**ABAC Roles and Rules: Zero Trust Access Control System**

**Project Overview**

This document provides the Attribute-Based Access Control (ABAC) roles, rules, and continuous verification strategies for a web application at a university. This system leverages Zero Trust security principles and integrates Distributed Federated Learning (FL) and machine learning to analyse security and access patterns across multiple campuses.

**User Roles and Attributes**

**1. Student**

* Attributes: Student ID, Department, Course Enrolments, Status (Active, Inactive), Device Type (Mobile, Laptop), Location (Specific Campus), Time of Access.

**2. Faculty**

* Attributes: Faculty ID, Department, Courses Teaching, Research Areas, Position (Professor, Assistant Professor, etc.), Status, Device Type, Location (Specific Campus), Time of Access.

**3. Staff**

* Attributes: Staff ID, Department, Position (Administrative, Technical Support, etc.), Special Permissions (Financial, HR), Status, Device Type, Location (Specific Campus), Time of Access.

**4. Visitor**

* Attributes: Visitor ID, Purpose (Tour, Meeting, Conference), Sponsor (Department), Duration of Visit, Device Type, Location (Specific Campus), Time of Access.

**5. IT Administrator**

* Attributes: Admin ID, Superuser Status, Departments Overseen, Device Type, Location (Specific Campus), Time of Access.

**University Campuses for Decentralized Federated Learning**

To support decentralized federated learning, the university maintains multiple campuses, each with distinct data storage and processing capabilities:

1. **Main Campus**
2. **Downtown Campus**
3. **Tech Park**
4. **Medical Campus**
5. **Research Institute**
6. **International Office**
7. **Remote Learning Center**
8. **Innovation Hub**
9. **Graduate School**
10. **Satellite Campus**

**Detailed ABAC Rules**

**1. Resource-Specific Access**

* Rule: Access to resources based on departmental alignment or specific permissions.
* Condition: **(User.Department == Resource.Department OR User.SpecialPermissions.includes(Resource.Type)) AND User.Status == "Active"**

**2. Sensitive Resource Access**

* Rule: Restricted access to sensitive data.
* Condition: **(User.Position in ["Administrative", "Professor"] AND User.SpecialPermissions.includes("Financial")) OR (Resource.Sensitivity == "High" AND User.AccessLevel == "High")**

**3. Educational Resource Access**

* Rule: Access based on roles and active status.
* Condition: **(User.Role in ["Student", "Faculty"] AND User.Status == "Active") AND (System.Time >= "08:00" AND System.Time <= "22:00")**

**4. Remote Access**

* Rule: Conditional off-campus access.
* Condition: **User.Location == "Off-Campus" AND (Resource.Sensitivity != "High" OR User.AuthenticatedWithMFA)**

**5. Visitor Access**

* Rule: Limited access based on visit purpose.
* Condition: **User.Role == "Visitor" AND (Resource.Type == "General" OR Resource.Purpose == User.VisitPurpose) AND System.Time <= User.DurationOfVisit**

**6. Emergency Access**

* Rule: Override controls for emergency services.
* Condition: **System.EmergencyStatus AND User.Role in ["IT Administrator", "Staff"]**

**Continuous Verification Strategies**

**1. Re-authentication and Re-authorization**

**Re-authentication** ensures that the identity of the user remains valid over time. Depending on the sensitivity of the application, you might require users to re-authenticate at regular intervals or in response to specific events.

* **Timed Re-authentication:** Users are prompted to re-authenticate after a predetermined period, such as every hour or at the start of a new session.
* **Event-driven Re-authentication:** Trigger re-authentication following specific events, such as accessing high-value resources, changes in user role, unusual activity, or from a new device/location.

**Re-authorization** checks whether the user's permissions are still valid under current conditions. This can be managed by:

* **Policy Engine Calls:** Each time a resource is requested, the policy engine re-evaluates the current context against the set policies to determine if access should still be granted.
* **Attribute Validation:** Regularly check that the user's attributes (role, department, location, etc.) haven't changed in a way that would affect their permissions.

**2. Session Management**

Manage user sessions with strict controls and monitoring:

* **Session Timeouts:** Automatically log out users after a period of inactivity.
* **Session Risk Assessment:** Continuously evaluate the risk level of a session based on factors such as user behavior, access patterns, and external threat intelligence.

**3. Contextual and Behavioural Monitoring**

Implement systems to monitor and react to changes in user behaviour or context:

* **Behavioural Analytics:** Use machine learning algorithms to establish a baseline of normal user behaviour and flag anomalies that might indicate a security risk.
* **Contextual Awareness:** Adjust access controls based on the context of the access request, such as time of day, geographic location, or device security posture.

**4. Multi-factor Authentication (MFA)**

Enhance security by requiring additional verification methods:

* **Adaptive MFA:** Deploy MFA that adapts based on the assessed risk of the access request. For routine requests from a known device and location, MFA might not be required, for high-risk requests, prompt for additional authentication factors.

**Conclusion**

The implementation of ABAC roles and rules across multiple campuses supports decentralized federated learning and ensures that the university’s Zero Trust access control system is robust, secure, and capable of accommodating diverse educational and research activities.