## **Entwicklung eines sicheren Authentifizierungssystems mit JWT und dynamischer Zugriffskontrolle**

* Das Ziel dieses Projekts ist die Entwicklung eines sicheren Authentifizierungssystems, das JWTs verwendet, aber zusätzlich eine dynamische Zugriffskontrolle implementiert. Diese Zugriffskontrolle basiert auf Benutzerverhalten und Kontextinformationen, um die Sicherheit der Authentifizierungsprozesse in modernen Webanwendungen zu erhöhen.

## **Selbst programmierte Anteile**

1. **Dynamische Zugriffskontrolle:**
   * Implementierung eines Moduls zur dynamischen Zugriffskontrolle, das basierend auf Benutzerverhalten (z.B. Anmeldezeitpunkt, Standort) entscheidet, ob der Zugriff auf bestimmte Ressourcen gewährt wird.
   * Beispiel: Wenn ein Benutzer sich von einem neuen Standort anmeldet, könnte das System zusätzliche Sicherheitsüberprüfungen (z.B. MFA) anfordern.
2. **Verhaltensbasierte Token-Generierung:**
   * Entwicklung einer Funktion zur Generierung von JWTs, die zusätzliche Claims basierend auf dem Verhalten des Benutzers enthält (z.B. Anzahl der fehlgeschlagenen Anmeldeversuche).
   * Diese Claims könnten verwendet werden, um die Berechtigungen des Benutzers dynamisch anzupassen.
3. **Token-Revocation-System:**
   * Implementierung eines Systems zur Widerrufung von Tokens, das es ermöglicht, bestimmte JWTs vor ihrem Ablaufdatum ungültig zu machen (z.B. bei Verdacht auf Kompromittierung).
   * Dies könnte durch eine zentrale Datenbank geschehen, die eine Liste von widerrufenen Tokens führt.

**"Wie kann eine dynamische Zugriffskontrolle in Kombination mit JSON Web Tokens die Sicherheit von Authentifizierungsprozessen in modernen Webanwendungen verbessern?"**

## **Theoretische Grundlagen**

* JWT-Sicherheit: Analyse der Sicherheitsmechanismen von JWTs und deren Rolle in Authentifizierungsprozessen.
* Dynamische Zugriffskontrolle: Untersuchung von Ansätzen zur dynamischen Anpassung von Berechtigungen basierend auf Benutzerverhalten und Kontextinformationen.
* Best Practices: Diskussion über bewährte Methoden zur Implementierung von JWTs und dynamischer Zugriffskontrolle.

## **Praktische Umsetzung**

## **Implementierungsschritte**

1. Benutzerregistrierung und -anmeldung:
   * Implementierung der grundlegenden Authentifizierungsrouten unter Verwendung von JWTs.
2. Entwicklung der dynamischen Zugriffskontrolle:
   * Programmierung eines Moduls zur Überprüfung des Benutzerverhaltens bei Anfragen und Anpassung der Berechtigungen.
   * Integration zusätzlicher Sicherheitsmaßnahmen wie MFA bei verdächtigen Anmeldungen.
3. Token-Revocation-System:
   * Implementierung einer Datenbanktabelle zur Verwaltung widerrufener Tokens und deren Überprüfung bei jeder Anfrage.

## **Dokumentation**

* Einführung in JWTs und deren Sicherheitsaspekte.
* Beschreibung des entwickelten Systems einschließlich aller selbst programmierten Funktionen.
* Analyse der Ergebnisse aus Tests zur Sicherheit und Benutzerfreundlichkeit.

2. **Entwicklung eines adaptiven Authentifizierungssystems mit JWT und Machine Learning**

* Ziel dieses Projekts ist die Entwicklung eines adaptiven Authentifizierungssystems, das JWTs verwendet und maschinelles Lernen integriert, um die Authentifizierungserfahrung dynamisch anzupassen. Das System analysiert das Benutzerverhalten in Echtzeit und passt die Authentifizierungsanforderungen basierend auf dem Risiko an, das mit der aktuellen Sitzung verbunden ist.

## **Selbst programmierte Anteile**

1. **Verhaltensanalyse-Modul:**
   * Entwicklung eines Moduls, das Benutzerverhalten analysiert, um Muster zu erkennen (z.B. Anmeldezeiten, Standort, Geräte).
   * Implementierung eines Algorithmus zur Klassifizierung von Benutzersitzungen in verschiedene Risikokategorien (niedrig, mittel, hoch).
2. **Adaptive Authentifizierungslogik:**
   * Programmierung einer Logik, die basierend auf der Risikokategorie unterschiedliche Authentifizierungsanforderungen stellt. Zum Beispiel:
     + Bei niedrigem Risiko: Standardanmeldung mit Benutzername und Passwort.
     + Bei mittlerem Risiko: Zusätzliche Sicherheitsfragen oder eine Bestätigung per E-Mail/SMS.
     + Bei hohem Risiko: Multi-Faktor-Authentifizierung (MFA) oder temporäre Sperrung des Zugriffs bis zur Klärung.
3. **Integration eines Machine Learning Modells:**
   * Implementierung eines einfachen maschinellen Lernmodells (z.B. Decision Tree oder Random Forest), das auf historischen Benutzerdaten trainiert wird, um das Risiko zukünftiger Anmeldungen vorherzusagen.
   * Nutzung von Bibliotheken wie scikit-learn für die Implementierung des Modells.

**"Wie kann ein adaptives Authentifizierungssystem unter Verwendung von JSON Web Tokens und maschinellem Lernen die Sicherheit und Benutzerfreundlichkeit in modernen Webanwendungen verbessern?"**

## **Theoretische Grundlagen**

* **JWT-Sicherheit:** Analyse der Rolle von JWTs in Authentifizierungsprozessen.
* **Maschinelles Lernen in der Cybersicherheit:** Untersuchung, wie ML-Algorithmen zur Verbesserung der Sicherheitsmaßnahmen eingesetzt werden können.
* **Adaptive Authentifizierung:** Diskussion über die Vorteile und Herausforderungen adaptiver Authentifizierungssysteme.

## 

## 

## **Praktische Umsetzung**

1. **Benutzerregistrierung und -anmeldung:**
   * Implementierung der grundlegenden Authentifizierungsrouten unter Verwendung von JWTs.
2. **Entwicklung des Verhaltensanalyse-Moduls:**
   * Programmierung des Moduls zur Analyse des Benutzerverhaltens und Klassifizierung in Risikokategorien.
3. **Implementierung der adaptiven Authentifizierungslogik:**
   * Programmierung der Logik zur Anpassung der Authentifizierungsanforderungen basierend auf dem ermittelten Risiko.
4. **Integration des Machine Learning Modells:**
   * Training des ML-Modells auf historischen Benutzerdaten zur Vorhersage des Anmelderisikos.

## **Dokumentation**

* Einführung in adaptive Authentifizierungssysteme und deren Vorteile.
* Beschreibung des entwickelten Systems einschließlich aller selbst programmierten Funktionen.
* Analyse der Ergebnisse aus Tests zur Sicherheit und Benutzerfreundlichkeit.