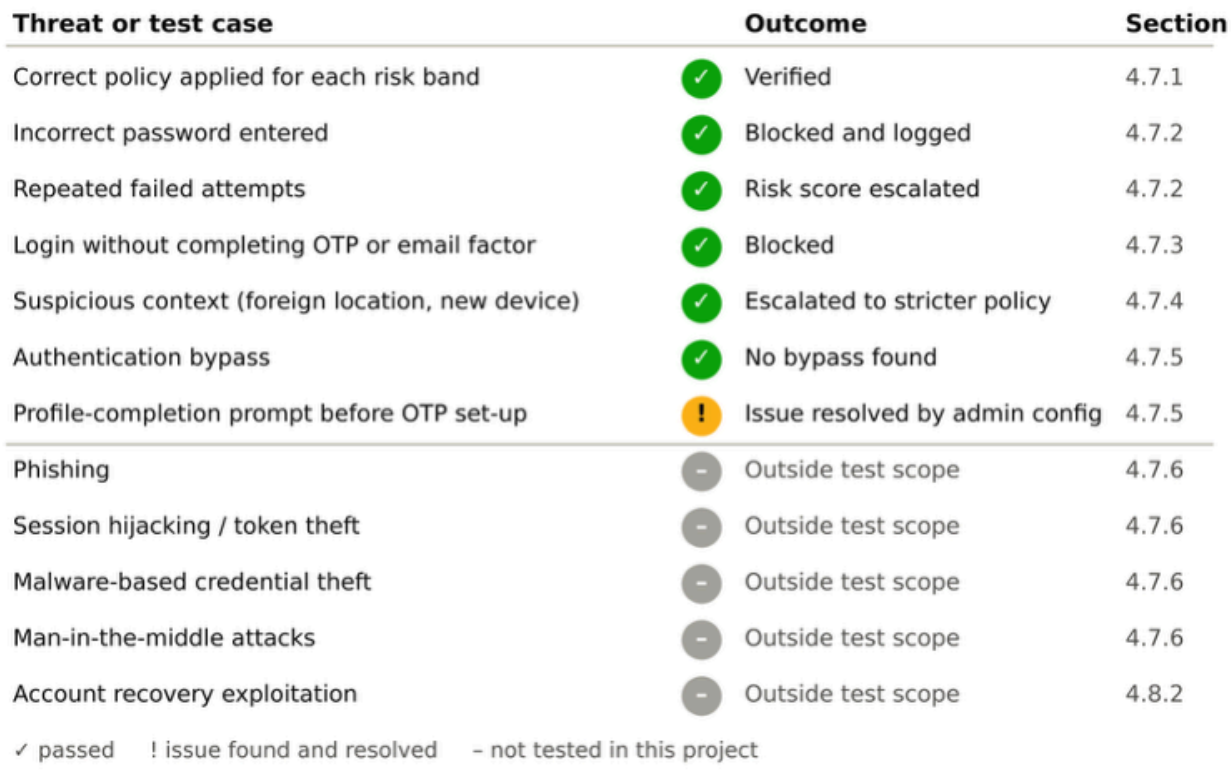
4.7.6 Security Coverage and Residual Risks

The assessment addressed authentication risks related to new device access, foreign login locations, repeated unsuccessful login attempts, and incorrect passwords. The adaptive MFA workflow and the risk-based policy selection procedure were validated using these threats.

However, several threat classes were not included in the assessment. Phishing, session hijacking, malware-based credential theft, man-in-the-middle attacks, token theft, and advanced identity attacks were not tested in this project. Additionally, instead of using real-time device,

geolocation, or behavioural telemetry data collection, the current prototype relies on simulated authentication context provided via predefined JSON inputs.

Even though the high-risk policy added more verification criteria, this does not imply that it is always more secure. The analysis shows that more stringent authentication requirements make it harder for unauthorised access to occur in the tested scenarios, but they also make authentication more difficult and require more work from the user. Therefore, when choosing authentication policies, security efficacy must be considered alongside usability and operational requirements.



Threat or test case		Outcome	Section
Correct policy applied for each risk band	✓	Verified	4.7.1
Incorrect password entered	✓	Blocked and logged	4.7.2
Repeated failed attempts	✓	Risk score escalated	4.7.2
Login without completing OTP or email factor	✓	Blocked	4.7.3
Suspicious context (foreign location, new device)	✓	Escalated to stricter policy	4.7.4
Authentication bypass	✓	No bypass found	4.7.5
Profile-completion prompt before OTP set-up	!	Issue resolved by admin config	4.7.5
Phishing	–	Outside test scope	4.7.6
Session hijacking / token theft	–	Outside test scope	4.7.6
Malware-based credential theft	–	Outside test scope	4.7.6
Man-in-the-middle attacks	–	Outside test scope	4.7.6
Account recovery exploitation	–	Outside test scope	4.8.2

✓ passed ! issue found and resolved – not tested in this project

Figure 4.6: *Security Policy Enforcement Results*

ID	Test case	Method	Expected result	Observed result	Status

ST-01	**Policy enforcement by risk band**	**Logins classified as low, medium and high; required factors checked**	**Each band requires its configured factors**	**Requirements applied consistently for every band**	**Pass**
ST-02	**Incorrect password**	**Deliberately incorrect credentials entered**	**Access blocked; failure logged**	**Blocked until valid credentials supplied; LOGIN_ERROR logged**	**Pass**
ST-03	**Repeated failed attempts**	**Failed attempts increased in scenario input**	**Risk score rises; stricter policy applied**	**Score increased and stronger requirements applied**	**Pass**
ST-04	**Incomplete MFA**	**Authentication stopped before OTP or email step**	**Access denied**	**Keycloak denied access and stopped authentication**	**Pass**
ST-05	**Suspicious context**	**Foreign location, new device and failed attempts combined**	**Escalation to medium or high risk**	**Score rose with each indicator; stricter policy applied**	**Pass**
ST-06	**Authentication bypass**	**Attempts to gain access without required steps**	**No bypass possible**	**No successful bypass found**	**Pass**
ST-07	**Profile-completion prompt**	**Observed during testing**	**OTP set-up without interruption**	**Prompt appeared before OTP configuration; fixed through administrative configuration**	**Resolved**
–	**Phishing, session hijacking, token theft, malware credential theft, man-in-the-middle, account recovery exploitation**	**–**	**–**	**Outside the scope of testing**	**Not tested**

Table 4.6: *Security Test Cases and Outcomes*

4.8 Security Usability Trade-Off Analysis

4.8.1 Comparison of Authentication Friction

In this section, the authentication friction from the three risk-adaptive policies will be compared. The metrics have been measured using the defined methodology.

Metric	Low risk	Medium risk	High risk
Average completion time	18.6 seconds	32.1 seconds	56.3 seconds
Authentication methods required	Password, SSO or passkey (single method)	Password + OTP	Password + OTP + email verification
Outcome	Lowest friction	Middle	Highest friction

It can be seen that the number of steps and the time taken for authentication increase with higher risk levels. From the low-risk policy to the medium policy, there is an average increase in time of 13.5 s, and from the medium policy to the high policy there is an average increase in time of 24.2s. Therefore, the high-risk policy will take approximately 3 times as long as the low-risk policy.

Overall, 91.6% of authentications were successful, 8.4% failed, and 34.8% used more than one authentication method. There were 21.7% abandonments; however, all were either medium or high risk. This is an important point for our project, as the additional steps to help prevent risk were the cause of all the abandonments. However, we do not know why additional authentication attempts were needed.

4.8.2 Comparison of Security Outcomes

In this section, the security results from the experiment will be compared. This can be seen in section 4.7.

The policies were all verified to have been applied. It was confirmed that the low-risk policy used only password authentication, the medium-risk policy used password and OTP verification, and the high-risk policy used password, OTP and email verification.

The credential-based tests showed various security measures. Firstly, incorrect passwords were blocked, and an entry was created in the Keycloak event logs. Only when correct credentials were entered was the user given access. Secondly, if there were multiple failed attempts, the Risk Engine score would increase, therefore creating more stringent authentication measures. Thirdly, if there was an unauthorised attempt to log in and they didn't complete the required factors, they were blocked. If there were an OTP or email verification requirement, Keycloak would block authentication.

The risk escalation was verified. As there were more high-risk factors (login from an outside location, a new device, and multiple failed attempts), the calculated risk score would increase, leading to more stringent authentication. Under a low-risk policy, the user would only be required to enter their password. For medium- and high-risk policies, they would be required to enter their password, OTP, and email verification. Lastly, there were no instances of bypassing authentication. One issue encountered was that the profile completion prompts appeared for some users before the OTP could be set up. However, this was resolved by the administrators because Keycloak required certain fields to be filled in user profiles.

This shows that the policies were enforced within the system and the system responded appropriately to the credential-based threats. However, this doesn't mean that it will prevent all types of attacks. The evaluation did not look at phishing attacks or the exploitation of account recovery. We also don't claim that the high-risk policy will always be more secure because it has more verification steps. It will only be more secure for the specific threat categories that were tested.

4.8.3 Policy-Level Trade-Offs

Here the trade-offs between the different policies will be explained.
The low risk policy had the lowest friction. There was only one authentication method, and the average time taken to complete was the lowest (18.6s); however, there were no abandons. Since there is only one valid credential, security will rely on it. Therefore, there would be more concern around the password being stolen or reused.

The medium risk policy had an additional verification step (OTP) and an average increase in time of 13.5s. However, it had an additional factor unrelated to the password. Therefore, even if an attacker had the stolen password, they wouldn't be able to complete the login without the OTP code. This was the middle ground of the trade-off.
The high-risk policy had the highest friction. There were three verification methods (password, OTP and email verification). The average time to complete was the longest (56.3s). There were also abandonments under the medium-risk policy; however, security increased. The tests showed that if the user didn't complete the required factors, they would be blocked. The risk score would also increase with more suspicious factors. The most severe scenarios would have the most stringent reaction. If the total risk score exceeded the threshold (70+), such as the impossible travel scenario, the account would be locked and flagged for investigation. This meant that the most suspicious logins would be blocked completely.
Overall, there was increased friction due to increased authentication requirements. The security tests showed the specific security measures. However, the system's adaptive nature meant that the increased friction was experienced only when there were actual risky login conditions.

4.8.4 Implications of Synthetic Persona Assumptions

The friction results were based on synthetic users with defined attributes. The users had technical confidence levels, friction tolerance, credentials, and a maximum number of authentication attempts. The abandonments were triggered by two rules. If the user exceeded their maximum friction tolerance or maximum number of authentication attempts. This meant that 21.7% of abandonments were due to the rules set in our model. A real user may have more or less patience. The time taken to complete authentications is useful for comparing policies in a controlled environment; however, we won't know how real users would feel about the friction.

The security results were dependent on the credentials each persona had. Since the synthetic user lacked the required factor, they would be blocked or challenged. A real attacker may have additional factors at play. This means that the security results only show that the policies were enforced in our scenarios.

Given this, the trade-off analysis should be viewed as a comparison of the policies in a controlled environment, based on our model's assumptions. This will be discussed further in sections 4.9 and 5.7.

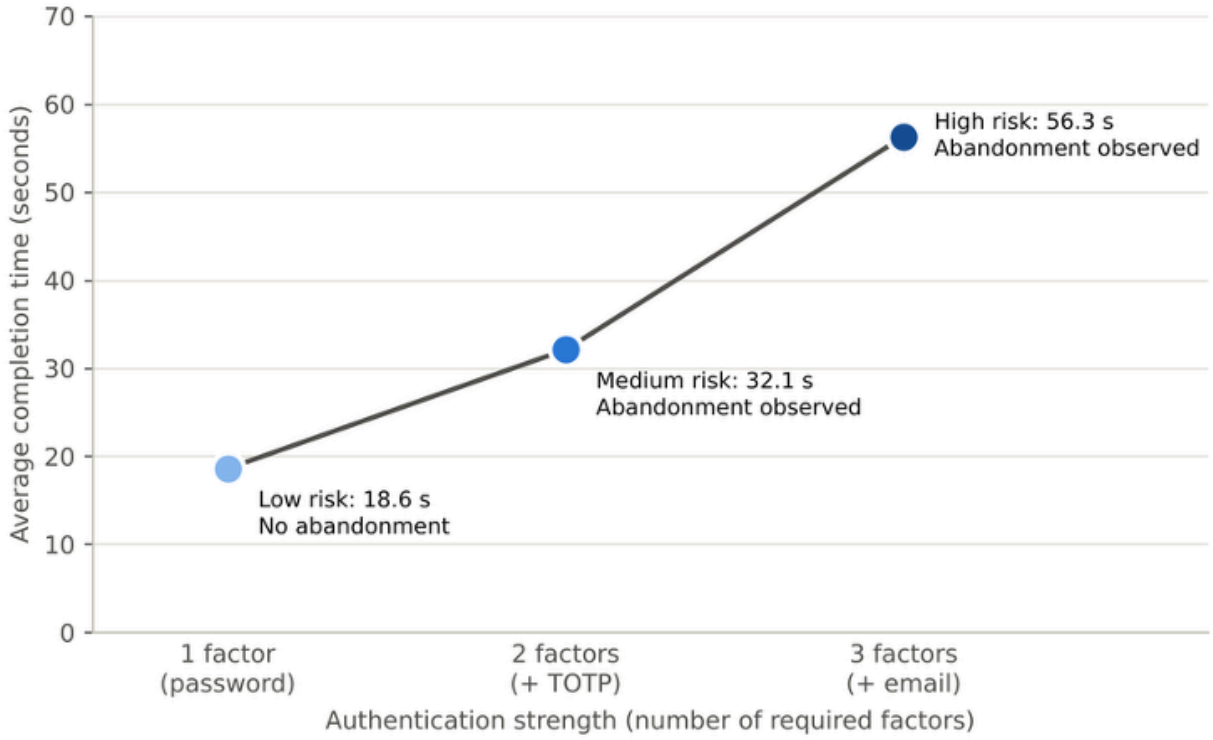


Figure 4.7: *Security-Usability Trade-Off Comparison*

Dimension	Low risk	Medium risk	High risk
Trigger (risk score)	0–1	2–4	5+
Required factors	Password	Password + TOTP	Password + TOTP + email
Average completion time	18.6 s (lowest)	32.1 s	56.3 s (highest)
Simulated abandonment	None	Observed	Observed
If password is stolen	No further protection	TOTP still required	TOTP and email still required
Response to tested threats	Incorrect passwords blocked; repeated failures raise score	As low risk, plus incomplete factors blocked	As medium risk; most severe scenarios locked and flagged (4.8.3)
Main residual risk	Stolen or reused password	Untested threats (e.g. phishing of codes)	Highest friction; untested threats still apply
Overall position	Lowest friction, weakest protection	Middle ground	Highest friction, strongest tested protection

Table 4.7: *Cross-Policy Security and Usability Comparison*

4.9 System Limitations and Observed Issues

The Keycloak instance was simulated in an isolated virtual environment with limited user capacity and scalability. However, this was predicted to be sufficient for the project and was found to meet the suitability requirements accordingly. A pitfall in this regard was the setup and use of Keycloak, which required copious amounts of copying and pasting of configuration files, commands, and environment variables. This need was unfortunately met with a failure of the hypervisor's 'Shared Clipboard' feature, meaning that a considerable number of commands had to be entered manually, substantially prolonging the required work.

In the future, the Keycloak simulation would likely benefit from additional coverage of authenticators (e.g., experimentation with TOTP, SMS, and USB security keys) to understand the impact on risk mitigation and the tangible impact on the risk assessment that occurs during login. A potential prediction in this respect is that the high degree of security offered by USB security keys could mitigate the risk levels inferred in Keycloak's risk assessments when user logins materialise as impossible travel (resulting from VPN use), as well as when using loan or shared devices.

ID	Requirement or issue	Evidence	Status
Requirements			
R1	Risk engine classifies each login into the correct band	Validated against scoring rules (4.3.5)	Met
R2	Keycloak enforces the policy for each band	Policy verification and enforcement tests (4.4.5, 4.7.1)	Met
R3	12 synthetic personas with human-factor attributes	Final persona dataset (4.3.1–4.3.2)	Met
R4	Brute-force, password and session controls configured	Table 4.1	Met
R5	Authentication events logged and results recorded	Keycloak events and spreadsheet (4.2.4, 4.5.4)	Met
R6	Execute planned authentication scenarios	12 of 20 planned scenarios executed (4.5.5)	Partially met
R7	Monitoring with Wazuh, Zeek and Suricata (Objective 4)	Deployment stated in 5.2.4; no monitoring results reported in Chapter 4	Partially met – evidence needed

R8	Evaluation against baseline MFA (Objective 5)	Three adaptive policies compared (4.8); no separate static baseline reported	Partially met
Outstanding issues			
I1	Profile-completion prompts before OTP set-up	4.7.5	Resolved
I2	Hypervisor shared-clipboard failure slowed configuration	4.9	Worked around (manual entry)
I3	Login context simulated via JSON; no real device, location or behavioural telemetry	4.7.6	Open – limitation
I4	Reasons for additional attempts not captured	4.6.3	Open
I5	Phishing, session hijacking, token theft, MITM and recovery attacks not tested	4.7.6, 4.8.2	Open – future work (5.8)
I6	Four personas lack TOTP enrolment required by their assigned policy	Table 4.2 (Chloe Martin, Frank Wilson, Kate Osei, Liam Chen)	Open

Table 4.8: *Requirements Verification and Outstanding Issues*

4.10 Summary of Key Findings

The results showed that the adaptive MFA model successfully modified authentication requirements based on the estimated risk level. Using location, device status, and unsuccessful login attempts as risk indicators, the Flask-based Risk Engine accurately categorised authentication attempts as Low, Medium, or High Risk. After that, Keycloak implemented the appropriate MFA policy for every risk category.

The results demonstrated that while low-risk scenarios retained a lower authentication burden, higher-risk authentication scenarios led to stricter verification requirements. Additionally, the project showed how adaptive authentication behaviour can be assessed in a controlled setting using test scenarios and artificial user profiles. Overall, the findings confirm that risk-based MFA is a useful tool for striking a balance between security requirements and usability.

4.11 Chapter Summary

This chapter implemented and evaluated the risk-adaptive MFA model designed in Chapter 3. Keycloak was deployed in an isolated virtual environment with an adaptive authentication extension (Bartoš, 2026) and a Flask risk engine. The engine classified each login attempt as low, medium, or high risk based on device familiarity, login location, and failed attempts, and Keycloak then applied the corresponding policy: password only; password and OTP; or password, OTP, and email verification. These indicators closely resemble those used in commercial adaptive authentication: Auth0's Adaptive MFA, for example, assesses new devices, impossible travel and untrusted IP addresses (Auth0, n.d.), and Microsoft Entra ID can require MFA when sign-in risk is medium or high (Microsoft, 2026). Twelve synthetic personas were each assigned an authentication scenario, and the resulting trials were used to measure login friction and security outcomes.

The results showed a clear trade-off between friction and protection. Average completion time rose from 18.6 seconds under the low-risk policy to 32.1 seconds under the medium-risk policy and 56.3 seconds under the high-risk policy. All simulated abandonment (21.7%) occurred in medium- and high-risk scenarios. This pattern matches earlier findings that stronger authentication controls increase authentication difficulty (Hastings et al., 2025). On the security side, each policy was enforced as configured: attempts that did not complete the required factors were blocked, risk scores rose as suspicious indicators accumulated, and no authentication bypasses were observed. Because the model is adaptive, additional friction was applied only when login conditions indicated elevated risk, while routine logins maintained the lowest authentication burden. This reflects current guidance that authentication should be designed with usability and security in mind (Temoshok et al., 2025).

These findings must be read within the boundaries of the study. Abandonment was driven by predefined friction tolerance and attempt thresholds rather than by real user behaviour. The login context was supplied through simulated inputs, and only credential-based and contextual threats were tested; phishing, session hijacking and account recovery attacks were not. The results therefore compare the policies under controlled conditions and the model's assumptions; they do not predict how real users would respond. Chapter 5 interprets these findings in relation to the research questions, discusses their implications for human factors and authentication policy design, compares them with the existing literature, and sets out the study's limitations and directions for future work.

Chapter 5 – Discussion, Conclusion and Future Work

Purpose: Interpret the Chapter 4 findings, discuss the implications of the security–usability trade-off, assess the contribution of synthetic human-factor modelling, acknowledge limitations and identify future improvements.

5.1 Introduction

This chapter covers the results of implementing and testing the adaptive MFA prototype built with Keycloak and a Flask-based Risk Engine. The outcomes are examined in light of the project's goals, which include the application of contextual and behavioural indicators, risk-based authentication choices, and artificial user modelling.

The project showed how calculated risk levels can be used to modify authentication requirements. The system was assessed under both typical and suspicious login circumstances using artificial user profiles and authentication scenarios. The chapter also examines the trade-off between security and usability, highlights the prototype's shortcomings, and offers suggestions for future enhancements.

5.2 Discussion of Findings Against Project Objectives

This section evaluates how well and to what extent our project has met the 5 objectives stated at the beginning of the document. We discuss each objective in relation to its implementation, testing and findings.

5.2.1 Objective 1 – Identification of Human Factors Affecting MFA

Our first objective was addressed during the initial stages of our project, within the literature research phase. This is where we learned and identified the different factors that can influence how an MFA affects effectiveness. This research provided the foundation for varying levels of

authorisation based on people's risk levels, both because we understood the usability and risk sides of it through human factors.

5.2.2 Objective 2 – Design of a Risk-Adaptive MFA Decision Model

Objective 2 was achieved by developing a risk-adaptive MFA Decision model using the Flask framework and Python. Our risk-adaptive MFA Decision model accounts for risk indicators, such as failed login attempts, unfamiliar devices and geographic location changes and assigns an overall risk score to each authentication attempt. As outlined in section 3.10, the overall risk score was then categorised as Low, Medium, or High, and, based on this categorisation, the decision model determined the authentication requirements for that scenario.

5.2.3 Objective 3 – Implementation of Adaptive MFA in Keycloak

The Third objective we focused on throughout this project was implementing the proposed Keycloak simulation. To test this, we created 12 different synthetic user profiles, with varying levels of risk. This provided us with a controlled way to test the deployment and implementation of our adaptive levels of authorisation across different scenarios and to assess how, or if, it works. However, we cannot be certain of the friction it will cause with humans, since these were AI-generated users.

5.2.4 Objective 4 – Deployment and Testing Within a Controlled Cyber Range

Our fourth objective was to deploy and test the adaptive MFA we had created in a controlled cyber range. To do this effectively and safely, we operated within virtual machines using VMware, and within them we used Keycloak. The use of Wazuh, Zeek, and Suricata helped us gather information about the events and provided a basis for reviewing the prototype. We deployed the 12 synthetic users that we created with AI, on which we based our results for this project.

5.2.5 Objective 5 – Evaluation Against Baseline MFA and Development of Recommendations

Our fifth objective was achieved through the evaluation of experiment results. Comparing results for authentication requirements, completion times, simulated abandonment, and successful and unsuccessful attempts across multiple synthetic user profiles helped us understand outcomes under different risk-based MFA policies. The findings from our experiment demonstrate that authentication requirements produce authentication friction but also provide stronger verification, particularly under suspicious conditions. Our experiment was limited, however, by the use of synthetic users and predefined attributes, and future research would likely benefit from human participant testing and a more complex risk classification system.

Research question	Finding	Supporting evidence	Section
RQ1. How can a risk-adaptive MFA model be implemented in Keycloak?	A Flask risk engine scored each login (F + 2D + 3L) and returned a risk band; Keycloak conditional flows enforced password, password + TOTP, or password + TOTP + email for low-, medium-, and high-risk.	Risk engine validated against scoring rules; policy verification confirmed each band's factors	4.2–4.4 4.3.5
RQ2. How do the authentication policies affect usability?	Friction rose with policy strength: average completion time of 18.6 s, 32.1 s and 56.3 s. Simulated abandonment (21.7%) occurred only in medium- and high-risk scenarios.	Completion times, steps, failure and retry rates, simulated abandonment	4.6 4.8.1
RQ3. How effective are the policies against the tested threats?	Each policy was enforced as configured: incorrect passwords and incomplete MFA were blocked, risk escalated with suspicious indicators, and no bypass was found. Effectiveness is limited to the threats tested.	Security test cases ST-01 to ST-07; residual risks listed	4.7 4.8.2
RQ4. How do synthetic human-factor assumptions influence the results?	Abandonment and retry outcomes were produced by preset friction-tolerance and attempt thresholds, so they reflect the model's assumptions rather than real user behaviour.	Persona attributes and abandonment rules; discussion of synthetic behaviour	4.8.4 5.4.4 5.7.2

Table 5.1: *Research Questions, Findings and Supporting Evidence*

5.3 Discussion of the Security–Usability Trade-Off

This section interprets the results from Chapter 4 on the tensions between security and usability. covers strengthening authentication and meeting user requirements based on risk bands for risk-adaptive Keycloak configuration. These configurations apply stronger authentication

requirements as risk increases. As each authentication requirement increases (password, TOTP, email verification), the friction heightens the tension in Security-usability trade-offs; increased MFA friction may push users to abandon or bypass MFA altogether (Arnold et al., 2022). This section further compares the friction caused by heightened authentication requirements across the risk bands (time, steps, attempts, and abandonment). Each security-heightened authentication level is demonstrated to compare security protections against friction, whilst recognising that these results are simulated rather than reflecting real user interactions and behaviours.

5.3.1 Benefits of Risk-Adaptive Authentication

Risk-Adaptive Authentication can provide benefits to university and/or business settings through giving users a frictionless experience that doesn't constrain their daily routines. Having a three-tier risk classification can impose stronger controls by providing a smooth experience for low-risk users while reserving aggressive, resource-intensive actions for high-risk users. By identifying high-risk users, we can isolate them from the network to conduct investigations such as email verifications, mandatory password resets or manual reviews. Additionally, Zero Trust controls are up to standard, as each request is never automatically verified; every request is always reviewed based on the user's geolocation, network, devices, and credentials (Robertson & Kosinski, 2026). If we were to impose the same authentication requirement on every user, it would frustrate legitimate users and hinder progress in authentication, reducing productivity.

5.3.2 Authentication Friction and User Experience

Medium-risk users require additional authentication steps compared to low-risk users, which take longer time to complete their scenario but would not trigger an automated fail because other factors could suggest that they are using a private browser but were in a familiar location or an appropriate geolocation shift, as demonstrated in M-06. Or if there were failures to log in

and suffered an account lockout after 3 failed attempts, then the user has another opportunity to do so by waiting for a recovery (shown in Code S5), then later re-entering new password credentials and email OTP, shown in M-03, but will have to wait even longer, further delaying completion time to become produced.

5.3.3 Security Improvements and Their Costs

The implemented policies have demonstrated stronger enforcement against tested threats, with remediation steps based on appropriate risk scoring. For example, (shown in R-6) an attacker has been shown to intercept an account through social engineering, which sets a risk score of 70, putting them as a risk user. Another threat we tested was phishing (shown in R-02), where phishing-resistant WebAuthn was implemented because an attacker held the user's phished password, and it was detected on an unrecognised device and network.

5.3.4 Implications for Authentication Policy Design

An organisation will need to balance authentication requirements and operational needs by relying heavily on Single Sign-On, where employees can log in once to access authorised applications, effectively meeting their operational needs without repeated friction or additional information (shown in L-02). This reflects the core user-centric security design, where it's for employees to take the necessary steps while stabilising their workflow and improving productivity, along with risk-adaptive, where if the user is low risk, they can operate smoothly. Furthermore, using contextual access control can factor in the employee's location, time of day, and network before attempting to log in, ensuring security is highly standardised.

5.4 Discussion of Human Factors

Human factors, including behaviour, habits and psychology, play a critical role in achieving effective Multi-Factor Authentication. However, MFA must adapt to certain human traits to allow authorised users to authenticate easily while making it difficult for attackers to intercept. Context-adaptive controls, such as risk-adaptive controls, can adapt to and respond to certain human traits by leveraging context. For example, repeated unintentional mistakes due to frustration can put users at high risk. Still, risk-adaptive controls could absorb and adapt to those users by collecting additional information, such as known IP addresses, trusted devices, and familiar geolocations, to distinguish between a trusted individual and a brute-force attacker. Furthermore, humans who consistently exhibit normal traits tend to have smooth experiences due to their low-risk factors.

5.4.1 Technical Confidence and MFA Interaction

The technical confidence assumptions were represented in the synthetic model by processing the mental model in which its users are able to navigate MFA interactions, understand how to identify verifications, and view MFA as necessary. Additionally, there were assumptions that users are highly confident in navigating MFA and successfully pass through without any friction or drawbacks, but also assumed that users may experience some friction due to minor mistakes, an unfamiliar IP address or using a new device, requiring additional verification but displaying a low rate of abandonment and eventually passing the MFA stage. Other assumptions could be significantly low, where users present huge difficulties due to frustration and pressure or attempting to brute-force, requiring additional verifications but leading to abandonment and an account lockout by the system.

5.4.2 Authentication Fatigue and Friction Tolerance

Throughout our simulation, 21.7% of simulated abandonments occurred when the AI agents took longer to complete a specific task. There was also a maximum limit on runs per person before usage limits were reached. We could only run a limited number of attempts for each individual/scenario to ensure we don’t overuse and get a fair value. Even if some attempts were incomplete or invalid, we could only log valid runs. There was also a repeated authentication requirement, in which most synthetic users used passwords, followed by TOTP as an additional factor. This indicates that TOTP was fast, easy to use, and secure due to its time-based nature, limits, and ease of reading and typing.

5.4.3 Accessibility and Diverse User Needs

The accessibility-related assumptions were that those who constantly have physical devices on them and repeatedly use authenticator apps or SMS codes for authentication are typically under no pressure and can effortlessly pass the authentication steps without problems or errors. Additionally, they have no problem navigating between their smartphone and computer and accurately typing their code quickly before the session expires. Another assumption is that users don't always have their mobile devices with them 24/7 and have to retrieve them to enter a code before the session expires. Both devices are on similar IP addresses and are both well-known. Furthermore, the assumption that users without a secondary device hasn’t been tested across different risk categories, although it was immediately assumed they would be locked out if they were high-risk users.

5.4.4 Limitations of Synthetic Human Behaviour

Synthetic users and predetermined thresholds cannot exhibit real human behavioural changes or emotional depth because they operate on statistical data rather than biological experience; they tend to rely on algorithmic rules rather than developing feelings in response to a given experience. Furthermore, AI agents tend to please their users, leading to prompt bias toward false-positive answers, a lack of in-depth discussion, and a failure to make any improvement. Synthetic users lack the emotions, depth and lived experience required to simulate real-life situations and instead use language patterns and text aggressiveness to determine the answers to prompts, which fails to fully capture the real world.

5.5 Comparison with Existing Literature and Authentication Guidance

Here we will discuss similarities between the project and previous work highlighted in Chapter 2. This includes a comparison of MFA usability, risk-based authentication, and security principles within authentication. This can be used to identify how this project has built upon previous research and to identify where there is limited evidence.

5.5.1 Comparison with MFA Usability Research

The importance of login duration, repetition and authentication errors is highlighted within the literature reviewed in Chapter 2. Several studies have investigated the impact of login duration on user satisfaction whilst using MFA (Kruzikova et al., 2024). Other studies have investigated how users perceive two-factor authentication (Marky et al., 2022). It was found that users felt that two-factor authentication would interrupt them from doing what they wanted to do. This gives us a reason to investigate the effort required for authentication with MFA.

5.5.2 Comparison with Risk-Adaptive Authentication Research

Chapter two covers risk-adaptive authentication. A key example studied in this area was a large-scale study on risk-based authentication (Wiefling et al., 2023). This study is useful to compare as it provides insight into how additional authentication can be applied.

The project uses the same underlying concept but on a smaller scale. The risk engine assigns specific values to different conditions such as new devices, different countries and failed login attempts. Depending on the chosen risk level, different authentication policies will be enforced. As this project is not operating on a large scale, there are only a few defined scenarios. Therefore, we can provide evidence of how the rules and flows have been set up, but not how they would perform in a real-world scenario.

5.5.3 Alignment with Authentication Security Guidance

In our project, we have considered how secure each authentication process is, rather than simply counting the number of steps. We have designed our policy to use passwords only for low-risk logins, passwords + TOTP for medium-risk logins, and finally passwords + TOTP + email verification for high-risk logins. This does not guarantee that the login will be secure against all attacks; however, we are drawing conclusions only from the tests we have completed.

We have also considered independence of factors. This is where we determine whether by compromising one factor, we would compromise another. For example, if an attacker had access to the user's email, they could access both the password reset and email verification. This would not be independent.

Another thing to consider is phishing. The attacker could create a fake login page and trick the user into entering their password and current TOTP code. They can then instantly use these details on the actual website. We therefore do not consider TOTP resilient to phishing attacks. NIST guidelines explain that one-time codes can be relayed by an attacker when they are manually entered.

We have also considered risks around recovering the account. Whilst there may be multiple steps to complete during the login process, there could be a weaker password reset or authenticator recovery process. Allowing the authenticator to be removed via a weak recovery process could reduce the security of the login process. We are only considering the recovery processes we have tested. We are not claiming our system is fully secure against all authentication attempts, as we have multiple verification steps.

5.5.4 Implications for Identity and Access Management

This project demonstrates how security and usability can be considered within Identity and Access Management. Previous work has examined how a system can cause friction by requiring users to wait longer to log in, repeat authentication steps, and feel disrupted. Other research has been done around risk-based authentication. This project combines these ideas into one Keycloak environment.

The system design requires users to enter their password when logging in at a low risk level. Users will have to provide their password plus a TOTP code if they are attempting to log in at a medium risk. If they are logging in at a high risk level, they will also have to verify their email

address. This reduces the number of steps users must complete during login. The additional verification steps are applied when there is a higher risk associated with the login attempt. This could help reduce the number of steps required for standard logins. Whilst also ensuring that additional steps are in place for any suspicious logins.

However, it should be noted that this project was tested in a controlled environment using synthetic users. This means that the results only demonstrate how the configured policies performed within the scenarios. It does not prove that the model will improve the satisfaction of real users or that it is ready to be implemented within all organisations.

The key takeaway for organisations will be that whilst increasing authentication steps, you also need to consider how much effort is being added, what is actually being protected by the steps and if the recovery processes might undermine the standard login process. Before implementing a risk-adaptive model, you would need to test with real users, ensure the risk levels are appropriate and ensure the security meets relevant guidance.

Study	Reported finding	This project's finding	Relationship
Kruzikova et al. (2024)	Actual and perceived login time shape users' satisfaction with 2FA and their view of its security (337 users).	Completion time rose from 18.6 s to 56.3 s as factors were added.	Consistent: time is a key measure of friction. This project did not measure satisfaction.
Marky et al. (2022)	Users found banking 2FA tedious and disruptive to their tasks (42 interviews).	Extra steps applied only to medium- and high-risk logins, where all abandonment occurred.	Consistent: the adaptive design confines interruption to riskier logins.
Hastings et al. (2025)	Moving to number matching raised mobile MFA failures from 10.2% to 17.9%.	Stronger policies produced longer completion times and simulated abandonment.	Consistent: stronger controls increase the difficulty of authentication.
Sakshi et al. (2026)	75% of surveyed users had quit an app over a complicated login; 54% were frustrated by separate devices.	21.7% simulated abandonment, all under multi-step policies.	Directionally consistent, but simulated rather than self-reported.
Jubur et al. (2025)	About 49.6% of simulated attack push prompts were approved, showing fatigue risk.	Push MFA was not used; fatigue was modelled only through tolerance thresholds.	Different method: MFA fatigue attacks were not tested.

Wiefling et al. (2023)	Risk-based authentication rarely re-challenged legitimate users while blocking most targeted attacks (31.3 million logins).	Same principle on 12 scenarios: low-risk logins kept to a password; suspicious logins escalated.	Consistent in concept; small scale, so real-world performance is not shown.
Son et al. (2025)	AI risk-adaptive MFA in Keycloak scored logins with 97% accuracy but did not measure human experience.	Rule-based engine; friction and abandonment measured for simulated users.	Extends: adds usability outcomes, with simpler risk classification.
Amft et al. (2023)	Account recovery often failed or offered a way around MFA.	Recovery was not tested; flagged as a residual risk.	Gap: not addressed by this project.
NIST SP 800-63B-4 (Temoshok et al., 2025)	Manually entered one-time codes can be relayed by an attacker and are not phishing-resistant.	TOTP was not treated as phishing-resistant.	Aligned with guidance.

Table 5.2: *Comparison of Findings with Existing Literature*

5.6 Research and Technical Contributions

Throughout this project, we have conducted a practical investigation into a risk-adaptive Multi-Factor Authentication system using Keycloak, focusing on how authentication requirements can be adjusted based on users' expected risk, informed by specific risk factors. By using Keycloak in a controlled, virtual environment, we can safely test and implement it. The main contributions to be able to do this are the controlled testbed, the risk-adaptive MFA model, the synthetic scenarios and the documented evaluation method.

5.6.1 Controlled Keycloak Authentication Testbed

This project has been carried out within a controlled virtual environment using Keycloak, which allows us to examine authentication configurations and different login scenarios without affecting any real users or systems. By using different virtual machines throughout the project, we created a structured environment for configuring, testing, and monitoring the authentication behaviour of this risk-adaptive MFA.

The central identity and access management testbed throughout this experiment is Keycloak, which handles user authentication and the associated MFA requirements. Keycloak allows us to apply our configured risk level to each scenario, thereby implementing different layers of authentication for users.

5.6.2 Risk-Adaptive Authentication Policy Model

One of the main components of this project is the risk-adaptive behaviour of our authentication system. To implement this, we configured the application to apply different authentication measures based on each user's estimated risk level.

Our risk score model operates on three levels: low risk is a calculated score of either 0 or 1. Medium risk, which is a calculated score of 2-4, and finally high risk has a score of 5+.

The formula that is followed, in order to calculate this level of risk for each individual user, is:

Risk Score = F + (2 x D) + (3 x L)

Where:
F = Number of failed Attempts
D = New Device
L = Different Country

Using this formula is a brilliant way for us to determine risk level, because as talked about in chapter 3 of this project, these three factors of F, D and L are three of the largest risk factors for authentication systems. We give a failed attempt a score of 1 here because it is the least risky action someone could take. A new device is given a score of 2 because it could potentially be a compromised account. Finally, a different country is given the highest score of 3 because it is extremely suspicious, especially when paired with a new device, indicating that an account is highly likely to have been compromised.

5.6.3 Synthetic Persona and Scenario Framework

A major contribution to the completion of this project was the creation of 12 synthetic user personas to support the 12 scenarios. Since this was in a controlled environment, we used AI to generate 12 personas at random, with varying levels of cyber knowledge and experience. Each

of these different personas had a different login scenario, displaying different numbers of failed login events or whether they were logging in from a new device or another country, in some cases both. These displayed the different levels of risk threats that a company can face and why having a good MFA is such an important aspect for a company.

The framework used allowed us to test multiple different scenarios and risk levels, rather than repeating the same example multiple times. This is crucial because it allows us to test our system across the different possible scenarios.

Overall, the persona model was created using an AI agent for 12 personas with varying personalities and levels of cyber experience; we used these in multiple scenarios to test our MFA and ensure it works across them.

5.6.4 Reproducible Security and Usability Evaluation

Another contribution to this project is the development of a structured evaluation plan to assess this risk-adaptive MFA under controlled conditions. Our evaluation uses the predefined scenarios, created through AI, which allows us to examine multiple different examples of threats and risks that will be faced. By documenting these results, we can analyse them over time to ensure the system is working correctly.

From a security perspective, this analysis and evaluation allow us to identify whether dynamic authorisation is working correctly, with the estimated risk score calculation and the application of the correct level of authorisation. We can also analyse whether implementing these layers of authorisation is necessary. Through the single layer- just the password for low-risk people- through to 3-layer authorisation, including password, TOTP and email verification for high-risk people.

From a usability perspective, the evaluation also considers the inclusion of extra authentication methods and how this may affect regular users. The introduction of these extra layers may add potential unnecessary friction for the regular user, which is where the trade off between security and usability is crucial. Comparing these policies with our results and other MFA policies will help us determine the usability-versus-security trade-off for our dynamic system.

The documented procedure provides a basis for us to work from; however, with only these 12 scenarios, it is hard to determine whether these policies strike the right balance between

usability and security. The aim of the dynamic system is to provide a high level of security for high-risk users while keeping it simple and reducing unnecessary friction for the everyday user.

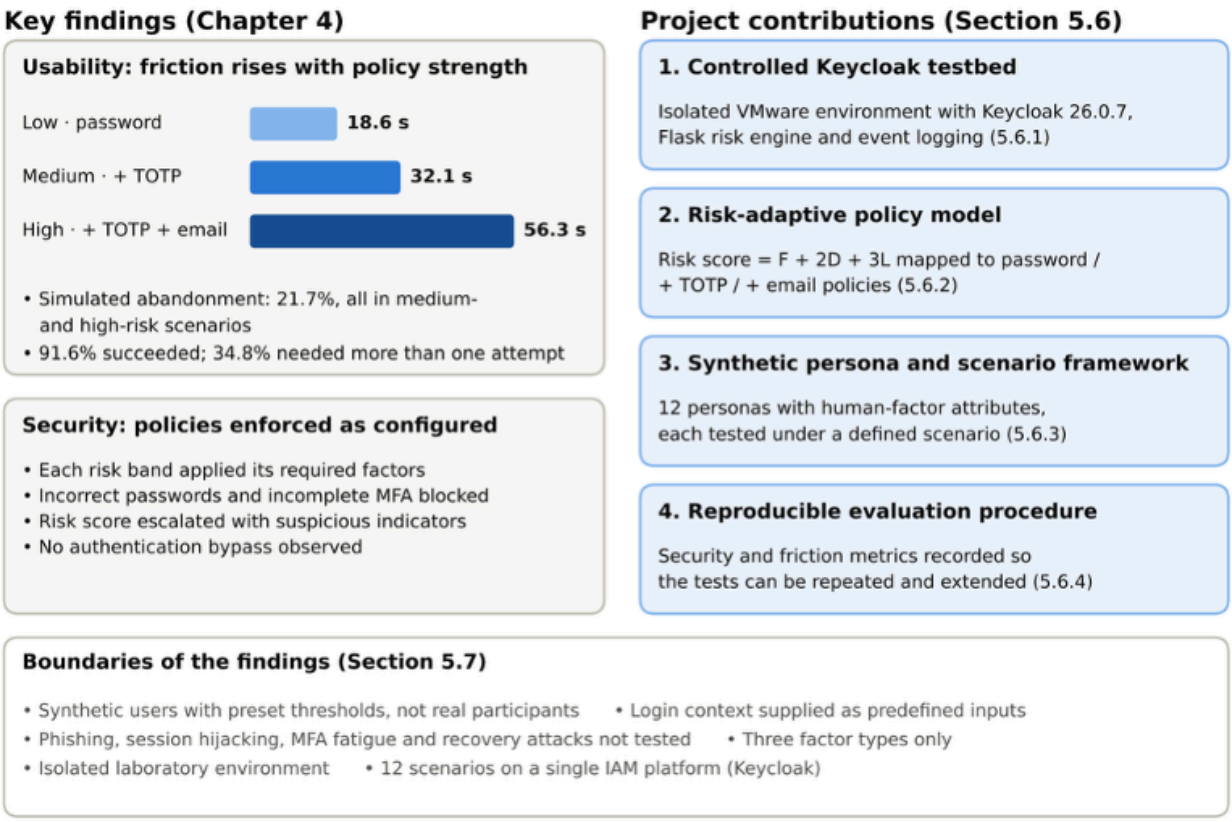


Figure 5.1: *Summary of Security-Usability Findings and Project Contributions*

Objective (Section 1.3)	Evidence	Status
1. Identify cognitive, behavioural and usability factors affecting MFA and use them to define simulated behaviours and risk indicators	Literature review (Chapter 2); factors mapped to persona attributes (technical confidence, friction tolerance, maximum attempts) and to the risk indicators F, D and L (3.5, 3.9, 5.2.1)	Achieved
2. Design a risk-adaptive MFA decision model	Risk score = F + 2D + 3L with low, medium and high bands mapped to authentication policies (3.7, 4.3.3, 5.2.2)	Achieved

3. Implement the model in Keycloak and develop synthetic user scenarios	Keycloak 26.0.7 conditional flows with risk-gateway client; 12 personas each assigned a scenario (4.2–4.4, 5.2.3)	Achieved (12 of 20 planned scenarios)
4. Deploy and test in a controlled Cyber Range and collect telemetry with monitoring tools	VMware environment deployed; Keycloak events logged (4.2.4). Wazuh, Zeek and Suricata are stated as used (5.2.4), but no monitoring results are reported	Partially achieved
5. Evaluate against a baseline MFA configuration using defined metrics, and develop recommendations	Three adaptive policies compared on time, steps, failures and abandonment (4.6, 4.8); recommendations in 5.8. No separate static baseline results are reported	Partially achieved

Table 5.3: *Project Objectives and Achievement Status*

5.7 Research Limitations

During the research for this study, challenges were encountered. While many other studies used synthetic users, others included human participants who could provide usability insights and identify constraints through different MFA methods. There was limited research that also used synthetic users, while Son et al. (2025) provided AI-risk-based authentication with 500,000 synthetic users; no other research provided in-depth value regarding synthetic users in risk-based MFA. While they also use authentic users, which lack real-time human behavioural aspects, other researchers who have used human participants had small sample sizes, which are very small compared to a large population such as a university and used surveys that do gain users' insight and satisfaction; users tend to forget or randomly select answers due to multiple long questions instead of a walkthrough to capture user frustrations in real time fully.

5.7.1 Absence of Human Participant Testing

This study cannot involve human participants due to the timeframe and ethical requirements, so it instead creates synthetic users to simulate different scenarios among low-, medium-, and high-risk users. The study was unable to establish a real behavioural relationship between users and Keycloak, making it difficult to discuss and improve user satisfaction. Without a practical walkthrough with real users, the study could miss usability insights, limiting its ability to identify

constraints and pinpoint emotions in real time. If those perceptions are unseen, authentication fatigue could be mistaken for inaccuracy or incompleteness, as synthetic users tend to give biased answers to please them, leading to a false positive.

5.7.2 Synthetic Persona Assumptions

This study integrates 12 synthetically created personas with assigned tolerance thresholds and technical confidence. The research is limited to synthetic personas and our manual input and biases. The abandonment and retry results depend on the simulated personality traits chosen for each user. Each threshold can ultimately influence which risk levels real users consider frustrating. The findings indicate that policies within Keycloak behave under predetermined assumptions, not on real user response.

5.7.3 Limited Authentication Methods

Password, TOTP, and email verification were implemented in our experiments. By not integrating further SMS, USB security keys, or phishing-resistant methods, we ultimately limit the ability for other MFA methods to be considered more efficient and effective.

5.7.4 Simplified Risk Classification

Within this study, only 4 rule-based indicators are used (device, IP, location/time, and network), each assigned a risk band based on the simulated user context (including overseas, VPN,n or unknown addresses). Because synthetic users are assigned predetermined risk-band tiers from their scenario, this does not test realistic policy classification accuracy; rather, it tests whether the policy is enforced.

5.7.5 Controlled Laboratory Environment

By testing our experiments only in a simulated, isolated virtual environment, we considerably limit our ability to obtain accurate results for user capacity and real network conditions. These limitations affect the ability to accurately measure completion times and failure rates under real-world conditions, including latency and email delays.

5.7.6 Security Test Coverage

The simulated threat scenarios include incorrect passwords, credential reuse, unauthorised attempts, and brute-force attempts. The design of our research limits the ability to represent only the enforcement of policies by intended factors we allowed. Not including phishing, proxy attacks, or MFA fatigue attacks limits the research's consideration of every possible attack.

Limitation	Description	Implication for the findings	Addressed in
Absence of human participant testing (5.7.1)	Synthetic users replaced real participants because of time and ethics constraints.	Actual satisfaction, perceptions and authentication fatigue cannot be established.	5.8.1
Synthetic persona assumptions (5.7.2)	Tolerance thresholds and technical confidence were set manually.	Abandonment and retry rates depend on the chosen thresholds and may not reflect real users.	5.8.4
Limited authentication methods (5.7.3)	Only password, TOTP and email verification were implemented.	Phishing-resistant and other methods could not be compared.	5.8.2
Simplified risk classification (5.7.4)	Rule-based indicators with predetermined risk bands per scenario.	The study tests policy enforcement, not real-world classification accuracy.	5.8.3
Controlled laboratory environment (5.7.5)	Isolated virtual environment without real network conditions or user load.	Completion times and failure rates may differ under latency, email delays and production load.	5.8.5
Security test coverage (5.7.6)	Phishing, proxy and MFA fatigue attacks were not tested.	Security conclusions apply only to the tested threats.	5.8.2 5.8.5
Single IAM platform (Section 1.4)	The evaluation was conducted in Keycloak only.	Findings may not transfer to other identity platforms.	5.8.6

Table 5.4: *Research Limitations and Implications*

5.8 Recommendations for Future Work

Recommendations within this study could be extended to the duration of the study to allow human participants to conduct a walkthrough within different locations, devices and IP addresses, observing them using and attempting the authentication process; risk calculations

would also be presented after the walkthrough without letting participants know what scores they got and which categories they were put into. This would help identify emotions and behaviours individuals experience in real time; we could also provide a timeframe for users who are successful to determine whether the timing was appropriate to minimise workflow disruptions. We can also use synthesis users to compare results and identify which users engage in the riskiest actions. Furthermore, we can gain more value from human participation by conducting a short survey, rating how Risk-based authentication Keycloak works for them.

5.8.1 Conduct Human Usability Testing

If the appropriate human ethics approval were obtained, the duration of this study would be longer; it could take us up to 6 months to recruit human participants, conduct a walkthrough, and establish and maintain Keycloak within a Virtual machine, which could introduce technical and scheduling difficulties along the way. This way, it will boost user confidence in getting comfortable with risk-adaptive if implemented, and shape perceptions through identifying and discussing participant insights. However, using human testing won't allow us to determine which participants are considered low, medium, and high risk before the walkthrough, which could lead to inconsistent baseline data and make it difficult to evaluate how effectively the three-risk models perform under various conditions.

Recommendation	Key actions	Addresses	Priority	Rationale
Conduct human usability testing (5.8.1)	Obtain ethics approval; run walkthroughs with real participants; measure completion time, errors, accessibility and perceptions	5.7.1, 5.7.2	High	Addresses the main limitation; needed before any claim about real users. Longest

				lead time (up to 6 months).
Evaluate phishing-resistant authentication (5.8.2)	Add passkeys, FIDO2 and WebAuthn to the high-risk policy; test phishing and relay scenarios	5.7.3, 5.7.6	High	TOTP and email codes are not phishing-resistant, so the highest-risk logins remain exposed.
Improve risk classification (5.8.3)	Weighted or continuous scoring; IP and device reputation; feed Wazuh, Zeek and Suricata alerts into the risk engine; validate false-positive rates	5.7.4	Medium	Moves the study from testing enforcement to testing classification accuracy.
Evaluate realistic operational conditions (5.8.5)	Emulate latency and packet loss; test recovery procedures end to end; run longitudinal simulations	5.7.5, 5.7.6	Medium	Checks whether timing and security results hold outside the laboratory.
Expand the synthetic persona model (5.8.4)	Add evidence-based accessibility, device and recovery scenarios; calibrate thresholds; run sensitivity analysis	5.7.2	Medium	Low cost and improves the realism of future simulations.
Compare alternative IAM platforms (5.8.6)	Repeat the scenarios on Authentik or ZITADEL, then commercial platforms such as Okta, Auth0 or Entra ID	Single-platform scope	Lower	Valuable for generalisability once the core model is validated.

Table 5.5: *Future Work Recommendations and Priorities*

5.8.2 Evaluate Phishing-Resistant Authentication

Passkeys are a recent authentication method that uses public-key cryptography to make logging into the system faster and easier. It provides multiple benefits, such as not being intercepted or

stolen because it doesn't use passwords, preventing brute-force attacks, credential stuffing, and phishing emails, and eliminating the need to remember passwords. Only using PIN verification and biometric methods can allow access locally on your device. However, Passkeys can be easily lost if the hardware device that stores them is lost or infected with malware, allowing attackers to gain unauthorised access to passkeys and, therefore, accounts. If a fingerprint is used, there is no way for users to change it (White, 2025). FIDO2 is a concept similar to passkeys that eliminates the need for passwords, but it is hardware-bound. FIDO2 has strict security policies, fits in a Zero Trust enterprise and provides protection against remote malware and phishing. However, losing a physical security key can result in a permanent lockout, and repeatedly inputting an incorrect PIN can trigger a secure defence that permanently wipes keys and stored credentials without manual review (Sloan, 2023). WebAuthn binds credentials to a specific domain name, resisting phishing. Instead of using passwords, it uses biometric authentication, providing a frictionless user experience. However, account recovery can be challenged if the physical security key is lost or stolen. It is complex to implement because it requires developers to handle complex cryptography, correctly register public keys, and requires further maintenance and patches.

5.8.3 Improve Risk Classification

The decision model used in this study considered login attempts within three fixed risk levels using four indicators: device familiarity, IP address, login location and time, and network trust. Each of these indicators was assigned to a predefined category, with this rule-based approach found to be suitable in real-world deployments, including, but not limited to, IP and device reputation, threat intelligence feeds, and/or behavioural patterns that develop over time. Future work could replace these fixed categories with adaptive, weighted and continuous risk scores that combine several signals. Examples of this include data on IP reputation, device fingerprinting and behavioural biometrics, such as keystroke dynamics and mouse movement (Shuford, 2024). Data of this nature can also be collected by Endpoint Detection and Response (EDR), Security Information and Event Management (SIEM), Network Intrusion Detection/Prevention System (NIDS/NIPS) and Network Detection and Response (NDR) software. Such tools include Wazuh, Zeek, and Suricata, which can feed into a risk engine of this nature, allowing network-level detection to directly impact the authentication policy applied to a login.

This can be further developed through machine learning, which offers a modern and adaptive pathway to better understanding human behaviour and mapping it to given risks.

5.8.4 Expand the Synthetic Persona Model

The synthetic persona model used in this study utilised twelve profiles. Each profile was defined by its role, enrolled credentials, technical confidence, friction tolerance and the maximum number of attempts before being locked out. These attributes were assigned manually as assumptions, with accessibility needs represented by an additional single persona with low vision who requires a screen reader. Further emphasis could be placed in the future as to accessibility, context of the device used and the scenario in which authentication is requested. Each additional consideration should be justified using published evidence rather than researcher judgement.

In terms of accessibility, additional personas could represent users with motor impairments, who may struggle to enter a time-based one-time password (TOTP) before it expires, and users with cognitive or learning disabilities, who may find multi-step flows or transcribing codes difficult. This would extend the earlier discussion of how motor, visual and cognitive needs affect users' ability to complete MFA independently (Section 2.3). These personas could also be used to test the Keycloak login flow against the Web Content Accessibility Guidelines (WCAG) 2.2. Success Criterion 3.3.8, Accessible Authentication (Minimum), requires that authentication does not depend on a cognitive function test, such as remembering or transcribing a code, unless an alternative or assisting mechanism is provided (W3C, 2023).

For device context, scenarios could include shared or hot-desk workstations, users without a personal smartphone, and lost, broken, or flat-battery devices (which could indicate automated or simulated devices). These address the frustration with separate-device requirements reported by Sakshi et al. (2026). Authentication scenarios could also consider account recovery after device loss, which Amft et al. (2023) found to be frequently unsuccessful, as well as delayed email one-time passcodes and repeated logins over the course of a working day to model cumulative friction.

Finally, persona attributes should be founded on empirical data. Friction tolerance and maximum attempts could be calibrated against measured completion and response times from studies such as Kruzikova et al. (2024) and Jubur et al. (2025), and the mix of personas could reflect

population data on disability and technology use. A sensitivity analysis that varies these thresholds would show how sensitive the study's findings are to the assumed values. This would address the limitation identified in Section 5.7.2.

5.8.5 Evaluate Realistic Operational Conditions

This study evaluated Keycloak in an isolated virtual environment with stable network conditions, fixed login contexts and no account recovery process. As per Section 5.7.5, results may therefore differ in production deployments. Future work should test the risk-adaptive model under conditions that more closely reflect day-to-day operation.

Network variability is the first of these. Network emulation tools, including Linux traffic control ('tc netem'), could introduce latency, packet loss, and intermittent connectivity into the testbed to measure their effects on authentication time, timeouts, and failed attempts, which are conditions particularly relevant to time-sensitive factors. Delayed email one-time passcodes may expire before use, and clock drift between devices can cause valid TOTP codes to be rejected. Keycloak's OTP "look-around window", which accepts codes from adjacent time steps, could be tuned and evaluated to balance tolerance to drift against code security. Network context can also affect risk classification itself. Users on mobile networks often share public IP addresses through carrier-grade NAT, and users connecting via VPNs may appear to have changed their location. Both of these factors could cause the IP and location indicators to misclassify legitimate logins.

Recovery procedures are the second area. Section 5.8.4 proposes recovery scenarios for individual personas; this work would instead evaluate the recovery process end-to-end. It would cover the use of recovery codes, administrator-assisted resets through a service desk, and the re-enrolment of MFA factors. This is because recovery serves as both a source of friction and a common target for attackers, since a reset process that is too permissive can be exploited through social engineering. As a result, every recovery path should therefore be assessed for both usability and security.

Finally, future work should examine how authentication contexts change over time, as this study classified each login independently rather than recognising longitudinal patterns. In practice, a user's devices, locations and working patterns shift gradually. A longitudinal simulation run over several weeks could test whether the model's baseline of normal behaviour adapts

appropriately, and how often legitimate users are reclassified as higher risk (Mthethwa et al., 2025). The simulation could also evaluate mid-session context changes, such as a user switching networks after logging in. This would link to the continuous authentication approaches discussed in Section 2.3 (Ayeswarya S & K, 2025).

5.8.6 Compare Alternative IAM Platforms

This study was conducted solely in Keycloak, primarily because it is open-source, freely deployable in an isolated environment, and extensible enough to support a custom risk-adaptive authentication flow. Its findings therefore describe one platform. Future work should assess whether the methodology can be transferred to other Identity and Access Management (IAM) systems, and whether the usability-security trade-offs observed here hold across them.

Several components of the methodology are platform-independent and could be reused with little change. These include the synthetic persona framework, the test scenarios and the evaluation metrics of time to authenticate, interaction steps, prompts and failures. Keycloak supports standard protocols such as OpenID Connect and SAML 2.0, as do most IAM platforms, meaning that the test applications and AI agents could authenticate against alternative platforms without redesign. The risk-adaptive decision model, however, was implemented through Keycloak's conditional authentication flows, an adaptive authentication extension (Bartoš, 2026) and a separate risk engine. This would need to be rebuilt for each platform. This rebuild would itself reveal how easily each platform supports adaptive authentication.

A comparison could begin with other open-source platforms, such as Authentik or ZITADEL, which offer the same level of control and inspection as Keycloak. Commercial platforms such as Okta, Auth0 and Microsoft Entra ID would offer a different comparison, as they provide built-in risk-based authentication rather than requiring it to be configured. These platforms' risk engines are proprietary, meaning that evaluation would compare observable outcomes, such as which scenarios trigger additional factors, rather than the scoring logic itself. Quispe Salazar and Vereau Jacobo (2025) compared Keycloak with Okta and Auth0 through a systematic review, and experimental testing would add first-hand evidence to that comparison. Practical constraints would also need to be addressed, including licensing costs for advanced risk features, limits on developer or trial tenants, and terms of service that may restrict automated login testing or trigger bot detection.

5.9 Overall Conclusion

This project set out to examine how human factors affect the balance between usability and security in Multi-Factor Authentication (MFA), and whether a risk-adaptive approach could reduce unnecessary authentication friction without weakening protection. To investigate this, the project developed a risk-adaptive MFA prototype using Keycloak 26.0.7 and deployed it in an isolated virtual environment. A Flask risk engine scored each login attempt using three indicators: failed attempts, new devices and changes in country. The resulting score placed each login in a low-, medium-, or high-risk band, and Keycloak's conditional authentication flows then enforced the corresponding policy: password-only, password and TOTP, or password, TOTP, and email verification. Twelve synthetic personas, each with defined technical confidence, friction tolerance and attempt limits, were used to test the system across twelve authentication scenarios.

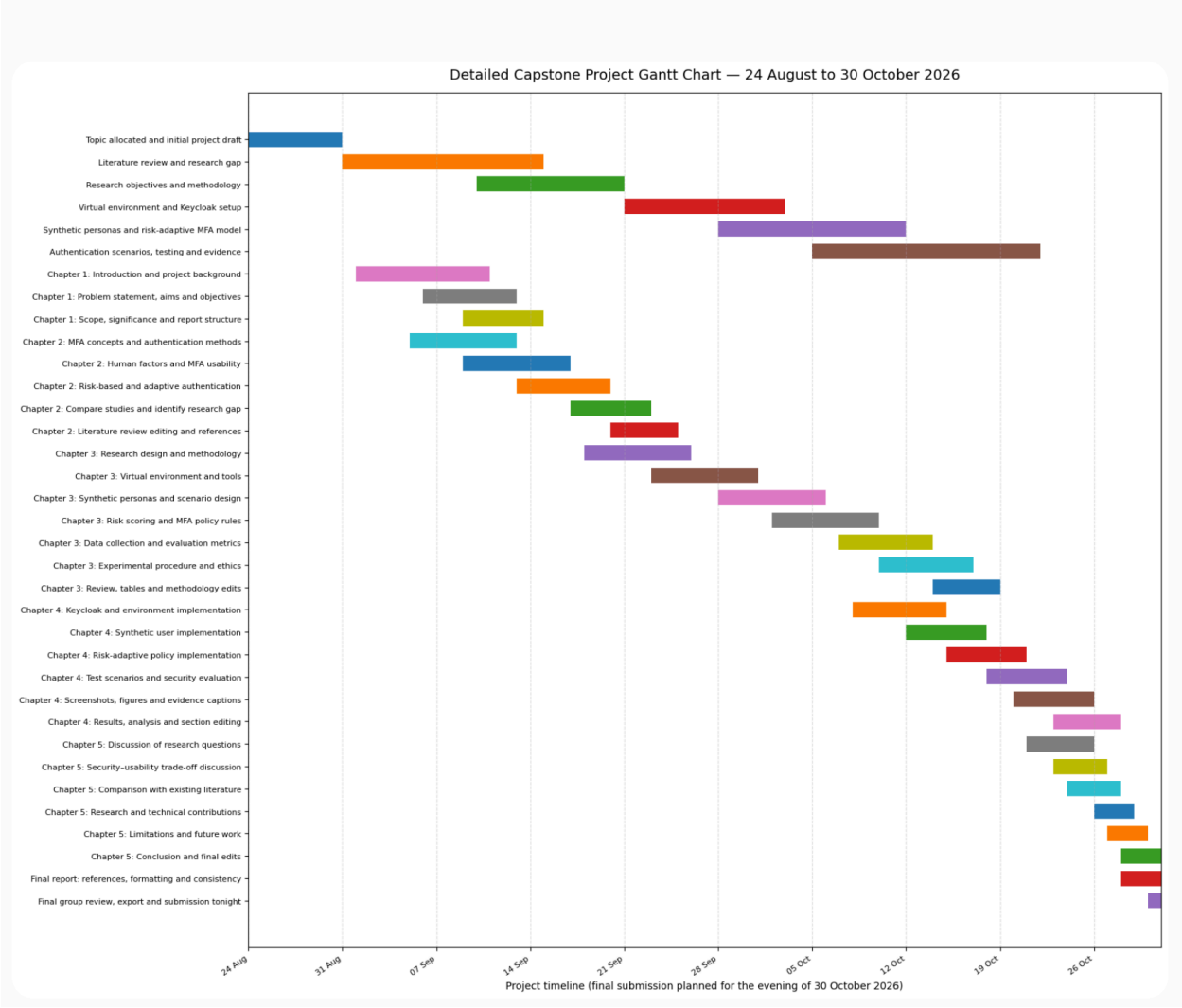
The results showed a clear trade-off between security and usability. Average completion time rose from 18.6 seconds under the low-risk policy to 32.1 seconds under the medium-risk policy and 56.3 seconds under the high-risk policy, roughly three times the low-risk time. All simulated abandonment (21.7%) occurred in medium- and high-risk scenarios, which suggests that each additional factor raises the likelihood that a user gives up. On the security side, every policy was enforced as configured. Incorrect passwords and incomplete MFA attempts were blocked, risk scores escalated as suspicious indicators accumulated, and no authentication bypass was observed. Because the model is adaptive, the additional friction was applied only when login conditions indicated elevated risk, and routine logins carried the lowest authentication burden.

The project makes four main contributions. It provides a controlled Keycloak testbed in which MFA configurations can be tested safely without affecting real users or systems. It defines a transparent risk-adaptive policy model that links contextual indicators to authentication requirements. It offers a synthetic persona and scenario framework for representing differences in user behaviour. Finally, it documents an evaluation procedure that records both security and friction outcomes, so the tests can be repeated and extended.

These findings must be read within the study's limitations. The personas were synthetic, and their abandonment and retry behaviour was produced by preset thresholds rather than by real users. The login context was supplied via predefined inputs, the experiment was conducted in a

laboratory environment, and only three authentication factors were implemented. Several important threats were not tested, including phishing, session hijacking, MFA fatigue attacks and account recovery exploitation. The results therefore show how the configured policies behave under controlled conditions; they do not show how real users would respond or how the model would perform in production. Future work should prioritise human usability testing, phishing-resistant authentication methods and more sophisticated risk classification. Overall, the project demonstrates that risk-adaptive MFA offers a practical way to reserve stronger authentication for riskier logins while keeping routine access simple. Any organisation adopting such a model should still validate it with real users and against current authentication guidance before deployment.

Project Management Timeline



Group Reflection

Introduction

When we started this capstone, our idea was fairly loose. We knew MFA frustrates people and that frustrated users find ways around it, but we didn't have a clear plan for how to study that. By the end, the seven of us had built a working risk-adaptive MFA system in Keycloak, tested it with twelve synthetic users, and written up what it showed, and also what it couldn't show. This reflection looks at how we got there, what went wrong along the way, and what we'd do differently.

What went well

The part we're proudest of is that the system actually works. Most of us had never touched Keycloak before this project, so getting it running in a virtual machine, setting up conditional authentication flows and connecting it to our Flask risk engine took a lot of trial and error. The first time a high-risk test login was asked for a password, a TOTP code and an email check, while a low-risk one got through with just a password, was a good moment for the group. It was the first time the idea in our proposal looked like something real.

Splitting the report by section worked well for us overall. Giving each person ownership of a chapter or part of the build meant people could go into real depth instead of everyone skimming everything. The persona framework improved a lot because of this. It started as a rough list of "types of users" and ended up as twelve profiles, each with its own technical confidence, friction tolerance and attempt limit, which gave the results section something consistent to measure against.

Our supervisor's feedback was also more useful than we expected at first. We were asked to cut the problem statement down to one paragraph, rewrite the scope in proper academic prose, and stop suggesting we were measuring real user engagement or fatigue. That last point stung a little, because a lot of our early writing leaned on it. But it was right. We were testing simulated users, and once we accepted that, the whole report became more honest and easier to defend.

Challenges

Time was our biggest problem. We planned twenty authentication scenarios and ended up running twelve. Cutting them felt like a setback at the time, but trying to do all twenty would have left the analysis and discussion rushed. Testing with real people was never realistic once we looked at the ethics process and our timeline, which is why we went with synthetic users. That choice let us run repeatable tests, but it also put a ceiling on what we could say about real behaviour.

The technical setup also threw up problems we didn't see coming. The shared clipboard in our hypervisor stopped working, so long configuration commands and environment variables had to be typed out by hand, which was slow and easy to get wrong. Keycloak also started showing profile-completion prompts before users could set up OTP, and it took a while to work out that this was a user-profile setting rather than a problem with our design. On top of that, usage limits on the AI agents capped how many runs we could do per persona.

The hardest lesson came near the end, when we put everyone's sections together. Things that made sense within one chapter didn't always match another. The policies, risk indicators, scoring thresholds and even figure numbers were described slightly differently depending on who wrote them, and some of our percentages didn't add up against each other. Nobody had done anything careless; we'd just been working in parallel without one agreed version of the system to check against. Fixing it at the end took far longer than agreeing on it at the start would have.

What we learned

On the technical side, we came away with hands-on experience of identity and access management, conditional flows, TOTP, brute-force protection and risk-based decisions. The bigger takeaway, though, was seeing the security–usability trade-off in our own numbers. Each extra factor made logins harder to compromise in our tests, but it also made them take longer, and every simulated abandonment happened under the stricter policies. Reading about that trade-off in the literature is one thing; watching login times go from about 19 seconds to over 56 made it much more concrete.

We also got better at being honest about our evidence. Our early drafts talked about results as if they applied to real users. Through revisions, we learned to separate what the system

measurably did from what we had assumed about the personas, and to say plainly which attacks we hadn't tested. We used to see the limitations section as a weakness; now we think it's part of what makes the rest of the report believable.

As a team, the main thing we learned is how much a shared reference document matters. One page agreed early on, with the risk formula, the policies, the persona list, which persona maps to which scenario, and how figures are numbered, would have saved us days of fixing inconsistencies later. We also learned that reading the whole report together regularly matters just as much as each person finishing their own part.

What we would do differently

If we did this again, we'd be more realistic about scope from the start and leave extra time for setup, because something always breaks. We'd write that shared reference document in the first couple of weeks and make everyone work from it. We'd also merge and read the full draft every few weeks instead of mostly at the end. And we'd start the ethics application much earlier, so that even a small usability study with real participants would be possible. That one change would have strengthened the human-factors side of the project more than anything else.

Conclusion

This capstone took us through the whole research process: finding a problem, reading around it, building something, testing it and being upfront about where it falls short. We finished with a working risk-adaptive MFA prototype and a framework that someone else could pick up and extend. We also came away with a much better sense of how people shape authentication security, and of how hard and how worthwhile it is to get seven people to produce one consistent piece of work.

References

Al-Omari, A. H., Alshanty, A. G., Alsarhan, A., & Omari, S. A. (2025). OTP security in wallet systems: A vulnerability assessment. International Journal of Innovative Research and Scientific Studies, 8(2), 2356–2371. https://doi.org/10.53894/ijirss.v8i2.5691

Amft, S., Höltervennhoff, S., Huaman, N., Krause, A., Simko, L., Acar, Y., & Fahl, S. (2023). “We’ve disabled MFA for you”: An evaluation of the security and usability of multi-factor authentication recovery deployments. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (pp. 3138–3152). Association for Computing Machinery. https://doi.org/10.1145/3576915.3623180

Arnold, D., Blackmon, B., Gibson, B., Moncivais, A. G., Powell, G. B., Skeen, M., Thorson, M. K., & Wade, N. B. (2022). The emotional impact of multi-factor authentication for university students. In CHI Conference on Human Factors in Computing Systems Extended Abstracts (pp. 1–4). Association for Computing Machinery. https://doi.org/10.1145/3491101.3516809

Australian Signals Directorate’s Australian Cyber Security Centre. (2026, September 9). Use unique and strong passwords (where the use of passkeys is unavailable). Cyber.gov.au. https://www.cyber.gov.au/publication/personal-cyber-security-handbook/use-unique-and-strong-passwords-where-the-use-of-passkeys-is-unavailable

Auth0. (n.d.). Adaptive MFA. Auth0 Docs. Retrieved October 30, 2026, from https://auth0.com/docs/secure/multi-factor-authentication/adaptive-mfa

Ayeswarya, S., & John Singh, K. (2025). Enhancing security and usability with context aware multi-biometric fusion for continuous user authentication. Scientific Reports, 15, Article 30627. https://doi.org/10.1038/s41598-025-14833-z

Bartoš, M. (2026). keycloak-adaptive-authn: Keycloak adaptive authentication extension [Computer software]. GitHub. https://github.com/mabartos/keycloak-adaptive-authn

Carjuman, N., Fook, L. C., & Hlaing, Z. C. (2025). Enhanced MFA framework against modern security threats. In Proceedings of the 2025 10th International Conference on Cloud Computing and Internet of Things (pp. 148–155). Association for Computing Machinery. https://doi.org/10.1145/3785520.3785523

Chatterjee, A., & Prinz, A. (2022). Applying Spring Security framework with KeyCloak-based OAuth2 to protect microservice architecture APIs: A case study. Sensors, 22(5), Article 1703. https://doi.org/10.3390/s22051703

Clark, B. (2025, December 30). Adaptive MFA. IBM. https://www.ibm.com/think/topics/adaptive-mfa

Divyabharathi, D., & Cholli, N. (2020). A review on identity and access management server (Keycloak). https://makhillpublications.co/files/published-files/mak-ijepe/2020/2-17-22.pdf

Gamayanto, I., Kurniawan, M. C., & Sanyoto, G. K. (2025). Security evaluation of Keycloak-based role-based access control in microservice architectures using the OWASP ASVS framework. Journal of Applied Informatics and Computing, 9(6), 3964–3973. https://doi.org/10.30871/jaic.v9i6.11604

Ghorbani Lyastani, S., Backes, M., & Bugiel, S. (2023). A systematic study of the consistency of two-factor authentication user journeys on top-ranked websites. In Proceedings of the 2023 Network and Distributed System Security Symposium. Internet Society. https://www.ndss-symposium.org/ndss-paper/a-systematic-study-of-the-consistency-of-two-factor-authentication-user-journeys-on-top-ranked-websites/

Gilsenan, C., Shakir, F., Alomar, N., & Egelman, S. (2023). Security and privacy failures in popular 2FA apps. In Proceedings of the 32nd USENIX Security Symposium (pp. 2079–2096). USENIX Association. https://www.usenix.org/conference/usenixsecurity23/presentation/gilsenan

Hastings, S., Moore, T., Gandal, N., & Barnir, N. (2026). Quantifying costs of enhanced security in multifactor authentication. Information Systems Frontiers, 28(2), 529–539. https://doi.org/10.1007/s10796-025-10641-y

Helena. (2023, September 27). Advantages and disadvantages of Keycloak. Intension. https://www.intension.de/en/infoblog/keycloak-advantages-disadvantages/

Hossain, M. A., & Raza, M. A. (2025). Exploring the effectiveness of multifactor authentication in preventing unauthorized access to online banking systems [Preprint]. SSRN. https://doi.org/10.2139/ssrn.5207142

Hossain, M. R. (2025). Identification of influencing factors for user mental models in novel authentication systems [Master's thesis, LUT University]. LUTPub. https://lutpub.lut.fi/handle/10024/171144

IBM. (2024). Cost of a data breach report 2024. https://www-api.ibm.com/adobe/assets/urn:aaid:aem:c4711149-da99-4654-9114-8ee524108540/original/as/cost_of_a_data_breach_report_2024.pdf

Jubur, M., Saxena, N., & Reegu, F. A. (2025). Usability and security analysis of the compare-and-confirm method in mobile push-based two-factor authentication. IEEE Transactions on Mobile Computing, 24(6), 4623–4638. https://doi.org/10.1109/TMC.2024.3524093

Karuppiah, S. P. (2025). Understanding the behaviour of business users in multi-factor authentication adoption [Master's thesis, LUT University]. LUTPub. https://lutpub.lut.fi/bitstream/handle/10024/169612/Masterthesis_Karuppiah_SivamPrasath.pdf

Kepkowski, M., Machulak, M., Wood, I., & Kaafar, D. (2023). Challenges with passwordless FIDO2 in an enterprise setting: A usability study (arXiv:2308.08096) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2308.08096

Keycloak. (n.d.). Keycloak. Retrieved September 14, 2026, from https://www.keycloak.org/

Keymate Team. (2026). Event-based OTP authenticator for Keycloak. Keymate. https://keymate.io/extensions/keycloak/event-based-otp

Kosinski, M. (2024, January 4). Multi-factor authentication. IBM. https://www.ibm.com/think/topics/multi-factor-authentication

Kovačić, S., & Bilić, I. (2026). AI-enhanced social engineering: Emerging threats and human-centric countermeasures. Revista Română de Informatică și Automatică, 36(1), 63–76. https://doi.org/10.33436/v36i1y202505

Kruzikova, A., Muzik, M., Knapova, L., Dedkova, L., Smahel, D., & Matyas, V. (2024). Two-factor authentication time: How time-efficiency and time-satisfaction are associated with perceived security and satisfaction. Computers & Security, 138, Article 103667. https://doi.org/10.1016/j.cose.2023.103667

Lassak, L., Pan, E., Ur, B., & Golla, M. (2024). Why aren't we using passkeys? Obstacles companies face deploying FIDO2 passwordless authentication. In Proceedings of the 33rd USENIX Security Symposium (pp. 7231–7248). USENIX Association. https://www.usenix.org/conference/usenixsecurity24/presentation/lassak

Liu, Y., & Huang, H. (2026). WiP: Characterizing and defending against mobile-agent-driven MFA automation (arXiv:2609.02154) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2609.02154

Marky, K., Ragozin, K., Chernyshov, G., Matviienko, A., Schmitz, M., Mühlhäuser, M., Eghtebas, C., & Kunze, K. (2022). "Nah, it's just annoying!" A deep dive into user perceptions of two-factor authentication. ACM Transactions on Computer-Human Interaction, 29(5), 1–32. https://doi.org/10.1145/3503514

Meyer, L. A., Romero, S., Bertoli, G., Burt, T., Weinert, A., & Lavista Ferres, J. (2023). How effective is multifactor authentication at deterring cyberattacks? Microsoft. https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/mfa-microsoft-research-paper-update-347347.pdf

Microsoft. (2024, May 13). Security at your organization: Multifactor authentication (MFA) statistics. Microsoft Learn. https://learn.microsoft.com/en-us/partner-center/security/security-at-your-organization

Microsoft. (2026, August 14). Microsoft Entra ID Protection risk-based access policies. Microsoft Learn. https://learn.microsoft.com/en-us/entra/id-protection/concept-identity-protection-policies

Mihai, D., Mihăilescu, M.-E., Carabaş, M., Năstase, V.-I., & Ţăpuş, N. (2024). Security posture improvement with high-availability Keycloak SSO. In 2024 23rd RoEduNet Conference: Networking in Education and Research (RoEduNet) (pp. 1–7). IEEE. https://doi.org/10.1109/roedunet64292.2024.10722696

Mohammed, A. H. Y., Dziyauddin, R. A., & Latiff, L. A. (2023). Current multi-factor of authentication: Approaches, requirements, attacks and challenges. International Journal of Advanced Computer Science and Applications, 14(1). https://doi.org/10.14569/ijacsa.2023.0140119

Mthethwa, S., Ndhlovu, N., Myaka, S., Ntshangase, S., Shadung, D., & Singano, T. (2025). Enhancing digital wallet security: A systematic comparison of passwordless and risk-based

authentication approaches. In IFIP Advances in Information and Communication Technology (pp. 230–242). Springer. https://doi.org/10.1007/978-3-032-13075-4_16

National Institute of Standards and Technology. (2026). CVE-2026-18963 detail. National Vulnerability Database. https://nvd.nist.gov/vuln/detail/cve-2026-18963

Nurhadi, N., Suhaidi, M., & Athariq, M. (2026). Design and quantitative evaluation of a Keycloak-based single sign-on architecture for integrated institutional information systems. ILKOM Jurnal Ilmiah, 18(2), 237–254. https://doi.org/10.33096/ilkom.v18i2.3368.237-254

Okello, M. (2023, January 13). What is Keycloak and why is it loved by developers? Medium. https://okellomarvinkevin.medium.com/what-is-keycloak-and-why-is-it-loved-by-developers-6ab465320cb2

OWASP. (2025). Multifactor authentication cheat sheet. OWASP Cheat Sheet Series. https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html

Parmar, V., Sanghvi, H. A., Patel, R. H., & Pandya, A. S. (2022). A comprehensive study on passwordless authentication. In 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) (pp. 1266–1275). IEEE. https://doi.org/10.1109/icscds53736.2022.9760934

Picard, C., & Pierre, S. (2023). RLAuth: A risk-based authentication system using reinforcement learning. IEEE Access, 11, 61129–61143. https://doi.org/10.1109/ACCESS.2023.3286376

Quispe Salazar, P. N., & Vereau Jacobo, E. W. (2025). Evaluating Keycloak as an identity server versus commercial solutions in multi-platform organizational environments. In Proceedings of the LACCEI International Multiconference on Entrepreneurship, Innovation and Regional Development (LEIRD 2025). LACCEI. https://doi.org/10.18687/LEIRD2025.1.1.407

Rivera-Dourado, M., Gestal, M., Pazos, A., & Vázquez-Naya, J. M. (2021). An analysis of the current implementations based on the WebAuthn and FIDO authentication standards. Engineering Proceedings, 7(1), Article 56. https://doi.org/10.3390/engproc2021007056

Robertson, D., & Kosinski, M. (2026, July 17). Guide to risk-based authentication. IBM. https://www.ibm.com/think/topics/risk-based-authentication

Sakshi, M., Sawant, S., Pradeepa, S., & Mba, S. (2026). The security-usability trade-off: A multi-dimensional analysis of multi-factor authentication and its impact on digital trust among diverse smartphone users. International Journal of Creative Research Thoughts, 14(6). https://www.ijcrt.org/papers/IJCRT2606100.pdf

Schechter, S., Brush, A. J. B., & Egelman, S. (2009). It's no secret. Measuring the security and reliability of authentication via "secret" questions. In 2009 30th IEEE Symposium on Security and Privacy (pp. 375–390). IEEE. https://doi.org/10.1109/sp.2009.11

Shuford, J. (2024). Exploring the efficacy of behavioral biometrics in cybersecurity. Journal of Artificial Intelligence General Science, 6(1), 577–593. https://doi.org/10.60087/jaigs.v6i1.285

Sloan, E. (2023, May 31). The benefits (and flaws) of FIDO2 web authentication. Packetlabs. https://www.packetlabs.net/posts/the-benefits-and-flaws-of-fido2-web-authentication/

Son, W., Kwon, S., Hwang, M.-N., & Lee, J.-H. (2025). An AI-based risk-adaptive MFA framework for Keycloak using behavioral login analysis. In Proceedings of the 2025 International Conference on Intelligent Computing and Its Emerging Applications (pp. 7–12). Association for Computing Machinery. https://doi.org/10.1145/3803817.3803827

Temoshok, D., Fenton, J., Choong, Y.-Y., Lefkovitz, N., Regenscheid, A., Galluzzo, R., & Richer, J. (2025). Digital identity guidelines: Authentication and authenticator management (NIST Special Publication 800-63B-4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-63b-4

Tinkoff, D., Van der Veken, L., Chui, M., & Balakrishnan, T. (2026, August 25). The state of AI in 2026: On the road to ROI. McKinsey & Company. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai

Tran-Truong, P. T., Pham, M. Q., Son, H. X., Nguyen, D. L. T., Nguyen, M. B., Tran, K. L., Van, L. C. P., Le, K. T., Vo, K. H., Kim, N. N. T., Nguyen, T. M., & Nguyen, A. T. (2025). A systematic review of multi-factor authentication in digital payment systems: NIST standards alignment and industry implementation analysis. Journal of Systems Architecture, 162, Article 103402. https://doi.org/10.1016/j.sysarc.2025.103402

White, M. (2025, April 28). Passkeys: Benefits, limitations, and will they replace passwords? Specops Software. https://specopssoft.com/blog/passkeys-benefits-limitations-passwords/

Whitman, D. R., Lawson, C. J., Reynolds, M. T., Collins, A. P., & Shekhar, S. (2025). Trade-offs between security strength and user convenience in multi-factor authentication systems [Preprint]. ResearchGate. https://www.researchgate.net/publication/400449475_Trade-offs_Between_Security_Strength_and_User_Convenience_in_Multi-Factor_Authentication_Systems

Wiefling, S., Jørgensen, P. R., Thunem, S., & Lo Iacono, L. (2023). Pump up password security! Evaluating and enhancing risk-based authentication on a real-world large-scale online service. ACM Transactions on Privacy and Security, 26(1), Article 6. https://doi.org/10.1145/3546069

World Wide Web Consortium. (2023). Web Content Accessibility Guidelines (WCAG) 2.2. https://www.w3.org/TR/WCAG22/

Würsching, L., Putz, F., Haesler, S., & Hollick, M. (2023). FIDO2 the rescue? Platform vs. roaming authentication on smartphones. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery. https://doi.org/10.1145/3544548.3580993W

Chapter 2 – Literature Table

Table 2.1 compares 20 studies on MFA usability, security and Keycloak, all published 2022-2026, using the eight columns required by the INFO3016 guidelines.

Author(s) & Year	Problem	Objective	Method	Tools / Technology	Findings	Gap	Improvement (our project)
Amft et al. (2023)	MFA recovery can lock legitimate users out or give attackers a way around MFA.	Evaluate the security and usability of MFA account recovery.	Review of recovery documentation on 1,303 websites plus hands-on recovery tests on 71 websites.	2FA Directory, Tranco rankings, test accounts, screen recordings.	Only 37 of 71 accounts (52%) were recovered, 30 were red, and 30 were lost. 24.6% of sites documented no recovery, and the documentation matched the real experience on only 9 of 71.	Tested by researchers rather than typical users. Websites only, with no IAM platform.	We include lockout and lost-device scenarios in Keycloak and measure how often users recover.
Chatterjee & Prinz (2022)	Microservice APIs in an e-health app can be called by unauthorised users.	Secure the APIs with Spring Security and Keycloak, including Keycloak two-factor login.	Case study with security and unit testing (22 test cases across 16 attack scenarios).	Spring Boot, Spring Security, Keycloak (OAuth2/OIDC, TOTP via Google Authenticator), Mockito.	All simulated attacks were blocked. Password + OTP responses were received in 0.1–1 s, and the system handled about 300 concurrent users.	Measures server speed, not human login time. No users tested, and MFA is the same for everyone.	We measure login time and errors from human-like agents and apply OTP only when the risk is higher.
Gamayanto et al. (2025)	Microservice apps need verified access control.	Assess Keycloak role-based access control against a security standard.	Security testing against OWASP ASVS Level 2.	Keycloak RBAC, Postman, OWASP ZAP.	No critical vulnerabilities were found.	Access control o—no MFA or usability evaluation.	We add MFA and usability metrics on top of Keycloak's role-based rules.

Ghorbani Lyastani et al. (2023)	Inconsistent 2FA setup and login steps across websites add cognitive friction and reduce adoption.	Compare how consistent 2FA discovery, setup, use and deactivation are across websites.	Systematic study of 85 top-ranked websites.	Screen-recorded walkthroughs, grounded-theory coding, clustering; SMS, TOTP, push and WebAuthn.	Websites differed in 6–7 of 14 factors on average. Only 31 offered device remembrance and only 6 made 2FA mandatory.	No user testing, and the risk context was fixed.	We design one consistent Keycloak flow that only changes its requirements when the risk changes.
Gilsenan et al. (2023)	Backups in TOTP authenticator apps may expose users' secret keys.	Assess the security and privacy of TOTP app backup and recovery.	Empirical analysis of 22 popular Android TOTP apps (over 180 million installs).	Network traffic analysis (mitmproxy, Frida), app decompilation (JADX).	2 apps sent plaintext TOTP secrets to their developers, 4 relied only on SMS for recovery, and most encrypted backups had serious flaws.	Examines the apps only, not user behaviour or the server side.	We include TOTP secret handling in our threat model and test OTP errors from the user's side.
Hastings et al. (2025)	Push MFA is vulnerable to fatigue attacks; number-matching was introduced to mitigate them.	Measured the impact of how number matching affected sign-in outcomes.	Analysis of university sign-in logs before and after the change.	Institutional mobile MFA logs (push vs number matching).	Mobile MFA failures rose from 10.2% to 17.9%.	Log data only, with no psychological data. User roles were unknown.	We link each failure to a simulated user profile and a behaviour cause.
Hossain & Raza (2025)	Online banking accounts face unauthorised access.	Evaluate how effective MFA is for online banking.	User perception study (confirm the method and sample in the full paper)†.	OTP, biometrics, security tokens.	OTP had the highest attack success rate (15%) and biometrics the lowest (5%).	Banking only. No IAM platform or adaptive authentication.	We test OTP inside Keycloak with risk-based rules.

Jubur et al. (2025)	Push 2FA is vulnerable to fatigue and concurrency attacks, in which users approve a fraudulent prompt.	Evaluate the usability and security of the compare-and-confirm push 2FA.	Lab study with 65 participants (24 login trials each, half of which were simulated attacks).	Custom web app, Firebase Cloud Messaging, Android phones, mixed-effects models.	Genuine logins were approved 97.12% of the time, but about 49.6% of attack prompts were still approved. Mean response time was 8.08 s.	Young, tech-savvy sample in a lab, with simulated attacks.	We use TOTP rather than push and simulate fatigue through agent behaviour via Keycloak lockouts.
Kepkowski et al. (2023)	Passwordless FIDO2 is hard to integrate into real enterprise IAM environments.	Identify the enterprise use cases that make FIDO2 difficult to implement.	Usability survey of 118 IT professionals.	FIDO2 passwordless authentication.	Over 60% named account recovery as a challenge. Remote workforces and legacy systems were also problems.	Professionals' opinions only, with no implementation or testing.	We implement and test MFA inside an IAM platform (Keycloak).
Kruzikova et al. (2024)	Little is known about how the time required for 2FA affects users' views of it.	Test how actual and perceived login time affects satisfaction and perceived security.	User study with 337 smartphone users aged 26–81 (path analysis).	NFC token, payment card with card reader, smartphone e-banking.	Login time depended on age, gender and experience. Users who were satisfied with the time also rated the method as more secure and more satisfying.	Only two token methods in a banking context. No TOTP or adaptive MFA.	We measure completion time for each risk profile, including slower agent behaviour.
Lassak et al. (2024)	Companies have been slow to deploy FIDO2 passwordless login.	Identify the obstacles companies face in deploying FIDO2.	28 semi-structured interviews (32 people) plus a card-sorting task.	FIDO2, WebAuthn, passkeys, Windows Hello, YubiKeys, Azure AD.	The main barriers were account recovery, friction, technical issues, regulation and security	Small qualitative sample, mostly from Germany and the US.	We provide measured evidence of friction from a controlled Keycloak simulation.

					culture. One deployment reached only 65% uptake after a year.		
Marky et al. (2022)	Banking 2FA may be hard for inexperienced users.	Understand users' experiences with banking 2FA.	Semi-structured interviews (42 participants).	Online-banking 2FA apps.	2FA felt tedious, and some users put both apps on the same device.	Qualitative only, with no system built or tested.	We turn these behaviours into measurable simulated actions.
Mihai et al. (2024)	Password-only login isn't enough for university services.	Deploy a high-availability Keycloak SSO with MFA for students and staff.	Implementation case study.	Keycloak SSO cluster, MFA, virtual machines.	A fault-tolerant MFA deployment that aims for a good user experience.	Little or no measured user experience (confirm in the full paper).	We measure usability outcomes: success rate, time and retries.
Nurhadi et al. (2026)	Separate logins across university systems cause repeated sign-ins and inconsistent security.	Design and evaluate a Keycloak-based SSO architecture.	Quantitative experiment in a university environment with 100 user accounts.	Keycloak, OAuth 2.0, OpenID Connect, JWT.	Average login time fell by 62%, throughput rose by 85%, the authentication success rate was 100%, and task completion time fell by 51%.	SSO only. MFA and human error were not evaluated.	We add risk-based MFA and error scenarios on top of Keycloak.
Quispe Salazar & Vereau Jacobo (2025)	Organisations must choose between open-source and commercial IAM.	Compare Keycloak with Okta and Auth0.	PRISMA systematic review (40 studies).	Keycloak, Okta, Auth0.	Compares the three on security, performance, usability, cost and scalability.	Review only, with no experiment.	We provide first-hand experimental data on Keycloak MFA.

Sawant & Pradeepa (2026)	MFA friction may reduce users' trust and engagement.	Analyse how MFA methods affect trust among smartphone users.	Quantitative survey (N = 221) with ANOVA and correlation.	Google Forms, 5-point Likert scale.	70% favoured biometrics, 75% had quit an app over a complicated login, and 54% were frustrated by using separate devices.	Self-reported, non-random regional sample. No technical testing.	We test these friction points technically in Keycloak.
Son, W., Kwon, S., Hwang, M.-N., & Lee, J.-H. (2025)	Static MFA adds friction to every login.	Build an AI risk-adaptive MFA framework in Keycloak.	Framework design and model evaluation.	Keycloak, logistic regression (login time, IP, user agent).	Real-time risk scoring with 97% accuracy.	No human experience, errors or fatigue measured.	We measure the usability and human-error outcomes of adaptive MFA.
Tran-Truong et al. (2025)	MFA in digital payments varies in quality and in how well it follows standards.	Review the MFA in payment systems against NIST standards.	Systematic review of 70 papers (2017–2024).	NIST SP 800-63 framework.	Research favours biometrics, while industry still relies heavily on OTP (confirm the 33% figure).	Payments only, with no experiment.	We improve OTP-based MFA with context-aware rules.
Wiefling et al. (2023)	Few validated real-world resources exist on risk-based authentication (RBA).	Evaluate and optimise RBA on a large real-world online service.	Long-term study of 3.3 million users and 31.3 million logins over more than a year.	RBA, machine-learning parameter tuning, IP address, round-trip time.	Users benefit from RBA after only a few logins, and legitimate users are rarely asked to re-authenticate.	A single service, not an IAM platform like Keycloak. No direct user-experience measures.	We implement risk-based rules in Keycloak and measure usability.

Würsching et al. (2023)	It is unclear whether smartphone users accept FIDO2 passwordless login.	Compare platform (Touch ID) and roaming (security key) FIDO2 authentication.	Lab study with 87 participants.	iPhone SE, Touch ID, YubiKey 5C NFC, WebAuthn.	Median SUS was 95 for the platform and 90 for roaming. 81% of the roaming group worried about losing the key.	Young sample, iOS only, and participants were briefed in advance.	Our simulation includes scenarios where the user's device is lost or unavailable.
Picard, C., & Pierre, S. (2023)	Static authentication policies can apply the same level of security to every login, which may create unnecessary friction and may not respond well to changes in user behaviour or context.	Develop and evaluate RLAuth, a risk-based authentication system that uses reinforcement learning to classify authentication contexts and decide when explicit authentication is required.	The study used four weeks of mobile-phone data from 42 users. A deep reinforcement-learning model classified authentication contexts as normal or abnormal using eight contextual variables.	Reinforcement learning, mobile contextual data, location, time of day, network connection, previous authenticator use, and application sensitivity.	RLAuth achieved a G-Mean of 92.62%, compared with 82.29% for a support vector machine, and required about 4.946 ms per authentication decision. Explicit authentication was requested only when abnormal behaviour was detected.	The study did not directly measure reductions in user friction. It also had limitations in familiar environments, and some confidence thresholds were selected arbitrarily.	Our project applies a simpler three-level risk model in **Keycloak** and evaluates human-oriented outcomes such as login time, retries, errors, and authentication friction, using synthetic users.

Table 2.1: *Comparison of existing MFA and Keycloak studies*