Threat Modeling Report

Created on 2021-09-30 12:10:33 PM

Threat Model Name:
Owner:
Reviewer:
Contributors:
Description:
Assumptions:
External Dependencies:

Threat Model Summary:

Not Started	42
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	42
Total Migrated	0

Diagram: Diagram 1

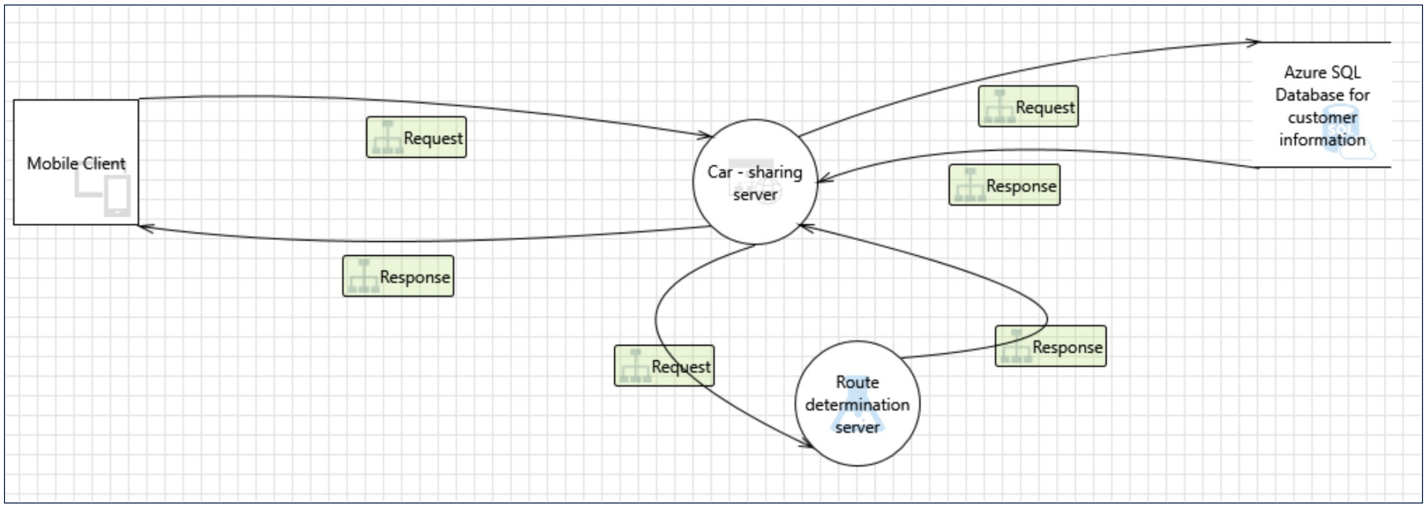
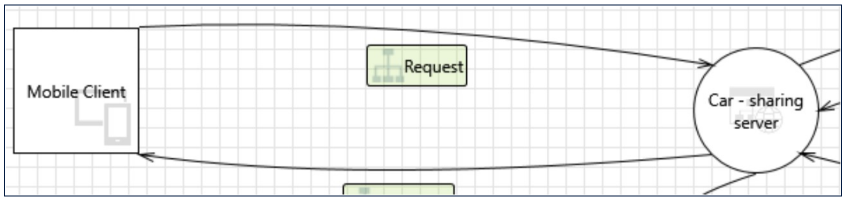


Diagram 1 Diagram Summary:

Not Started	42
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	42
Total Migrated	0

Interaction: Request



1. 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

2. An adversary may spoof Mobile Client and gain access to Web Application [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):	Consider using a standard authentication mechanism to authenticate to Web Application. Refer: <a href="https://aka.ms/tmtauthn#standard-authn-web-app">https://aka.ms/tmtauthn#standard-authn-web-app</a>
SDL Phase:	Design

3. An adversary can create a fake website and launch phishing attacks [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication
Justification:	<no mitigation provided>
Possible Mitigation(s):	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a> Ensure that authenticated ASP.NET pages incorporate UI Redressing or clickjacking defences. Refer: <a href="https://aka.ms/tmtconfigmgmt#ui-defenses">https://aka.ms/tmtconfigmgmt#ui-defenses</a> Validate all redirects within the application are closed or done safely. Refer: <a href="https://aka.ms/tmtinputval#redirect-safe">https://aka.ms/tmtinputval#redirect-safe</a>
SDL Phase:	Implementation

4. An adversary can steal sensitive data like user credentials [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor,
Justification:	<no mitigation provided>
Possible Mitigation(s):	Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Refer: <a href="https://aka.ms/tmtdata#autocomplete-input">https://aka.ms/tmtdata#autocomplete-input</a> Perform input validation and filtering on all string type Model properties. Refer: <a href="https://aka.ms/tmtinputval#typemodel">https://aka.ms/tmtinputval#typemodel</a> Validate all redirects within the application are closed or done safely. Refer: <a href="https://aka.ms/tmtinputval#redirect-safe">https://aka.ms/tmtinputval#redirect-safe</a> Enable step up or adaptive authentication. Refer: <a href="https://aka.ms/tmtauthn#step-up-adaptive-authn">https://aka.ms/tmtauthn#step-up-adaptive-authn</a> Implement forgot password functionalities securely. Refer: <a href="https://aka.ms/tmtauthn#forgot-pword-fxn">https://aka.ms/tmtauthn#forgot-pword-fxn</a> Ensure that password and account policy are implemented. Refer: <a href="https://aka.ms/tmtauthn#pword-account-policy">https://aka.ms/tmtauthn#pword-account-policy</a> Implement input validation on all string type parameters accepted by Controller methods. Refer: <a href="https://aka.ms/tmtinputval#string-method">https://aka.ms/tmtinputval#string-method</a>
SDL Phase:	Implementation

5. An adversary can spoof the target web application due to insecure TLS certificate configuration [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Ensure that TLS certificate parameters are configured with correct values
Justification:	<no mitigation provided>
Possible Mitigation(s):	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a>
SDL Phase:	Implementation

6. Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues [State: Not Started] [Priority: Medium]

Category:	Repudiation
Description:	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that auditing and logging is enforced on the application. Refer: <a href="https://aka.ms/tmtauditlog#auditing">https://aka.ms/tmtauditlog#auditing</a> Ensure that log rotation and separation are in place. Refer: <a href="https://aka.ms/tmtauditlog#log-rotation">https://aka.ms/tmtauditlog#log-rotation</a> Ensure that Audit and Log Files have Restricted Access. Refer: <a href="https://aka.ms/tmtauditlog#log-restricted-access">https://aka.ms/tmtauditlog#log-restricted-access</a> Ensure that User Management Events are Logged. Refer: <a href="https://aka.ms/tmtauditlog#user-management">https://aka.ms/tmtauditlog#user-management</a>
SDL Phase:	Implementation

7. An adversary can gain access to sensitive information 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):	Do not expose security details in error messages. Refer: <a href="https://aka.ms/tmtxmgmt#messages">https://aka.ms/tmtxmgmt#messages</a> Implement Default error handling page. Refer: <a href="https://aka.ms/tmtxmgmt#default">https://aka.ms/tmtxmgmt#default</a> Set Deployment Method to Retail in IIS. Refer: <a href="https://aka.ms/tmtxmgmt#deployment">https://aka.ms/tmtxmgmt#deployment</a> Exceptions should fail safely. Refer: <a href="https://aka.ms/tmtxmgmt#fail">https://aka.ms/tmtxmgmt#fail</a> ASP.NET applications must disable tracing and debugging prior to deployment. Refer: <a href="https://aka.ms/tmtconfigmgmt#trace-deploy">https://aka.ms/tmtconfigmgmt#trace-deploy</a> Implement controls to prevent username enumeration. Refer: <a href="https://aka.ms/tmtauthn#controls-username-enum">https://aka.ms/tmtauthn#controls-username-enum</a>
SDL Phase:	Implementation

8. 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

9. 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

10. An adversary may gain access to sensitive data from log files [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary may gain access to sensitive data from log files
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that the application does not log sensitive user data. Refer: <a href="https://aka.ms/tmtauditlog#log-sensitive-data">https://aka.ms/tmtauditlog#log-sensitive-data</a> Ensure that Audit and Log Files have Restricted Access. Refer: <a href="https://aka.ms/tmtauditlog#log-restricted-access">https://aka.ms/tmtauditlog#log-restricted-access</a>
SDL Phase:	Implementation

11. An adversary can reverse weakly encrypted or hashed content [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary can reverse weakly encrypted or hashed content
Justification:	<no mitigation provided>
Possible Mitigation(s):	Do not expose security details in error messages. Refer: <a href="https://aka.ms/tmtxmgmt#messages">https://aka.ms/tmtxmgmt#messages</a> Implement Default error handling page. Refer: <a href="https://aka.ms/tmtxmgmt#default">https://aka.ms/tmtxmgmt#default</a> Set Deployment Method to Retail in IIS. Refer: <a href="https://aka.ms/tmtxmgmt#deployment">https://aka.ms/tmtxmgmt#deployment</a> Use only approved symmetric block ciphers and key lengths. Refer: <a href="https://aka.ms/tmtcrypto#cipher-length">https://aka.ms/tmtcrypto#cipher-length</a> Use approved block cipher modes and initialization vectors for symmetric ciphers. Refer: <a href="https://aka.ms/tmtcrypto#vector-ciphers">https://aka.ms/tmtcrypto#vector-ciphers</a> Use approved asymmetric algorithms, key lengths, and padding. Refer: <a href="https://aka.ms/tmtcrypto#padding">https://aka.ms/tmtcrypto#padding</a> Use approved random number generators. Refer: <a href="https://aka.ms/tmtcrypto#numgen">https://aka.ms/tmtcrypto#numgen</a> Do not use symmetric stream ciphers. Refer: <a href="https://aka.ms/tmtcrypto#stream-ciphers">https://aka.ms/tmtcrypto#stream-ciphers</a> Use approved MAC/HMAC/keyed hash algorithms. Refer: <a href="https://aka.ms/tmtcrypto#mac-hash">https://aka.ms/tmtcrypto#mac-hash</a> Use only approved cryptographic hash functions. Refer: <a href="https://aka.ms/tmtcrypto#hash-functions">https://aka.ms/tmtcrypto#hash-functions</a> Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a>
SDL Phase:	Implementation

12. 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

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

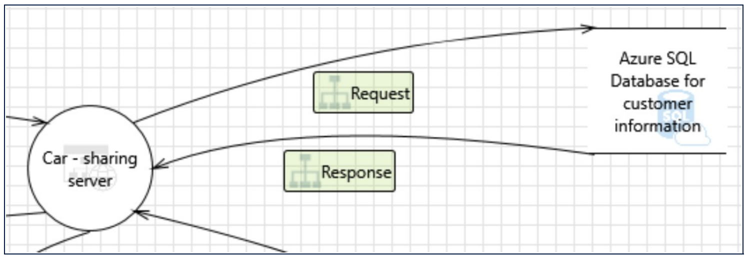
Category:	Tampering
Description:	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated 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 Application for data access. Refer: <a href="https://aka.ms/tmtinputval#typesafe">https://aka.ms/tmtinputval#typesafe</a>
SDL Phase:	Implementation

14. An adversary can gain access to sensitive data stored in Web App's config files [State: Not Started] [Priority: High]

Category:	Tampering
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 App's configuration files that contain sensitive data. Refer: <a href="https://aka.ms/tmtdata#encrypt-data">https://aka.ms/tmtdata#encrypt-data</a>
SDL Phase:	Implementation

Interaction: Request



15. An adversary can gain unauthorized access to Azure SQL database due to weak account policy [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	Due to poorly configured account policies, adversary can launch brute force attacks on Azure SQL Database for customer information
Justification:	<no mitigation provided>
Possible Mitigation(s):	When possible use Azure Active Directory Authentication for connecting to SQL Database. Refer: <a href="https://aka.ms/tmt-th10a">https://aka.ms/tmt-th10a</a> Ensure that least-privileged accounts are used to connect to Database server. Refer: <a href="https://aka.ms/tmt-th10b">https://aka.ms/tmt-th10b</a> and <a href="https://aka.ms/tmt-th10c">https://aka.ms/tmt-th10c</a>
SDL Phase:	Implementation

16. An adversary can gain unauthorized access to Azure SQL DB instances due to weak network security configuration. [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	An adversary can gain unauthorized access to Azure SQL DB instances due to weak network security configuration.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Restrict access to Azure SQL Database instances by configuring server-level and database-level firewall rules to permit connections from selected networks (e.g. a virtual network or a custom set of IP addresses) where possible. Refer:<a href="https://aka.ms/tmt-th143">https://aka.ms/tmt-th143</a>
SDL Phase:	Implementation

17. An adversary can read confidential data due to weak connection string configuration [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary can read confidential data due to weak connection string configuration.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Clients connecting to an Azure SQL Database instance using a connection string should ensure encrypt=true and trustservercertificate=false are set. This configuration ensures that connections are encrypted only if there is a verifiable server certificate (otherwise the connection attempt fails). This helps protect against Man-In-The-Middle attacks. Refer: <a href="https://aka.ms/tmt-th144">https://aka.ms/tmt-th144</a>
SDL Phase:	Implementation

18. An adversary having access to the storage container (e.g. physical access to the storage media) may be able to read sensitive data [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary having access to the storage container (e.g. physical access to the storage media) may be able to read sensitive data.
Justification:	<no mitigation provided>

Possible Mitigation(s):	Enable Transparent Data Encryption (TDE) on Azure SQL Database instances to have data encrypted at rest. Refer:<a href="https://aka.ms/tmt-th145a">https://aka.ms/tmt-th145a</a>. Use the Always Encrypted feature to allow client applications to encrypt sensitive data before it is sent to the Azure SQL Database. Refer: <a href="https://aka.ms/tmt-th145b">https://aka.ms/tmt-th145b</a>
SDL Phase:	Implementation

19. A compromised identity may permit more privileges than intended to an adversary due to weak permission and role assignments [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	A compromised identity may permit more privileges than intended to an adversary due to weak permission and role assignments.
Justification:	<no mitigation provided>
Possible Mitigation(s):	It is recommended to review permission and role assignments to ensure the users are granted the least privileges necessary. Refer: <a href="https://aka.ms/tmt-th146">https://aka.ms/tmt-th146</a>
SDL Phase:	Implementation

20. An adversary can deny actions performed on Azure SQL Database for customer information due to a lack of auditing [State: Not Started] [Priority: Medium]

Category:	Repudiation
Description:	An adversary can deny actions performed on Azure SQL Database for customer information due to a lack of auditing.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Enable auditing on Azure SQL Database instances to track and log database events. After configuring and customizing the audited events, enable threat detection to receive alerts on anomalous database activities indicating potential security threats. Refer: <a href="https://aka.ms/tmt-th147">https://aka.ms/tmt-th147</a>
SDL Phase:	Design

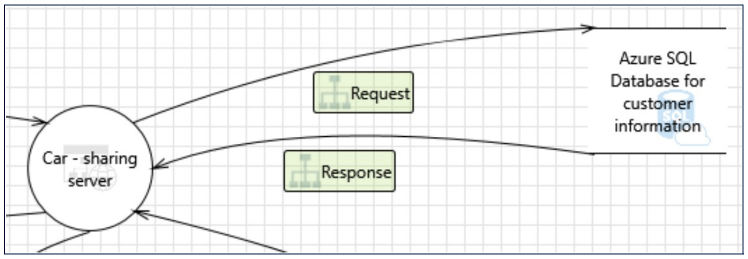
21. An adversary can gain long term, persistent access to an Azure SQL DB instance through the compromise of local user account password(s) [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	An adversary can gain long term, persistent access to an Azure SQL DB instance through the compromise of local user account password(s).
Justification:	<no mitigation provided>
Possible Mitigation(s):	It is recommended to rotate user account passwords (e.g. those used in connection strings) regularly, in accordance with your organization's policies. Store secrets in a secret storage solution (e.g. Azure Key Vault).
SDL Phase:	Implementation

22. An adversary may abuse weak Azure SQL Database for customer information configuration [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	An adversary may abuse weak Azure SQL Database for customer information configuration.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Enable SQL Vulnerability Assessment to gain visibility into the security posture of your Azure SQL Database instances. Acting on the assessment results help reduce attack surface and enhance your database security. Refer: <a href="https://aka.ms/tmt-th149">https://aka.ms/tmt-th149</a>
SDL Phase:	Implementation

Interaction: Response



23. An adversary can reverse weakly encrypted or hashed content [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary can reverse weakly encrypted or hashed content
Justification:	<no mitigation provided>
Possible Mitigation(s):	Do not expose security details in error messages. Refer: <a href="https://aka.ms/tmtxmgmt#messages">https://aka.ms/tmtxmgmt#messages</a> Implement Default error handling page. Refer: <a href="https://aka.ms/tmtxmgmt#default">https://aka.ms/tmtxmgmt#default</a> Set Deployment Method to Retail in IIS. Refer: <a href="https://aka.ms/tmtxmgmt#deployment">https://aka.ms/tmtxmgmt#deployment</a> Use only approved symmetric block ciphers and key lengths. Refer: <a href="https://aka.ms/tmtcrypto#cipher-length">https://aka.ms/tmtcrypto#cipher-length</a> Use approved block cipher modes and initialization vectors for symmetric ciphers. Refer: <a href="https://aka.ms/tmtcrypto#vector-ciphers">https://aka.ms/tmtcrypto#vector-ciphers</a> Use approved asymmetric algorithms, key lengths, and padding. Refer: <a href="https://aka.ms/tmtcrypto#padding">https://aka.ms/tmtcrypto#padding</a> Use approved random number generators. Refer: <a href="https://aka.ms/tmtcrypto#numgen">https://aka.ms/tmtcrypto#numgen</a> Do not use symmetric stream ciphers. Refer: <a href="https://aka.ms/tmtcrypto#stream-ciphers">https://aka.ms/tmtcrypto#stream-ciphers</a> Use approved MAC/HMAC/keyed hash algorithms. Refer: <a href="https://aka.ms/tmtcrypto#mac-hash">https://aka.ms/tmtcrypto#mac-hash</a> Use only approved cryptographic hash functions. Refer: <a href="https://aka.ms/tmtcrypto#hash-functions">https://aka.ms/tmtcrypto#hash-functions</a> Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a>
SDL Phase:	Implementation

24. An adversary may gain access to sensitive data from log files [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary may gain access to sensitive data from log files
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that the application does not log sensitive user data. Refer: <a href="https://aka.ms/tmtauditlog#log-sensitive-data">https://aka.ms/tmtauditlog#log-sensitive-data</a> Ensure that Audit and Log Files have Restricted Access. Refer: <a href="https://aka.ms/tmtauditlog#log-restricted-access">https://aka.ms/tmtauditlog#log-restricted-access</a>
SDL Phase:	Implementation

25. An adversary can gain access to sensitive information 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):	Do not expose security details in error messages. Refer: <a href="https://aka.ms/tmtxmgmt#messages">https://aka.ms/tmtxmgmt#messages</a> Implement Default error handling page. Refer: <a href="https://aka.ms/tmtxmgmt#default">https://aka.ms/tmtxmgmt#default</a> Set Deployment Method to Retail in IIS. Refer: <a href="https://aka.ms/tmtxmgmt#deployment">https://aka.ms/tmtxmgmt#deployment</a> Exceptions should fail safely. Refer: <a href="https://aka.ms/tmtxmgmt#fail">https://aka.ms/tmtxmgmt#fail</a> ASP.NET applications must disable tracing and debugging prior to deployment. Refer: <a href="https://aka.ms/tmtconfigmgmt#trace-deploy">https://aka.ms/tmtconfigmgmt#trace-deploy</a> Implement controls to prevent username enumeration. Refer: <a href="https://aka.ms/tmtauthn#controls-username-enum">https://aka.ms/tmtauthn#controls-username-enum</a>
SDL Phase:	Implementation

26. Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues [State: Not Started] [Priority: Medium]

Category:	Repudiation
Description:	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that auditing and logging is enforced on the application. Refer: <a href="https://aka.ms/tmtauditlog#auditing">https://aka.ms/tmtauditlog#auditing</a> Ensure that log rotation and separation are in place. Refer: <a href="https://aka.ms/tmtauditlog#log-rotation">https://aka.ms/tmtauditlog#log-rotation</a> Ensure that Audit and Log Files have Restricted Access. Refer: <a href="https://aka.ms/tmtauditlog#log-restricted-access">https://aka.ms/tmtauditlog#log-restricted-access</a> Ensure that User Management Events are Logged. Refer: <a href="https://aka.ms/tmtauditlog#user-management">https://aka.ms/tmtauditlog#user-management</a>
SDL Phase:	Implementation

27. An adversary can spoof the target web application due to insecure TLS certificate configuration [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Ensure that TLS certificate parameters are configured with correct values
Justification:	<no mitigation provided>
Possible Mitigation(s):	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a>
SDL Phase:	Implementation

28. An adversary can steal sensitive data like user credentials [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor,
Justification:	<no mitigation provided>
Possible Mitigation(s):	Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Refer: <a href="https://aka.ms/tmtdata#autocomplete-input">https://aka.ms/tmtdata#autocomplete-input</a> Perform input validation and filtering on all string type Model properties. Refer: <a href="https://aka.ms/tmtinputval#typemodel">https://aka.ms/tmtinputval#typemodel</a> Validate all redirects within the application are closed or done safely. Refer: <a href="https://aka.ms/tmtinputval#redirect-safe">https://aka.ms/tmtinputval#redirect-safe</a> Enable step up or adaptive authentication. Refer: <a href="https://aka.ms/tmtauthn#step-up-adaptive-authn">https://aka.ms/tmtauthn#step-up-adaptive-authn</a> Implement forgot password functionalities securely. Refer: <a href="https://aka.ms/tmtauthn#forgot-pword-fxn">https://aka.ms/tmtauthn#forgot-pword-fxn</a> Ensure that password and account policy are implemented. Refer: <a href="https://aka.ms/tmtauthn#pword-account-policy">https://aka.ms/tmtauthn#pword-account-policy</a> Implement input validation on all string type parameters accepted by Controller methods. Refer: <a href="https://aka.ms/tmtinputval#string-method">https://aka.ms/tmtinputval#string-method</a>
SDL Phase:	Implementation

29. An adversary can create a fake website and launch phishing attacks [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication
Justification:	<no mitigation provided>
Possible Mitigation(s):	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a> Ensure that authenticated ASP.NET pages incorporate UI Redressing or clickjacking defences. Refer: <a href="https://aka.ms/tmtconfigmgmt#ui-defenses">https://aka.ms/tmtconfigmgmt#ui-defenses</a> Validate all redirects within the application are closed or done safely. Refer: <a href="https://aka.ms/tmtinputval#redirect-safe">https://aka.ms/tmtinputval#redirect-safe</a>
SDL Phase:	Implementation

30. An adversary may spoof Azure SQL Database for customer information and gain access to Web Application [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):	Consider using a standard authentication mechanism to authenticate to Web Application. Refer: <a href="https://aka.ms/tmtauthn#standard-authn-web-app">https://aka.ms/tmtauthn#standard-authn-web-app</a>
SDL Phase:	Design

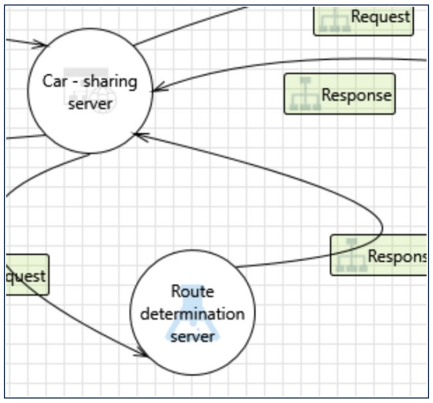
31. An adversary can gain access to sensitive data by performing SQL injection through Web App [State: Not Started] [Priority: High]

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 Application for data access. Refer: <a href="https://aka.ms/tmtinputval#typesafe">https://aka.ms/tmtinputval#typesafe</a>
SDL Phase:	Implementation

32. An adversary can gain access to sensitive data stored in Web App's config files [State: Not Started] [Priority: High]

Category:	Tampering
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 App's configuration files that contain sensitive data. Refer: <a href="https://aka.ms/tmtdata#encrypt-data">https://aka.ms/tmtdata#encrypt-data</a>
SDL Phase:	Implementation

Interaction: Response



33. An adversary can reverse weakly encrypted or hashed content [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary can reverse weakly encrypted or hashed content
Justification:	<no mitigation provided>
Possible Mitigation(s):	Do not expose security details in error messages. Refer: <a href="https://aka.ms/tmtxmgmt#messages">https://aka.ms/tmtxmgmt#messages</a> Implement Default error handling page. Refer: <a href="https://aka.ms/tmtxmgmt#default">https://aka.ms/tmtxmgmt#default</a> Set Deployment Method to Retail in IIS. Refer: <a href="https://aka.ms/tmtxmgmt#deployment">https://aka.ms/tmtxmgmt#deployment</a> Use only approved symmetric block ciphers and key lengths. Refer: <a href="https://aka.ms/tmtcrypto#cipher-length">https://aka.ms/tmtcrypto#cipher-length</a> Use approved block cipher modes and initialization vectors for symmetric ciphers. Refer: <a href="https://aka.ms/tmtcrypto#vector-ciphers">https://aka.ms/tmtcrypto#vector-ciphers</a> Use approved asymmetric algorithms, key lengths, and padding. Refer: <a href="https://aka.ms/tmtcrypto#padding">https://aka.ms/tmtcrypto#padding</a> Use approved random number generators. Refer: <a href="https://aka.ms/tmtcrypto#numgen">https://aka.ms/tmtcrypto#numgen</a> Do not use symmetric stream ciphers. Refer: <a href="https://aka.ms/tmtcrypto#stream-

	ciphers">https://aka.ms/tmtcrypto#stream-ciphers</a> Use approved MAC/HMAC/keyed hash algorithms. Refer: <a href="https://aka.ms/tmtcrypto#mac-hash">https://aka.ms/tmtcrypto#mac-hash</a> Use only approved cryptographic hash functions. Refer: <a href="https://aka.ms/tmtcrypto#hash-functions">https://aka.ms/tmtcrypto#hash-functions</a> Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a>
SDL Phase:	Implementation

34. An adversary may gain access to sensitive data from log files [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary may gain access to sensitive data from log files
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that the application does not log sensitive user data. Refer: <a href="https://aka.ms/tmtauditlog#log-sensitive-data">https://aka.ms/tmtauditlog#log-sensitive-data</a> Ensure that Audit and Log Files have Restricted Access. Refer: <a href="https://aka.ms/tmtauditlog#log-restricted-access">https://aka.ms/tmtauditlog#log-restricted-access</a>
SDL Phase:	Implementation

35. An adversary can gain access to sensitive information 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):	Do not expose security details in error messages. Refer: <a href="https://aka.ms/tmtxmgmt#messages">https://aka.ms/tmtxmgmt#messages</a> Implement Default error handling page. Refer: <a href="https://aka.ms/tmtxmgmt#default">https://aka.ms/tmtxmgmt#default</a> Set Deployment Method to Retail in IIS. Refer: <a href="https://aka.ms/tmtxmgmt#deployment">https://aka.ms/tmtxmgmt#deployment</a> Exceptions should fail safely. Refer: <a href="https://aka.ms/tmtxmgmt#fail">https://aka.ms/tmtxmgmt#fail</a> ASP.NET applications must disable tracing and debugging prior to deployment. Refer: <a href="https://aka.ms/tmtconfigmgmt#trace-deploy">https://aka.ms/tmtconfigmgmt#trace-deploy</a> Implement controls to prevent username enumeration. Refer: <a href="https://aka.ms/tmtauthn#controls-username-enum">https://aka.ms/tmtauthn#controls-username-enum</a>
SDL Phase:	Implementation

36. Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues [State: Not Started] [Priority: Medium]

Category:	Repudiation
Description:	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that auditing and logging is enforced on the application. Refer: <a href="https://aka.ms/tmtauditlog#auditing">https://aka.ms/tmtauditlog#auditing</a> Ensure that log rotation and separation are in place. Refer: <a href="https://aka.ms/tmtauditlog#log-rotation">https://aka.ms/tmtauditlog#log-rotation</a> Ensure that Audit and Log Files have Restricted Access. Refer: <a href="https://aka.ms/tmtauditlog#log-restricted-access">https://aka.ms/tmtauditlog#log-restricted-access</a> Ensure that User Management Events are Logged. Refer: <a href="https://aka.ms/tmtauditlog#user-management">https://aka.ms/tmtauditlog#user-management</a>
SDL Phase:	Implementation

37. An adversary can spoof the target web application due to insecure TLS certificate configuration [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Ensure that TLS certificate parameters are configured with correct values
Justification:	<no mitigation provided>
Possible Mitigation(s):	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a>
SDL Phase:	Implementation

38. An adversary can steal sensitive data like user credentials [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor,
Justification:	<no mitigation provided>
Possible Mitigation(s):	Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Refer: <a href="https://aka.ms/tmtdata#autocomplete-input">https://aka.ms/tmtdata#autocomplete-input</a> Perform input validation and filtering on all string type Model properties. Refer: <a href="https://aka.ms/tmtinputval#typemodel">https://aka.ms/tmtinputval#typemodel</a> Validate all redirects within the application are closed or done safely. Refer: <a href="https://aka.ms/tmtinputval#redirect-safe">https://aka.ms/tmtinputval#redirect-safe</a> Enable step up or adaptive authentication. Refer: <a href="https://aka.ms/tmtauthn#step-up-adaptive-authn">https://aka.ms/tmtauthn#step-up-adaptive-authn</a> Implement forgot password functionalities securely. Refer: <a href="https://aka.ms/tmtauthn#forgot-pword-fxn">https://aka.ms/tmtauthn#forgot-pword-fxn</a> Ensure that password and account policy are implemented. Refer: <a href="https://aka.ms/tmtauthn#pword-account-policy">https://aka.ms/tmtauthn#pword-account-policy</a> Implement input validation on all string type parameters accepted by Controller methods. Refer: <a href="https://aka.ms/tmtinputval#string-method">https://aka.ms/tmtinputval#string-method</a>
SDL Phase:	Implementation

39. An adversary can create a fake website and launch phishing attacks [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication
Justification:	<no mitigation provided>
Possible Mitigation(s):	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: <a href="https://aka.ms/tmtcommsec#x509-ssltls">https://aka.ms/tmtcommsec#x509-ssltls</a> Ensure that authenticated ASP.NET pages incorporate UI Redressing or clickjacking defences. Refer: <a href="https://aka.ms/tmtconfigmgmt#ui-defenses">https://aka.ms/tmtconfigmgmt#ui-defenses</a> Validate all redirects within the application are closed or done safely. Refer: <a href="https://aka.ms/tmtinputval#redirect-safe">https://aka.ms/tmtinputval#redirect-safe</a>
SDL Phase:	Implementation

40. An adversary may spoof Route determination server and gain access to Web Application [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):	Consider using a standard authentication mechanism to authenticate to Web Application. Refer: <a href="https://aka.ms/tmtauthn#standard-authn-web-app">https://aka.ms/tmtauthn#standard-authn-web-app</a>
SDL Phase:	Design

41. An adversary can gain access to sensitive data by performing SQL injection through Web App [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 Application for data access. Refer: <a href="https://aka.ms/tmtinputval#typesafe">https://aka.ms/tmtinputval#typesafe</a>
SDL Phase:	Implementation

42. An adversary can gain access to sensitive data stored in Web App's config files [State: Not Started] [Priority: High]

Category:	Tampering

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 App's configuration files that contain sensitive data. Refer: <a href="https://aka.ms/tmtdata#encrypt-data">https://aka.ms/tmtdata#encrypt-data</a>
SDL Phase:	Implementation