**Current Date:** Feb 18, 2025

**I. SQL Server Security Assessment**

A. Authentication & Authorization

1. Strong Password Policies:

\* Verify password complexity, length, and history requirements are enforced for SQL Logins.

\* Check if 'CHECK\_POLICY' and 'CHECK\_EXPIRATION' are enabled for SQL Logins.

\* Audit for weak or default passwords, especially for privileged accounts.

2. Windows Authentication vs. SQL Server Authentication:

\* Assess the use of Windows Authentication (preferred) versus SQL Server Authentication.

\* If SQL Server Authentication is used, ensure it's justified and that logins are managed securely.

\* Scrutinize the usage of the 'sa' (System Administrator) login:

\* Is it disabled if Windows Authentication is exclusively used?

\* If enabled, is it renamed and protected with an exceptionally strong password?

\* Is its usage audited?

3. Principle of Least Privilege:

\* Review server-level roles (e.g., sysadmin, serveradmin, securityadmin). Are memberships minimized and justified?

\* Audit database-level roles and permissions. Do users and applications have only the necessary permissions for their tasks?

\* Check for excessive use of db\_owner or dbo user permissions.

\* Review permissions granted to the PUBLIC role.

4. Orphaned Users:

\* Identify and remediate orphaned database users (users without a corresponding login).

5. Guest User Access:

\* Verify the guest user is disabled in all user databases unless explicitly required and approved.

\* Check permissions granted to the guest user if enabled.

6. Contained Databases:

\* If used, assess the security implications and ensure contained database users are managed appropriately.

7. Application Role Security:

\* If application roles are used, review their passwords and activation procedures.

8. SQL Injection Prevention:

\* While primarily an application-level concern, review stored procedures and dynamic SQL usage for vulnerabilities.

\* Ensure developers are trained on parameterized queries and input validation.

B. Network Security & Connectivity

1. SQL Server Browser Service:

\* Verify if the SQL Server Browser service is disabled if not required (e.g., using static ports, single instance).

2. Port Configuration:

\* Check if SQL Server is running on a non-default port if security through obscurity is part of the strategy.

\* Ensure firewall rules (Windows Firewall and network firewalls) only allow access to the SQL Server port from authorized sources/subnets.

3. Protocols:

\* Disable unused protocols (e.g., VIA, Named Pipes if not necessary).

\* Ensure TCP/IP is configured securely.

4. Force Encryption / SSL/TLS:

\* Verify if connections to SQL Server are encrypted using SSL/TLS ('Force Encryption' server property or client-initiated).

\* Check the validity and strength of certificates used for SQL Server encryption (see Section D).

5. Endpoint Security:

\* Review permissions on SQL Server endpoints (e.g., Database Mirroring, Service Broker).

\* Ensure default trace is enabled or a suitable server-side trace/extended event session is capturing security-relevant events.

C. Surface Area Reduction & Configuration Management

1. Feature Disablement:

\* Disable unused SQL Server features (e.g., xp\_cmdshell, Database Mail, CLR integration, OLE Automation, Ad Hoc Distributed Queries) unless explicitly required, approved, and secured.

\* Review the configuration of sp\_configure options for security implications.

2. Patch Management:

\* Verify SQL Server instances are regularly patched with the latest cumulative updates (CUs) and security updates.

\* Document the patching process and history.

3. Service Account Privileges:

\* Ensure SQL Server service accounts (Engine, Agent, Browser, etc.) run with the least necessary privileges.

\* Avoid using domain administrator or local administrator accounts for SQL Server services.

\* Use Managed Service Accounts (MSAs) or Group Managed Service Accounts (gMSAs) where possible.

4. SQL Server Agent Security:

\* Secure Agent proxies and ensure jobs are owned by appropriate, non-sysadmin principals where possible.

\* Review job step commands for potential security risks (e.g., use of xp\_cmdshell).

5. Dedicated Admin Connection (DAC):

\* Ensure the DAC is enabled and its usage is understood for emergency access.

6. Server Configuration Options:

\* Audit settings like 'Hide Instance', 'Remote Admin Connections'.

7. Error Log and Security Log Review:

\* Check that SQL Server error logs are regularly reviewed for suspicious activity.

\* Verify that login auditing (successful and failed) is configured appropriately.

D. Linked Servers

1. Necessity and Justification:

\* Verify each linked server is necessary and documented with a business justification.

\* Remove unused or unauthorized linked servers.

2. Authentication Methods:

\* Review the authentication method used for each linked server.

\* Avoid using "Be made using the login's current security context" with highly privileged accounts.

\* Prefer "Be made using this security context" with a dedicated, least-privilege SQL login or mapped Windows login where possible.

\* Scrutinize linked servers configured with sysadmin privileges.

3. Provider String Security:

\* Review provider strings for any embedded credentials or insecure options.

4. RPC and RPC Out Configuration:

\* Ensure Remote Procedure Call (RPC) and RPC Out options are enabled only if necessary for the linked server's functionality.

5. Data Access and Collation:

\* Understand what data can be accessed via the linked server and ensure it aligns with security policies.

\* Be aware of collation compatibility issues that might indirectly lead to information disclosure or errors.

E. Certificates Management

1. Purpose of Certificates:

\* Identify all certificates used by SQL Server (e.g., for connection encryption, Transparent Data Encryption (TDE), database mirroring, Always On Availability Groups, Service Broker).

2. Certificate Validity:

\* Verify certificates are within their validity period (not expired).

\* Ensure certificates are issued by a trusted Certificate Authority (CA) or are properly managed self-signed certificates (with appropriate risk assessment).

3. Certificate Strength:

\* Check for strong algorithms (e.g., SHA-256 or higher for signature, appropriate key length for RSA/ECC).

\* Avoid using certificates with known weak algorithms (e.g., MD5, SHA-1).

4. Private Key Protection:

\* Ensure private keys for certificates are adequately protected.

\* If using TDE, ensure the Database Master Key (DMK) and Service Master Key (SMK) are backed up and protected. The certificate used for TDE should also be backed up.

5. Certificate Chains:

\* Verify the entire certificate chain is valid and trusted by clients and servers.

6. Certificate Revocation:

\* Ensure a process is in place to revoke compromised certificates and that SQL Server/clients check for revocation (CRL or OCSP).

7. Regular Review and Renewal Process:

\* Verify there's a documented process for reviewing and renewing certificates before they expire.

F. Data Protection

1. Transparent Data Encryption (TDE):

\* Assess if TDE is implemented for databases containing sensitive data (data at rest encryption).

\* Verify TDE certificate and master key backups are securely stored and regularly tested.

2. Always Encrypted:

\* Determine if Always Encrypted is used for specific sensitive columns, providing client-side encryption.

\* Review key management for column master keys and column encryption keys.

3. Backup Encryption:

\* Verify if database backups are encrypted.

\* Ensure backup encryption certificates/keys are managed securely and backed up.

4. Data Masking:

\* Check if Dynamic Data Masking or other techniques are used to obfuscate sensitive data for non-privileged users.

5. Row-Level Security (RLS):

\* Assess if RLS is implemented to restrict data access at the row level based on user context.

6. Data Classification:

\* Verify if sensitive data has been identified and classified within the databases.

\* SQL Server Data Classification features can assist here.

G. Auditing & Logging

1. SQL Server Audit:

\* Verify SQL Server Audit is configured to capture critical security events (e.g., logins, failed logins, schema changes, permission changes, data access to sensitive tables).

\* Ensure audit logs are written to a secure location, preferably a separate, hardened server or Security Information and Event Management (SIEM) system.

\* Review audit specifications for completeness.

2. Login Auditing:

\* Ensure login auditing (successful and/or failed) is enabled at the server level.

3. C2 Auditing / Common Criteria Compliance:

\* If required for compliance, verify if these are configured (though SQL Server Audit is generally preferred over C2).

4. Default Trace / Extended Events:

\* Ensure default trace or a comprehensive Extended Events session is running to capture important system and security events.

5. Log Retention and Review:

\* Verify a policy is in place for audit log retention.

\* Ensure logs are regularly reviewed by security personnel.

H. Object Security

1. Ownership Chaining:

\* Understand and review database object ownership chains to prevent unintended privilege escalation.

2. Schema Security:

\* Ensure objects are appropriately organized into schemas, and permissions are granted at the schema level where possible, rather than individual object level.

\* Prevent users from creating objects in schemas owned by privileged users (e.g., dbo).

3. Stored Procedure and Function Security:

\* Review permissions on stored procedures and functions.

\* Check for use of EXECUTE AS clause and its security implications (context switching).

4. Triggers:

\* Review DDL and DML triggers for any unintended security consequences.

I. Backup and Recovery

1. Regular Backups:

\* Verify all databases have a regular, tested backup schedule (full, differential, transaction log).

2. Backup Storage Security:

\* Ensure backup files are stored in a secure location with restricted access.

\* Consider encrypting backups (see F.3).

3. Restore Procedures:

\* Verify documented and tested restore procedures are in place.

\* Regularly test database restores.

4. Service Master Key (SMK) and Database Master Key (DMK) Backup:

\* Ensure SMKs and DMKs are backed up and stored securely, separate from the server.

**II. Virtual Machine (VM) Security Assessment (Host for SQL Server)**

A. Host Operating System Security

1. Patch Management:

\* Verify the VM's operating system is regularly patched with the latest security updates.

2. User Account Control (UAC) (Windows):

\* Ensure UAC is enabled and configured appropriately.

3. Local Administrator Accounts:

\* Restrict membership in the local Administrators group.

\* Ensure strong, unique passwords for local administrator accounts.

4. Unnecessary Services and Features:

\* Disable any unnecessary OS services and features to reduce the attack surface.

5. File System Permissions (NTFS):

\* Ensure appropriate NTFS permissions are set on SQL Server installation directories, data file directories, log file directories, and backup directories to restrict access to SQL Server service accounts and authorized administrators.

6. Antivirus/Anti-malware:

\* Verify antivirus/anti-malware software is installed, up-to-date, and configured with appropriate exclusions for SQL Server processes and files to avoid performance issues and corruption.

7. Host-based Firewall:

\* Ensure a host-based firewall (e.g., Windows Firewall) is enabled and configured to allow only necessary traffic to/from the SQL Server VM.

8. Remote Access Security (RDP/SSH):

\* Secure remote access protocols (e.g., use strong authentication, Network Level Authentication for RDP, restrict RDP/SSH access to specific IPs/jump boxes).

\* Audit remote logins.

9. Time Synchronization:

\* Ensure the VM's time is synchronized with a reliable time source (critical for Kerberos authentication and log correlation).

B. Hypervisor Level Security (e.g., VMware vSphere, Microsoft Hyper-V)

While a full hypervisor audit is extensive, key areas impacting the SQL Server VM include:

1. Hypervisor Patching:

\* Ensure the hypervisor itself is patched and up-to-date.

2. VM Isolation:

\* Verify proper network segmentation and isolation between VMs, especially between production SQL Servers and less trusted VMs.

3. Virtual Network Configuration:

\* Review virtual switch configurations for security best practices.

4. VM Snapshot Management:

\* Assess policies for VM snapshot creation, retention, and deletion (snapshots are not backups and can have performance/security implications if mismanaged).

5. Access Control to Hypervisor Management:

\* Ensure strong access controls and auditing for the hypervisor management interface.

6. VM Tools:

\* Ensure VM tools (e.g., VMware Tools, Hyper-V Integration Services) are installed and kept up-to-date on the SQL Server VM.

C. VM Configuration & Resource Management

1. Resource Allocation:

\* Ensure the VM has adequate and appropriately configured CPU, memory, and disk resources for SQL Server performance and stability, but not excessively over-provisioned if it impacts licensing or density security.

2. Secure VM Template/Image:

\* If VMs are deployed from templates, ensure the template is hardened according to security best practices.

3. Physical Host Security (if applicable/known):

\* While often out of scope for a SQL-focused audit, if on-premises, be aware that physical security of the host servers is fundamental.

III. Operational Security & Documentation

1. Change Management:

\* Verify a formal change management process is in place for any modifications to the SQL Server or VM configuration.

2. Incident Response Plan:

\* Ensure an incident response plan is in place that covers SQL Server-related security incidents.

3. Disaster Recovery (DR) Plan:

\* Review DR plans for SQL Server, including RPO/RTO objectives and testing frequency.

\* Assess security of the DR site and data replication mechanisms.

4. Security Documentation:

\* Verify that security configurations, policies, and procedures related to SQL Server and its VMs are documented and kept up-to-date.

5. Regular Security Assessments:

\* Establish a schedule for periodic security assessments and penetration testing.

This list should provide a solid framework for your SQL Server/VM security assessment. Remember to tailor it to your specific environment, SQL Server versions, and any relevant compliance requirements (e.g., GDPR, HIPAA, PCI-DSS). Good luck with your audit!

Okay, this is a crucial task for maintaining a secure SQL Server estate. A thorough security assessment will help you identify and mitigate potential vulnerabilities. Here's a detailed and defined list of issues you should cover for your SQL Server / VM Security assessment, suitable for an audit.

This list is broken down into key areas:

**I. Virtual Machine (VM) and Host Security (Applicable to both Hyper-V & VMware)**

1. **Host Operating System Security:**
   * **Patch Management:**
     + Is the host OS (Windows Server, Hyper-V Server, ESXi) regularly patched with the latest security updates?
     + Is there a documented patch management process with an auditable trail?
     + Are firmware and device drivers on the host up to date?
   * **Minimization & Hardening:**
     + Is the host OS installed with the minimum required roles and features (e.g., Server Core for Windows Server if appropriate)?
     + Are unnecessary services, applications, and protocols disabled or removed from the host?
     + Are security baselines (e.g., Microsoft Security Baselines, VMware vSphere Security Configuration Guide) applied and enforced?
     + Is access to the host management interface restricted (e.g., dedicated management network, limited IP addresses)?
   * **Authentication & Access Control (Host):**
     + Are strong, unique passwords enforced for all host administrator accounts?
     + Is Multi-Factor Authentication (MFA) used for host administrator access where possible?
     + Is Role-Based Access Control (RBAC) implemented to limit administrative privileges based on job function (Principle of Least Privilege)?
     + Are default administrator accounts renamed or disabled?
     + Is access to the host console (physical or virtual) physically and logically secured?
     + Are host login attempts (successful and failed) logged and monitored?
     + Are Hyper-V Administrators group memberships tightly controlled?
     + For VMware, are ESXi Lockdown Modes appropriately configured (Normal, Strict)?
   * **Network Security (Host):**
     + Is the host protected by a firewall (host-based and network)?
     + Are only necessary ports open on the host?
     + Is the management network segmented from VM traffic and other networks?
     + For Hyper-V, is live migration traffic on a dedicated, secured network (consider IPSec)?
     + For VMware, is vMotion traffic on an isolated, non-routable network?
     + Are virtual switches configured securely (e.g., preventing MAC spoofing, promiscuous mode unless specifically required)?
   * **Storage Security (Host):**
     + Is storage for VM configuration files, virtual hard disks (VHDs/VMDKs), and snapshots secured with appropriate permissions?
     + Is BitLocker (for Hyper-V hosts) or VM encryption (for VMware) considered or implemented for host volumes storing sensitive VM data?
   * **Anti-Malware (Host):**
     + Is an enterprise-grade anti-malware solution installed and kept up-to-date on the host OS (especially for Windows-based hosts)?
     + Are appropriate exclusions configured for Hyper-V or VMware processes and storage locations to prevent performance issues and corruption?
   * **Host Auditing & Logging:**
     + Are security event logs on the host enabled, collected, and regularly reviewed?
     + Are logs forwarded to a central Security Information and Event Management (SIEM) system?
   * **Hyper-V Specific:**
     + Is Credential Guard used to protect derived domain credentials on the host?
     + Are code integrity policies (Device Guard) considered or implemented?
     + Are users managing Hyper-V hosts part of the "Hyper-V Administrators" group and not local administrators on the host OS unless necessary?
     + Are unknown VHDs prevented from being mounted?
     + Is nesting disabled in production unless explicitly required and secured?
   * **VMware Specific:**
     + Is vCenter Server secured (patched, hardened, access controls)?
     + Is SSH access to ESXi hosts disabled unless strictly necessary and controlled?
     + Are VIB acceptance levels configured appropriately?
     + Is SLP (Service Location Protocol) disabled if not needed?
     + Is CIM access controlled?
     + Are management session restrictions in place?
     + Is the Managed Object Browser (MOB) disabled on ESXi hosts in production?
2. **Virtual Machine (Guest OS) Security:**
   * **Template and Deployment Security:**
     + Are VM templates hardened and regularly updated with patches and security configurations before deployment?
     + Are Generation 2 VMs used for supported guest OSes (Hyper-V) for enhanced security features like Secure Boot?
     + Is Secure Boot enabled for guest VMs where supported?
   * **Guest OS Hardening:**
     + Is the guest OS (Windows, Linux) hardened according to security best practices and organizational baselines (e.g., CIS Benchmarks)?
     + Are unnecessary services, applications, and features removed or disabled within the guest OS?
   * **Patch Management (Guest OS):**
     + Is there a robust patch management process for guest operating systems?
     + Are critical security patches applied in a timely manner?
   * **Integration Services/VMware Tools:**
     + Are the latest versions of Hyper-V Integration Services or VMware Tools installed and kept up-to-date in all VMs?
     + Are features like copy/paste, device connection/disconnection from the VM console controlled as per security policy (e.g., isolation.tools.\* settings in VMware)?
   * **Network Security (Guest VM):**
     + Do VMs connect to the correct virtual switches with appropriate security settings and VLAN segmentation?
     + Is a host-based firewall (e.g., Windows Firewall, iptables) configured and active within each VM?
     + Is network traffic between VMs and to/from external networks appropriately restricted?
   * **Storage Security (Guest VM):**
     + Are virtual hard disks and snapshot files stored in secure locations with appropriate ACLs (even within the datastore/volume)?
     + Is guest OS-level disk encryption (e.g., BitLocker, LUKS) considered or implemented for VMs storing highly sensitive data?
   * **Anti-Malware (Guest VM):**
     + Is an enterprise-grade anti-malware solution installed, active, and up-to-date on all guest VMs running an OS susceptible to malware?
   * **VM Console Access:**
     + Is access to the VM console restricted?
     + For VMware, is the guest OS configured to lock when the last remote user disconnects?
     + Is the number of simultaneous connections to a VM console limited if applicable?

**II. SQL Server Security (Within the Guest VM)**

1. **Installation & Configuration:**
   * **Principle of Least Privilege for Service Accounts:**
     + Are dedicated, low-privilege domain or local accounts used for SQL Server services (Engine, Agent, Browser, etc.)? Avoid using Local System, Network Service, or highly privileged domain accounts.
     + Are service accounts granted only the necessary permissions on the OS and within Active Directory?
     + Are different service accounts used for different SQL Server instances?
     + Are service SIDs used where possible for granting permissions within SQL Server?
   * **Surface Area Reduction:**
     + Are unused SQL Server features and components disabled or uninstalled (e.g., Database Mail, Service Broker, CLR integration, xp\_cmdshell, OLE Automation, etc., if not explicitly needed)?
     + Are sample databases (e.g., AdventureWorks, Northwind) removed from production instances?
     + Is the SQL Server Browser service disabled if not required (e.g., if all connections use a specific port and instance name is not relied upon for discovery)?
   * **Network Configuration:**
     + Is SQL Server configured to listen on a non-default port if security policy dictates?
     + Are only necessary network protocols enabled (typically TCP/IP)? Are Named Pipes, Shared Memory, VIA disabled if not used?
     + Is "Hide instance" configured if it's a named instance and direct port connections are enforced?
     + Is Force Encryption enabled for SQL Server connections to enforce SSL/TLS?
   * **SQL Server Patching:**
     + Is the SQL Server instance (including all components like SSAS, SSRS, SSIS) regularly patched with the latest Cumulative Updates (CUs) and Security Updates (SUs)?
     + Is there a documented patching process and rollback plan?
2. **Authentication:**
   * **Authentication Mode:**
     + Is Windows Authentication mode used exclusively where possible? This is generally more secure than SQL Server Authentication.
     + If Mixed Mode (Windows and SQL Server Authentication) is necessary, is its use justified and documented?
   * **SA Account (SQL Authentication):**
     + If SQL Authentication is used, is the 'sa' account disabled or renamed?
     + Is the 'sa' password extremely strong and stored securely, with access tightly controlled?
     + Is the use of the 'sa' account by applications or users strictly prohibited or heavily audited?
   * **Password Policies (SQL Logins):**
     + If SQL Logins are used, are strong password policies enforced (complexity, history, expiration)?
     + Is "CHECK\_POLICY" and "CHECK\_EXPIRATION" enabled for SQL Logins?
   * **Login Management:**
     + Are unused or dormant SQL Server logins and Windows logins/groups regularly reviewed and disabled or removed?
     + Are default or guest accounts disabled or their permissions severely restricted?
     + Is MFA considered for privileged user access to SQL Server (e.g., via Azure MFA for SQL Server on Azure VM, or third-party solutions)?
   * **Public Role:**
     + Are permissions granted to the public server role and database role reviewed and minimized? By default, public should have very limited permissions.
3. **Authorization (Permissions & Roles):**
   * **Principle of Least Privilege:**
     + Are server-level and database-level permissions granted based on the principle of least privilege? Users and applications should only have the permissions necessary to perform their tasks.
     + Are permissions granted to roles rather than individual logins/users wherever possible?
     + Are custom server roles and database roles used to group permissions effectively?
   * **Server Roles:**
     + Is membership in fixed server roles (e.g., sysadmin, serveradmin, securityadmin) tightly controlled and minimized?
     + Are alternatives to sysadmin (e.g., more granular permissions, CONTROL SERVER) used where appropriate?
   * **Database Roles:**
     + Is membership in fixed database roles (e.g., db\_owner, db\_ddladmin, db\_datawriter, db\_datareader) appropriate and minimized?
     + Is the guest user disabled in all user databases unless specifically required and understood?
   * **Object Permissions:**
     + Are permissions on database objects (tables, views, stored procedures, functions) granted granularly?
     + Is ownership chaining understood and managed to prevent privilege escalation?
     + Are database owners (dbo) legitimate and secured accounts? Is the database owner set to 'sa' or an individual user account? (Best practice is often a dedicated, disabled login or sa if its password is secure and not used for connections).
   * **Cross-Database Ownership Chaining:**
     + Is cross-database ownership chaining disabled at the server level unless explicitly required and its security implications understood?
   * **Impersonation (EXECUTE AS):**
     + Is the use of IMPERSONATE server permission and EXECUTE AS clause in modules (stored procedures, functions, triggers) carefully controlled and reviewed for potential privilege escalation?
   * **Dynamic SQL:**
     + If dynamic SQL is used, is it implemented securely to prevent SQL injection (e.g., using sp\_executesql with parameterized queries, sanitizing inputs)?
4. **Data Protection:**
   * **Encryption at Rest:**
     + Is Transparent Data Encryption (TDE) implemented for databases containing sensitive data?
     + If TDE is used, is the Database Encryption Key (DEK) backed up and protected by a certificate or asymmetric key? Is the TDE certificate/key backed up securely and separately?
     + Are database backups encrypted (either via TDE or native backup encryption)?
   * **Encryption in Transit:**
     + Are connections to SQL Server encrypted using SSL/TLS? (See "Network Configuration" and "Certificates").
     + Is "Force Strict Encryption" enabled on the server if all clients support it?
   * **Column-Level Encryption / Always Encrypted:**
     + For highly sensitive data elements (e.g., PII, credit card numbers), is Always Encrypted or column-level encryption considered or implemented?
     + If Always Encrypted is used, where are the column master keys stored and how is access to them managed?
   * **Data Masking:**
     + Is Dynamic Data Masking used to limit exposure of sensitive data to non-privileged users in query results?
   * **Row-Level Security:**
     + Is Row-Level Security (RLS) implemented where necessary to control access to specific rows based on user context?
   * **Backup Security:**
     + Are SQL Server backups stored in a secure location with restricted access?
     + Are backup files encrypted?
     + Is there a tested disaster recovery plan that includes restoring backups?
     + Who has access to backup encryption keys/passwords?
5. **Auditing & Logging:**
   * **SQL Server Audit:**
     + Is SQL Server Audit configured to capture critical security events? This should include:
       - Failed and successful logins.
       - Changes to server configuration (sp\_configure).
       - Changes to audit configuration.
       - Security-related DDL (e.g., GRANT, DENY, REVOKE, CREATE/ALTER/DROP LOGIN/USER/ROLE).
       - Access to sensitive data or objects (if required by compliance).
       - DBCC commands.
       - EXECUTE on xp\_cmdshell or other powerful extended stored procedures.
     + Are audit logs written to a secure location (e.g., separate drive, Windows Security Event Log, or a central SIEM)?
     + Are audit logs regularly reviewed, and are alerts configured for critical events?
     + Is the audit specification and audit itself enabled?
   * **Login Auditing (Server Property):**
     + Is login auditing (failed logins, successful logins, or both) configured at the server level in SQL Server Properties (Security page)?
   * **Default Trace / System Health Session:**
     + Is the default trace or system health extended event session enabled and reviewed for security-relevant information? (Though SQL Server Audit is preferred for formal auditing).
   * **Error Logs:**
     + Are SQL Server error logs and SQL Server Agent logs regularly reviewed for suspicious activity or security-related errors?
6. **Linked Servers:**
   * **Necessity and Minimization:**
     + Are linked servers only created when absolutely necessary?
     + Are unused linked servers removed?
   * **Security Context / Authentication:**
     + How is authentication configured for linked servers?
       - Is "Be made using the login's current security context" (self-mapping) used, and is this appropriate for the trust level between servers?
       - Is "Be made using this security context" (mapping to a specific remote login/password) used? If so, is the remote login a least-privileged account on the target server? Are credentials stored securely?
       - Avoid using the option "Be made without using a security context" for connections requiring authentication.
     + Are mappings defined for specific local logins rather than a global mapping if "Be made using this security context" is chosen?
     + Is delegation (Kerberos) configured correctly if Windows pass-through authentication is intended?
   * **Permissions:**
     + What permissions does the account used by the linked server have on the remote server? Adhere to the principle of least privilege.
   * **Provider Security:**
     + Is the OLE DB provider for the linked server from a trusted source and up-to-date?
     + Are provider-specific security settings reviewed?
   * **Data Access:**
     + Is access to execute queries via the linked server restricted to necessary logins/roles on the local server?
   * **Loopback Linked Servers:**
     + Are loopback linked servers (pointing to the same instance) avoided unless there's a very specific, justified, and secured reason?
7. **Certificates (for SSL/TLS, TDE, Always Encrypted, etc.):**
   * **Issuance and Trust:**
     + Are certificates used by SQL Server (for connection encryption, TDE, etc.) issued by a trusted internal Certificate Authority (CA) or a reputable public CA?
     + Are self-signed certificates avoided in production environments, especially for connection encryption that needs to be trusted by clients?
   * **Certificate Properties:**
     + Does the certificate Subject Name (Common Name) or Subject Alternative Name (SAN) correctly match the FQDN or virtual name of the SQL Server instance/listener?
     + Is the certificate within its validity period (not expired and not yet valid)?
     + Does the certificate have the "Server Authentication" purpose in its Enhanced Key Usage (EKU) extension (for SSL/TLS)?
     + Is a strong key length (e.g., RSA 2048-bit or higher) and signature algorithm (e.g., SHA256 or higher) used?
   * **Private Key Protection:**
     + Is the private key associated with the certificate protected with strong ACLs?
     + Does the SQL Server service account have 'Read' permission to the private key?
   * **Certificate Management:**
     + Is there a process for renewing certificates before they expire?
     + Is there a process for revoking and replacing compromised certificates?
     + Are certificates deployed correctly in clustered or Availability Group environments (i.e., installed on all relevant nodes with consistent settings)?
     + Is SQL Server Configuration Manager used to assign the certificate to the SQL Server instance for connection encryption?
   * **TDE Certificates:**
     + Are TDE certificates backed up (certificate file and private key password) and stored in a highly secure, separate location?
     + Is the master key encrypted by the service master key, and is the service master key backed up?
   * **Always Encrypted Certificates:**
     + Are Column Master Keys (often certificates) stored securely (e.g., Windows Certificate Store, Azure Key Vault, HSM) and access to them tightly controlled?
8. **SQL Server Agent Security:**
   * **Proxy Accounts:**
     + If SQL Server Agent proxies are used, are they configured with least privilege credentials?
     + Is access to use specific proxies restricted to trusted logins/roles?
     + Are subsystem access permissions for proxies reviewed (e.g., CmdExec, PowerShell)? CmdExec should be particularly scrutinized.
   * **Job Ownership:**
     + Are SQL Agent job owners appropriate and secured accounts (typically sa or a dedicated, secured administrative account)? Avoid individual user accounts as job owners.
   * **Job Step Security:**
     + Are job steps that execute operating system commands or scripts reviewed for security implications?
   * **Alerts and Operators:**
     + Are operators configured for important alerts (e.g., severe errors, job failures related to security)?
9. **Other SQL Server Components:**
   * **SQL Server Reporting Services (SSRS):**
     + Is access to the SSRS web portal and web service secured?
     + Are report data sources configured with least privilege credentials?
     + Is the SSRS service account a low-privilege account?
   * **SQL Server Analysis Services (SSAS):**
     + Is access to SSAS databases and cubes controlled via roles?
     + Is the SSAS service account a low-privilege account?
     + Is Windows Authentication preferred?
   * **SQL Server Integration Services (SSIS):**
     + If the SSIS Catalog (SSISDB) is used, is access to it secured?
     + Are package protection levels appropriate (e.g., EncryptSensitiveWithUserKey, EncryptAllWithPassword, ServerStorage)?
     + Are credentials within packages managed securely (e.g., using parameters, environment variables, or Azure Key Vault for sensitive connection strings/passwords)?

**III. General Security Practices & Procedures**

1. **Physical Security:**
   * Are servers (physical hosts) located in a secure data center with restricted physical access, environmental controls, and surveillance?
2. **Change Management:**
   * Is there a formal change management process for any changes to the VM, host, or SQL Server configuration?
3. **Incident Response Plan:**
   * Is there a documented incident response plan that covers security breaches related to the SQL Server estate?
4. **Backup and Disaster Recovery:**
   * Is there a comprehensive, documented, and regularly tested backup and disaster recovery plan for the SQL Server instances and their underlying VMs?
   * Are backups stored off-site or in a separate fault domain?
   * Are recovery time objectives (RTO) and recovery point objectives (RPO) defined and met?
5. **Security Awareness Training:**
   * Are administrators and relevant personnel trained on security best practices for VMs and SQL Server?
6. **Regular Security Assessments:**
   * Are periodic security assessments (like this one) and vulnerability scans performed?
   * Are findings tracked and remediated in a timely manner?
7. **Documentation:**
   * Is the security configuration of the SQL Server estate well-documented?