# Added Security Improvements List:

|  |  |
| --- | --- |
| Security Issue | Solution |
| API Gateway Single Point of Failure | Deploy multiple instances with a load balancer |
| No Rate Limiting (DDoS Risk) | Implement rate limiting with Redis/WAF |
| Refresh Token Theft Risk | Securely store and validate refresh tokens |
| Missing Role-Based Access Control | Use @PreAuthorize("hasRole('ADMIN')") for admin endpoints |
| No HTTPS (MITM Attack Risk) | Enable HTTPS with SSL/TLS |
| No Service-to-Service Authorization | Validate token claims for internal service calls |
| No Logging of Unauthorized Access | Add security logs for failed authentication attempts |
| Checkmark with solid fillSensitive Data | Store as environment variables, sensitive data should not be hardcoded |

# 1.Sensitive Data Improvement:

All sensitive information, such as urls, user credentials, servers, keys, ids, etc. That were added as hardcoded data in the code, should be stored as environment variables.

Total hardcoded parameters with sensitive data in all application.properties:

|  |
| --- |
| **api-gateway.service.url**  **eureka.client.service-url.defaultZone**  **spring.security.oauth2.resourceserver.jwt.issuer-uri**  **spring.security.oauth2.resourceserver.jwt.jwk-set-uri**  **spring.datasource.username**  **spring.datasource.password** |

## 1.1.How to Implement:

1. Start PowerShell (since we are using windows) as admin
2. Run below commands one by one:

|  |
| --- |
| >>[System.Environment]::SetEnvironmentVariable(**'KEYCLOAK\_SERVER\_URL'**, 'http://localhost:8080', 'Machine')  >> [System.Environment]::SetEnvironmentVariable(**'KEYCLOAK\_CWSMU\_REALM'**, 'customer-app-realm', 'Machine')  >> [System.Environment]::SetEnvironmentVariable(**'KEYCLOAK\_CWSMU\_CLIENT\_SECRET'**, 'IQMO1LDePpY3VtpA3dRBktD4UKNCXTqF', 'Machine')  >> [System.Environment]::SetEnvironmentVariable(**'KEYCLOAK\_CWSMU\_CLIENT\_ID'**, 'customer-service-client', 'Machine')  >> [System.Environment]::SetEnvironmentVariable(**'EUREKA\_SERVER\_URL'**, 'http://localhost:8761/eureka', 'Machine')  >> [System.Environment]::SetEnvironmentVariable(**'ORACLE\_DB\_USERNAME'**, 'system', 'Machine')  >> [System.Environment]::SetEnvironmentVariable(**'ORACLE\_DB\_PASSWORD'**, 'password', 'Machine')  >> [System.Environment]::SetEnvironmentVariable(**'API\_GW\_SERVICE\_URL'**, 'http://api-gateway', 'Machine') |

This provides to store related parameters as environment variables **permanantly**.

1. Restart the system.
2. You can check environment variable via PowerShell by “**echo $env:<parameter\_name>**”
3. Update related parameters in the application parameters like below example:

|  |  |
| --- | --- |
| Before | **spring.datasource.username=username** |
| After | **spring.datasource.username=${ORACLE\_DB\_USERNAME}** |

# 2.Refresh Token Theft Risk Improvement:

Since the project include refreshing token in the code, then the refresh token parameters are required to visible by the classes. The solution for this risk is to securly store and validate refresh tokens. Especially for a case to automate token refresh inside the project structure, it is more important to securely store client credentials. Since the microservicess (customer-service, address-service) need to refresh the token without exposing data like “client\_secret”, DO NOT harcode secrets.

It is secure way to use Spring Boot Configuration Files (application.properties) to store secrets securly:

|  |
| --- |
| keycloak.auth-server-url=${KEYCLOAK\_SERVER\_URL}  keycloak.realm=${KEYCLOAK\_CWSMU\_REALM}  keycloak.client-id=${KEYCLOAK\_CWSMU\_CLIENT\_ID}  keycloak.client-secret=${KEYCLOAK\_CWSMU\_CLIENT\_SECRET}  keycloak.token-endpoint=${KEYCLOAK\_SERVER\_URL}/realms/${KEYCLOAK\_CWSMU\_REALM}/protocol/openid-connect/token |

Check the refresh token all related parameters that have sensitive data and add as environment variables instead of hardcoding. Open PowerShell, run below command:

|  |
| --- |
| Temporarily:  For a specific user:  [System.Environment]::SetEnvironmentVariable('KEYCLOAK\_CLIENT\_SECRET', '<real\_parameter>’, 'User')  Permanently – required admin user  >>[System.Environment]::SetEnvironmentVariable('KEYCLOAK\_CLIENT\_SECRET', '<real\_parameter>’, 'Machine') |

# Additional Security Updates:

1. **Using Spring Boot Secrets Management**:

keycloak.client-secret=${KEYCLOAK\_CLIENT\_SECRET:default\_value}

\*The “*default\_value*” ensures the application doesn’t crash if the environment variable is missing.

1. **Using a Secrets Manager**:

If deploying in a cloud environment, use AWS Secrets Manager, Azure Key Vault, or Vault by HashiCorp to manage secrets securely