Threat Modeling Report

Created on 31/03/2025 17:59:26

Threat Model Name: TFG Ines

Owner: Inés López Giménez

Reviewer: Ambrosio Toval

Contributors: Ambrosio Toval

Description: Ejemplo práctico de una arquitectura sistema de pago en línea para mostrar en mi TFG el funcionamiento de la herramienta Microsoft Tool.

Assumptions:

External Dependencies:

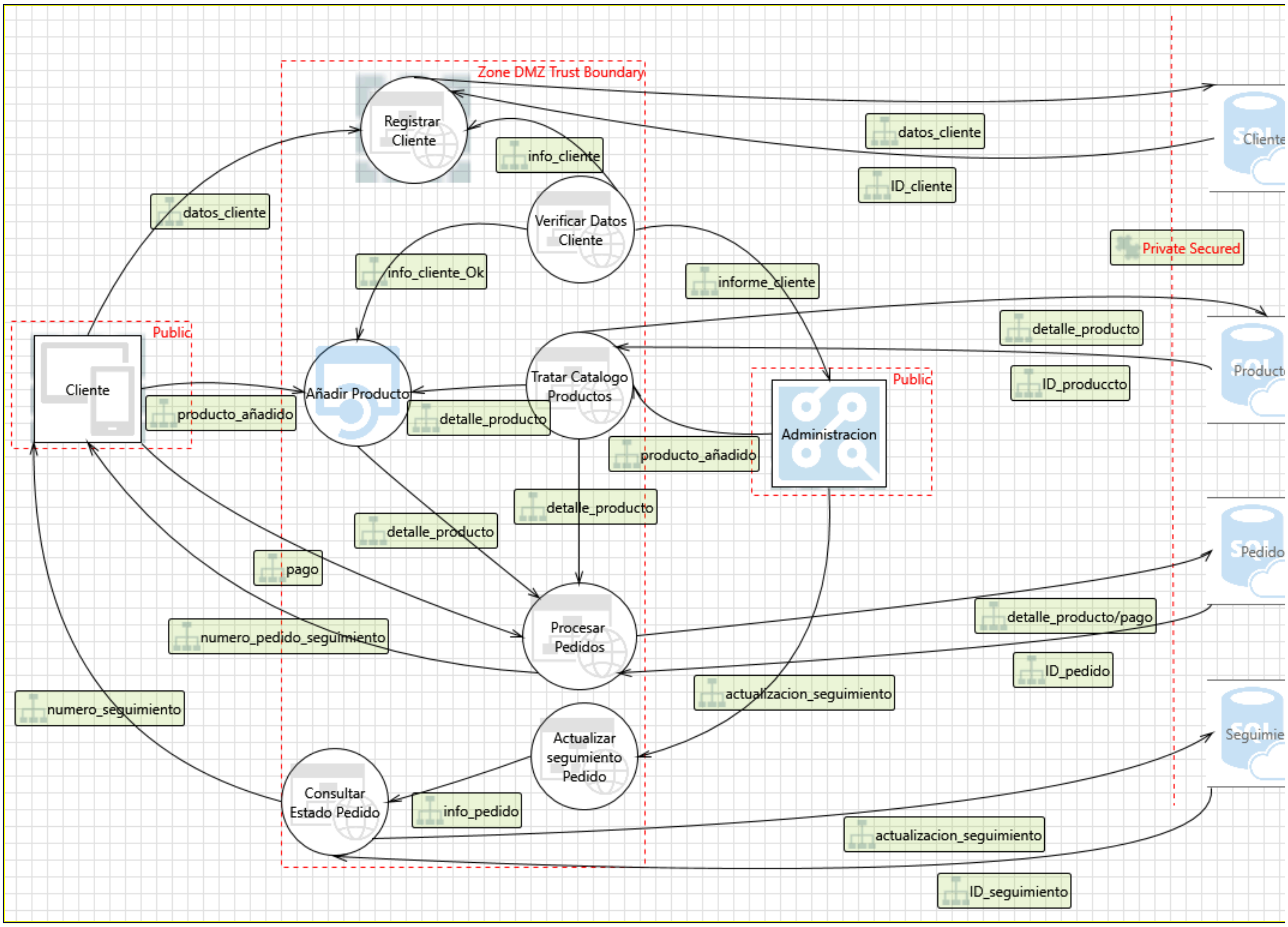
Notes:

Id	Note	Date	Added By
1		06/02/2025 18:41:06	DESKTOP-5UU3R4J\Nesi

Threat Model Summary:

Not Started	300
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	300
Total Migrated	0

Diagram: TFG Ines

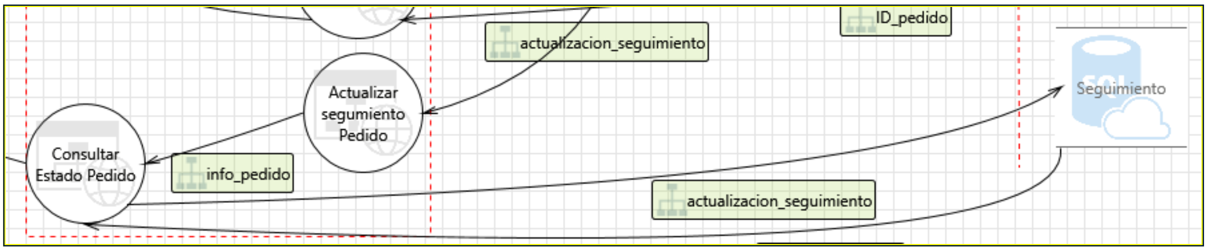


TFG Ines Diagram Summary:

Not Started	300

Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	300
Total Migrated	0

Interaction: actualizacion_seguimiento


ID_pedido
actualizacion_seguimiento
Actualizar seguimiento Pedido
Seguimiento
Consultar Estado Pedido
info_pedido
actualizacion_seguimiento

1. An adversary may abuse weak Seguimiento configuration [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	An adversary may abuse weak Seguimiento 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

2. 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

3. 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

4. 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

5. 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

6. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

7. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

8. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

9. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

10. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

11. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

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

Category: Elevation of Privileges

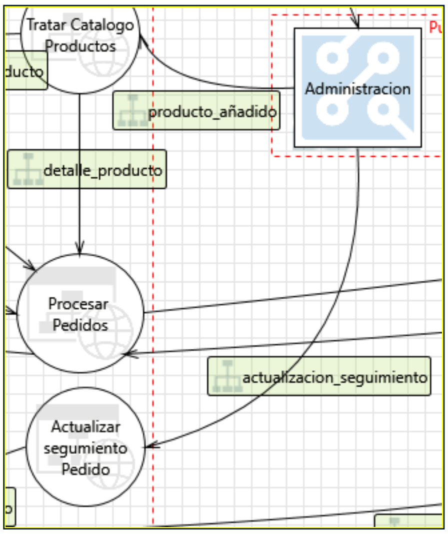
Description: Due to poorly configured account policies, adversary can launch brute force attacks on Seguimiento

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

Interaction: actualizacion_seguimiento


Tratar Catalogo Productos
ducto
producto_añadido
Administracion
Pu
detalle_producto
Procesar Pedidos
actualizacion_seguimiento
Actualizar seguimiento Pedido
o

13. An adversary may gain unauthorized access to privileged features on Administracion [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: An adversary may get access to admin interface or privileged services like WiFi, SSH, File shares, FTP etc., on a device

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that all admin interfaces are secured with strong credentials. Refer: <a href="https://aka.ms/tmtconfigmgmt#admin-strong">https://aka.ms/tmtconfigmgmt#admin-strong</a>

SDL Phase: Implementation

14. An adversary may exploit unused services or features in Actualizar segumiento Pedido [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: An adversary may use unused features or services on Actualizar segumiento Pedido such as UI, USB port etc. Unused features increase the attack surface and serve as additional entry points for the adversary

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that only the minimum services/features are enabled on devices. Refer: <a href="https://aka.ms/tmtconfigmgmt#min-enable">https://aka.ms/tmtconfigmgmt#min-enable</a>

SDL Phase: Implementation

15. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

16. 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

17. 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

18. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

19. An adversary can gain access to sensitive data by sniffing traffic to Azure Web App [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: An adversary may conduct man in the middle attack and downgrade TLS connection to clear text protocol, or forcing browser communication to pass through a proxy server that he controls. This may happen because the application may use mixed content or HTTP Strict Transport Security policy is not ensured.

Justification: <no mitigation provided>

Possible Mitigation(s): Configure SSL certificate for custom domain in Azure App Service. Refer: <a href="https://aka.ms/tmtcommsec#ssl-appservice">https://aka.ms/tmtcommsec#ssl-appservice</a> Force all traffic to Azure App Service over HTTPS connection . Refer: <a href="https://aka.ms/tmtcommsec#appservice-https">https://aka.ms/tmtcommsec#appservice-https</a>

SDL Phase: Implementation

20. An adversary can fingerprint an Azure web application by leveraging server header information [State: Not Started] [Priority: Low]

Category: Information Disclosure

Description: An adversary can fingerprint web application by leveraging server header information

Justification: <no mitigation provided>

Possible Mitigation(s): Remove standard server headers on Windows Azure Web Sites to avoid fingerprinting. Refer: <a href="https://aka.ms/tmtconfigmgmt#standard-finger">https://aka.ms/tmtconfigmgmt#standard-finger</a>

SDL Phase: Implementation

21. 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

22. 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

23. An adversary can deny actions on Azure App Service due to lack of auditing [State: Not Started] [Priority: High]

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): Enable diagnostics logging for web apps in Azure App Service. Refer: <a href="https://aka.ms/tmtauditlog#diagnostics-logging">https://aka.ms/tmtauditlog#diagnostics-logging</a>

SDL Phase: Implementation

24. 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

25. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

26. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

27. 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

28. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

29. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

30. 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

31. An adversary may spoof Administracion 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

32. An adversary may exploit known vulnerabilities in unpatched devices [State: Not Started] [Priority: High]

Category: Tampering

Description: An adversary may leverage known vulnerabilities and exploit a device if the firmware of the device is not updated

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that the Cloud Gateway implements a process to keep the connected devices firmware up to date. Refer: <a href="https://aka.ms/tmtconfigmgmt#cloud-firmware">https://aka.ms/tmtconfigmgmt#cloud-firmware</a>

SDL Phase: Design

33. An adversary may tamper Administracion and extract cryptographic key material from it [State: Not Started] [Priority: High]

Category: Tampering

Description: An adversary may partially or wholly replace the software running on Actualizar segumiento Pedido, potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material.

Justification: <no mitigation provided>

Possible Mitigation(s): Store Cryptographic Keys securely on IoT Device. Refer: <a href="https://aka.ms/tmtcrypto#keys-iot">https://aka.ms/tmtcrypto#keys-iot</a>

SDL Phase: Design

34. An adversary may tamper the OS of a device and launch offline attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: An adversary may launch offline attacks made by disabling or circumventing the installed operating system, or made by physically separating the storage media from the device in order to attack the data separately.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt OS and additional partitions of IoT Device with Bitlocker. Refer: <a href="https://aka.ms/tmtconfigmgmt#partition-iot">https://aka.ms/tmtconfigmgmt#partition-iot</a>

SDL Phase: Design

35. 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

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

Category: Tampering

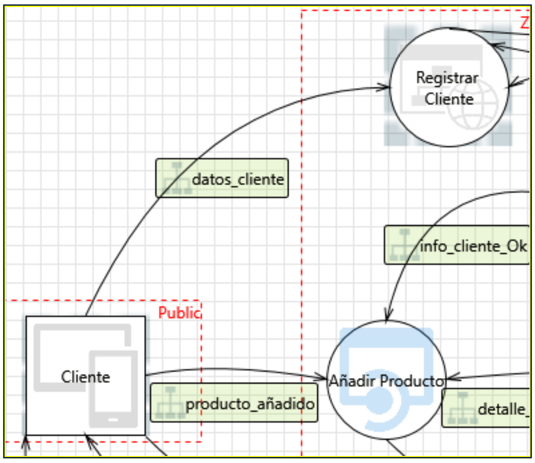
Description: An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.

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: datos_cliente



37. 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

38. 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

39. 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

40. 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

41. 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

42. An adversary can gain access to sensitive data by sniffing traffic to Azure Web App [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: An adversary may conduct man in the middle attack and downgrade TLS connection to clear text protocol, or forcing browser communication to pass through a proxy server that he controls. This may happen because the application may use mixed content or HTTP Strict Transport Security policy is not ensured.

Justification: <no mitigation provided>

Possible Mitigation(s): Configure SSL certificate for custom domain in Azure App Service. Refer: <a href="https://aka.ms/tmtcommsec#ssl-appservice">https://aka.ms/tmtcommsec#ssl-appservice</a> Force all traffic to Azure App Service over HTTPS connection . Refer: <a href="https://aka.ms/tmtcommsec#appservice-https">https://aka.ms/tmtcommsec#appservice-https</a>

SDL Phase: Implementation

43. An adversary can fingerprint an Azure web application by leveraging server header information [State: Not Started] [Priority: Low]

Category: Information Disclosure

Description: An adversary can fingerprint web application by leveraging server header information

Justification: <no mitigation provided>

Possible Mitigation(s): Remove standard server headers on Windows Azure Web Sites to avoid fingerprinting. Refer: <a href="https://aka.ms/tmtconfigmgmt#standard-finger">https://aka.ms/tmtconfigmgmt#standard-finger</a>

SDL Phase: Implementation

44. 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

45. 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

46. An adversary can deny actions on Azure App Service due to lack of auditing [State: Not Started] [Priority: High]

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): Enable diagnostics logging for web apps in Azure App Service. Refer: <a href="https://aka.ms/tmtauditlog#diagnostics-logging">https://aka.ms/tmtauditlog#diagnostics-logging</a>

SDL Phase: Implementation

47. 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

48. 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

49. 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

50. An adversary may spoof Cliente 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

51. 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

52. 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

53. 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

54. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

55. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

56. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

57. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

58. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

59. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

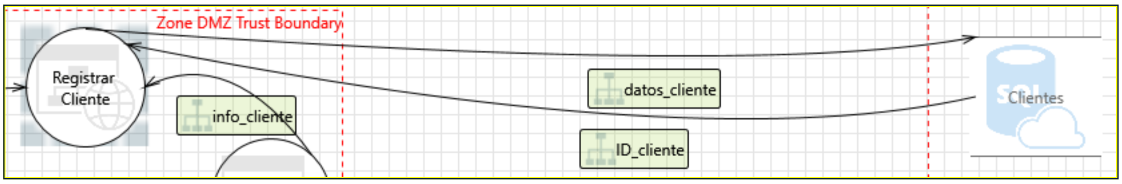
Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

Interaction: datos_cliente



60. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

61. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

62. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

63. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

64. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

65. 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 Clientes

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

66. An adversary may abuse weak Clientes configuration [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: An adversary may abuse weak Clientes 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

67. 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

68. 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

69. 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

70. 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

71. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

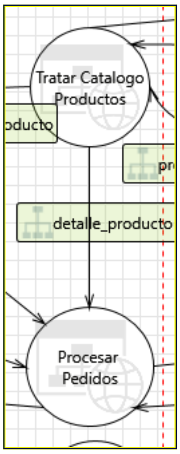
Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

Interaction: detalle_producto



72. 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

73. 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

74. 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

75. 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

76. 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

77. 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

78. 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

79. An adversary may spoof Tratar Catalogo Productos 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

80. 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

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

Category: Tampering

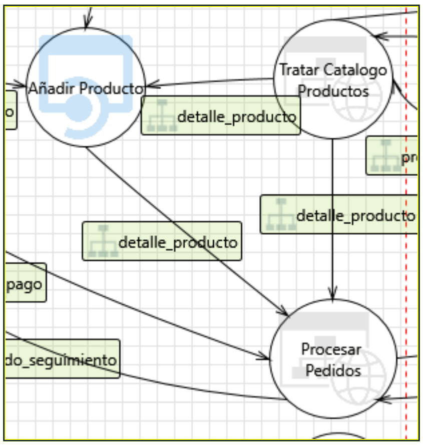
Description: An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.

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: detalle_producto



82. 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

83. 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

84. 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

85. 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

86. 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

87. 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

88. 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

89. An adversary may spoof Añadir Producto 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

90. 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

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

Category: Tampering

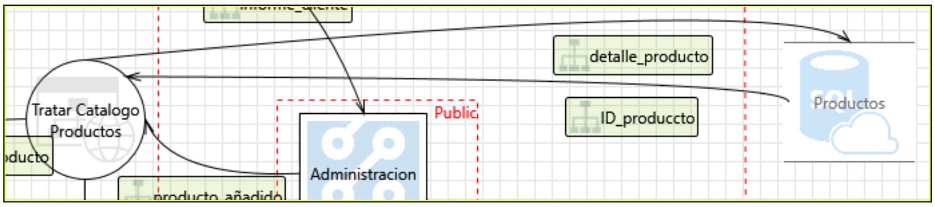
Description: An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.

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: detalle_producto



92. 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 Productos

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

93. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

94. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

95. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

96. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

97. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

98. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

99. 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

100. 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

101. 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

102. 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

103. An adversary may abuse weak Productos configuration [State: Not Started] [Priority: High]

Category: Elevation of Privileges

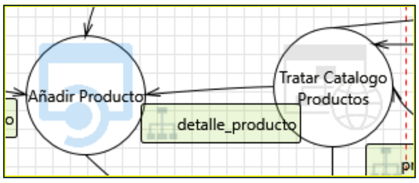
Description: An adversary may abuse weak Productos configuration.

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: detalle_producto



104. 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

105. 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

106. 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

107. 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

108. 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

109. An adversary may spoof Tratar Catalogo Productos 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

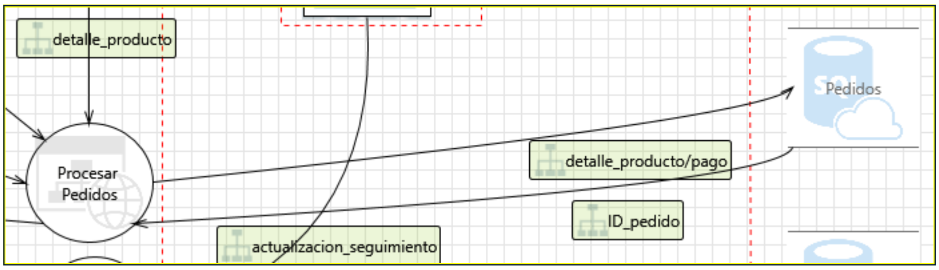
110. An adversary may inject malicious inputs into an API and affect downstream processes [State: Not Started] [Priority: High]

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

111. 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: detalle_producto/pago



112. 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 Pedidos
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

113. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category:	Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

114. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

115. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

116. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

117. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

118. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

119. 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

120. 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

121. 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

122. 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

123. An adversary may abuse weak Pedidos configuration [State: Not Started] [Priority: High]

Category: Elevation of Privileges

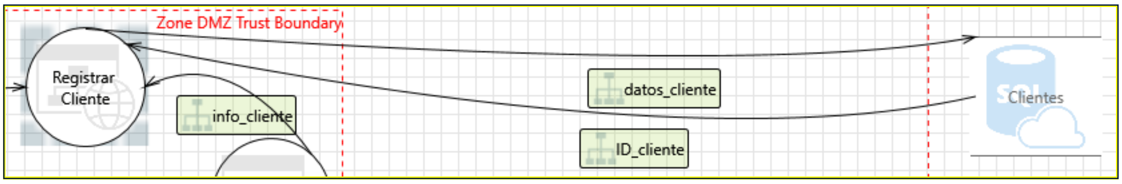
Description: An adversary may abuse weak Pedidos configuration.

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: ID_cliente



124. 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

125. 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

126. An adversary may spoof Clientes 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

127. 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

128. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

129. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

130. 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

131. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

132. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

133. 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

134. 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

135. 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

136. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

137. 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

138. 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

139. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

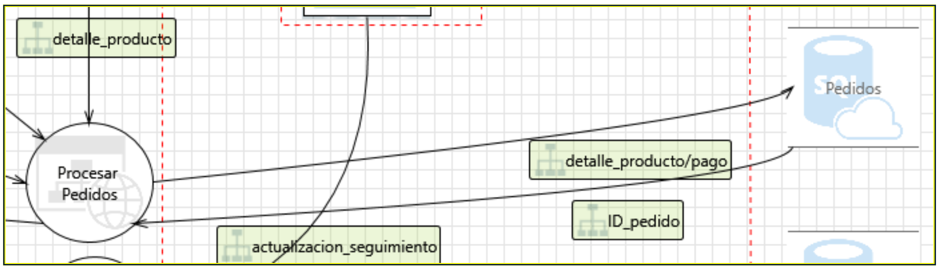
Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

Interaction: ID_pedido



140. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

141. 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

142. 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

143. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

144. 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

145. 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

146. 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

147. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

148. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

149. 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

150. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

151. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

152. 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

153. An adversary may spoof Pedidos 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

154. 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

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

Category: Tampering

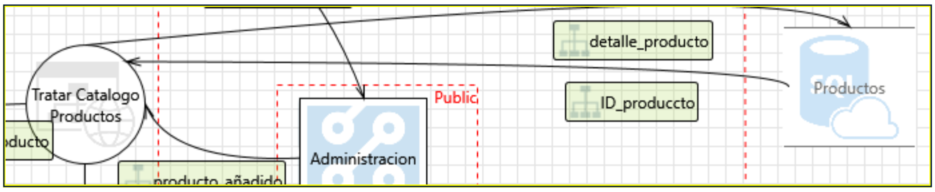
Description: An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.

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: ID_produccto



156. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

157. 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

158. 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

159. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

160. 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

161. 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

162. 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

163. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

164. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

165. 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

166. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

167. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

168. 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

169. An adversary may spoof Productos 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

170. 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

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

Category: Tampering

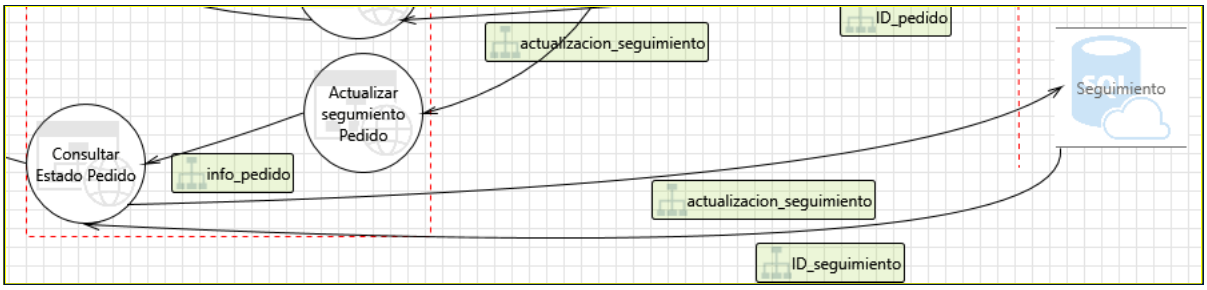
Description: An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.

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: ID_seguimiento


actualizacion_seguimiento
ID_pedido
Actualizar seguimiento Pedido
Seguimiento
Consultar Estado Pedido
info_pedido
actualizacion_seguimiento
ID_seguimiento

172. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

173. 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

174. 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

175. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

176. 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

177. 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

178. 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

179. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

180. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

181. 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

182. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

183. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

184. 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

185. An adversary may spoof Seguimiento 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

186. 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

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

Category: Tampering

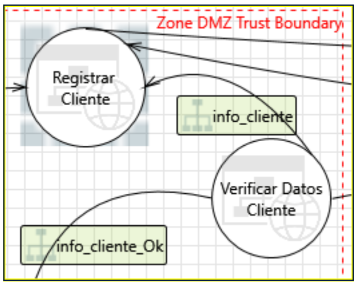
Description: An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.

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: info_cliente



188. 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

189. 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

190. An adversary may spoof Verificar Datos Cliente 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

191. 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

192. 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

193. 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

194. 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

195. 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

196. 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

197. 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

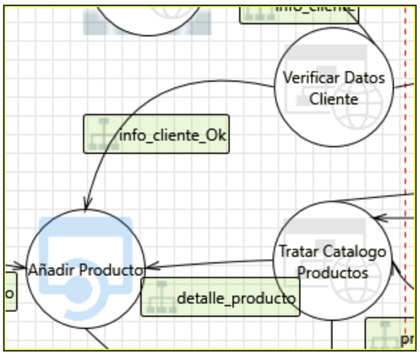
Justification: <no mitigation provided>

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

Interaction: info_cliente_Ok



198. 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

199. 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

200. An adversary may spoof Verificar Datos Cliente 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

201. 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

202. 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

203. 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

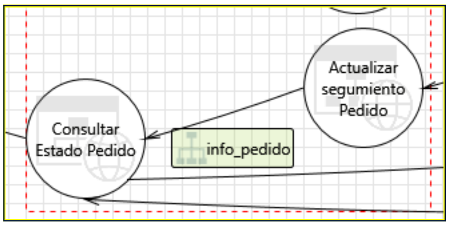
204. An adversary can gain access to sensitive information from an API through error messages [State: Not Started] [Priority: High]

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

205. 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

Interaction: info_pedido



206. 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

207. 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

208. An adversary may spoof Actualizar segumiento Pedido 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

209. 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

210. 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

211. 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

212. 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

213. 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

214. 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

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

Category: Information Disclosure

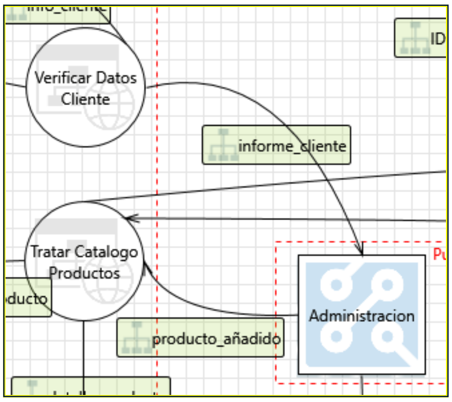
Description: An adversary can reverse weakly encrypted or hashed content

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

Interaction: informe_cliente



216. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

217. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

218. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

219. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

220. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

221. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

222. An adversary may execute unknown code on Administracion [State: Not Started] [Priority: High]

Category: Tampering

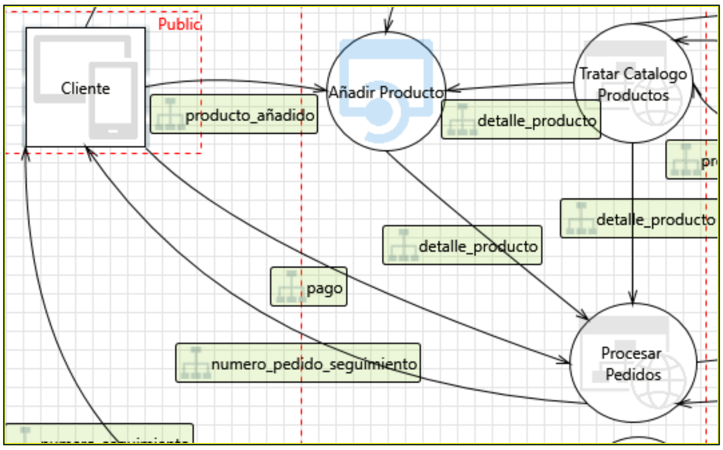
Description: An adversary may launch malicious code into Administracion and execute it

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that unknown code cannot execute on devices. Refer: <a href="https://aka.ms/tmtconfigmgmt#unknown-exe">https://aka.ms/tmtconfigmgmt#unknown-exe</a>

SDL Phase: Design

Interaction: numero_pedido_seguimiento



223. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations
Justification:	<no mitigation provided>
Possible Mitigation(s):	Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>
SDL Phase:	Design

224. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>
SDL Phase:	Implementation

225. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>
SDL Phase:	Implementation

226. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>
SDL Phase:	Implementation

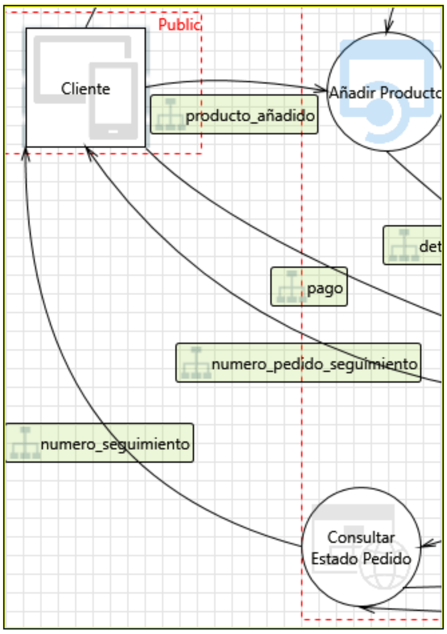
227. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric
Justification:	<no mitigation provided>
Possible Mitigation(s):	Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>
SDL Phase:	Design

228. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>
SDL Phase:	Design

Interaction: numero_seguimiento



229. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations
Justification:	<no mitigation provided>
Possible Mitigation(s):	Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>
SDL Phase:	Design

230. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>
SDL Phase:	Implementation

231. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>
SDL Phase:	Implementation

232. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category:	Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

233. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

234. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

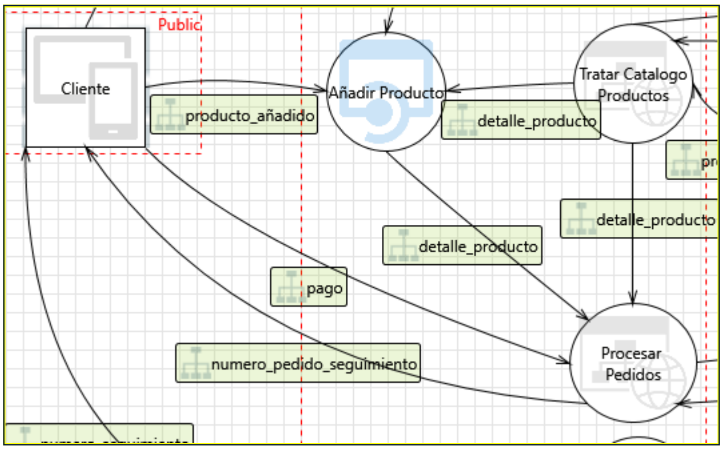
Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

Interaction: pago



235. 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

236. 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

237. 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

238. An adversary may spoof Cliente 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

239. 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

240. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>
SDL Phase:	Design

241. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric
Justification:	<no mitigation provided>
Possible Mitigation(s):	Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>
SDL Phase:	Design

242. 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

243. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

244. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

245. 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

246. An adversary can deny actions on Azure App Service due to lack of auditing [State: Not Started] [Priority: High]

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): Enable diagnostics logging for web apps in Azure App Service. Refer: <a href="https://aka.ms/tmtauditlog#diagnostics-logging">https://aka.ms/tmtauditlog#diagnostics-logging</a>

SDL Phase: Implementation

247. 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

248. 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

249. An adversary can fingerprint an Azure web application by leveraging server header information [State: Not Started] [Priority: Low]

Category: Information Disclosure

Description: An adversary can fingerprint web application by leveraging server header information

Justification: <no mitigation provided>

Possible Mitigation(s): Remove standard server headers on Windows Azure Web Sites to avoid fingerprinting. Refer: <a href="https://aka.ms/tmtconfigmgmt#standard-finger">https://aka.ms/tmtconfigmgmt#standard-finger</a>

SDL Phase: Implementation

250. An adversary can gain access to sensitive data by sniffing traffic to Azure Web App [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: An adversary may conduct man in the middle attack and downgrade TLS connection to clear text protocol, or forcing browser communication to pass through a proxy server that he controls. This may happen because the application may use mixed content or HTTP Strict Transport Security policy is not ensured.

Justification: <no mitigation provided>

Possible Mitigation(s): Configure SSL certificate for custom domain in Azure App Service. Refer: <a href="https://aka.ms/tmtcommsec#ssl-appservice">https://aka.ms/tmtcommsec#ssl-appservice</a> Force all traffic to Azure App Service over HTTPS connection . Refer: <a href="https://aka.ms/tmtcommsec#appservice-https">https://aka.ms/tmtcommsec#appservice-https</a>

SDL Phase: Implementation

251. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

252. 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

253. 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

254. 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

255. 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

256. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

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

Category: Elevation of Privileges

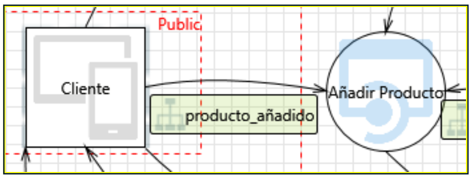
Description: An adversary may jail break into a mobile device and gain elevated privileges

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: producto_añadido



258. 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

259. 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

260. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>

SDL Phase: Design

261. 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

262. 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

263. 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

264. 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

265. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>

SDL Phase: Implementation

266. 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

267. 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

268. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

269. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

270. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

271. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

272. An adversary obtains refresh or access tokens from Cliente and uses them to obtain access to the Añadir Producto API [State: Not Started] [Priority: High]

Category: Spoofing

Description: On a public client (e.g. a mobile device), refresh tokens may be stolen and used by an attacker to obtain access to the API. Depending on the client type, there are different ways that tokens may be revealed to an attacker and therefore different ways to protect them, some involving how the software using the tokens requests, stores and refreshes them.

Justification: <no mitigation provided>

Possible Mitigation(s): Use ADAL libraries to manage token requests from OAuth2 clients to AAD (or on-premises AD). Refer: <a href="https://aka.ms/tmtauthn#adal-oauth2">https://aka.ms/tmtauthn#adal-oauth2</a>

SDL Phase: Implementation

273. An adversary may spoof Cliente 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

274. 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

275. 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

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

Category: Tampering

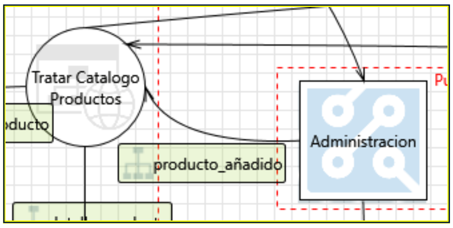
Description: SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands 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: producto_añadido



277. An adversary may gain unauthorized access to privileged features on Administracion [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: An adversary may get access to admin interface or privileged services like WiFi, SSH, File shares, FTP etc., on a device

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that all admin interfaces are secured with strong credentials. Refer: <a href="https://aka.ms/tmtconfigmgmt#admin-strong">https://aka.ms/tmtconfigmgmt#admin-strong</a>

SDL Phase: Implementation

278. An adversary may exploit unused services or features in Tratar Catalogo Productos [State: Not Started] [Priority: High]

Category: Elevation of Privileges

Description: An adversary may use unused features or services on Tratar Catalogo Productos such as UI, USB port etc. Unused features increase the attack surface and serve as additional entry points for the adversary

Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that only the minimum services/features are enabled on devices. Refer: <a href="https://aka.ms/tmtconfigmgmt#min-enable">https://aka.ms/tmtconfigmgmt#min-enable</a>
SDL Phase:	Implementation

279. An adversary may gain unauthorized access to Service Fabric cluster operations [State: Not Started] [Priority: High]

Category:	Elevation of Privileges
Description:	If RBAC is not implemented on Service Fabric, clients may have over-privileged access on the fabric's cluster operations
Justification:	<no mitigation provided>
Possible Mitigation(s):	Restrict client's access to cluster operations using RBAC. Refer: <a href="https://aka.ms/tmtauthz#cluster-rbac">https://aka.ms/tmtauthz#cluster-rbac</a>
SDL Phase:	Design

280. 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

281. 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

282. An adversary can gain access to unencrypted secrets in Service Fabric applications [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. If secrets are not encrypted, an adversary who can gain access to them can abuse them.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Encrypt secrets in Service Fabric applications. Refer: <a href="https://aka.ms/tmtdata#fabric-apps">https://aka.ms/tmtdata#fabric-apps</a>
SDL Phase:	Implementation

283. An adversary can gain access to sensitive data by sniffing traffic to Azure Web App [State: Not Started] [Priority: High]

Category:	Information Disclosure
Description:	An adversary may conduct man in the middle attack and downgrade TLS connection to clear text protocol, or forcing browser communication to pass through a proxy server that he controls. This may happen because the application may use mixed content or HTTP Strict Transport Security policy is not ensured.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Configure SSL certificate for custom domain in Azure App Service. Refer: <a href="https://aka.ms/tmtcommsec#ssl-appservice">https://aka.ms/tmtcommsec#ssl-appservice</a> Force all traffic to Azure App Service over HTTPS connection . Refer: <a href="https://aka.ms/tmtcommsec#appservice-https">https://aka.ms/tmtcommsec#appservice-https</a>
SDL Phase:	Implementation

284. An adversary can fingerprint an Azure web application by leveraging server header information [State: Not Started] [Priority: Low]

Category: Information Disclosure

Description: An adversary can fingerprint web application by leveraging server header information

Justification: <no mitigation provided>

Possible Mitigation(s): Remove standard server headers on Windows Azure Web Sites to avoid fingerprinting. Refer: <a href="https://aka.ms/tmtconfigmgmt#standard-finger">https://aka.ms/tmtconfigmgmt#standard-finger</a>

SDL Phase: Implementation

285. 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

286. 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

287. An adversary can deny actions on Azure App Service due to lack of auditing [State: Not Started] [Priority: High]

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): Enable diagnostics logging for web apps in Azure App Service. Refer: <a href="https://aka.ms/tmtauditlog#diagnostics-logging">https://aka.ms/tmtauditlog#diagnostics-logging</a>

SDL Phase: Implementation

288. 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

289. An adversary may gain unauthorized access to resources in Service Fabric [State: Not Started] [Priority: High]

Category: Spoofing

Description: If a service fabric cluster is not secured, it allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.

Justification: <no mitigation provided>

Possible Mitigation(s): Restrict anonymous access to Service Fabric Cluster. Refer: <a href="https://aka.ms/tmtauthn#anon-access-cluster">https://aka.ms/tmtauthn#anon-access-cluster</a>

SDL Phase: Implementation

290. An adversary can spoof a node and access Service Fabric cluster [State: Not Started] [Priority: High]

Category: Spoofing

Description: If the same certificate that is used for node-to-node security is used for client-to-node security, it will be easy for an adversary to spoof and join a new node, in case the client-to-node certificate (which is often stored locally) is compromised

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate. Refer: <a href="https://aka.ms/tmtauthn#fabric-cn-nn">https://aka.ms/tmtauthn#fabric-cn-nn</a>

SDL Phase: Implementation

291. 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

292. An adversary can potentially spoof a client if weaker client authentication channels are used [State: Not Started] [Priority: High]

Category: Spoofing

Description: Azure AD authentication provides better control on identity management and hence it is a better alternative to authenticate clients to Service Fabric

Justification: <no mitigation provided>

Possible Mitigation(s): Use AAD to authenticate clients to service fabric clusters. Refer: <a href="https://aka.ms/tmtauthn#aad-client-fabric">https://aka.ms/tmtauthn#aad-client-fabric</a>

SDL Phase: Design

293. An adversary can spoof a node in Service Fabric cluster by using stolen certificates [State: Not Started] [Priority: High]

Category: Spoofing

Description: If self-signed or test certificates are stolen, it would be difficult to revoke them. An adversary can use stolen certificates and continue to get access to Service Fabric cluster.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA). Refer: <a href="https://aka.ms/tmtauthn#fabric-cert-ca">https://aka.ms/tmtauthn#fabric-cert-ca</a>

SDL Phase: Design

294. 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

295. An adversary may spoof Administracion 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

296. An adversary may exploit known vulnerabilities in unpatched devices [State: Not Started] [Priority: High]

Category:	Tampering
Description:	An adversary may leverage known vulnerabilities and exploit a device if the firmware of the device is not updated
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that the Cloud Gateway implements a process to keep the connected devices firmware up to date. Refer: <a href="https://aka.ms/tmtconfigmgmt#cloud-firmware">https://aka.ms/tmtconfigmgmt#cloud-firmware</a>
SDL Phase:	Design

297. An adversary may tamper Administracion and extract cryptographic key material from it [State: Not Started] [Priority: High]

Category:	Tampering
Description:	An adversary may partially or wholly replace the software running on Tratar Catalogo Productos, potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Store Cryptographic Keys securely on IoT Device. Refer: <a href="https://aka.ms/tmtcrypto#keys-iot">https://aka.ms/tmtcrypto#keys-iot</a>
SDL Phase:	Design

298. An adversary may tamper the OS of a device and launch offline attacks [State: Not Started] [Priority: High]

Category:	Tampering
Description:	An adversary may launch offline attacks made by disabling or circumventing the installed operating system, or made by physically separating the storage media from the device in order to attack the data separately.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Encrypt OS and additional partitions of IoT Device with Bitlocker. Refer: <a href="https://aka.ms/tmtconfigmgmt#partition-iot">https://aka.ms/tmtconfigmgmt#partition-iot</a>
SDL Phase:	Design

299. 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

300. 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