PA2562: Secure Software Engineering

Threat Modeling Assignment

Name	Social Security Number	E-Mail
Denim Deshmukh	980401-6450	dede19@student.bth.se
Sai Pranav Koyyada	990429-9311	saky19@student.bth.se

EVV System

The application is meant to serve as an identifier for the Exam leader, with regards to theExam taker. We log into the application as an Exam taker, wanting to take the exam, provide our picture in exchange for a QR code. When we take the exam, we present our code andthe Exam leader may scan it to retrieve the picture, verifying our intention to take the exam.

1. Decomposition of the Application Ecosystem

    - **External Entity**
      Student, Examiner using Mobile Application

    - **Process**
      Server 1 - REST API
      Server 2 - Backend Service hosting Credentials Data Source in Postgresql DB connected to Server 1.
      Server 3 - Backend Service for storage and retrieval of pictures connected to Server 1.

- **Data Flow**

  From Row Entities to Column Entities

Entity	Student Application	Examiner Application	Server 1	Server 2	Server 3	Local Data Store	Credential Data Store	Picture Store
Student Application			Request			Data		
Examiner Application			Request					
Server 1	Response	Response		Data	Data			
Server 2			Data				SQL Query	
Server 3			Data					Data
Local Data Store	Data							
Credential Data Source				Data				
Picture Store					Data			

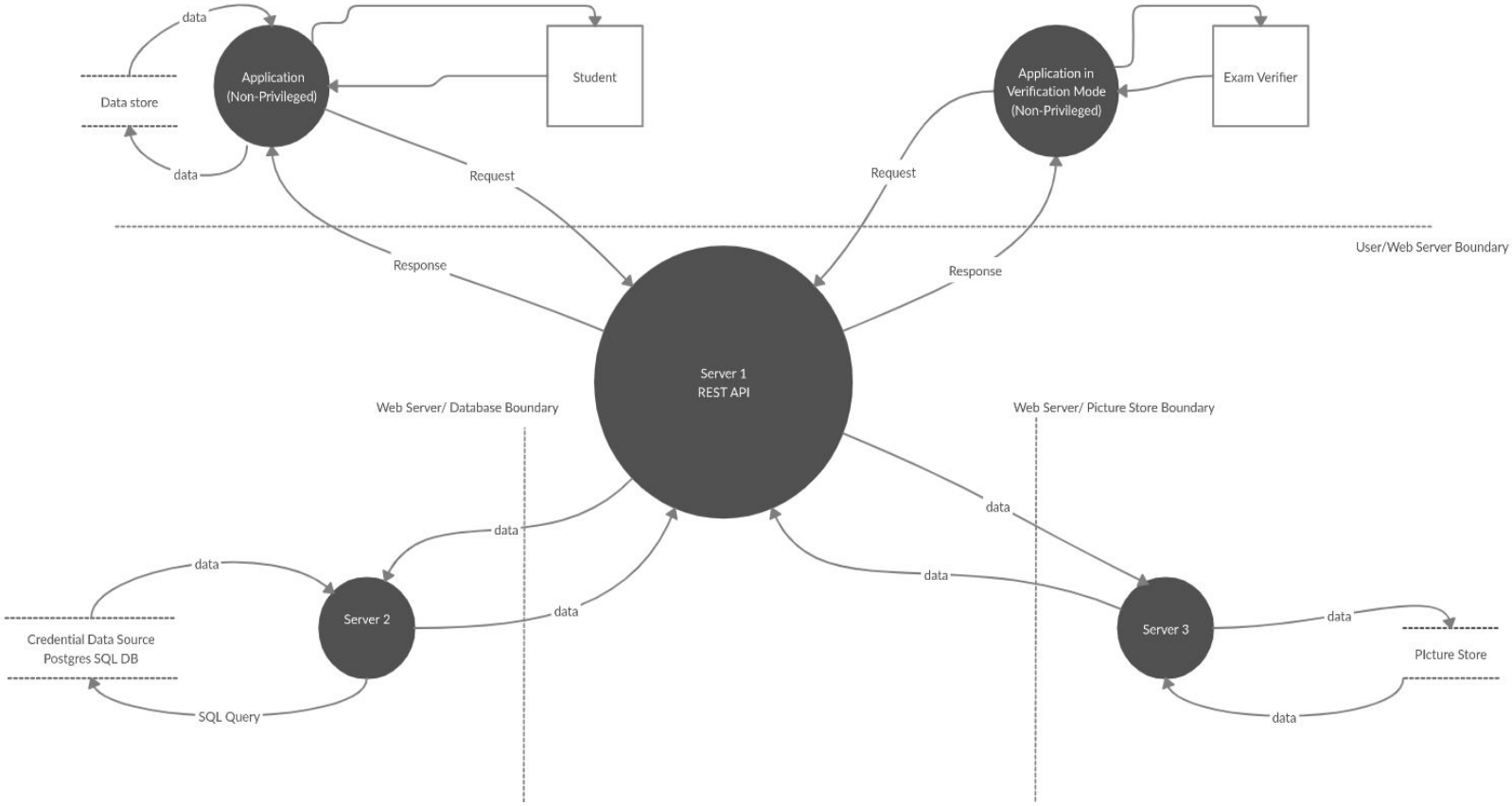
- **Data Store**

  Local Data Store (Student Mobile Application), Credentials Data Store Postgresql db, Picture Store

- **Trust Boundaries**

  User/Web Server Boundary, Web Server/ Credential Data Store Boundary , Web Server/ Picture Store Boundary

2. Data Flow Diagram



Entities

1. Student
2. Exam Verifier

Process

3. Application
4. Application in Verification Mode
5. Server 1- REST API
6. Server 2 - Backend Service for Credential Hosting
7. Server 3 - Backend Service for Picture Store

Data Stores

8. Local Data Store on Student Application
9. Credential Data Store
10. Picture Store

Data Flow

11. Student Application → Server 1 : Request
12. Student Application → Local Data Store : Data
13. Examiner Application → Server 1 : Request
14. Server 1 → Student Application : Response
15. Server 1 → Examiner Application : Response
16. Server 1 → Server 2 : Data
17. Server 1 → Server 3 : Data
18. Server 2 → Server 1 : Data
19. Server 2 → Credential Data Source : SQL Query
20. Server 3 → Server 1 : Data
21. Server 3 → Picture Store : Data
22. Local Data Store → Student Application : Data
23. Credential Data Source → Server 2 : Data
24. Picture Store → Server 3 : Data

3. STRIDE

Entity	S	T	R	I	D	E
1	✓		✓			
2	✓		✓			
3	✓	✓	✓			✓
4	✓	✓	✓	✓		✓
5	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓
7	✓	✓	✓	✓	✓	✓
8		✓	✓	✓	✓	
9		✓		✓	✓	

10		✓		✓	✓	
11		✓		✓	✓	
12		✓		✓		
13		✓		✓	✓	
14		✓		✓		
15		✓		✓		
16		✓			✓	
17		✓			✓	
18		✓		✓	✓	
19		✓		✓	✓	
20		✓		✓	✓	
21		✓		✓	✓	
22		✓		✓		
23		✓		✓	✓	
24		✓		✓	✓	

4. Recommendations of security mitigations to eliminate or minimize the threats.

Spoofing
Proper authentication must be implemented to avoid impersonations.

Tampering
The integrity must be preserved using:

- User input validation and output encoding.
- Encryption of data and resources over network.
- Identify and resolve 3rd party dependencies vulnerabilities with composition analysis tools
- Identify security bugs
- Parse prepared SQL statements to nullify SQL injections.

Repudiation

Proper auditing and logging must be practiced to track the activity of entities over the system.

Information Disclosure

Confidentiality of the system must be ensured with:

- Standard Encryption
- Binding certificates issued by trusted Certificate Authority(CA)
- Include trusted dependencies to the system if required.

Denial of Service

Availability of the system can be assured by

- Mitigating resource consumption
- Proper traffic monitoring and log rotation
- Notifying disk overflows.

Elevation of Privileges

Authorization must be segmented with standard mechanism and privileges must be distributed as required by the role of the user. It is always advisable to follow the least privilege principle. Application dependencies and 3rd party libraries must be examined.