## **Abstract**

## In einer zunehmend digitalisierten Welt, in der Cyberangriffe und Identitätsdiebstahl an der Tagesordnung sind, ist die Sicherheit von Authentifizierungsprozessen von entscheidender Bedeutung. Dieses Projekt zielt darauf ab, ein adaptives Authentifizierungssystem zu entwickeln, das JSON Web Tokens (JWT) nutzt und Machine Learning (ML) integriert, um die Sicherheitsanforderungen moderner Webanwendungen zu erfüllen. Die Relevanz dieses Themas liegt in der Notwendigkeit, Authentifizierungsmethoden zu verbessern, um unbefugten Zugriff zu verhindern und gleichzeitig die Benutzerfreundlichkeit zu gewährleisten. Die Zielsetzung des Projekts umfasst die Entwicklung eines Systems, das in der Lage ist, die Authentifizierungsanforderungen dynamisch anzupassen, basierend auf dem Risiko eines Anmeldeversuchs. Zu den messbaren Zielen gehören die Implementierung eines funktionalen Prototyps, der eine sichere Benutzerregistrierung und -anmeldung ermöglicht sowie eine adaptive Authentifizierung bietet. Darüber hinaus soll ein Verhaltensanalysator entwickelt werden, der mithilfe von ML-Modellen Muster im Benutzerverhalten erkennt und Risikobewertungen erstellt. Das Vorgehen umfasst eine systematische Analyse bestehender Authentifizierungsmethoden und Sicherheitsanforderungen. Zunächst wird eine grundlegende JWT-Implementierung erstellt, gefolgt von der Entwicklung des adaptiven Authentifizierungsmoduls und des Verhaltensanalysators. Die Integration von Machine Learning zur Risikobewertung wird ebenfalls Teil des Entwicklungsprozesses sein. Abschließend erfolgt eine umfassende Testphase zur Validierung der Sicherheitsmechanismen und der Benutzerfreundlichkeit des Systems. Die erwarteten Ergebnisse des Projekts sind sowohl theoretischer als auch praktischer Natur. Theoretisch wird das Projekt neue Erkenntnisse über die Implementierung adaptiver Authentifizierungsmechanismen in Verbindung mit Machine Learning liefern. Praktisch wird ein funktionierendes Prototyp-System bereitgestellt, das die Sicherheit von Webanwendungen durch intelligente Anpassungen in den Authentifizierungsprozessen verbessert und gleichzeitig den Benutzern eine intuitive Interaktion ermöglicht.

## **Projektplan**

## **1. Projektvorbereitung**

* **Inhalte:**
  + Definition der Projektziele und -anforderungen.
  + Erstellung eines Lastenhefts.
  + Recherche zu bestehenden Authentifizierungsmethoden und Sicherheitsanforderungen.
* **Dauer:** Woche 1

## **2. Systemarchitektur und Design**

* **Inhalte:**
  + Entwurf der Systemarchitektur.
  + Auswahl der Technologien (z.B. Node.js, Express, MongoDB).
  + Erstellung von Diagrammen (z.B. UML-Diagramme) zur Veranschaulichung der Architektur.
* **Dauer:** Woche 1-2

## **3. Implementierung der Benutzerregistrierung und -anmeldung**

* **Inhalte:**
  + Entwicklung der API-Routen für die Registrierung und Anmeldung.
  + Implementierung der JWT-Generierung und -Verifizierung.
  + Dokumentation des Codes.
* **Dauer:** Woche 2-3

## **4. Entwicklung des Moduls zur adaptiven Authentifizierung**

* **Inhalte:**
  + Programmierung des Moduls zur dynamischen Anpassung der Authentifizierungsanforderungen basierend auf Risikobewertungen.
  + Integration von Machine Learning zur Analyse des Benutzerverhaltens.
  + Dokumentation des Moduls.
* **Dauer:** Woche 3-5

## **5. Integration von Sicherheitsrichtlinien**

* **Inhalte:**
  + Entwicklung von adaptiven Sicherheitsrichtlinien für verschiedene Risikostufen.
  + Implementierung zusätzlicher Sicherheitsmaßnahmen (z.B. MFA bei verdächtigen Anmeldungen).
  + Dokumentation der Sicherheitsrichtlinien.
* **Dauer:** Woche 5-6

## **6. Testen und Validieren des Systems**

* **Inhalte:**
  + Durchführung von Funktionstests und Sicherheitstests.
  + Behebung von Fehlern und Optimierung des Systems basierend auf Testergebnissen.
  + Dokumentation der Testergebnisse.
* **Dauer:** Woche 6

## **7. Erstellung der Ausarbeitung**

* **Inhalte:**
  + Zusammenstellung aller Projektdokumentationen.
  + Erstellung einer abschließenden Projektdokumentation mit theoretischen Grundlagen, Implementierungsdetails und Testergebnissen.
* **Dauer:** Woche 6

## **8. Vorbereitung der Präsentation**

* **Inhalte:**
  + Erstellung einer Präsentation zur Vorstellung des Projekts.
  + Übung der Präsentation im Team.
* **Dauer:** Woche 6

## **Zeitplan**

Hier ist eine Übersicht über den Zeitplan in Form einer Tabelle:

| **Arbeitspaket** | **Dauer** | **Verantwortlich** |
| --- | --- | --- |
| 1. Projektvorbereitung | Woche 1 | Projektleiter |
| 2. Systemarchitektur und Design | Woche 1-2 | Technischer Architekt |
| 3. Implementierung Benutzerregistrierung | Woche 2-3 | Entwickler |
| 4. Entwicklung Modul adaptive Authentifizierung | Woche 3-5 | Entwickler |
| 5. Integration Sicherheitsrichtlinien | Woche 5-6 | Sicherheitsspezialist |
| 6. Testen und Validieren | Woche 6 | QA-Ingenieur |
| 7. Erstellung der Ausarbeitung | Woche 6 | Projektleiter |
| 8. Vorbereitung der Präsentation | Woche 6 | Projektleiter |

## **Meilensteine**

1. Abschluss von Konzept und Anforderungsanalyse (Ende Woche 1)
2. Abschluss des Systemdesigns (Ende Woche 2)
3. Fertigstellung der Benutzerregistrierungs- und Anmeldemodule (Ende Woche 3)
4. Abschluss des Moduls zur adaptiven Authentifizierung (Ende Woche 5)
5. Integration der Sicherheitsrichtlinien (Ende Woche 6)
6. Abschluss aller Tests und Validierungen (Ende Woche 6)
7. Fertigstellung der Projektdokumentation (Ende Woche 6)
8. Durchführung der Präsentation (Ende Woche 6)

## **1. Welche Fragestellung soll im Rahmen des Projekts bearbeitet werden?**

Die zentrale wissenschaftliche Fragestellung lautet: **"Wie kann ein adaptives Authentifizierungssystem, das JSON Web Tokens (JWT) verwendet und durch Machine Learning unterstützt wird, die Sicherheit und Benutzerfreundlichkeit in modernen Webanwendungen verbessern?"** Diese Frage untersucht, wie adaptive Authentifizierungsmethoden implementiert werden können, um das Risiko unbefugten Zugriffs zu minimieren, während gleichzeitig die Benutzererfahrung optimiert wird.

## **2. Welche Grundlagen werden beschrieben und ausgearbeitet?**

Im Rahmen des Projekts werden folgende Grundlagen beschrieben und ausgearbeitet:

* **JSON Web Tokens (JWT):** Die Struktur, Funktionsweise und Sicherheitsaspekte von JWTs, einschließlich der Herausforderungen bei ihrer Verwendung.
* **Adaptive Authentifizierung:** Definition und Merkmale adaptiver Authentifizierungssysteme, einschließlich der Verwendung von Kontextinformationen (z.B. Standort, Gerät) zur Risikobewertung.
* **Machine Learning:** Grundlagen des maschinellen Lernens, insbesondere wie ML-Modelle zur Analyse von Benutzerverhalten eingesetzt werden können, um Risikoprofile zu erstellen.
* **Sicherheitsmechanismen:** Eine Analyse der Sicherheitsmechanismen, die in Authentifizierungssystemen implementiert werden können, um unbefugten Zugriff zu verhindern.

## **3. Welche (weiterführenden) aktuellen (Forschungs-)Arbeiten gibt es dazu?**

Es gibt mehrere aktuelle Forschungsarbeiten und Artikel, die sich mit den Themen adaptive Authentifizierung und Machine Learning befassen:

* **Adaptive Authentication and Machine Learning:** Dieser Artikel beschreibt, wie adaptive Authentifizierungsmethoden konfiguriert werden können, um die Benutzerfreundlichkeit zu verbessern und gleichzeitig hohe Sicherheitsstandards aufrechtzuerhalten. Es wird erläutert, wie ML-Algorithmen verwendet werden können, um Risikoprofile zu erstellen und die Authentifizierungsanforderungen dynamisch anzupassen
* **Adaptive Multifaktor-Authentifizierung:** Studien zeigen, dass adaptive MFA über herkömmliche Methoden hinausgeht, indem sie kontextuelle Informationen analysiert und die Authentifizierungsstufen entsprechend anpasst. Dies verbessert nicht nur die Sicherheit, sondern auch die Benutzererfahrung
* **Verhaltensbiometrie in der adaptiven Authentifizierung:** Forschungsarbeiten untersuchen den Einsatz von Verhaltensbiometrie zur Identifikation von Anomalien im Benutzerverhalten und deren Anwendung in adaptiven Authentifizierungssystemen
* **Sicherheitsrisiken von JWTs:** Es gibt zahlreiche Studien über die Sicherheitsrisiken bei der Verwendung von JWTs und Best Practices zur Minimierung dieser Risiken. Diese Arbeiten sind wichtig für das Verständnis der Sicherheitsarchitektur des geplanten Systems

## **1. Was soll die fertige Software an Funktionalitäten bieten?**

Die fertige Software soll folgende Funktionalitäten bieten:

* **Benutzerregistrierung und -authentifizierung:** Ermöglicht es Benutzern, sich zu registrieren und anzumelden, wobei ein JWT generiert wird.
* **Adaptive Authentifizierung:** Dynamische Anpassung der Authentifizierungsanforderungen basierend auf dem Risiko, das mit dem aktuellen Anmeldeversuch verbunden ist. Dies kann zusätzliche Sicherheitsfragen oder Multi-Faktor-Authentifizierung (MFA) umfassen.
* **Verhaltensanalyse:** Verwendung von Machine Learning zur Überwachung und Analyse des Benutzerverhaltens, um Muster zu erkennen und Risikobewertungen zu erstellen.
* **Token-Management:** Verwaltung von JWTs, einschließlich Ablaufzeiten und Widerrufmechanismen.
* **API-Schutz:** Sicherer Zugriff auf geschützte Routen durch Überprüfung der JWTs in den Autorisierungsheadern.

## **2. Welches (Anwender-/Anwendungs-)Problem soll die Software lösen?**

Die Software soll das Problem der unzureichenden Sicherheit in Authentifizierungsprozessen lösen, indem sie:

* Die Risiken von unbefugtem Zugriff durch adaptive Authentifizierungsmechanismen minimiert.
* Die Benutzerfreundlichkeit verbessert, indem sie nur dann zusätzliche Informationen anfordert, wenn das Risiko erhöht ist.
* Eine effektive Verhaltensanalyse implementiert, um potenzielle Bedrohungen frühzeitig zu erkennen und zu adressieren.

## **3. Welche (Nutzer-/System-)Schnittstellen gibt es? Wie sehen diese aus?**

## **Schnittstellen**

* **Benutzerschnittstelle (UI):** Eine Webanwendung mit Formularen für die Registrierung und Anmeldung sowie einer Benutzeroberfläche zur Anzeige von Benutzerprofilinformationen und Sicherheitsoptionen.
* **API-Schnittstelle:**
  + **POST /register:** Registrierung eines neuen Benutzers (Eingabe: E-Mail, Passwort).
  + **POST /login:** Authentifizierung des Benutzers (Eingabe: E-Mail, Passwort; Ausgabe: JWT).
  + **GET /protected:** Zugriff auf geschützte Ressourcen (Überprüfung des JWT im Header).
  + **POST /token/refresh:** Erneuerung eines abgelaufenen Tokens.

## **Grobarchitektur**

+-------------------+ +-------------------+

| Benutzer- | | Authentifizierungs|

| oberfläche | <------> | API |

| (Frontend) | | (Backend) |

+-------------------+ +-------------------+

|

|

v

+---------------------------+

| Datenbank |

| (Benutzerdaten) |

+---------------------------+

## **4. Welche Technologien sollen verwendet werden?**

## **Frameworks und Bibliotheken**

* **Backend:**
  + **Kotlin**
  + **MySQL DB**
  + **jsonwebtoken** für die Erstellung und Verifizierung von JWTs.
  + **TensorFlow.js** oder **scikit-learn** für die Implementierung von Machine Learning Modellen zur Verhaltensanalyse.
* **Frontend:**
  + **HTML und CSS** für die Entwicklung der Benutzeroberfläche.

## **5. Eigenanteil: Was ist vorhanden? Was muss selbst programmiert werden?**

## **Vorhandene Komponenten**

* Es können Open-Source-Bibliotheken wie jsonwebtoken, Express.js und ML-Bibliotheken wie TensorFlow.js verwendet werden, um grundlegende Authentifizierungsfunktionen zu implementieren.

## **Selbst programmierte Anteile**

1. **Adaptive Authentifizierung:**
   * Programmierung des Moduls zur dynamischen Anpassung der Authentifizierungsanforderungen basierend auf Risikobewertungen.
2. **Verhaltensanalyse-Modul:**
   * Entwicklung eines Moduls zur Analyse des Benutzerverhaltens mithilfe von Machine Learning, um Muster zu erkennen und Risikobewertungen zu erstellen.
3. **Token-Management-System:**
   * Entwicklung eines Systems zur Verwaltung von Token-Ablaufzeiten und Widerrufmechanismen.

## **6. Beschreibung der selbst programmierten Anteile, inkl. deren Funktionalität und Bedeutung für das Gesamtprojekt**

Die selbst programmierten Anteile sind entscheidend für die Verbesserung der Sicherheit und Benutzerfreundlichkeit des Authentifizierungssystems:

* **Adaptive Authentifizierung:** Dieses Modul ermöglicht es dem System, sich an unterschiedliche Nutzungsszenarien anzupassen und erhöht somit die Sicherheit durch kontextbasierte Entscheidungen über den Zugriff auf Ressourcen.
* **Verhaltensanalyse-Modul:** Durch die Analyse des Benutzerverhaltens können potenzielle Bedrohungen frühzeitig erkannt werden. Dies verbessert nicht nur die Sicherheit, sondern ermöglicht auch eine personalisierte Benutzererfahrung.
* **Token-Management-System:** Durch effektives Management von Tokens wird sichergestellt, dass nur autorisierte Benutzer Zugang zu geschützten Ressourcen haben. Dies trägt zur Gesamtintegrität des Systems bei.