Azure Cloud Audit Checklist

SACHIN HISSARIA
CA | CISA | DISA | CEH | COBIT-19 | ISO27001:2022 | RPA | Trainer

Sr. No	Control	Risk	Auditors Remarks
1	**Ensure That 'All users with the following roles' is set to 'Owner' (Automated)** Enable security alert emails to subscription owners.	Enabling security alert emails to subscription owners ensures that they receive security alert emails from Microsoft. This ensures that they are aware of any potential security issues and can mitigate the risk in a timely fashion.	
2	**Ensure 'Additional email addresses' is Configured with a Security Contact Email (Automated)** Microsoft Defender for Cloud emails the subscription owners whenever a high-severity alert is triggered for their subscription. You should provide a security contact email address as an additional email address.	Microsoft Defender for Cloud emails the Subscription Owner to notify them about security alerts. Adding your Security Contact's email address to the 'Additional email addresses' field ensures that your organization's Security Team is included in these alerts. This ensures that the proper people are aware of any potential compromise in order to mitigate the risk in a timely fashion.	
3	**Ensure That 'Notify about alerts with the following severity' is Set to 'High' (Automated)** Enables emailing security alerts to the subscription owner or other designated security contact.	Enabling security alert emails ensures that security alert emails are received from Microsoft. This ensures that the right people are aware of any potential security issues and are able to mitigate the risk.	
4	**Ensure that Microsoft Defender for Cloud Apps integration with Microsoft Defender for Cloud is Selected (Manual)** This integration setting enables Microsoft Defender for Cloud Apps (formerly 'Microsoft Cloud App Security' or 'MCAS' - see additional info) to communicate with Microsoft Defender for Cloud.	Microsoft Defender for Cloud offers an additional layer of protection by using Azure Resource Manager events, which is considered to be the control plane for Azure. By analyzing the Azure Resource Manager records, Microsoft Defender for Cloud detects unusual or potentially harmful operations in the Azure subscription environment. Several of the preceding analytics are powered by Microsoft Defender for Cloud Apps. To benefit from these analytics, subscription must have a Cloud App Security license. Microsoft Defender for Cloud Apps works only with Standard Tier subscriptions.	
5	**Ensure that Microsoft Defender for Endpoint integration with Microsoft Defender for Cloud is selected (Manual)** This integration setting enables Microsoft Defender for Endpoint (formerly 'Advanced Threat Protection' or 'ATP' or 'WDATP' - see additional info) to communicate with Microsoft Defender for Cloud. IMPORTANT: When enabling integration between DfE & DfC it needs to be taken into account that this will have some side effects that may be undesirable. 1. For server 2019 & above if defender is installed (default for these server SKU's) this will trigger a deployment of the new unified agent and link to any of the extended configuration in the Defender portal. 2. If the new unified agent is required for server SKU's of Win 2016 or Linux and lower there is additional integration that needs to be switched on and agents need to be aligned.	Microsoft Defender for Endpoint integration brings comprehensive Endpoint Detection and Response (EDR) capabilities within Microsoft Defender for Cloud. This integration helps to spot abnormalities, as well as detect and respond to advanced attacks on endpoints monitored by Microsoft Defender for Cloud. MDE works only with Standard Tier subscriptions.	
6	**Ensure That Microsoft Defender for IoT Hub Is Set To 'On' (Manual)** Microsoft Defender for IoT acts as a central security hub for IoT devices within your organization.	IoT devices are very rarely patched and can be potential attack vectors for enterprise networks. Updating their network configuration to use a central security hub allows for detection of these breaches.	
7	**Ensure that 'Secure transfer required' is set to 'Enabled' (Automated)** Enable data encryption in transit.	The secure transfer option enhances the security of a storage account by only allowing requests to the storage account by a secure connection. For example, when calling REST APIs to access storage accounts, the connection must use HTTPS. Any requests using HTTP will be rejected when 'secure transfer required' is enabled. When using the Azure files service, connection without encryption will fail, including scenarios using SMB 2.1, SMB 3.0 without encryption, and some flavors of the Linux SMB client. Because Azure storage doesn’t support HTTPS for custom domain names, this option is not applied when using a custom domain name.	

Sr. No	Control	Risk	Auditors Remarks
8	**Ensure that 'Enable Infrastructure Encryption' for Each Storage Account in Azure Storage is Set to 'enabled' (Automated)** Enabling encryption at the hardware level on top of the default software encryption for Storage Accounts accessing Azure storage solutions.	Azure Storage automatically encrypts all data in a storage account at the network level using 256-bit AES encryption, which is one of the strongest, FIPS 140-2-compliant block ciphers available. Customers who require higher levels of assurance that their data is secure can also enable 256-bit AES encryption at the Azure Storage infrastructure level for double encryption. Double encryption of Azure Storage data protects against a scenario where one of the encryption algorithms or keys may be compromised. Similarly, data is encrypted even before network transmission and in all backups. In this scenario, the additional layer of encryption continues to protect your data. For the most secure implementation of key based encryption, it is recommended to use a Customer Managed asymmetric RSA 2048 Key in Azure Key Vault.	
9	**Ensure that 'Enable key rotation reminders' is enabled for each Storage Account (Manual)** Access Keys authenticate application access requests to data contained in Storage Accounts. A periodic rotation of these keys is recommended to ensure that potentially compromised keys cannot result in a long-term exploitable credential. The "Rotation Reminder" is an automatic reminder feature for a manual procedure.	Reminders such as those generated by this recommendation will help maintain a regular and healthy cadence for activities which improve the overall efficacy of a security program. Cryptographic key rotation periods will vary depending on your organization's security requirements and the type of data which is being stored in the Storage Account. For example, PCI DSS mandates that cryptographic keys be replaced or rotated 'regularly,' and advises that keys for static data stores be rotated every 'few months.' For the purposes of this recommendation, 90 days will prescribed for the reminder. Review and adjustment of the 90 day period is recommended, and may even be necessary. Your organization's security requirements should dictate the appropriate setting.	
10	**Ensure that Storage Account Access Keys are Periodically Regenerated (Manual)** For increased security, regenerate storage account access keys periodically.	When a storage account is created, Azure generates two 512-bit storage access keys which are used for authentication when the storage account is accessed. Rotating these keys periodically ensures that any inadvertent access or exposure does not result from the compromise of these keys. Cryptographic key rotation periods will vary depending on your organization's security requirements and the type of data which is being stored in the Storage Account. For example, PCI DSS mandates that cryptographic keys be replaced or rotated 'regularly,' and advises that keys for static data stores be rotated every 'few months.' For the purposes of this recommendation, 90 days will prescribed for the reminder. Review and adjustment of the 90 day period is recommended, and may even be necessary. Your organization's	
11	**Ensure Storage Logging is Enabled for Queue Service for 'Read', 'Write', and 'Delete' requests (Automated)** The Storage Queue service stores messages that may be read by any client who has access to the storage account. A queue can contain an unlimited number of messages, each of which can be up to 64KB in size using version 2011-08-18 or newer. Storage Logging happens server-side and allows details for both successful and failed requests to be recorded in the storage account. These logs allow users to see the details of read, write, and delete operations against the queues. Storage Logging log entries contain the following information about individual requests: Timing information such as start time, end-to-end latency, and server latency, authentication details, concurrency information, and the sizes of the request and response messages.	Storage Analytics logs contain detailed information about successful and failed requests to a storage service. This information can be used to monitor individual requests and to diagnose issues with a storage service. Requests are logged on a best-effort basis. Storage Analytics logging is not enabled by default for your storage account.	

Sr. No	Control	Risk	Auditors Remarks
12	**Ensure that Shared Access Signature Tokens Expire Within an Hour (Manual)** Expire shared access signature tokens within an hour.	A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. A shared access signature can be provided to clients who should not be trusted with the storage account key but for whom it may be necessary to delegate access to certain storage account resources. Providing a shared access signature URI to these clients allows them access to a resource for a specified period of time. This time should be set as low as possible and preferably no longer than an hour.	
13	**Ensure that 'Public access level' is disabled for storage accounts with blob containers (Automated)** Disallowing public access for a storage account overrides the public access settings for individual containers in that storage account.	The default configuration for a storage account permits a user with appropriate permissions to configure public (anonymous) access to containers and blobs in a storage account. Keep in mind that public access to a container is always turned off by default and must be explicitly configured to permit anonymous requests. It grants read-only access to these resources without sharing the account key, and without requiring a shared access signature. It is recommended not to provide anonymous access to blob containers until, and unless, it is strongly desired. A shared access signature token or Azure AD RBAC should be used for providing controlled and timed access to blob containers. If no anonymous access is needed on any container in the storage account, it's recommended to set allowBlobPublicAccess false at the account level, which forbids any container to accept anonymous access in the future.	
14	**Ensure Default Network Access Rule for Storage Accounts is Set to Deny (Automated)** Restricting default network access helps to provide a new layer of security, since storage accounts accept connections from clients on any network. To limit access to selected networks, the default action must be changed.	Storage accounts should be configured to deny access to traffic from all networks (including internet traffic). Access can be granted to traffic from specific Azure Virtual networks, allowing a secure network boundary for specific applications to be built. Access can also be granted to public internet IP address ranges to enable connections from specific internet or on-premises clients. When network rules are configured, only applications from allowed networks can access a storage account. When calling from an allowed network, applications continue to require proper authorization (a valid access key or SAS token) to access the storage account.	
15	**Ensure 'Allow Azure services on the trusted services list to access this storage account' is Enabled for Storage Account Access (Automated)** Some Azure services that interact with storage accounts operate from networks that can't be granted access through network rules. To help this type of service work as intended, allow the set of trusted Azure services to bypass the network rules. These services will then use strong authentication to access the storage account. If the Allow trusted Azure services exception is enabled, the following services are granted access to the storage account: Azure Backup, Azure Site Recovery, Azure DevTest Labs, Azure Event Grid, Azure Event Hubs, Azure Networking, Azure Monitor, and Azure SQL Data Warehouse (when registered in the subscription).	Turning on firewall rules for storage account will block access to incoming requests for data, including from other Azure services. We can re-enable this functionality by enabling "Trusted Azure Services" through networking exceptions.	
16	**Ensure Private Endpoints are used to access Storage Accounts (Automated)** Use private endpoints for your Azure Storage accounts to allow clients and services to securely access data located over a network via an encrypted Private Link. To do this, the private endpoint uses an IP address from the VNet for each service. Network traffic between disparate services securely traverses encrypted over the VNet. This VNet can also link addressing space, extending your network and accessing resources on it. Similarly, it can be a tunnel through public networks to connect remote infrastructures together. This creates further security through segmenting network traffic and preventing outside sources from accessing it.	Securing traffic between services through encryption protects the data from easy interception and reading.	

Sr. No	Control	Risk	Auditors Remarks
17	**Ensure Soft Delete is Enabled for Azure Containers and Blob Storage (Automated)** The Azure Storage blobs contain data like ePHI or Financial, which can be secret or personal. Data that is erroneously modified or deleted by an application or other storage account user will cause data loss or unavailability. It is recommended that both Azure Containers with attached Blob Storage and standalone containers with Blob Storage be made recoverable by enabling the soft delete configuration. This is to save and recover data when blobs or blob snapshots are deleted.	Containers and Blob Storage data can be incorrectly deleted. An attacker/malicious user may do this deliberately in order to cause disruption. Deleting an Azure Storage blob causes immediate data loss. Enabling this configuration for Azure storage ensures that even if blobs/data were deleted from the storage account, Blobs/data objects are recoverable for a particular time which is set in the "Retention policies," ranging from 7 days to 365 days.	
18	**Ensure Storage for Critical Data are Encrypted with Customer Managed Keys (Manual)** Enable sensitive data encryption at rest using Customer Managed Keys rather than Microsoft Managed keys.	By default, data in the storage account is encrypted using Microsoft Managed Keys at rest. All Azure Storage resources are encrypted, including blobs, disks, files, queues, and tables. All object metadata is also encrypted. If you want to control and manage this encryption key yourself, however, you can specify a customer-managed key. That key is used to protect and control access to the key that encrypts your data. You can also choose to automatically update the key version used for Azure Storage encryption whenever a new version is available in the associated Key Vault.	
19	**Ensure Storage logging is Enabled for Blob Service for 'Read', 'Write', and 'Delete' requests (Automated)** The Storage Blob service provides scalable, cost-efficient object storage in the cloud. Storage Logging happens server-side and allows details for both successful and failed requests to be recorded in the storage account. These logs allow users to see the details of read, write, and delete operations against the blobs. Storage Logging log entries contain the following information about individual requests: timing information such as start time, end-to-end latency, and server latency; authentication details; concurrency information; and the sizes of the request and response messages.	Storage Analytics logs contain detailed information about successful and failed requests to a storage service. This information can be used to monitor each individual request to a storage service for increased security or diagnostics. Requests are logged on a best-effort basis. Storage Analytics logging is not enabled by default for your storage account.	
20	**Ensure Storage Logging is Enabled for Table Service for 'Read', 'Write', and 'Delete' Requests (Automated)** Azure Table storage is a service that stores structured NoSQL data in the cloud, providing a key/attribute store with a schema-less design. Storage Logging happens server-side and allows details for both successful and failed requests to be recorded in the storage account. These logs allow users to see the details of read, write, and delete operations against the tables. Storage Logging log entries contain the following information about individual requests: timing information such as start time, end-to-end latency, and server latency; authentication details; concurrency information; and the sizes of the request and response messages.	Storage Analytics logs contain detailed information about successful and failed requests to a storage service. This information can be used to monitor each individual request to a storage service for increased security or diagnostics. Requests are logged on a best-effort basis. Storage Analytics logging is not enabled by default for your storage account.	
21	**Ensure the "Minimum TLS version" for storage accounts is set to "Version 1.2" (Automated)** In some cases, Azure Storage sets the minimum TLS version to be version 1.0 by default. TLS 1.0 is a legacy version and has known vulnerabilities. This minimum TLS version can be configured to be later protocols such as TLS 1.2.	TLS 1.0 has known vulnerabilities and has been replaced by later versions of the TLS protocol. Continued use of this legacy protocol affects the security of data in transit.	
22	**Ensure that 'Auditing' is set to 'On' (Automated)** Enable auditing on SQL Servers.	The Azure platform allows a SQL server to be created as a service. Enabling auditing at the server level ensures that all existing and newly created databases on the SQL server instance are audited. Auditing policy applied on the SQL database does not override auditing policy and settings applied on the particular SQL server where the database is hosted. Auditing tracks database events and writes them to an audit log in the Azure storage account. It also helps to maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that could indicate business concerns or suspected security violations.	

Sr. No	Control	Risk	Auditors Remarks
23	**Ensure no Azure SQL Databases allow ingress from 0.0.0.0/0 (ANY IP) (Automated)** Ensure that no SQL Databases allow ingress from 0.0.0.0/0 (ANY IP).	Azure SQL Server includes a firewall to block access to unauthorized connections. More granular IP addresses can be defined by referencing the range of addresses available from specific datacenters. By default, for a SQL server, a Firewall exists with StartIp of 0.0.0.0 and EndIP of 0.0.0.0 allowing access to all the Azure services. Additionally, a custom rule can be set up with StartIp of 0.0.0.0 and EndIP of 255.255.255.255 allowing access from ANY IP over the Internet. In order to reduce the potential attack surface for a SQL server, firewall rules should be defined with more granular IP addresses by referencing the range of addresses available from specific datacenters.	
24	**Ensure SQL server's Transparent Data Encryption (TDE) protector is encrypted with Customer-managed key (Automated)** Transparent Data Encryption (TDE) with Customer-managed key support provides increased transparency and control over the TDE Protector, increased security with an HSM-backed external service, and promotion of separation of duties. With TDE, data is encrypted at rest with a symmetric key (called the database encryption key) stored in the database or data warehouse distribution. To protect this data encryption key (DEK) in the past, only a certificate that the Azure SQL Service managed could be used. Now, with Customer-managed key support for TDE, the DEK	Customer-managed key support for Transparent Data Encryption (TDE) allows user control of TDE encryption keys and restricts who can access them and when. Azure Key Vault, Azure's cloud-based external key management system, is the first key management service where TDE has integrated support for Customer-managed keys. With Customer-managed key support, the database encryption key is protected by an asymmetric key stored in the Key Vault. The asymmetric key is set at the server level and inherited by all databases under that server.	
25	**Ensure that Azure Active Directory Admin is Configured for SQL Servers (Automated)** Use Azure Active Directory Authentication for authentication with SQL Database to manage credentials in a single place.	Azure Active Directory authentication is a mechanism to connect to Microsoft Azure SQL Database and SQL Data Warehouse by using identities in Azure Active Directory (Azure AD). With Azure AD authentication, identities of database users and other Microsoft services can be managed in one central location. Central ID management provides a single place to manage database users and simplifies permission management. • It provides an alternative to SQL Server authentication. • Helps stop the proliferation of user identities across database servers. • Allows password rotation in a single place. • Customers can manage database permissions using external (AAD) groups. • It can eliminate storing passwords by enabling integrated Windows authentication and other forms	
26	**Ensure that 'Data encryption' is set to 'On' on a SQL Database (Automated)** Enable Transparent Data Encryption on every SQL server.	Azure SQL Database transparent data encryption helps protect against the threat of malicious activity by performing real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application.	
27	**Ensure that 'Auditing' Retention is 'greater than 90 days' (Automated)** SQL Server Audit Retention should be configured to be greater than 90 days.	Audit Logs can be used to check for anomalies and give insight into suspected breaches or misuse of information and access.	
28	**Ensure that Microsoft Defender for SQL is set to 'On' for critical SQL Servers (Automated)** Enable "Microsoft Defender for SQL" on critical SQL Servers.	Microsoft Defender for SQL is a unified package for advanced SQL security capabilities. Microsoft Defender is available for Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics. It includes functionality for discovering and classifying sensitive data, surfacing and mitigating potential database vulnerabilities, and detecting anomalous activities that could indicate a threat to your database. It provides a single go-to location for enabling and managing these capabilities.	

Sr. No	Control	Risk	Auditors Remarks
29	**Ensure that Vulnerability Assessment (VA) is enabled on a SQL server by setting a Storage Account (Automated)** Enable Vulnerability Assessment (VA) service scans for critical SQL servers and corresponding SQL databases.	Enabling Microsoft Defender for SQL server does not enables Vulnerability Assessment capability for individual SQL databases unless storage account is set to store the scanning data and reports. The Vulnerability Assessment service scans databases for known security vulnerabilities and highlights deviations from best practices, such as misconfigurations, excessive permissions, and unprotected sensitive data. Results of the scan include actionable steps to resolve each issue and provide customized remediation scripts where applicable. Additionally, an assessment report can be customized by setting an acceptable baseline for permission configurations, feature configurations, and database settings.	
30	**Ensure that Vulnerability Assessment (VA) setting 'Periodic recurring scans' is set to 'on' for each SQL server (Automated)** Enable Vulnerability Assessment (VA) Periodic recurring scans for critical SQL servers and corresponding SQL databases.	VA setting 'Periodic recurring scans' schedules periodic (weekly) vulnerability scanning for the SQL server and corresponding Databases. Periodic and regular vulnerability scanning provides risk visibility based on updated known vulnerability signatures and best practices.	
31	**Ensure that Vulnerability Assessment (VA) setting 'Send scan reports to' is configured for a SQL server (Automated)** Configure 'Send scan reports to' with email addresses of concerned data owners/stakeholders for a critical SQL servers.	Vulnerability Assessment (VA) scan reports and alerts will be sent to email addresses configured at 'Send scan reports to'. This may help in reducing time required for identifying risks and taking corrective measures.	
32	**Ensure that Vulnerability Assessment (VA) setting 'Also send email notifications to admins and subscription owners' is set for each SQL Server (Automated)** Enable Vulnerability Assessment (VA) setting 'Also send email notifications to admins and subscription owners'.	VA scan reports and alerts will be sent to admins and subscription owners by enabling setting 'Also send email notifications to admins and subscription owners'. This may help in reducing time required for identifying risks and taking corrective measures.	
33	**Ensure 'Enforce SSL connection' is set to 'ENABLED' for PostgreSQL Database Server (Automated)** Enable SSL connection on PostgreSQL Servers.	SSL connectivity helps to provide a new layer of security by connecting database server to client applications using Secure Sockets Layer (SSL). Enforcing SSL connections between database server and client applications helps protect against "man in the middle" attacks by encrypting the data stream between the server and application.	
34	**Ensure Server Parameter 'log_checkpoints' is set to 'ON' for PostgreSQL Database Server (Automated)** Enable log_checkpoints on PostgreSQL Servers.	Enabling log_checkpoints helps the PostgreSQL Database to Log each checkpoint in turn generates query and error logs. However, access to transaction logs is not supported. Query and error logs can be used to identify, troubleshoot, and repair configuration errors and sub-optimal performance.	
35	**Ensure server parameter 'log_connections' is set to 'ON' for PostgreSQL Database Server (Automated)** Enable log_connections on PostgreSQL Servers.	Enabling log_connections helps PostgreSQL Database to log attempted connection to the server, as well as successful completion of client authentication. Log data can be used to identify, troubleshoot, and repair configuration errors and suboptimal performance.	
36	**Ensure server parameter 'log_disconnections' is set to 'ON' for PostgreSQL Database Server (Automated)** Enable log_disconnections on PostgreSQL Servers.	Enabling log_disconnections helps PostgreSQL Database to Logs end of a session, including duration, which in turn generates query and error logs. Query and error logs can be used to identify, troubleshoot, and repair configuration errors and sub-optimal performance.	
37	**Ensure server parameter 'connection_throttling' is set to 'ON' for PostgreSQL Database Server (Automated)** Enable connection_throttling on PostgreSQL Servers.	Enabling connection_throttling helps the PostgreSQL Database to Set the verbosity of logged messages. This in turn generates query and error logs with respect to concurrent connections that could lead to a successful Denial of Service (DoS) attack by exhausting connection resources. A system can also fail or be degraded by an overload of legitimate users. Query and error logs can be used to identify, troubleshoot, and repair configuration errors and sub-optimal performance.	

Sr. No	Control	Risk	Auditors Remarks
38	**Ensure Server Parameter 'log_retention_days' is greater than 3 days for PostgreSQL Database Server (Automated)** Ensure log_retention_days on PostgreSQL Servers is set to an appropriate value.	Configuring log_retention_days determines the duration in days that Azure Database for PostgreSQL retains log files. Query and error logs can be used to identify, troubleshoot, and repair configuration errors and sub-optimal performance.	
39	**Ensure 'Allow access to Azure services' for PostgreSQL Database Server is disabled (Automated)** Disable access from Azure services to PostgreSQL Database Server.	If access from Azure services is enabled, the server's firewall will accept connections from all Azure resources, including resources not in your subscription. This is usually not a desired configuration. Instead, set up firewall rules to allow access from specific network ranges or VNET rules to allow access from specific virtual networks.	
40	**Ensure 'Infrastructure double encryption' for PostgreSQL Database Server is 'Enabled' (Automated)** Azure Database for PostgreSQL servers should be created with 'infrastructure double encryption' enabled.	If Double Encryption is enabled, another layer of encryption is implemented at the hardware level before the storage or network level. Information will be encrypted before it is even accessed, preventing both interception of data in motion if the network layer encryption is broken and data at rest in system resources such as memory or processor cache. Encryption will also be in place for any backups taken of the database, so the key will secure access the data in all forms. For the most secure implementation of key based encryption, it is recommended to use a Customer Managed asymmetric RSA 2048 Key in Azure Key Vault.	
41	**Ensure 'Enforce SSL connection' is set to 'Enabled' for Standard MySQL Database Server (Automated)** Enable SSL connection on MYSQL Servers.	SSL connectivity helps to provide a new layer of security by connecting database server to client applications using Secure Sockets Layer (SSL). Enforcing SSL connections between database server and client applications helps protect against "man in the middle" attacks by encrypting the data stream between the server and application.	
42	**Ensure 'TLS Version' is set to 'TLSV1.2' for MySQL flexible Database Server (Automated)** Ensure TLS version on MySQL flexible servers is set to the default value.	TLS connectivity helps to provide a new layer of security by connecting database server to client applications using Transport Layer Security (TLS). Enforcing TLS connections between database server and client applications helps protect against "man in the middle" attacks by encrypting the data stream between the server and application.	
43	**Ensure server parameter 'audit_log_enabled' is set to 'ON' for MySQL Database Server (Manual)** Enable audit_log_enabled on MySQL Servers.	Enabling audit_log_enabled helps MySQL Database to log items such as connection attempts to the server, DDL/DML access, and more. Log data can be used to identify, troubleshoot, and repair configuration errors and suboptimal performance.	
44	**Ensure server parameter 'audit_log_events' has 'CONNECTION' set for MySQL Database Server (Manual)** Set audit_log_enabled to include CONNECTION on MySQL Servers.	Enabling CONNECTION helps MySQL Database to log items such as successful and failed connection attempts to the server. Log data can be used to identify, troubleshoot, and repair configuration errors and suboptimal performance.	
45	**Ensure That 'Firewalls & Networks' Is Limited to Use Selected Networks Instead of All Networks (Automated)** Limiting your Cosmos DB to only communicate on whitelisted networks lowers its attack footprint.	Selecting certain networks for your Cosmos DB to communicate restricts the number of networks including the internet that can interact with what is stored within the database.	
46	**Ensure That Private Endpoints Are Used Where Possible (Manual)** Private endpoints limit network traffic to approved sources.	For sensitive data, private endpoints allow granular control of which services can communicate with Cosmos DB and ensure that this network traffic is private. You set this up on a case by case basis for each service you wish to be connected.	
47	**Use Azure Active Directory (AAD) Client Authentication and Azure RBAC where possible. (Manual)** Cosmos DB can use tokens or AAD for client authentication which in turn will use Azure RBAC for authorization. Using AAD is significantly more secure because AAD handles the credentials and allows for MFA and centralized management, and the Azure RBAC better integrated with the rest of Azure.	AAD client authentication is considerably more secure than token-based authentication because the tokens must be persistent at the client. AAD does not require this.	

Sr. No	Control	Risk	Auditors Remarks
48	**Ensure that a 'Diagnostic Setting' exists (Manual)** Enable Diagnostic settings for exporting activity logs. Diagnostic settings are available for each individual resource within a subscription. Settings should be configured for all appropriate resources for your environment.	A diagnostic setting controls how a diagnostic log is exported. By default, logs are retained only for 90 days. Diagnostic settings should be defined so that logs can be exported and stored for a longer duration in order to analyze security activities within an Azure subscription.	
49	**Ensure Diagnostic Setting captures appropriate categories (Automated)** **Prerequisite:** A Diagnostic Setting must exist. If a Diagnostic Setting does not exist, the navigation and options within this recommendation will not be available. Please review the recommendation at the beginning of this subsection titled: "Ensure that a 'Diagnostic Setting' exists." The diagnostic setting should be configured to log the appropriate activities from the control/management plane.	A diagnostic setting controls how the diagnostic log is exported. Capturing the diagnostic setting categories for appropriate control/management plane activities allows proper alerting.	
50	**Ensure the Storage Container Storing the Activity Logs is not Publicly Accessible (Automated)** The storage account container containing the activity log export should not be publicly accessible.	Allowing public access to activity log content may aid an adversary in identifying weaknesses in the affected account's use or configuration.	