Information Security (CS-3002)

Assignment # 4

Privacy, Trust & the CIA Triad in Modern Information Systems

Pre-Assignment Learning Material

Before beginning this assignment, all students must review the following learning resources:

- **Video Lecture:**
  Privacy Past and Present: Look at Data Privacy (https://youtu.be/PkkuS3RJRnI?si=1XiHxecvaOjw7MHp)

- **Supporting Reading:**
  RSA Conference Presentation (2024)
  "Privacy Past and Present: A Father-Daughter Dive into Data Privacy Evolution"
  (See attached slides on GCR)

These materials explore how privacy norms evolved from the early *"Right to Be Let Alone"* to modern digital privacy laws such as GDPR and CCPA, emphasizing trust, transparency and control over personal data.

GDPR-Compliant Mini Hospital Management System Implementing the CIA Triad

Inspired by the lecture's focus on data protection, privacy laws, and the evolution of privacy expectations, this project allows students to apply these concepts practically in a small-scale system that upholds the principles of lawful, fair, and transparent data processing

Scenario

A community hospital is transitioning from paper-based records to a digital management system. The hospital must ensure that personal data is processed in *compliance with GDPR* ensuring *privacy*, minimizing *data exposure* and keeping *audit trails* of *who* accesses *what* data and *why*. The Hospital administration require a privacy-centric dashboard that ensures:

- Confidentiality: Patient identities and medical data are hidden or encrypted.
- Integrity: Only authorized roles can modify or audit data; any change must be logged.
- Availability: The system remains functional and data retrievable to authorized users.

Your IS team (2 members) has been tasked to develop and demonstrate this system using Streamlit, Python, and a simple database (SQLite/MySQL).

Project Description

Develop a Streamlit-based Hospital Management Dashboard that:

1. Connects to a secure database (SQLite or MySQL).
2. Uses role-based access control (RBAC) to regulate permissions.
3. Applies data anonymization/masking to protect patient information.
4. Maintains secure logs to ensure integrity and auditability.
5. Ensures system availability through stable data retrieval and exception handling.

Core Functional Requirements

1. Confidentiality (Data Protection: Privacy)

i. Replace or encrypt personal data using hashlib or Fernet.

ii. Implement data masking for sensitive identifiers (names, contacts, diagnoses). For example:

- Name → ANON_1021
- Contact → XXX-XXX-4592

iii. Restrict access based on user roles:

a. **Admin:** Full access to raw & anonymized data.

b. **Doctor:** Access to anonymized data only.

c. **Receptionist:** Add/edit records but cannot view sensitive data.

iv. Include a login page for user authentication.

2. Integrity (Data Accuracy and Accountability)

i. Maintain activity logs to record all user actions:

- Log user role, timestamp, and action type (login, anonymization, update, view).

ii. Use database constraints or code validation to prevent unauthorized changes.

iii. Display an **"**Integrity Audit Log**"** (Admin only).

3. Availability (System Access & Reliability)

i. Ensure the dashboard and database remain responsive.
ii. Implement error handling (try/except) for failed logins or DB errors.

iii. Include a data backup/export option (CSV download) for recovery.

iv. Display system uptime or last synchronization time in the dashboard footer.

Suggested Database Schema

Table: users

user_id	**username**	**password**	**role**
1	admin	admin123	admin
2	Dr. Bob	doc123	doctor
3	Alice_recep	rec123	receptionist

Table:patients
| patient_id | name | contact | diagnosis | anonymized_name | anonymized_contact |date_added |

Table: logs
| log_id | user_id | role | action | timestamp | details |

Example Workflow

1. User logs in → Authentication verifies credentials and assigns role.
2. Role defines permitted views/actions (RBAC).
3. Admin triggers "Anonymize Data" → sensitive fields are masked or encrypted.
4. Doctor views anonymized patient data.

5. Receptionist adds/edit records but cannot view masked data.
6. All actions are timestamped and stored in logs.
7. Admin can review audit logs and export them securely.

Deliverables

Each group (2 students) must submit:

1. **Source Code Folder** (.py files + database file)
2. **PDF Report** (3 to 5 pages) including:
   i. System overview diagram (showing CIA layers)
   ii. Screenshots of login, anonymization, and log screens
   iii. Discussion on CIA implementation & GDPR alignment
3. **Short Demo Video (Optional, 2–3 mins)** – Upload video on drive and Copy-Paste the drive link in your PDF report.
4. **Assignment4.ipynb (or .py)** file with proper steps/comments.

Bonus (Optional +2 Weightage)

- Add Fernet encryption for reversible anonymization.
- Display real-time activity graphs (e.g., user actions per day).
- Implement GDPR features like:
  - Data retention timer
  - User consent banner

Evaluation Rubric (Total: 100 Marks)

Component	**Marks**
Privacy & GDPR Compliance	20
Confidentiality Implementation	20
Integrity (Logging & Validation)	20
Availability & Reliability	15
Dashboard Functionality & Design	10
Documentation & Screenshots	10
Presentation/Demo/Video	5