Vidzemes University of Applied Sciences

**Faculty of Engineering**

cybersecurity requirements engineering

project work assignment

“FIRST Assignment”

Valmiera, 2023

CONTENT

[1 Introduction 3](#__RefHeading___Toc885_2932776666)

[2 Chapter (example “System decomposition”) 3](#__RefHeading___Toc887_2932776666)

[Chapter-2.1 (example “Data flow diagrams”) 3](#__RefHeading___Toc889_2932776666)

[Chapter-2.1.1 (example “Use case diagram”) 3](#__RefHeading___Toc891_2932776666)

[Chapter-2.1.2 3](#__RefHeading___Toc893_2932776666)

[Chapter-2.2 (example “Process flow diagrams”) 3](#__RefHeading___Toc895_2932776666)

[3 Chapter-3 (example “Attack trees”) 3](#__RefHeading___Toc897_2932776666)

[Chapter-3.1 (example “Grumpy cat data storage asset”) 3](#__RefHeading___Toc1029_4288059415)

[References 4](#__RefHeading___Toc899_2932776666)

[Attachment 1 5](#__RefHeading___Toc901_2932776666)

# Introduction

**Goal and briefly about the system**.

Example: The goal of this assignment is to analyze cybersecurity requirements of the software “Processing Center”. This systems consists of several parts or modules which can work on one server or several through network. This document was created in order to identify security vulnerabilities and provide preventive actions. This system has created not only for cloud environment, but it can works in Kubernets with virtual servers infrastructure. The system has two main parts: **cloud environment** which has a DB instance of processing center, a web-server with 2 web-applications and ActiveMQ broker. **Local environment** also has DB instance, in this case, for cardissue system and works on standalone server, a web-server which has webAPI to work with internal request, for example a request to print new payment card’s pool. This systems can work with external request from mobile and web banking application. To handle all request fast and correct, there is an ActiveMQ broker which works with client’s requests. It is a simple non-commercial example without any additional components, I mean load balansers, firewalls, route networks, etc. More information below on diagram.

**Remember that this assignment should not include trade secrets**System decomposition and full report below:

# Threat Modeling Report

Created on 21.02.2023 23:08:35

**Threat Model Name:**

**Owner:**

**Reviewer:**

**Contributors:**

**Description:**

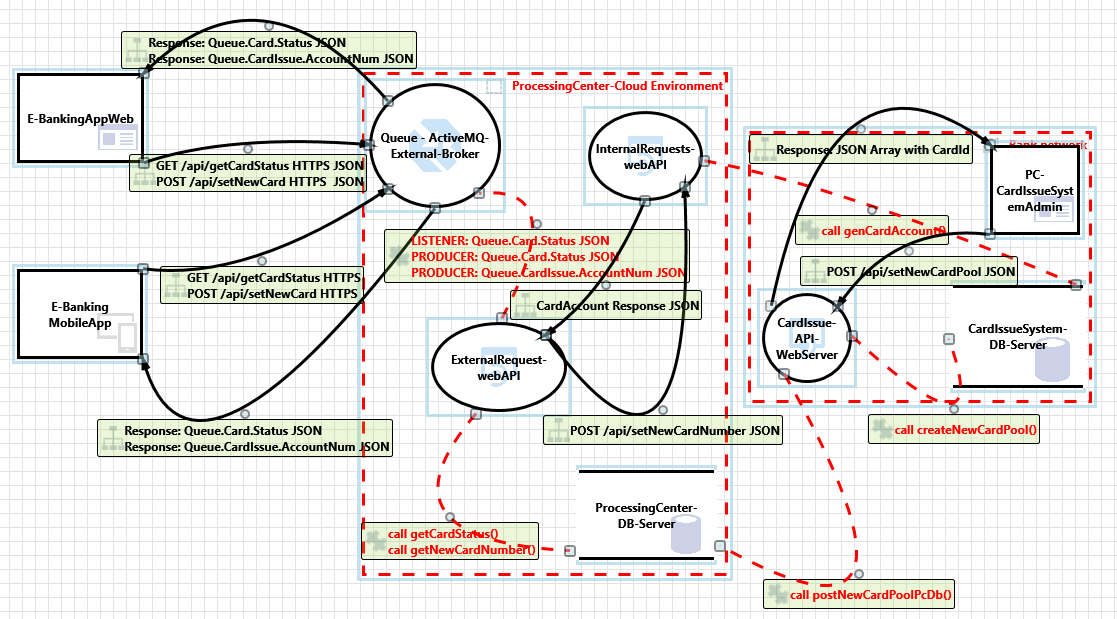
**Assumptions:**

**External Dependencies:**

### Threat Model Summary:

|  |  |
| --- | --- |
| Not Started | 46 |
| Not Applicable | 0 |
| Needs Investigation | 0 |
| Mitigation Implemented | 0 |
| Total | 46 |
| Total Migrated | 0 |

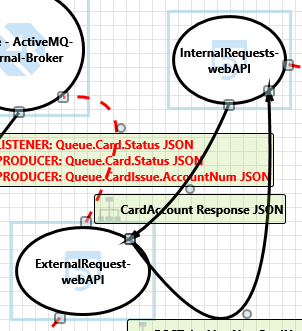
## Diagram: ORM-Hibernate-http-port-1522



### ORM-Hibernate-http-port-1522 Diagram Summary:

|  |  |
| --- | --- |
| Not Started | 46 |
| Not Applicable | 0 |
| Needs Investigation | 0 |
| Mitigation Implemented | 0 |
| Total | 46 |
| Total Migrated | 0 |

### Interaction: CardAccount Response JSON



#### 1. An adversary may gain unauthorized access to Web API due to poor access control checks  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | An adversary may gain unauthorized access to Web API due to poor access control checks |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Implement proper authorization mechanism in ASP.NET Web API. Refer: <a href="https://aka.ms/tmtauthz#authz-aspnet">https://aka.ms/tmtauthz#authz-aspnet</a> |
| **SDL Phase:** | Implementation |

#### 2. An adversary can gain access to sensitive information from an API through error messages  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that proper exception handling is done in ASP.NET Web API. Refer: <a href="https://aka.ms/tmtxmgmt#exception">https://aka.ms/tmtxmgmt#exception</a> |
| **SDL Phase:** | Implementation |

#### 3. An adversary can gain access to sensitive data by sniffing traffic to Web API  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary can gain access to sensitive data by sniffing traffic to Web API |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Force all traffic to Web APIs over HTTPS connection. Refer: <a href="https://aka.ms/tmtcommsec#webapi-https">https://aka.ms/tmtcommsec#webapi-https</a> |
| **SDL Phase:** | Implementation |

#### 4. An adversary can gain access to sensitive data stored in Web API's config files  [State: Not Started]  [Priority: Medium]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Encrypt sections of Web API's configuration files that contain sensitive data. Refer: <a href="https://aka.ms/tmtconfigmgmt#config-sensitive">https://aka.ms/tmtconfigmgmt#config-sensitive</a> |
| **SDL Phase:** | Implementation |

#### 5. Attacker can deny a malicious act on an API leading to repudiation issues  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Repudiation |
| **Description:** | Attacker can deny a malicious act on an API leading to repudiation issues |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that auditing and logging is enforced on Web API. Refer: <a href="https://aka.ms/tmtauditlog#logging-web-api">https://aka.ms/tmtauditlog#logging-web-api</a> |
| **SDL Phase:** | Design |

#### 6. An adversary may spoof InternalRequests-webAPI and gain access to Web API  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that standard authentication techniques are used to secure Web APIs. Refer: <a href="https://aka.ms/tmtauthn#authn-secure-api">https://aka.ms/tmtauthn#authn-secure-api</a> |
| **SDL Phase:** | Design |

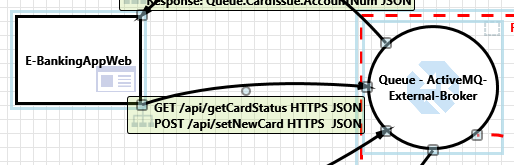
#### 7. An adversary may inject malicious inputs into an API and affect downstream processes  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may inject malicious inputs into an API and affect downstream processes |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that model validation is done on Web API methods. Refer: <a href="https://aka.ms/tmtinputval#validation-api">https://aka.ms/tmtinputval#validation-api</a> Implement input validation on all string type parameters accepted by Web API methods. Refer: <a href="https://aka.ms/tmtinputval#string-api">https://aka.ms/tmtinputval#string-api</a> |
| **SDL Phase:** | Implementation |

#### 8. An adversary can gain access to sensitive data by performing SQL injection through Web API  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that type-safe parameters are used in Web API for data access. Refer: <a href="https://aka.ms/tmtinputval#typesafe-api">https://aka.ms/tmtinputval#typesafe-api</a> |
| **SDL Phase:** | Implementation |

### Interaction: GET /api/getCardStatus HTTPS JSON POST /api/setNewCard HTTPS JSON



#### 9. An adversary may spoof the service or service endpoints by leveraging stale CNAME DNS records and executing a subdomain hijack attack  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | An adversary may spoof the service or service endpoints by leveraging stale CNAME DNS records and executing a subdomain hijack attack |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Address stale CNAME DNS records mapping custom domain names to the domain name of the Azure Traffic Manager instance. In some cases, deleting the stale CNAME records may be sufficient, while in other cases, the domain name of the Azure Traffic Manager instance should be kept to prevent subdomain hijack attacks. Refer: <a href="https://aka.ms/tmt-th178 ">https://aka.ms/tmt-th178 </a> |
| **SDL Phase:** | Implementation |

#### 10. An adversary may tamper deployed binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may tamper deployed binaries |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that deployed application's binaries are digitally signed. Refer: <a href="https://aka.ms/tmtauthn#binaries-signed">https://aka.ms/tmtauthn#binaries-signed</a> |
| **SDL Phase:** | Design |

#### 11. An adversary may reverse engineer deployed binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may reverse engineer deployed binaries |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that binaries are obfuscated if they contain sensitive information. Refer: <a href="https://aka.ms/tmtdata#binaries-info">https://aka.ms/tmtdata#binaries-info</a> |
| **SDL Phase:** | Implementation |

#### 12. An adversary may spread malware, steal or tamper data due to lack of endpoint protection on devices  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may spread malware, steal or tamper data due to lack of endpoint protection on devices. Scenarios such as stealing a user's laptop and extracting data from hard disk, luring users to install malware, exploit unpatched OS etc. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that devices have end point security controls configured as per organizational policies. Refer: <a href="https://aka.ms/tmtconfigmgmt#controls-policies">https://aka.ms/tmtconfigmgmt#controls-policies</a> |
| **SDL Phase:** | Design |

#### 13. An adversary may gain access to sensitive data stored on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary may gain access to sensitive data stored on host machines |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Consider using Encrypted File System (EFS) is used to protect confidential user-specific data. Refer: <a href="https://aka.ms/tmtdata#efs-user">https://aka.ms/tmtdata#efs-user</a> Ensure that sensitive data stored by the application on the file system is encrypted. Refer: <a href="https://aka.ms/tmtdata#filesystem">https://aka.ms/tmtdata#filesystem</a> |
| **SDL Phase:** | Design |

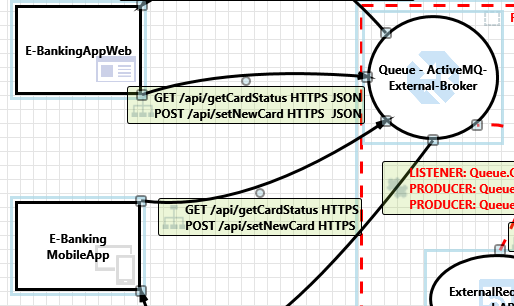
#### 14. An adversary may gain elevated privileges and execute malicious code on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | If an application runs under a high-privileged account, it may provide an opportunity for an adversary to gain elevated privileges and execute malicious code on host machines. E.g., If the developed executable runs under the logged-in user's identity and the user has admin rights on the machine, the executable will be running with administrator privileges. Any unnoticed vulnerability in the application could be used by adversaries to execute malicious code on the host machines that run the application. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that the deployed applications are run with least privileges. . Refer: <a href="https://aka.ms/tmtauthz#deployed-privileges">https://aka.ms/tmtauthz#deployed-privileges</a> |
| **SDL Phase:** | Implementation |

#### 15. An adversary may gain unauthorized access to data on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | An adversary may gain unauthorized access to data on host machines |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that proper ACLs are configured to restrict unauthorized access to data on the device. Refer: <a href="https://aka.ms/tmtauthz#acl-restricted-access">https://aka.ms/tmtauthz#acl-restricted-access</a> Ensure that sensitive user-specific application content is stored in user-profile directory. Refer: <a href="https://aka.ms/tmtauthz#sensitive-directory">https://aka.ms/tmtauthz#sensitive-directory</a> |
| **SDL Phase:** | Implementation |

### Interaction: GET /api/getCardStatus HTTPS POST /api/setNewCard HTTPS



#### 16. An adversary may spoof the service or service endpoints by leveraging stale CNAME DNS records and executing a subdomain hijack attack  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | An adversary may spoof the service or service endpoints by leveraging stale CNAME DNS records and executing a subdomain hijack attack |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Address stale CNAME DNS records mapping custom domain names to the domain name of the Azure Traffic Manager instance. In some cases, deleting the stale CNAME records may be sufficient, while in other cases, the domain name of the Azure Traffic Manager instance should be kept to prevent subdomain hijack attacks. Refer: <a href="https://aka.ms/tmt-th178 ">https://aka.ms/tmt-th178 </a> |
| **SDL Phase:** | Implementation |

#### 17. An adversary can reverse engineer and tamper binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary can use various tools, reverse engineer binaries and abuse them by tampering |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Obfuscate generated binaries before distributing to end users. Refer: <a href="https://aka.ms/tmtdata#binaries-end">https://aka.ms/tmtdata#binaries-end</a> |
| **SDL Phase:** | Design |

#### 18. An adversary may tamper deployed binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may tamper deployed binaries |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that deployed application's binaries are digitally signed. Refer: <a href="https://aka.ms/tmtauthn#binaries-signed">https://aka.ms/tmtauthn#binaries-signed</a> |
| **SDL Phase:** | Design |

#### 19. An adversary may reverse engineer deployed binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may reverse engineer deployed binaries |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that binaries are obfuscated if they contain sensitive information. Refer: <a href="https://aka.ms/tmtdata#binaries-info">https://aka.ms/tmtdata#binaries-info</a> |
| **SDL Phase:** | Implementation |

#### 20. An adversary may spread malware, steal or tamper data due to lack of endpoint protection on devices  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may spread malware, steal or tamper data due to lack of endpoint protection on devices. Scenarios such as stealing a user's laptop and extracting data from hard disk, luring users to install malware, exploit unpatched OS etc. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that devices have end point security controls configured as per organizational policies. Refer: <a href="https://aka.ms/tmtconfigmgmt#controls-policies">https://aka.ms/tmtconfigmgmt#controls-policies</a> |
| **SDL Phase:** | Design |

#### 21. An adversary can gain sensitive data from mobile device  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | If application saves sensitive PII or HBI data on phone SD card or local storage, then it ay get stolen. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Encrypt sensitive or PII data written to phones local storage. Refer: <a href="https://aka.ms/tmtdata#pii-phones">https://aka.ms/tmtdata#pii-phones</a> |
| **SDL Phase:** | Implementation |

#### 22. An adversary can gain access to sensitive data by sniffing traffic from Mobile client  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary can gain access to sensitive data by sniffing traffic from Mobile client |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Implement Certificate Pinning. Refer: <a href="https://aka.ms/tmtcommsec#cert-pinning">https://aka.ms/tmtcommsec#cert-pinning</a> |
| **SDL Phase:** | Implementation |

#### 23. An adversary may gain access to sensitive data stored on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary may gain access to sensitive data stored on host machines |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Consider using Encrypted File System (EFS) is used to protect confidential user-specific data. Refer: <a href="https://aka.ms/tmtdata#efs-user">https://aka.ms/tmtdata#efs-user</a> Ensure that sensitive data stored by the application on the file system is encrypted. Refer: <a href="https://aka.ms/tmtdata#filesystem">https://aka.ms/tmtdata#filesystem</a> |
| **SDL Phase:** | Design |

#### 24. An adversary may gain elevated privileges and execute malicious code on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | If an application runs under a high-privileged account, it may provide an opportunity for an adversary to gain elevated privileges and execute malicious code on host machines. E.g., If the developed executable runs under the logged-in user's identity and the user has admin rights on the machine, the executable will be running with administrator privileges. Any unnoticed vulnerability in the application could be used by adversaries to execute malicious code on the host machines that run the application. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that the deployed applications are run with least privileges. . Refer: <a href="https://aka.ms/tmtauthz#deployed-privileges">https://aka.ms/tmtauthz#deployed-privileges</a> |
| **SDL Phase:** | Implementation |

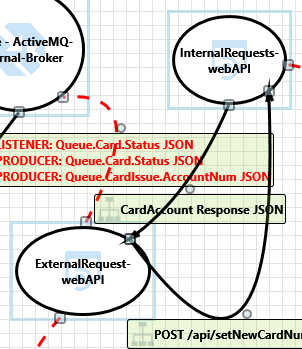
#### 25. An adversary may gain unauthorized access to data on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | An adversary may gain unauthorized access to data on host machines |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that proper ACLs are configured to restrict unauthorized access to data on the device. Refer: <a href="https://aka.ms/tmtauthz#acl-restricted-access">https://aka.ms/tmtauthz#acl-restricted-access</a> Ensure that sensitive user-specific application content is stored in user-profile directory. Refer: <a href="https://aka.ms/tmtauthz#sensitive-directory">https://aka.ms/tmtauthz#sensitive-directory</a> |
| **SDL Phase:** | Implementation |

#### 26. An adversary may jail break into a mobile device and gain elevated privileges  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | An adversary may jail break into a mobile device and gain elevated privileges |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Implement implicit jailbreak or rooting detection. Refer: <a href="https://aka.ms/tmtauthz#rooting-detection">https://aka.ms/tmtauthz#rooting-detection</a> |
| **SDL Phase:** | Design |

### Interaction: POST /api/setNewCardNumber JSON



#### 27. An adversary may gain unauthorized access to Web API due to poor access control checks  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | An adversary may gain unauthorized access to Web API due to poor access control checks |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Implement proper authorization mechanism in ASP.NET Web API. Refer: <a href="https://aka.ms/tmtauthz#authz-aspnet">https://aka.ms/tmtauthz#authz-aspnet</a> |
| **SDL Phase:** | Implementation |

#### 28. An adversary can gain access to sensitive information from an API through error messages  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that proper exception handling is done in ASP.NET Web API. Refer: <a href="https://aka.ms/tmtxmgmt#exception">https://aka.ms/tmtxmgmt#exception</a> |
| **SDL Phase:** | Implementation |

#### 29. An adversary can gain access to sensitive data by sniffing traffic to Web API  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary can gain access to sensitive data by sniffing traffic to Web API |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Force all traffic to Web APIs over HTTPS connection. Refer: <a href="https://aka.ms/tmtcommsec#webapi-https">https://aka.ms/tmtcommsec#webapi-https</a> |
| **SDL Phase:** | Implementation |

#### 30. An adversary can gain access to sensitive data stored in Web API's config files  [State: Not Started]  [Priority: Medium]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Encrypt sections of Web API's configuration files that contain sensitive data. Refer: <a href="https://aka.ms/tmtconfigmgmt#config-sensitive">https://aka.ms/tmtconfigmgmt#config-sensitive</a> |
| **SDL Phase:** | Implementation |

#### 31. Attacker can deny a malicious act on an API leading to repudiation issues  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Repudiation |
| **Description:** | Attacker can deny a malicious act on an API leading to repudiation issues |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that auditing and logging is enforced on Web API. Refer: <a href="https://aka.ms/tmtauditlog#logging-web-api">https://aka.ms/tmtauditlog#logging-web-api</a> |
| **SDL Phase:** | Design |

#### 32. An adversary may spoof ExternalRequest-webAPI and gain access to Web API  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that standard authentication techniques are used to secure Web APIs. Refer: <a href="https://aka.ms/tmtauthn#authn-secure-api">https://aka.ms/tmtauthn#authn-secure-api</a> |
| **SDL Phase:** | Design |

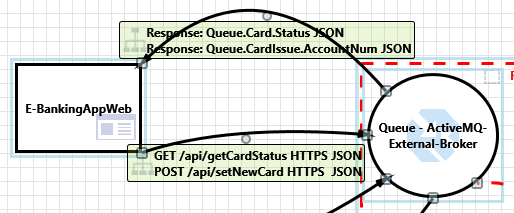
#### 33. An adversary may inject malicious inputs into an API and affect downstream processes  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may inject malicious inputs into an API and affect downstream processes |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that model validation is done on Web API methods. Refer: <a href="https://aka.ms/tmtinputval#validation-api">https://aka.ms/tmtinputval#validation-api</a> Implement input validation on all string type parameters accepted by Web API methods. Refer: <a href="https://aka.ms/tmtinputval#string-api">https://aka.ms/tmtinputval#string-api</a> |
| **SDL Phase:** | Implementation |

#### 34. An adversary can gain access to sensitive data by performing SQL injection through Web API  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that type-safe parameters are used in Web API for data access. Refer: <a href="https://aka.ms/tmtinputval#typesafe-api">https://aka.ms/tmtinputval#typesafe-api</a> |
| **SDL Phase:** | Implementation |

### Interaction: Response: Queue.Card.Status JSON Response: Queue.CardIssue.AccountNum JSON



#### 35. An adversary may gain unauthorized access to data on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | An adversary may gain unauthorized access to data on host machines |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that proper ACLs are configured to restrict unauthorized access to data on the device. Refer: <a href="https://aka.ms/tmtauthz#acl-restricted-access">https://aka.ms/tmtauthz#acl-restricted-access</a> Ensure that sensitive user-specific application content is stored in user-profile directory. Refer: <a href="https://aka.ms/tmtauthz#sensitive-directory">https://aka.ms/tmtauthz#sensitive-directory</a> |
| **SDL Phase:** | Implementation |

#### 36. An adversary may gain elevated privileges and execute malicious code on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | If an application runs under a high-privileged account, it may provide an opportunity for an adversary to gain elevated privileges and execute malicious code on host machines. E.g., If the developed executable runs under the logged-in user's identity and the user has admin rights on the machine, the executable will be running with administrator privileges. Any unnoticed vulnerability in the application could be used by adversaries to execute malicious code on the host machines that run the application. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that the deployed applications are run with least privileges. . Refer: <a href="https://aka.ms/tmtauthz#deployed-privileges">https://aka.ms/tmtauthz#deployed-privileges</a> |
| **SDL Phase:** | Implementation |

#### 37. An adversary may gain access to sensitive data stored on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary may gain access to sensitive data stored on host machines |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Consider using Encrypted File System (EFS) is used to protect confidential user-specific data. Refer: <a href="https://aka.ms/tmtdata#efs-user">https://aka.ms/tmtdata#efs-user</a> Ensure that sensitive data stored by the application on the file system is encrypted. Refer: <a href="https://aka.ms/tmtdata#filesystem">https://aka.ms/tmtdata#filesystem</a> |
| **SDL Phase:** | Design |

#### 38. An adversary may spread malware, steal or tamper data due to lack of endpoint protection on devices  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may spread malware, steal or tamper data due to lack of endpoint protection on devices. Scenarios such as stealing a user's laptop and extracting data from hard disk, luring users to install malware, exploit unpatched OS etc. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that devices have end point security controls configured as per organizational policies. Refer: <a href="https://aka.ms/tmtconfigmgmt#controls-policies">https://aka.ms/tmtconfigmgmt#controls-policies</a> |
| **SDL Phase:** | Design |

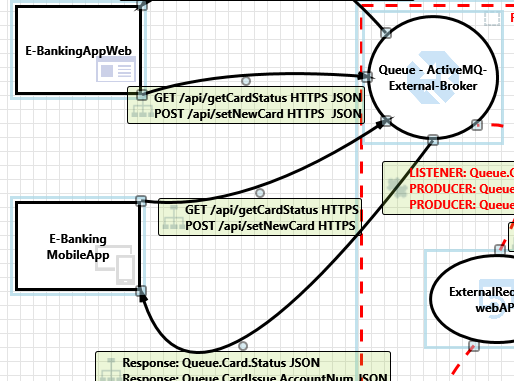
#### 39. An adversary may reverse engineer deployed binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may reverse engineer deployed binaries |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that binaries are obfuscated if they contain sensitive information. Refer: <a href="https://aka.ms/tmtdata#binaries-info">https://aka.ms/tmtdata#binaries-info</a> |
| **SDL Phase:** | Implementation |

#### 40. An adversary may tamper deployed binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may tamper deployed binaries |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that deployed application's binaries are digitally signed. Refer: <a href="https://aka.ms/tmtauthn#binaries-signed">https://aka.ms/tmtauthn#binaries-signed</a> |
| **SDL Phase:** | Design |

### Interaction: Response: Queue.Card.Status JSON Response: Queue.CardIssue.AccountNum JSON



#### 41. An adversary may gain unauthorized access to data on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | An adversary may gain unauthorized access to data on host machines |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that proper ACLs are configured to restrict unauthorized access to data on the device. Refer: <a href="https://aka.ms/tmtauthz#acl-restricted-access">https://aka.ms/tmtauthz#acl-restricted-access</a> Ensure that sensitive user-specific application content is stored in user-profile directory. Refer: <a href="https://aka.ms/tmtauthz#sensitive-directory">https://aka.ms/tmtauthz#sensitive-directory</a> |
| **SDL Phase:** | Implementation |

#### 42. An adversary may gain elevated privileges and execute malicious code on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Elevation of Privileges |
| **Description:** | If an application runs under a high-privileged account, it may provide an opportunity for an adversary to gain elevated privileges and execute malicious code on host machines. E.g., If the developed executable runs under the logged-in user's identity and the user has admin rights on the machine, the executable will be running with administrator privileges. Any unnoticed vulnerability in the application could be used by adversaries to execute malicious code on the host machines that run the application. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that the deployed applications are run with least privileges. . Refer: <a href="https://aka.ms/tmtauthz#deployed-privileges">https://aka.ms/tmtauthz#deployed-privileges</a> |
| **SDL Phase:** | Implementation |

#### 43. An adversary may gain access to sensitive data stored on host machines  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | An adversary may gain access to sensitive data stored on host machines |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Consider using Encrypted File System (EFS) is used to protect confidential user-specific data. Refer: <a href="https://aka.ms/tmtdata#efs-user">https://aka.ms/tmtdata#efs-user</a> Ensure that sensitive data stored by the application on the file system is encrypted. Refer: <a href="https://aka.ms/tmtdata#filesystem">https://aka.ms/tmtdata#filesystem</a> |
| **SDL Phase:** | Design |

#### 44. An adversary may spread malware, steal or tamper data due to lack of endpoint protection on devices  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may spread malware, steal or tamper data due to lack of endpoint protection on devices. Scenarios such as stealing a user's laptop and extracting data from hard disk, luring users to install malware, exploit unpatched OS etc. |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that devices have end point security controls configured as per organizational policies. Refer: <a href="https://aka.ms/tmtconfigmgmt#controls-policies">https://aka.ms/tmtconfigmgmt#controls-policies</a> |
| **SDL Phase:** | Design |

#### 45. An adversary may reverse engineer deployed binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may reverse engineer deployed binaries |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that binaries are obfuscated if they contain sensitive information. Refer: <a href="https://aka.ms/tmtdata#binaries-info">https://aka.ms/tmtdata#binaries-info</a> |
| **SDL Phase:** | Implementation |

#### 46. An adversary may tamper deployed binaries  [State: Not Started]  [Priority: High]

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | An adversary may tamper deployed binaries |
| **Justification:** | <no mitigation provided> |
| **Possible Mitigation(s):** | Ensure that deployed application's binaries are digitally signed. Refer: <a href="https://aka.ms/tmtauthn#binaries-signed">https://aka.ms/tmtauthn#binaries-signed</a> |
| **SDL Phase:** | Design |

# Chapter (example “System decomposition”)

AAA BBB CCC.

Example: “System decomposition”

## Chapter-2.1 (example “Data flow diagrams”)

AAA.

Example: Data flow diagrams shows how grumpy cat software is being constructed

### Chapter-2.1.1 (example “Use case diagram”)

AAA BB CCC.

### Chapter-2.1.2

AAA BBB CCC.

## Chapter-2.2 (example “Process flow diagrams”)

AAA BBB CCC.

Example:

# Chapter-3 (example “Attack trees”)

AAA BBB CCC.

## Chapter-3.1 (example “**Grumpy cat data storage asset**”)

# References

AAA.

Example

1. Pattern and Security Requirements: Engineering-Based Establishment of Security Standards. Krisstian Beckers, Springer, 2015.

2. Sci-kit learn, How to choose correct estimator for your model, Available https://scikit-learn.org/stable/tutorial/machine\_learning\_map/index.html

# Attachment 1

AAA.