**Design Document for Zero Trust Learning Management System (LMS)**

**1. System Overview**

Our LMS will provide a secure, minimalistic interface for disseminating educational materials with strict access controls and automated notification functionalities. The system will prioritize security, aligning with Zero Trust principles, to ensure that access to resources is continuously verified and strictly controlled based on dynamic policies.

**2. Key Features and Components**

**a. Minimalistic Interface**

* **Objective:** Ensure ease of use with a clean, intuitive user interface focusing on essential functionalities.
* **Implementation:** Use a responsive design framework to create a user-friendly interface with clear navigation paths for accessing and managing educational materials.

**b. Dynamic Access Control**

* **Objective:** Implement strict access control that adapts to user context and behavior.
* **Implementation:**
  + Integrate an ABAC (Attribute-Based Access Control) system to define access controls based on attributes such as user role, location, device security posture, and time of access.
  + Set access limits on resources (e.g., viewable only three times per user), with the system dynamically adjusting access based on real-time evaluations of user behavior and environmental factors.

**c. Multi-Factor Authentication (MFA)**

* **Objective:** Strengthen user authentication to ensure that access is secure and verified.
* **Implementation:**
  + Require MFA for all user logins and sensitive actions within the LMS. This may include OTPs (One-Time Passwords) sent via SMS or email, or app-based authentication methods.
  + Develop a policy for enforcing MFA based on user role, access location, and the sensitivity of accessed resources.

**d. Automated Notification System**

* **Objective:** Keep users informed about their access status and system changes in real-time.
* **Implementation:**
  + Develop a system to send automated notifications via email or SMS, alerting users about the remaining access opportunities for restricted resources.
  + Implement real-time alerts for administrators when unusual access patterns are detected, facilitating quick response to potential security incidents.

**3. Security and Compliance**

* **Data Encryption:** Use end-to-end encryption to protect data in transit and at rest. Implement HTTPS for all communications within the LMS.

**Overview of Dynamic Access Control**

**Objective:**  
To provide secure, context-sensitive access to educational resources, adapting permissions in real time based on predefined security policies and user attributes.

**Key Components of Implementation**

1. **Attribute Definition:**
   * **User Attributes:** Include user role (e.g., student, instructor, admin), department, course enrollment, and history of access.
   * **Resource Attributes:** Type of educational material (e.g., video, document), sensitivity level, and intended audience.
   * **Environmental Attributes:** User's location, time of access, device type, and security posture of the device.
   * **Behavioral Attributes:** Patterns of access such as frequency, regularity, and deviations from normal behavior.
2. **Policy Engine:**
   * **Role:** Acts as the decision-making core of the DAC system, evaluating access requests based on policies that incorporate the defined attributes.
   * **Policy Definition:** Develop comprehensive policies that specify how attributes affect access. For example, a policy might allow instructors unlimited access to course materials during the semester, but restrict access to three times for students, and only from secure campus networks during daytime hours.
3. **Access Enforcement Mechanism:**
   * **Role:** Enforces the decisions made by the policy engine.
   * **Implementation:** Typically integrated with the system’s authentication and authorization processes. This mechanism checks each access request against the policy engine's decisions before granting or denying access.
4. **Context-Aware Security:**
   * **Real-Time Evaluation:** The system evaluates each access request in real-time, considering any changes in user attributes or environmental conditions.
   * **Adaptability:** The system can modify access rights dynamically as new information becomes available or as user behaviors change.

**Steps to Implement Dynamic Access Control**

**Step 1: Define Attributes and Policies**

* Gather comprehensive details about what attributes are relevant for each user and resource. This involves collaboration with various stakeholders within the institution to understand different needs and security requirements.
* Define clear, actionable policies that specify how these attributes interact to determine access rights. For example, a policy might state that only students enrolled in a course can access its materials and only during the semester.

**Step 2: Build or Integrate a Policy Engine**

* Develop or integrate an existing ABAC engine capable of processing complex policies involving multiple attributes.
* The engine should evaluate access requests against the defined policies to make real-time decisions about whether to allow or deny access.

**Step 3: Implement Access Enforcement**

* Ensure that every access request passes through the policy engine for evaluation.
* Integrate the access enforcement mechanism with the system’s login and resource request workflows to ensure that no access is possible without proper authorization according to the current policies and attributes.

**Step 4: Monitor and Adapt Policies**

* Implement logging and monitoring to track access patterns and detect anomalies.
* Regularly review and update policies based on feedback from monitoring systems and changes in organizational needs or security landscapes.

**User Attributes & Roles**

**User Attributes**

1. **Role:** Identifies whether the user is a student, instructor, or administrative staff. This attribute determines the level of access to different examination functionalities (e.g., students can take exams, instructors can create or grade them).
2. **Course Enrollment:** Indicates whether a student or instructor is associated with a specific course that includes the exam. Access to an exam is granted only if the user is enrolled in or teaching the course.
3. **Authentication Status:** Reflects whether the user has successfully completed the multi-factor authentication process. Access to exams requires successful authentication to ensure the user’s identity.

**Resource Attributes**

1. **Exam ID:** A unique identifier for each exam, used to link access requests to specific examination content.
2. **Examination Availability:** Indicates whether the exam is currently active and within the scheduled availability window set by the instructor.
3. **Exam Sensitivity Level:** Classifies the security level of the exam based on its content and importance, influencing the strictness of access controls and monitoring.

**Environmental Attributes**

1. **Access Location:** Identifies the physical or network location from which the access request is made. Access might be restricted to certain locations, such as on-campus computer labs or secure networks, to prevent unauthorized access from external or less secure environments.
2. **Time of Access:** Ensures that access requests align with the scheduled exam times. Attempts to access the exam outside of these times are denied, supporting the integrity and scheduling of the examination process.
3. **Device Security Posture:** Assesses whether the device used to access the exam meets the university’s security standards, such as up-to-date antivirus software, operating system patches, and prohibited software (e.g., screen capture tools).

**Behavioural Attributes**

1. **Access Frequency:** Monitors how often a user attempts to access the exam. Repeated access attempts could indicate suspicious behaviour, triggering additional security checks or alerts.
2. **Historical Access Patterns:** Analyses the user’s typical access patterns and flags deviations that might suggest impersonation or other fraudulent activities.

**How These Attributes Work Together**

* **Access Request Processing:** When a user attempts to access an exam, the system evaluates the request based on all relevant attributes. The ABAC system checks each attribute against predefined policies to determine whether the access should be granted.
* **Dynamic Policy Application:** The policies consider multiple attributes simultaneously. For example, a policy could allow a student to access the exam if the user’s role is 'student', they are enrolled in the course, the exam is currently active, the access is from an approved location, and the access time matches the scheduled window. Additionally, the system would verify that the device security posture is adequate.
* **Real-Time Decision Making:** The system makes access decisions in real-time, allowing immediate entry if all conditions are met, or denying access if any criteria are not fulfilled. This process includes logging all access attempts for security monitoring and auditing.
* **Continuous Evaluation:** The examination system continuously evaluates access conditions during the exam session. If any attribute changes unfavorably (e.g., the device becomes compromised), the system can dynamically revoke access to protect the examination’s integrity.

**Policy 1: Role and Course Enrollment Verification**

**Objective:** Ensure that only students and instructors who are associated with a specific course can access its exams.

* **Condition:** User's role must be either 'Student' or 'Instructor.'
* **Condition:** User must be enrolled in or teaching the course associated with the exam.
* **Action:** Grant access if both conditions are satisfied; otherwise, deny.

**Policy 2: Multi-Factor Authentication Requirement**

**Objective:** Strengthen security by requiring strong authentication before exam access.

* **Condition:** User must have completed multi-factor authentication successfully.
* **Action:** Grant access if the condition is met; otherwise, deny.

**Policy 3: Examination Availability Window**

**Objective:** Restrict exam access to the scheduled time frames set by the course instructor.

* **Condition:** The current system time must fall within the examination's scheduled start and end times.
* **Action:** Grant access if the condition is met; otherwise, deny.

**Policy 4: Secure Location Access**

**Objective:** Allow exam access only from secure, approved locations.

* **Condition:** The access location must be either an on-campus location designated for exams or a recognized secure network (e.g., secured VPN).
* **Action:** Grant access if the condition is met; otherwise, deny.

**Policy 5: Device Security Compliance**

**Objective:** Ensure that the device used for accessing the exam complies with university security standards.

* **Condition:** The device must pass security checks including up-to-date antivirus, absence of prohibited software, and latest security patches.
* **Action:** Grant access if the condition is met; otherwise, deny.

**Policy 6: Exam Sensitivity Level and Access Frequency**

**Objective:** Implement stricter controls for exams with higher sensitivity levels.

* **Condition:** If the exam's sensitivity level is high, the system will monitor access frequency more stringently.
* **Sub-condition:** Users attempting access more frequently than normal patterns suggest may trigger additional verification or temporary access suspension.
* **Action:** Monitor and adjust access permissions based on frequency of access and exam sensitivity level.

**Policy 7: Behavioral Anomaly Detection**

**Objective:** Detect and respond to unusual access patterns that may indicate potential security threats.

* **Condition:** Access patterns deviating significantly from the user’s historical norms will be flagged.
* **Action:** Trigger a security alert and require additional authentication; potentially suspend access until the anomaly is investigated.

**Implementation Tips for Policies**

* **Dynamic Policy Enforcement:** Use a policy decision point (PDP) integrated with the LMS that evaluates each access request against these policies in real-time.
* **Real-Time Monitoring:** Implement a monitoring system that continuously evaluates access conditions and can dynamically alter access rights based on real-time data.
* **Logging and Auditing:** Ensure all access attempts, successful or denied, and security incidents are logged for auditing purposes and future analysis.