Conduct an Application Security Review

Security Assessment Template

Executive Summary

The Customer Information Management System (CIMS) is designed to facilitate better communication and support for customers by allowing them to update their information via a portal, enabling Customer Relationship Managers to manage customer data through an internal web interface, and providing data scientists with access to query the database for analysis.

Several risks were identified in the initial design, including the use of a default PostgreSQL configuration without encryption or audit logging, unsecured communication channels, potential SQL injection vulnerabilities, and inadequate firewall configurations. To mitigate these risks, it is recommended to configure the database with SSL, enable audit logging, and encrypt sensitive data. Additionally, secure authentication methods should replace hardcoded credentials, parameterized queries should be used to prevent SQL injection and firewall rules should be tightened. Ensuring compliance with SOX and PCI DSS, such as by implementing strong access controls and regular security assessments, will further enhance the system's security.

Risk 1: *Default Database Configuration*

Risk Rating: *High*

Related Security Frameworks (if any): *PCI DSS, SOX*

Explanation of Risk
The PostgreSQL database, which contains sensitive customer information including credit card details, has been left in its default configuration without encryption, lacking critical security features such as SSL encryption, audit logging, and encryption of sensitive data. This means that data stored in the database is vulnerable to unauthorized access, data breaches, and non-compliance with PCI DSS and SOX requirements. If an attacker gains access to the database, they could easily extract and misuse sensitive customer data. Given the critical nature of the data and compliance requirements with PCI DSS and SOX, the absence of encryption poses a significant risk to data security and regulatory compliance.

Recommendations	How does the recommendation mitigate the risk?
Enable SSL Encryption: *Configure the database to enforce SSL connections to ensure that all data transmitted between the application and the database is encrypted.*	***Mitigation***: *This protects data in transit from being intercepted or tampered with by malicious actors.*
Implement Audit Logging: *Enable comprehensive audit logging to track all access and modifications to the database.*	***Mitigation***: *Audit logs provide a record of activities, which is essential for detecting unauthorized access and maintaining data integrity, thereby supporting SOX compliance.*
Encrypt Sensitive Data at Rest: *Use strong encryption algorithms to encrypt sensitive data stored in the database.*	***Mitigation:*** *This ensures that even if the database is compromised, the encrypted data remains protected, aligning with PCI DSS requirements.*

Risk 2: *Unsecured Communication Channels*

Risk Rating: *Critical*

Related Security Frameworks (if any): *PCI DSS, SOX*

Explanation of Risk
The communication between the AWS portal server and the on-premises database involves POST requests with hardcoded usernames and passwords, which can be intercepted and exploited. This creates a significant vulnerability that could lead to unauthorized access to the database and exposure of sensitive information.

Recommendations	How does the recommendation mitigate the risk?
Implement Secure Authentication Methods: *Replace hardcoded credentials with secure authentication methods such as OAuth or JWT.*	***Mitigation***: *These methods provide more secure ways of authenticating users, reducing the risk of credential theft and unauthorized access.*
Enforce TLS for Data in Transit: *Ensure that all communications between components are encrypted using TLS.*	***Mitigation***: *TLS encryption protects data in transit from being intercepted or altered, safeguarding sensitive information and maintaining compliance with PCI DSS and SOX.*

Risk 3: *Direct SQL Queries from Web Interfaces*

Risk Rating: *High*

Related Security Frameworks (if any): PCI DSS

Explanation of Risk
Direct SQL queries from web interfaces increase the risk of SQL injection attacks, which can be exploited to manipulate the database, extract sensitive data, or cause service disruptions. This is a critical security concern as it can lead to data breaches and undermine the integrity of the database.

Recommendations	How does the recommendation mitigate the risk?
Use Parameterized Queries: *Ensure all SQL queries are parameterized or use prepared statements.*	***Mitigation***: *Parameterized queries prevent SQL injection attacks by separating SQL code from data inputs, thereby protecting the database from malicious inputs.*
Implement Input Validation and Sanitization: *Validate and sanitize all user inputs to ensure they meet expected formats before processing.*	***Mitigation***: *This reduces the risk of injecting harmful code into the application, thereby protecting the database and maintaining data integrity.*

Risk 4: *Firewall Configuration*

Risk Rating: *Medium*

Related Security Frameworks (if any): PCI DSS

Explanation of Risk
A firewall exception for port 5432 exposes the database to potential unauthorized access. If the firewall rules are too permissive, malicious actors can exploit this open port to gain entry to the database, leading to data breaches and service disruptions.

Recommendations	How does the recommendation mitigate the risk?
***Restrict Access with IP Whitelisting**: Configure the firewall to only allow connections from trusted IP addresses.*	***Mitigation**: This limits access to the database, ensuring only authorized devices can connect, thereby reducing the attack surface.*
***Use VPNs for Secure Connections**: Require the use of VPNs for any external access to the database.*	***Mitigation**: VPNs provide a secure tunnel for data transmission, ensuring that communications are protected from eavesdropping and tampering, which enhances compliance with security frameworks like PCI DSS.*

Risk 5: *Data Scientist Access*

Risk Rating: *Medium*

Related Security Frameworks (if any): PCI DSS, SOX

Explanation of Risk
Data scientists have read-only access to the database from their local machines, which could lead to data exposure if their machines are compromised. This poses a risk of unauthorized access to sensitive data, potentially leading to data breaches and non-compliance with PCI DSS and SOX.

Recommendations	How does the recommendation mitigate the risk?
Ensure Endpoint Security: *Implement and enforce strict security policies on data scientists' machines, including antivirus software, firewalls, and regular security updates.*	***Mitigation***: *Securing the endpoints reduces the risk of data breaches from compromised devices, protecting sensitive information.*
Use VPNs for Access: *Require data scientists to use VPNs when accessing the database.*	***Mitigation***: *VPNs ensure that data transmitted between the data scientists' machines and the database is encrypted, protecting against interception and unauthorized access.*

Cost Analysis for Recommended Solutions

1. Enable SSL Encryption for the Database

- ***Time**:*
  - *Initial setup: 1-2 days.*
  - *Ongoing maintenance: Minimal, with periodic updates and renewals.*
- ***Money**:*
  - *Cost of SSL certificates: $100-$500 per year.*
  - *Possible need for upgraded infrastructure to handle encryption overhead.*
- ***Complexity**:*
  - *Medium. Requires changes to database configuration and testing to ensure all connections use SSL.*

2. Implement Audit Logging for the Database

- ***Time**:*
  - *Initial setup: 2-3 days.*
  - *Ongoing maintenance: Moderate, including periodic log reviews and storage management.*
- ***Money**:*
  - *Increased storage costs for maintaining logs: Variable, potentially $50-$200 per month depending on log volume.*
  - *Possible additional costs for log management tools.*
- ***Complexity**:*
  - *Medium. Requires configuration of logging parameters and ensuring logs are securely stored and accessible for audits.*

3. Encrypt Data at Rest in the Database

- ***Time**:*
  - *Initial setup: 3-5 days.*
  - *Ongoing maintenance: Minimal, mostly ensuring continued encryption with database updates.*
- ***Money**:*
  - *Potential cost of encryption tools or features if not natively supported.*
  - *Increased computational overhead may require upgraded hardware: $500-$2000 one-time cost.*
- ***Complexity**:*
  - *High. Requires careful planning to avoid data loss, potential downtime during implementation, and verification that all data is properly encrypted.*

4. Implement Secure Authentication Methods (OAuth/JWT)

- ***Time**:*
  - *Initial setup: 1-2 weeks.*
  - *Ongoing maintenance: Moderate, including regular token management and security updates.*
- ***Money**:*
  - *Development costs: $5000-$20000 depending on scope and complexity.*
  - *Potential licensing costs for authentication libraries or services.*
- ***Complexity**:*

    - *High. Requires integration with existing authentication mechanisms, user management updates, and extensive testing to ensure security and functionality.*

5. Enforce VPN Access for Secure Communications

- ***Time**:*
    - *Initial setup: 1-2 weeks.*
    - *Ongoing maintenance: Moderate, including regular updates and monitoring.*
- ***Money**:*
    - *VPN service costs: $100-$500 per month.*
    - *Potential costs for additional hardware (VPN routers): $500-$2000 one-time cost.*
- ***Complexity**:*
    - *Medium. Requires network configuration, user setup, and ensuring all necessary communications routes through the VPN.*

Final Architecture Recommendation

Provide a final architecture recommendation that:

- Explain how the architecture changes in your final architecture remediate each of the identified risks
- Discusses how business requirements and customer needs are met by the proposed architecture
- Discuss any costs associated with the changes.

Proposed Architecture Changes:

1. ***Database Configuration and Security:***
    - ***Implement SSL Encryption****: Configure the PostgreSQL database to enforce SSL connections.*
    - ***Enable Audit Logging****: Enable comprehensive audit logging for all access and modifications.*
    - ***Encrypt Data at Rest****: Use strong encryption algorithms for sensitive data stored in the database.*
    - ***Remediation****: These changes will address the default configuration risk by ensuring data in transit and at rest is protected, audit trails are maintained for compliance, and sensitive information is encrypted, mitigating the risk of unauthorized access and data breaches.*

2. ***Secure Communication Channels:***
    - ***Secure Authentication Methods****: Replace hardcoded credentials with OAuth or JWT for secure authentication.*
    - ***TLS Enforcement****: Enforce TLS for all communications between application components.*
    - ***Remediation****: These measures will secure the communication channels, preventing credential interception and ensuring data integrity during transmission, thus mitigating risks related to unsecured communication.*

3. ***Web Interface Security:***
    - ***Parameterized Queries****: Use parameterized queries or prepared statements for all SQL operations.*
    - ***Input Validation and Sanitization****: Implement robust input validation and sanitization processes.*
    - ***Remediation****: These practices will prevent SQL injection attacks and ensure data inputs are safe, addressing the risks of direct SQL queries from web interfaces.*

4. ***Network and Firewall Configuration:***
    - ***IP Whitelisting****: Restrict database access to trusted IP addresses using firewall rules.*
    - ***VPN Access****: Require VPN for external database access to ensure secure connections.*
    - ***Remediation****: These changes will limit database access to authorized users and secure the connection, reducing the risk of unauthorized access through firewall vulnerabilities.*

5. ***Data Scientist Access Controls:***
    - ***Endpoint Security****: Enforce strict security policies on data scientists' devices.*
    - ***VPN for Database Access****: Require VPN usage for accessing the database.*

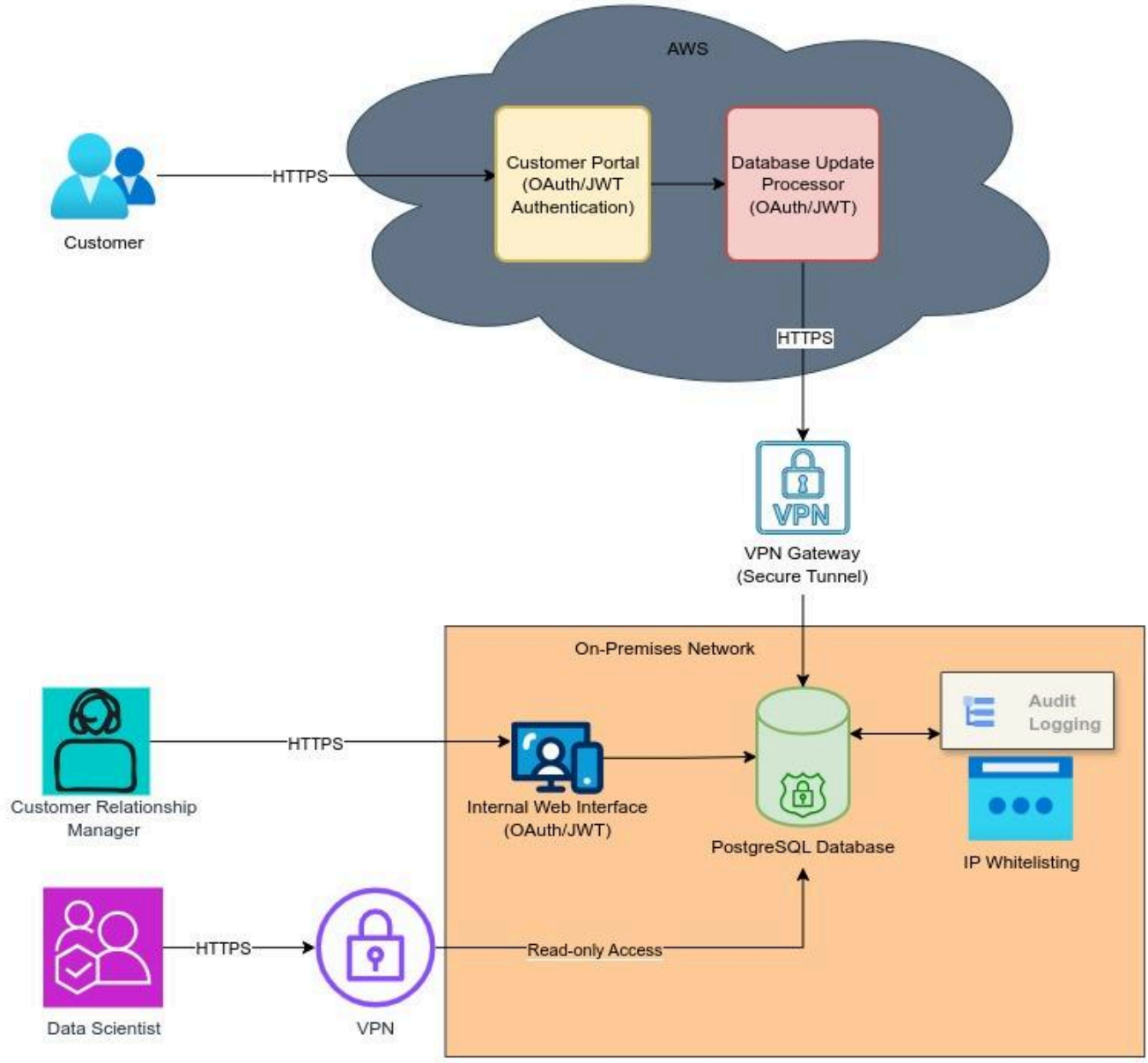
- ***Remediation**: These measures will protect sensitive data from being compromised through data scientists' devices, ensuring secure data access and compliance with PCI DSS and SOX.*

Business Requirements and Customer Needs:

- ***Customer Self-Service**: The enhanced security measures for the customer portal ensure that customers can safely update their information without risking data breaches.*
- ***Internal Management**: Secure authentication and communication methods allow Customer Relationship Managers to update customer information securely and efficiently.*
- ***Data Analysis**: Data scientists can continue to query the database while adhering to stricter security protocols, ensuring their work complies with security and compliance requirements.*
- ***Compliance**: The proposed architecture ensures compliance with SOX and PCI DSS by implementing necessary security controls, audit logging, and data encryption, thereby meeting regulatory requirements.*

Associated Costs:

1. ***Infrastructure and Configuration:***
   - ***SSL Certificates**: Purchasing and maintaining SSL certificates for the database.*
   - ***Encryption**: Implementing encryption for data at rest may require additional computational resources and potential database downtime during setup.*
   - ***Audit Logging**: Enabling and maintaining audit logs could increase storage costs.*
2. ***Software and Tools:***
   - ***Secure Authentication Solutions**: Implementing OAuth or JWT may involve licensing or development costs.*
   - ***VPN Services**: Setting up and maintaining VPN infrastructure incurs both initial setup and ongoing operational costs.*
3. ***Operational Costs:***
   - ***Training**: Training staff on new security protocols and tools.*
   - ***Maintenance**: Regular security assessments, audits, and updates to maintain compliance and security standards.*


AWS
Customer
HTTPS
Customer Portal
(OAuth/JWT
Authentication)
Database Update
Processor
(OAuth/JWT)
HTTPS
VPN
VPN Gateway
(Secure Tunnel)
On-Premises Network
Customer Relationship
Manager
HTTPS
Internal Web Interface
(OAuth/JWT)
PostgreSQL Database
Audit
Logging
IP Whitelisting
Data Scientist
HTTPS
VPN
Read-only Access

Architecture Diagram Description

Components:

1. **Customer Portal (AWS)**
    - Secure log-in using OAuth/JWT
    - HTTPS enforced for all communications
2. **Internal Web Interface**
    - Secure log-in using OAuth/JWT
    - Direct connection to the PostgreSQL database within the corporate network
3. **Database Update Processor (AWS)**
    - Enforces HTTPS for POST requests
    - Uses secure authentication methods (OAuth/JWT)
    - Communicates with on-premises PostgreSQL database over a secure VPN
4. **PostgreSQL Database (On-Premises)**
    - SSL/TLS encryption enforced
    - Data encryption at rest
    - Audit logging enabled
    - Firewall configured with IP whitelisting
5. **Data Scientist Access**
    - Secure VPN access to the corporate network
    - Read-only access with strong authentication
    - Endpoint security measures (antivirus, firewall)
6. **Network Security**
    - VPN Gateway for secure communications between AWS and on-premises infrastructure
    - Firewall with strict rules and IP whitelisting