Microsoft Azure Security Engineer Associate (AZ-500) Study Notes

Exam AZ-500: Microsoft Azure Security Technologies - Certifications | Microsoft Learn

Skills measured as of May 2, 2023

Manage identity and access (25–30%)

Manage identities in Azure AD

Secure users in Azure AD

Multi-Factor Authentication (MFA): Azure AD supports MFA for users to add an extra layer of security to their accounts. This requires users to provide two or more forms of authentication before they can access their accounts.

Conditional Access: Conditional Access policies allow you to set conditions that must be met before a user can access Azure AD or other resources. This can include things like location, device type, or risk level.

Password Policies: You can set password policies for users to ensure that their passwords are strong and not easily guessed. Azure AD also supports passwordless authentication methods like Microsoft Authenticator.

Identity Protection: Azure AD Identity Protection helps detect and prevent identity-based attacks by analyzing user behavior and risk factors.

Privileged Identity Management (PIM): PIM allows you to control and monitor access to privileged roles in Azure AD. You can require approvals for access, set time limits for access, and receive alerts for unusual activity.

Azure AD Connect Health: Azure AD Connect Health provides monitoring and insights into the health of your on-premises Active Directory environment and its synchronization with Azure AD.

Security Reports: Azure AD provides security reports that allow you to monitor and analyze user activity, sign-ins, and other security-related events.

Secure directory groups in Azure AD

Group membership management: It's important to carefully manage group membership in Azure AD. Only assign necessary permissions and access to users who require it.

Group Naming Conventions: Implement a naming convention for your groups that's consistent and easy to understand. This can help prevent confusion and make it easier to manage your groups.

Role-Based Access Control (RBAC): RBAC can be used to manage access to Azure AD resources. Assign roles to groups based on the access they require, and ensure that roles are reviewed and updated regularly.

Conditional Access policies: Use Conditional Access policies to restrict access to groups based on specific conditions, such as the location or device used to access the resource.

Group Owner management: Ensure that group owners are responsible and trusted individuals who can manage and control group membership effectively. Group owners should be trained and aware of the responsibilities that come with managing groups.

Group Expiration: Implement an expiration policy for groups to ensure that they're reviewed regularly and deactivated if they're no longer required.

Monitor group activity: Monitor group activity regularly to identify and address any unauthorized or unusual behavior.

Recommend when to use external identities

When you need to collaborate with external users: If you need to collaborate with users outside of your organization, such as partners, vendors, or customers, external identities can be useful. You can use Azure AD B2B collaboration to allow external users to access your resources securely.

When you want to reduce administrative overhead: External identities can help reduce administrative overhead by allowing external users to manage their own identities and passwords. This can free up IT resources and reduce the risk of password-related security issues.

When you want to leverage existing identities: If your external users already have existing identities, such as social media accounts or other corporate identities, you can use external identities to allow them to access your resources without creating new accounts.

When you want to provide a better user experience: External identities can help provide a better user experience by allowing external users to use familiar identities to access your resources. This can help reduce friction and increase adoption of your services.

When you want to scale your applications globally: External identities can help you scale your applications globally by allowing users from different regions to access your resources easily. Azure AD supports many external identity providers, which can help you reach a wider audience.

Secure external identities

Multi-Factor Authentication (MFA): Use MFA to ensure that external users provide two or more forms of authentication before they can access your resources. This can help prevent unauthorized access and reduce the risk of account compromises.

Conditional Access: Use Conditional Access policies to restrict access to external users based on specific conditions, such as location, device type, or risk level.

Identity Protection: Azure AD Identity Protection can help detect and prevent identity-based attacks by analyzing user behavior and risk factors. This can help protect your external users and resources from security threats.

Password Policies: Set password policies for external users to ensure that their passwords are strong and not easily guessed. Consider implementing passwordless authentication methods like Microsoft Authenticator to further reduce the risk of password-related security issues.

Guest User Management: Carefully manage guest user accounts in Azure AD. Only assign necessary permissions and access to guest users who require it, and monitor guest user activity regularly.

Audit and Monitor: Regularly audit and monitor external user activity to identify and address any unauthorized or unusual behavior. Azure AD provides security reports that allow you to monitor and analyze user activity, sign-ins, and other security-related events.

Data Protection: Implement appropriate data protection controls to ensure that external users only have access to the data they require. Use Azure AD's Access Review feature to regularly review and update access permissions for external users.

Implement Azure AD Identity Protection

Enable Azure AD Identity Protection: Azure AD Identity Protection is a feature that helps protect user identities and detect identity-based attacks. To use Identity Protection, you'll need to enable it in your Azure AD tenant.

Configure Risk Policies: Azure AD Identity Protection includes pre-defined risk policies that can help you detect and prevent identity-based attacks. Configure these policies based on your organization's security requirements.

Create Custom Policies: You can create custom policies in Azure AD Identity Protection to address specific security scenarios that aren't covered by the pre-defined policies.

Enable User Risk Policy: User Risk Policy helps you detect risky user behavior by analyzing user activity, such as failed sign-ins and risky sign-ins. If a user is deemed risky, you can take action to reduce the risk of a successful attack.

Enable Sign-In Risk Policy: Sign-In Risk Policy helps you detect sign-in attempts that are deemed risky, such as sign-ins from unfamiliar locations or devices. You can configure the policy to block access or require additional authentication if a sign-in attempt is deemed risky.

Use the Azure AD Identity Protection Dashboard: The Identity Protection Dashboard provides a view of the risk posture of your organization and enables you to investigate security events and take appropriate action.

Implement Adaptive Authentication: Adaptive Authentication uses risk-based policies to determine the level of authentication required for a user based on their risk level. You can use this feature to help prevent unauthorized access to your resources.

Manage authentication by using Azure AD

Configure Microsoft Entra Verified ID

Microsoft Entra Verified ID is a service that allows you to issue and verify verifiable credentials. Verifiable credentials are a new type of digital identity that can be used to prove your identity to others. They are more secure and privacy-preserving than traditional forms of identification, such as passwords and security questions.

To configure Microsoft Entra Verified ID, you will need to:

- Create a Microsoft Entra Verified ID tenant.
- Create a verifiable credential issuer.
- Create a verifiable credential verifier.
- Issue a verifiable credential.
- Verify a verifiable credential.

Here are the steps in more detail:

To create a Microsoft Entra Verified ID tenant, you will need to go to the Microsoft Entra Verified ID website and sign in with your Microsoft account.

Once you have created a tenant, you will need to create a verifiable credential issuer. A verifiable credential issuer is a software application that can be used to issue verifiable credentials.

To create a verifiable credential issuer, you will need to use the Microsoft Entra Verified ID SDK. The SDK is available for a variety of programming languages, including Java, Python, and C#.

Once you have created a verifiable credential issuer, you will need to create a verifiable credential verifier. A verifiable credential verifier is a software application that can be used to verify verifiable credentials.

To create a verifiable credential verifier, you will need to use the Microsoft Entra Verified ID SDK.

Once you have created a verifiable credential issuer and a verifiable credential verifier, you can issue and verify verifiable credentials.

To issue a verifiable credential, you will need to use the verifiable credential issuer. The issuer will generate a verifiable credential and then send it to the user.

To verify a verifiable credential, you will need to use the verifiable credential verifier. The verifier will check the signature on the verifiable credential and then verify the contents of the credential.

For more information on how to configure Microsoft Entra Verified ID, please refer to the Microsoft Entra Verified ID documentation.

Implement multi-factor authentication (MFA)

Enable MFA: To use MFA in Azure AD, you need to first enable it for your organization. You can do this in the Azure portal by going to the "Azure Active Directory" section and selecting "Security."

Configure MFA Settings: Once you have enabled MFA, you can configure the settings for your organization. This includes specifying which users are required to use MFA and which authentication methods they can use.

Choose Authentication Methods: Azure AD supports several authentication methods for MFA, including phone call, text message, mobile app notification, and third-party authenticator apps like Microsoft Authenticator.

Educate Users: It's important to educate your users on the importance of MFA and how to use it. Make sure they understand how to set up MFA on their devices and how to use it to verify their identities when accessing your organization's resources.

Monitor Usage: Regularly monitor MFA usage to ensure that your organization's resources are properly protected. Use the Azure AD portal to view MFA usage and manage MFA settings for your users.

Customize Policies: Azure AD allows you to customize MFA policies based on your organization's security requirements. This includes specifying which users are required to use MFA, which authentication methods they can use, and under what circumstances MFA is required.

Test MFA: Before rolling out MFA to your entire organization, it's important to test it thoroughly. Try logging in to a test account and verifying your identity using MFA to make sure it's working properly.

Implement passwordless authentication

Enable Passwordless Authentication: To use passwordless authentication in Azure AD, you need to first enable it for your organization. You can do this in the Azure portal by going to the "Azure Active Directory" section and selecting "Authentication methods."

Choose Authentication Methods: Azure AD supports several passwordless authentication methods, including Windows Hello for Business, Microsoft Authenticator, and FIDO2 security keys.

Educate Users: It's important to educate your users on the importance of passwordless authentication and how to use it. Make sure they understand how to set up passwordless authentication on their devices and how to use it to verify their identities when accessing your organization's resources.

Configure Authentication Methods: Once you have chosen your passwordless authentication methods, you can configure the settings for each method. This includes specifying which users are required to use passwordless authentication and which authentication methods they can use.

Monitor Usage: Regularly monitor passwordless authentication usage to ensure that your organization's resources are properly protected. Use the Azure AD portal to view authentication usage and manage authentication settings for your users.

Customize Policies: Azure AD allows you to customize passwordless authentication policies based on your organization's security requirements. This includes specifying which users are required to use passwordless authentication, which authentication methods they can use, and under what circumstances passwordless authentication is required.

Test Passwordless Authentication: Before rolling out passwordless authentication to your entire organization, it's important to test it thoroughly. Try logging in to a test account and verifying your identity using passwordless authentication to make sure it's working properly.

Implement password protection

Enable Password Protection: To use password protection in Azure AD, you need to first enable it for your organization. You can do this in the Azure portal by going to the "Azure Active Directory" section and selecting "Password protection."

Configure Password Protection Policies: Once you have enabled password protection, you can configure the settings for your organization's password protection policies. This includes specifying the minimum password length, complexity requirements, and password expiration policies.

Educate Users: It's important to educate your users on the importance of strong passwords and how to create them. Make sure they understand the password protection policies in place and how to create passwords that meet those requirements.

Monitor Password Usage: Regularly monitor password usage to ensure that your organization's resources are properly protected. Use the Azure AD portal to view password usage and manage password protection policies for your users.

Customize Password Policies: Azure AD allows you to customize password policies based on your organization's security requirements. This includes specifying the minimum password length, complexity requirements, and password expiration policies.

Test Password Protection: Before rolling out password protection to your entire organization, it's important to test it thoroughly. Try creating test accounts with different password policies to make sure they are working properly.

Implement Additional Security Measures: Password protection is just one part of a comprehensive security strategy. Consider implementing additional security measures, such as multi-factor authentication, to further protect your organization's resources.

Implement single sign-on (SSO)

Enable SSO: To use SSO in Azure AD, you need to first enable it for your organization. You can do this in the Azure portal by going to the "Azure Active Directory" section and selecting "Enterprise applications."

Configure Enterprise Applications: Once you have enabled SSO, you can configure the settings for each enterprise application in your organization. This includes specifying the SSO method and the user sign-in and sign-out URLs.

Add Users: Add users to the enterprise application so they can access it using SSO. You can add users individually or in bulk using a CSV file.

Customize SSO Settings: Azure AD allows you to customize SSO settings based on your organization's security requirements. This includes specifying which users are allowed to use SSO and under what circumstances SSO is required.

Test SSO: Before rolling out SSO to your entire organization, it's important to test it thoroughly. Try logging in to an enterprise application and verifying that SSO is working properly.

Monitor SSO Usage: Regularly monitor SSO usage to ensure that your organization's resources are properly protected. Use the Azure AD portal to view SSO usage and manage SSO settings for your users.

Implement Additional Security Measures: SSO is just one part of a comprehensive security strategy. Consider implementing additional security measures, such as multi-factor authentication, to further protect your organization's resources.

Integrate single sign on (SSO) and identity providers

Identify Identity Providers: Before integrating SSO and identity providers, identify which identity providers you want to use. Azure AD supports a variety of identity providers, including Microsoft accounts, Google, Facebook, and Twitter.

Recommend and enforce modern authentication protocols

Enable Modern Authentication: To use modern authentication in Azure AD, you need to enable it for your organization. This can be done in the Azure portal by going to the "Azure Active Directory" section and selecting "Security."

Recommend Modern Authentication: Once modern authentication is enabled, it's important to recommend its use to your users. Modern authentication protocols, such as OAuth 2.0 and OpenID Connect, provide stronger security and greater flexibility than older protocols like Basic Authentication.

Enforce Modern Authentication: To ensure that your organization's resources are properly protected, it's important to enforce the use of modern authentication protocols. This can be done by creating Conditional Access policies that require the use of modern authentication for specific applications or users.

Monitor Authentication Usage: Regularly monitor authentication usage to ensure that your organization's resources are properly protected. Use the Azure AD portal to view authentication logs and manage authentication settings for your users.

Implement Additional Security Measures: Modern authentication is just one part of a comprehensive security strategy. Consider implementing additional security measures, such as multi-factor authentication and password protection policies, to further protect your organization's resources.

Manage authorization by using Azure AD

Configure Azure role permissions for management groups, subscriptions, resource groups, and resources

Understand Azure RBAC: Azure Role-Based Access Control (RBAC) is used to manage access to Azure resources. RBAC allows you to assign roles to users, groups, or applications, which determine their permissions to manage resources.

Identify the Scope of Permissions: Before configuring Azure role permissions, identify the scope of permissions needed. Azure RBAC allows you to assign roles at the management group, subscription, resource group, or resource level.

Assign Built-In Roles: Azure provides a set of built-in roles that can be assigned to users or groups. These roles include Owner, Contributor, Reader, and User Access Administrator, among others. Each role provides a set of permissions that allow users to manage Azure resources.

Create Custom Roles: Azure also allows you to create custom roles with specific permissions tailored to your organization's needs. Custom roles can be created using Azure PowerShell or the Azure portal.

Use Resource Locks: Resource locks can be used to prevent accidental deletion or modification of resources. Resource locks can be applied at the resource group or resource level, and can be configured to be either read-only or delete-protected.

Monitor Role Permissions: Regularly monitor role permissions to ensure that users and groups have the appropriate level of access to Azure resources. Use the Azure portal to view role assignments and audit logs.

Use Azure Policy: Azure Policy can be used to enforce compliance with organizational policies and standards. Azure Policy can be used to enforce specific permissions, such as requiring multi-factor authentication for certain roles.

Assign built-in roles in Azure AD

Understand Azure AD Roles: Azure AD provides a set of built-in roles that can be assigned to users or groups. These roles provide permissions to manage Azure AD resources, such as users, groups, applications, and policies.

Identify the Scope of Permissions: Before assigning Azure AD roles, identify the scope of permissions needed. Azure AD roles can be assigned at the directory, application, or resource level.

Assign Built-In Roles: Azure AD provides a set of built-in roles, such as Global Administrator, User Administrator, Application Administrator, and Helpdesk Administrator, among others. Each role provides a set of permissions that allow users to manage Azure AD resources.

Create Custom Roles: Azure AD also allows you to create custom roles with specific permissions tailored to your organization's needs. Custom roles can be created using Azure AD PowerShell or the Azure portal.

Assign Roles to Users or Groups: Once roles have been created or identified, they can be assigned to users or groups. This can be done using the Azure AD portal or Azure AD PowerShell.

Monitor Role Assignments: Regularly monitor role assignments to ensure that users and groups have the appropriate level of access to Azure AD resources. Use the Azure AD portal to view role assignments and audit logs.

Use Azure AD Privileged Identity Management: Azure AD Privileged Identity Management can be used to manage and monitor access to privileged roles in Azure AD. Azure AD Privileged Identity Management allows you to grant just-in-time access to roles and monitor usage to ensure that privileged access is properly managed.

Assign built-in roles in Azure

Understand Azure RBAC: Azure Role-Based Access Control (RBAC) is used to manage access to Azure resources. RBAC allows you to assign roles to users, groups, or applications, which determine their permissions to manage resources.

Identify the Scope of Permissions: Before assigning Azure roles, identify the scope of permissions needed. Azure roles can be assigned at the management group, subscription, resource group, or resource level.

Assign Built-In Roles: Azure provides a set of built-in roles, such as Owner, Contributor, Reader, and User Access Administrator, among others. Each role provides a set of permissions that allow users to manage Azure resources.

Create Custom Roles: Azure also allows you to create custom roles with specific permissions tailored to your organization's needs. Custom roles can be created using Azure PowerShell or the Azure portal.

Assign Roles to Users or Groups: Once roles have been created or identified, they can be assigned to users or groups. This can be done using the Azure portal, Azure PowerShell, or Azure CLI.

Use Resource Locks: Resource locks can be used to prevent accidental deletion or modification of resources. Resource locks can be applied at the resource group or resource level, and can be configured to be either read-only or delete-protected.

Monitor Role Assignments: Regularly monitor role assignments to ensure that users and groups have the appropriate level of access to Azure resources. Use the Azure portal to view role assignments and audit logs.

Create and assign custom roles, including Azure roles and Azure AD roles

Understand the Need for Custom Roles: Custom roles can be created to provide specific permissions tailored to your organization's needs. This can be useful when the built-in roles do not provide the necessary level of access or when you want to restrict access to certain resources.

Identify the Scope of Permissions: Before creating custom roles, identify the scope of permissions needed. Custom roles can be created at the management group, subscription, resource group, or resource level.

Create Custom Roles: Azure provides several options for creating custom roles, including Azure Portal, Azure PowerShell, and Azure CLI. In Azure AD, custom roles can be created using Azure AD PowerShell or the Azure portal.

Define Permissions: When creating custom roles, define the specific permissions required for the role. This can be done using role-based access control (RBAC) definitions, which define the actions that can be performed on resources.

Assign Roles to Users or Groups: Once custom roles have been created, they can be assigned to users or groups. This can be done using the Azure portal, Azure PowerShell, or Azure CLI.

Monitor Role Assignments: Regularly monitor role assignments to ensure that users and groups have the appropriate level of access to Azure resources. Use the Azure portal to view role assignments and audit logs.

Use Resource Locks: Resource locks can be used to prevent accidental deletion or modification of resources. Resource locks can be applied at the resource group or resource level, and can be configured to be either read-only or delete-protected.

Implement and manage Microsoft Entra Permissions Management

Microsoft Entra Permissions Management (PPM) is a cloud-based service that helps you manage permissions in your cloud infrastructure. PPM provides a number of features that can help you improve your security posture, including:

Visibility: PPM provides visibility into all permissions assigned to identities, actions, and resources across your cloud infrastructure. This helps you identify potential security risks, such as unused or excessive permissions.

Policy enforcement: PPM can be used to enforce least privilege policies across your cloud infrastructure. This helps you ensure that users only have access to the resources they need.

Remediation: PPM can be used to remediate security risks, such as unused or excessive permissions. This helps you reduce your attack surface and improve your security posture.

To implement and manage Microsoft Entra PPM, you will need to:

Create a PPM tenant: You will need to create a PPM tenant in the Microsoft Entra portal.

Integrate PPM with your cloud infrastructure: You will need to integrate PPM with your cloud infrastructure. This can be done using the PPM SDK or the PPM CLI.

Create policies: You will need to create policies that define the permissions that users should have.

Assign policies to users: You will need to assign policies to users.

Monitor PPM: You will need to monitor PPM to ensure that it is working properly and that it is detecting and remediating security risks.

For more information on how to implement and manage Microsoft Entra PPM, please refer to the Microsoft Entra PPM documentation.

Here are some additional tips for implementing and managing Microsoft Entra PPM:

Start small: Don't try to implement PPM for your entire organization all at once. Start with a small subset of users and resources and then gradually expand your implementation over time.

Get buy-in from stakeholders: It is important to get buy-in from stakeholders before you implement PPM. This will help ensure that you have the support you need to be successful.

Use automation: There are a number of tools that can be used to automate the implementation and management of PPM. This can help you save time and effort.

Monitor and adjust: It is important to monitor PPM to ensure that it is working properly and that it is detecting and remediating security risks. You may need to adjust your policies and procedures over time as your organization changes.

Configure Azure AD Privileged Identity Management (PIM)

Understand the Need for Privileged Identity Management: In many organizations, there are certain privileged accounts that have access to critical resources and data. These accounts are high-value targets for attackers, so it's important to monitor and manage them closely.

Configure PIM: Azure AD Privileged Identity Management allows you to manage privileged access to resources in Azure AD and other Microsoft Online Services. To configure PIM, you need to be a Global Administrator or a Privileged Role Administrator.

Create Roles: PIM allows you to create custom roles based on the permissions required for a specific job function. For example, you can create a role that allows users to reset passwords but not create new users.

Assign Roles: Once roles are created, they can be assigned to users or groups. PIM allows you to assign roles permanently or for a specific duration of time.

Use Approval Workflows: PIM also allows you to configure approval workflows for high-risk actions. For example, if a user requests elevated access to a resource, an approval workflow can be triggered that requires another user to approve the request.

Monitor Activity: PIM provides detailed activity logs that allow you to monitor privileged access to resources. You can also configure alerts to notify you of unusual activity.

Conduct Audits: PIM provides reports that allow you to conduct audits of privileged access to resources. These reports can be used to demonstrate compliance with regulations or internal policies.

Configure role management and access reviews by using Microsoft Entra Identity Governance

Microsoft Entra Identity Governance (IDG) is a cloud-based service that helps you manage identity and access for your organization. IDG provides a number of features that can help you improve your security posture, including:

Role management: IDG allows you to define roles and assign them to users. This helps you control who has access to what resources.

Access reviews: IDG allows you to periodically review user access to resources. This helps you ensure that users only have access to the resources they need.

To configure role management and access reviews by using Microsoft Entra IDG, you will need to:

- Create a Microsoft Entra IDG tenant.
- Create roles and assign them to users.
- Create access reviews and assign them to users.

Here are the steps in more detail:

To create a Microsoft Entra IDG tenant, you will need to go to the Microsoft Entra IDG website and sign in with your Microsoft account.

Once you have created a tenant, you will need to create roles. A role is a collection of permissions that can be assigned to a user. To create a role, you will need to:

- Give the role a name.
- Select the permissions that the role will have.

Once you have created roles, you will need to assign them to users. To assign a role to a user, you will need to:

- Select the user.
- Select the role.

Once you have assigned roles to users, you will need to create access reviews. An access review is a process that you can use to periodically review user access to resources. To create an access review, you will need to:

- Give the access review a name.
- Select the users who will be included in the review.
- Select the resources that will be reviewed.
- Select the frequency of the review.

Once you have created access reviews, you will need to assign them to users. To assign an access review to a user, you will need to:

- Select the user.
- Select the access review.

For more information on how to configure role management and access reviews by using Microsoft Entra IDG, please refer to the Microsoft Entra IDG documentation.

Implement Conditional Access policies

Understand the Need for Conditional Access: Conditional Access allows you to control access to your organization's resources based on conditions such as the user's location, device, or risk level. This helps ensure that only authorized users with trusted devices are able to access sensitive resources.

Configure Conditional Access Policies: Azure provides a wide range of conditions and controls that can be used to configure Conditional Access policies. For example, you can require multi-factor authentication for users accessing resources from a new or untrusted device, or block access to resources from certain geographic locations.

Apply Policies to Specific Users and Groups: Conditional Access policies can be applied to specific users or groups, allowing you to tailor access controls based on individual roles and responsibilities.

Monitor and Report on Policy Activity: Azure provides detailed reporting on Conditional Access policy activity, allowing you to monitor user access and policy effectiveness. You can also configure alerts to notify you of policy violations.

Leverage Third-Party Integrations: Azure supports integration with a range of third-party services, such as security information and event management (SIEM) platforms, to provide enhanced monitoring and threat detection capabilities.

Stay Up-to-Date with Best Practices: As the threat landscape evolves, it's important to stay up-to-date with best practices for configuring Conditional Access policies. Microsoft regularly updates its recommendations based on the latest threat intelligence.

Manage application access in Azure AD

Manage access to enterprise applications in Azure AD, including OAuth permission grants

Understand the Need for Access Management: Enterprise applications are critical assets that contain sensitive information and resources. To maintain security, it's important to manage access to these applications based on users' roles and responsibilities.

Configure Application Access: Azure AD provides a range of options to manage access to enterprise applications, such as requiring multi-factor authentication, limiting access based on user location, or requiring device compliance.

Manage OAuth Permission Grants: OAuth is an authorization framework that allows users to grant third-party applications access to their resources without giving out their credentials. Azure AD provides the ability to manage OAuth permission grants to ensure only authorized applications have access to user data.

Use Azure AD App Proxy: Azure AD App Proxy provides secure remote access to on-premises applications without requiring VPN connectivity. This allows users to access applications from anywhere, while maintaining a secure connection and consistent access controls.

Monitor Application Access: Azure provides detailed reporting and analytics on application access activity, allowing you to monitor user access and policy effectiveness. You can also configure alerts to notify you of policy violations.

Leverage Third-Party Integrations: Azure supports integration with a range of third-party services, such as security information and event management (SIEM) platforms, to provide enhanced monitoring and threat detection capabilities.

Manage app registrations in Azure AD

Understand App Registrations: App registrations in Azure AD are used to represent apps that need to access resources, such as APIs, on behalf of a user or organization. These apps can be registered as either single-tenant or multi-tenant, depending on the intended use case.

Create App Registrations: You can create app registrations in Azure AD using the Azure portal or programmatically using Azure AD Graph API or Microsoft Graph API.

Configure App Permissions: App permissions are used to grant access to resources, such as APIs, on behalf of users or organizations. You can configure app permissions for app registrations in Azure AD, which allows you to manage access to resources at the application level.

Manage Secrets: Secrets, such as passwords or certificates, are used to authenticate an app with Azure AD. You can manage app secrets for app registrations in Azure AD to ensure that only authorized apps can access your resources.

Monitor App Activity: Azure provides detailed reporting and analytics on app activity, allowing you to monitor app registrations and their access to resources. You can also configure alerts to notify you of potential threats.

Leverage Third-Party Integrations: Azure supports integration with a range of third-party services, such as security information and event management (SIEM) platforms, to provide enhanced monitoring and threat detection capabilities.

Configure app registration permission scopes

Understand App Permissions: App permissions are used to grant an app access to resources, such as APIs, on behalf of a user or organization. When an app is registered in Azure AD, it can be assigned specific permissions to access resources.

Configure Permission Scopes: Permission scopes are used to define the level of access an app has to specific resources. For example, an app may be granted read-only access to a particular API, or it may be granted read and write access.

Grant Permissions: Once permission scopes have been defined for an app registration, permissions can be granted to the app. Permissions can be granted either by an administrator or by a user who has consented to the app's access.

Manage Permissions: Azure provides tools to manage app permissions, including the ability to revoke permissions if necessary. This is important to maintain the security of your organization's resources and prevent unauthorized access.

Monitor App Activity: Azure provides detailed reporting and analytics on app activity, allowing you to monitor app registrations and their access to resources. You can also configure alerts to notify you of potential threats.

Leverage Third-Party Integrations: Azure supports integration with a range of third-party services, such as security information and event management (SIEM) platforms, to provide enhanced monitoring and threat detection capabilities.

Manage app registration permission consent

Understand App Consent: When an app is registered in Azure AD, it can request permission to access certain resources on behalf of a user or organization. Before access is granted, the user or administrator must consent to the requested permissions.

Manage Consent Settings: Azure AD provides various options to manage app consent settings. For example, you can configure whether users are required to consent to permissions, and whether administrators are allowed to consent on behalf of users.

Monitor Consent Activity: Azure AD provides detailed reporting and analytics on app consent activity, allowing you to monitor the types of permissions being requested and the frequency of consent.

Revoke Consent: Azure AD allows administrators to revoke consent for an app at any time, which immediately revokes the app's access to resources. This is important to maintain the security of your organization's resources and prevent unauthorized access.

Configure App Permissions: App permissions can be configured in Azure AD to define the level of access an app has to specific resources. This is important to ensure that apps are only granted the necessary permissions to perform their intended functions.

Leverage Third-Party Integrations: Azure AD supports integration with a range of third-party services, such as security information and event management (SIEM) platforms, to provide enhanced monitoring and threat detection capabilities.

Manage and use service principals

Understand Service Principals: A service principal is an identity in Azure AD that represents a service or application. It is used to authenticate and authorize access to resources in Azure.

Create a Service Principal: A service principal can be created through the Azure Portal or programmatically using Azure CLI or Azure PowerShell.

Assign Permissions: Once a service principal is created, it can be assigned permissions to access resources in Azure. This can be done through Azure Portal or programmatically using Azure CLI or Azure PowerShell.

Manage Service Principals: Azure AD provides various tools to manage service principals. For example, you can view and update the properties of a service principal, reset the credentials, or delete a service principal.

Use Service Principals: Service principals can be used to authenticate and authorize access to resources in Azure. For example, a service principal can be used to access an Azure Key Vault or Azure Storage Account.

Secure Service Principals: It's important to secure service principals to prevent unauthorized access to resources. This can be done by following security best practices, such as limiting permissions to only what's necessary, rotating credentials regularly, and monitoring service principal activity.

Manage managed identities for Azure resources

Understand Managed Identities: A managed identity is an identity in Azure AD that is automatically managed by Azure. It can be used to authenticate and authorize access to resources in Azure, such as Azure Virtual Machines, Azure Functions, and Azure App Service.

Create a Managed Identity: A managed identity can be created through the Azure Portal or programmatically using Azure CLI or Azure PowerShell.

Assign Permissions: Once a managed identity is created, it can be assigned permissions to access resources in Azure. This can be done through Azure Portal or programmatically using Azure CLI or Azure PowerShell.

Use Managed Identities: Managed identities can be used to authenticate and authorize access to resources in Azure. For example, a managed identity can be used to access an Azure Key Vault or Azure Storage Account.

Secure Managed Identities: It's important to secure managed identities to prevent unauthorized access to resources. This can be done by following security best practices, such as limiting permissions to only what's necessary, rotating credentials regularly, and monitoring managed identity activity.

Manage Managed Identities: Azure provides various tools to manage managed identities. For example, you can view and update the properties of a managed identity, reset the credentials, or delete a managed identity.

Recommend when to use and configure an Azure AD Application Proxy, including authentication

Understand Azure AD Application Proxy: The Azure AD Application Proxy is a service that allows organizations to securely publish internal web applications and access them from anywhere using a web browser or mobile device.

Use Cases: Organizations can use Azure AD Application Proxy to publish on-premises web applications, such as SharePoint or Outlook Web Access, and make them accessible to external users without the need for a VPN connection. It can also be used to provide secure access to applications hosted in the cloud, such as Microsoft 365 or Salesforce.

Authentication: The Azure AD Application Proxy can be configured to provide authentication for published applications using Azure AD or other identity providers. This ensures that only authorized users can access the applications.

Configuration: To configure an application for Azure AD Application Proxy, the organization needs to install a connector on a server in the internal network. The connector communicates with the Azure AD Application Proxy service and forwards traffic to the published application.

Security Considerations: Organizations should consider security best practices when configuring Azure AD Application Proxy. For example, they should limit access to the connector server, ensure that published applications are secured with HTTPS, and monitor traffic to detect any suspicious activity.

Benefits: Azure AD Application Proxy provides benefits such as simplified application access for external users, reduced infrastructure requirements, and improved security.

Secure networking (20–25%)

Plan and implement security for virtual networks

Plan and implement Network Security Groups (NSGs) and Application Security Groups (ASGs)

Network Security Groups (NSGs) filter traffic based on IP addresses, ports, and protocols.

Application Security Groups (ASGs) group resources together based on application requirements.

Inbound and outbound security rules can be created for each NSG.

NSGs can be associated with a virtual network, a subnet, or a network interface.

Allow and deny rules can be created, and rules can be prioritized.

NSGs can be used to restrict traffic to and from the internet.

ASGs can simplify the management of NSGs by applying rules to a group of resources.

Plan and implement user-defined routes (UDRs)

User-defined routes (UDRs) allow you to override Azure's default routing for network traffic.

UDRs can be created for virtual networks, subnets, and individual network interfaces.

UDRs can be used to route traffic through a virtual appliance or VPN connection.

UDRs can be created using Azure Portal, Azure CLI, or Azure PowerShell.

UDRs can be assigned to virtual machines or network interfaces.

When creating UDRs, it's important to consider the order of route tables, as well as the priority and weight of individual routes.

UDRs can be used in conjunction with Azure ExpressRoute to establish private connections between on-premises infrastructure and Azure.

Plan and implement VNET peering or VPN gateway

VNET peering enables connectivity between two virtual networks in the same region.

Peering can be configured between virtual networks in different subscriptions and between virtual networks in different Azure Active Directory tenants.

When peering virtual networks, traffic flows directly between them without passing through a gateway or over the public internet.

Peered virtual networks can communicate using their private IP addresses.

Azure VPN Gateway enables secure connectivity between on-premises infrastructure and Azure virtual networks.

VPN Gateway can be configured using either Policy-based VPN or Route-based VPN.

VPN Gateway can be used to establish site-to-site VPN connections or point-to-site VPN connections.

When using VPN Gateway, traffic is encrypted and travels over the public internet.

Plan and implement Virtual WAN, including secured virtual hub

Virtual WAN (Wide Area Network) is a networking service that simplifies large-scale connectivity and enables optimized routing of traffic.

Virtual WAN allows you to connect and manage multiple VPN and Azure ExpressRoute connections.

Virtual WAN supports hub-and-spoke topology, where hubs represent central locations and spokes represent remote locations.

Secured Virtual Hub is a hub in Virtual WAN that provides security and connectivity features for spokes.

Secured Virtual Hub can be configured with Firewall, Network Virtual Appliances, and Azure Firewall Manager to provide advanced network security features.

Secured Virtual Hub also supports automatic failover and global reachability for spokes.

Virtual WAN can be configured using Azure Portal, Azure PowerShell, or Azure CLI.

Virtual WAN can be monitored using Azure Monitor, and logs can be exported to Azure Log Analytics or third-party SIEM solutions.

Secure VPN connectivity, including point-to-site and site-to-site

Virtual Private Network (VPN) is a technology that provides secure connectivity between on-premises infrastructure and Azure virtual networks over the public internet.

VPN connections can be configured using Point-to-Site (P2S) or Site-to-Site (S2S) VPN.

P2S VPN enables clients to connect to Azure virtual networks from remote locations using VPN clients.

S2S VPN enables connectivity between on-premises infrastructure and Azure virtual networks using VPN gateways.

Azure VPN Gateway can be configured using either Policy-based VPN or Route-based VPN.

VPN Gateway supports P2S VPN connections and S2S VPN connections.

VPN Gateway can be configured with various types of VPN devices, such as Cisco ASA, Juniper, and Check Point.

VPN Gateway can be configured for high availability and can be monitored using Azure Monitor.

VPN Gateway can be integrated with Azure AD for user authentication and authorization.

Implement encryption over ExpressRoute

Azure ExpressRoute provides private, dedicated, high-throughput network connectivity between on-premises infrastructure and Azure datacenters.

ExpressRoute can be used to establish connections between on-premises infrastructure and Azure virtual networks or Azure services, such as Azure Storage or Azure SQL Database.

Encryption can be implemented over ExpressRoute to ensure data confidentiality and integrity.

ExpressRoute supports two types of encryption: IPsec encryption and MACsec encryption.

IPsec encryption is a network-layer encryption that encrypts the entire IP packet, including the header and payload.

MACsec encryption is a link-layer encryption that encrypts the Ethernet frame.

IPsec encryption can be used to encrypt traffic between on-premises infrastructure and Azure virtual networks.

MACsec encryption can be used to encrypt traffic between Azure ExpressRoute Direct circuits and Azure services.

Encryption over ExpressRoute can be implemented using Azure ExpressRoute Premium or ExpressRoute Global Reach.

Configure firewall settings on PaaS resources

Platform as a Service (PaaS) is a cloud computing model that provides a platform for developing, deploying, and managing applications without the need to manage the underlying infrastructure.

PaaS resources include services such as Azure App Service, Azure SQL Database, and Azure Storage.

PaaS resources can be secured using various Azure security features, such as Network Security Groups (NSGs) and Application Security Groups (ASGs).

NSGs can be used to allow or deny traffic to PaaS resources based on source and destination IP addresses, source and destination ports, and protocol.

ASGs can be used to group resources together based on common attributes, such as application name or location, and apply NSG rules to the group.

Azure Firewall can also be used to secure PaaS resources by providing centralized network security policy management and threat protection.

Azure Firewall can be deployed in either a standalone or high-availability configuration, and can be integrated with Azure Virtual Network or Azure ExpressRoute to provide secure connectivity.

Monitor network security by using Network Watcher, including NSG flow logging

Network Watcher is an Azure service that provides a set of tools to monitor and diagnose network issues in Azure.

Network Watcher can be used to monitor network security by providing visibility into network traffic and security policies.

Network Watcher provides various features to monitor and diagnose network issues, such as Topology, Connection Monitor, and Packet Capture.

NSG flow logging is a feature of Network Watcher that allows you to capture and log network traffic that flows through an NSG.

NSG flow logging can be used to monitor and analyze network traffic patterns, troubleshoot network issues, and identify security threats.

NSG flow logs can be stored in Azure Storage, Azure Event Hubs, or Azure Log Analytics.

NSG flow logs can be analyzed using Azure Log Analytics or third-party SIEM solutions.

Network Watcher can also be used to monitor and diagnose network performance issues, such as latency and packet loss, using features such as Connection Monitor and Network Performance Monitor.

Plan and implement security for private access to Azure resources

Plan and implement virtual network Service Endpoints

Azure Virtual Network Service Endpoints allow you to extend your virtual network private address space to Azure services over a private connection.

Service Endpoints provide secure connectivity to Azure services without exposing them to the public internet.

Service Endpoints enable traffic to flow directly from your virtual network to Azure services, bypassing the internet and improving security and performance.

Service Endpoints are available for various Azure services, such as Azure Storage, Azure SQL Database, and Azure Key Vault.

Service Endpoints can be configured using the Azure portal, Azure CLI, Azure PowerShell, or Azure Resource Manager templates.

When you create a Service Endpoint, a subnet is required to be associated with the Service Endpoint.

A Service Endpoint is accessed by using the IP address of the Azure service, which is resolved to the private IP address of the Service Endpoint.

Service Endpoints can be secured using Network Security Groups (NSGs) and Application Security Groups (ASGs).

You can configure Service Endpoints to allow traffic only from specific virtual networks or subnets, and deny traffic from all other sources.

Plan and implement Private Endpoints

Private Endpoints enable you to access Azure services over a private connection using a private IP address within your virtual network.

Private Endpoints allow you to access Azure services securely from your virtual network without exposing them to the public internet.

Private Endpoints can be used to connect to various Azure services, such as Azure Storage, Azure SQL Database, and Azure Key Vault.

Private Endpoints provide secure connectivity by using Azure Private Link technology, which establishes a private, dedicated connection between your virtual network and the Azure service.

Private Endpoints are created in the same virtual network as the resource that you want to access privately.

A Private Endpoint is assigned a private IP address from the IP address range of the subnet in which it is created.

A Private Endpoint can be associated with one or more network interfaces, which can be used to connect to the Azure service over the private connection.

Private Endpoints can be secured using Network Security Groups (NSGs) and Application Security Groups (ASGs).

You can configure Private Endpoints to allow traffic only from specific virtual networks or subnets, and deny traffic from all other sources.

Private Endpoints can be configured using the Azure portal, Azure CLI, Azure PowerShell, or Azure Resource Manager templates.

Plan and implement Private Link services

Private Link services enable you to expose your own services as a private endpoint within your virtual network.

Private Link services allow you to provide your customers or users with secure and private access to your services, without exposing them to the public internet.

Private Link services use Azure Private Link technology to establish a private, dedicated connection between the user's virtual network and the service.

Private Link services can be used to provide secure access to various services, such as web applications, APIs, and databases.

Private Link services can be created using the Azure portal, Azure CLI, Azure PowerShell, or Azure Resource Manager templates.

When you create a Private Link service, you define a service name, IP configuration, and resource group.

After creating a Private Link service, you can configure the service to allow access only from specific virtual networks or subnets, and deny traffic from all other sources.

You can also configure Private Link services to use Network Security Groups (NSGs) and Application Security Groups (ASGs) to further secure access to the service.

Private Link services can be associated with Azure Load Balancer, Azure Application Gateway, or Azure Traffic Manager to provide high availability and scalability.

Private Link services can be monitored using Azure Monitor and Network Watcher, which provide visibility into traffic patterns, performance, and security.

Plan and implement network integration for Azure App Service and Azure Functions

Azure App Service and Azure Functions can be integrated with virtual networks to securely access resources within the virtual network.

Network integration for App Service and Functions can be achieved using either VNet Integration or Regional VNet Integration.

VNet Integration enables App Service or Functions to access resources within a virtual network through a point-to-site VPN connection or an ExpressRoute circuit.

Regional VNet Integration allows you to access resources within a specific region of your virtual network.

VNet Integration and Regional VNet Integration can be configured using the Azure portal, Azure CLI, Azure PowerShell, or Azure Resource Manager templates.

When you integrate App Service or Functions with a virtual network, you need to specify the virtual network and subnet that you want to use.

After integration, App Service or Functions can access resources within the virtual network using their private IP addresses.

App Service and Functions can also be configured to use Service Endpoints or Private Endpoints to securely access Azure services over a private connection within the virtual network.

Network Security Groups (NSGs) and Application Security Groups (ASGs) can be used to secure traffic between App Service or Functions and resources within the virtual network.

You can also monitor network traffic between App Service or Functions and the virtual network using Network Watcher and NSG flow logging.

Plan and implement network security configurations for an App Service Environment (ASE)

An App Service Environment (ASE) is a fully isolated and dedicated environment for running App Service apps at scale.

ASEs can be configured to integrate with virtual networks and to enforce network security configurations to protect the apps running in the environment.

Network security configurations for ASEs can include configuring Network Security Groups (NSGs), Service Endpoints, Private Endpoints, and Virtual Network Service Endpoints (VNET-SE) for resources within the virtual network.

NSGs can be used to filter inbound and outbound traffic for the ASE and to control access to resources within the virtual network.

Service Endpoints and Private Endpoints can be used to allow secure and private access to Azure services and to on-premises resources over a private connection.

VNET-SE can be used to allow access to services hosted within the ASE, such as databases or APIs, over a private connection within the virtual network.

Network security configurations for ASEs can be managed using the Azure portal, Azure PowerShell, or Azure CLI.

After configuring network security for an ASE, you can monitor traffic between the ASE and the virtual network using NSG flow logging and Network Watcher.

Additionally, ASEs can be configured with Web Application Firewall (WAF) to provide an additional layer of security for web applications running in the environment. WAF can be configured to block common web application attacks, such as SQL injection and cross-site scripting (XSS).

Plan and implement network security configurations for an Azure SQL Managed Instance

Azure SQL Managed Instance is a fully managed platform-as-a-service (PaaS) offering for running SQL Server workloads in the cloud.

To secure the network for an Azure SQL Managed Instance, you can implement network security configurations, such as configuring Virtual Network Service Endpoints (VNET-SE), Service Endpoints, Private Endpoints, and Firewall rules.

VNET-SE can be used to allow access to the Azure SQL Managed Instance over a private connection within the virtual network.

Service Endpoints and Private Endpoints can be used to allow secure and private access to Azure SQL Managed Instance over a private connection, bypassing the public internet.

Firewall rules can be used to control traffic to and from the Azure SQL Managed Instance. You can configure the firewall to allow access from specific IP addresses or ranges and block all other traffic.

Network security configurations for Azure SQL Managed Instance can be managed using the Azure portal, Azure PowerShell, or Azure CLI.

After configuring network security for Azure SQL Managed Instance, you can monitor traffic using the Azure SQL Analytics feature or the Azure Monitor logs.

You can also use Azure Active Directory (Azure AD) authentication to authenticate users and applications accessing the Azure SQL Managed Instance, which adds an additional layer of security.

Plan and implement security for public access to Azure resources

Plan and implement TLS to applications, including Azure App Service and API Management

Transport Layer Security (TLS) is a protocol used to provide secure communication between applications over a network.

To secure applications hosted in Azure, you can implement TLS to encrypt communication between the application and clients.

Azure App Service and API Management both support TLS encryption for incoming requests and outgoing responses.

You can configure App Service and API Management to require TLS for all incoming requests or for specific endpoints, using custom domain names or the default azurewebsites.net or *.azure-api.net domain names.

To enable TLS for App Service, you can configure SSL certificates for custom domain names or use the default SSL certificate provided by Azure.

To enable TLS for API Management, you can upload SSL certificates for custom domain names or use the default SSL certificate provided by Azure.

You can also configure TLS termination on Azure Application Gateway or Azure Front Door, which can be used to offload the encryption and decryption of TLS traffic and improve performance.

When implementing TLS, it's important to use strong encryption protocols, such as TLS 1.2 or later, and to keep SSL certificates up-to-date to prevent security vulnerabilities.

You can monitor TLS configuration and compliance using Azure Security Center, which can detect misconfigurations and vulnerabilities related to TLS.

Plan, implement, and manage an Azure Firewall, including Azure Firewall Manager and firewall policies

Azure Firewall is a fully managed, cloud-based network security service that provides network traffic filtering and protection against threats.

Azure Firewall allows you to create, enforce, and log application and network connectivity policies across subscriptions and virtual networks.

To implement Azure Firewall, you can create a firewall instance in a virtual network and configure firewall rules to allow or deny traffic based on source, destination, port, and protocol.

You can also use Azure Firewall Manager to centrally manage multiple firewall instances and to create and apply firewall policies across subscriptions and virtual networks.

Firewall policies are used to define and enforce network security and application-level policies across multiple Azure Firewall instances. You can create a policy to define rules for specific applications, ports, protocols, and IP addresses.

Firewall policies can be created using Azure Policy or Azure Firewall Manager, and can be applied to multiple firewall instances in different regions and subscriptions.

You can also configure Azure Firewall to integrate with Azure Monitor to monitor firewall activity and generate alerts for potential security threats or policy violations.

Azure Firewall supports various features such as Application Rules, Network Rules, NAT Rules, TLS Inspection, DNS Proxy, and Forced Tunneling.

Azure Firewall can be integrated with Azure Security Center for additional threat intelligence and security recommendations.

Plan and implement an Azure Application Gateway

Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications.

Application Gateway provides features such as SSL termination, cookie-based session affinity, URL path-based routing, Web Application Firewall (WAF), and autoscaling.

To implement Application Gateway, you need to create an instance of it and then configure it to route traffic to one or more backend servers or pools of servers.

Application Gateway can route traffic based on various factors such as HTTP headers, URL paths, query strings, or host headers.

You can also configure Application Gateway to perform SSL offloading, which allows it to terminate SSL/TLS connections and decrypt traffic before forwarding it to the backend servers.

Application Gateway can also be configured with a Web Application Firewall (WAF) to protect web applications from common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and others.

Application Gateway supports autoscaling, which allows you to dynamically adjust the number of instances based on traffic load. You can configure autoscaling based on metrics such as CPU usage, memory usage, or requests per second.

You can also configure Application Gateway with a custom domain name, which enables you to use your own SSL certificate and improve branding.

Application Gateway can be integrated with Azure Monitor to monitor its performance, track requests and responses, and generate alerts for specific conditions or errors.

Azure Application Gateway can also be deployed with Azure Firewall or Azure Front Door for additional security and functionality.

Plan and implement an Azure Front Door, including Content Delivery Network (CDN)

Azure Front Door is a global, cloud-based content delivery network (CDN) and application delivery platform that provides secure and scalable routing of user traffic to your applications and services.

Azure Front Door uses anycast protocol, which enables traffic to be routed to the nearest Front Door point of presence (POP) and ensures low-latency and high-performance delivery of content.

Azure Front Door supports various traffic routing methods such as weighted round-robin, performance-based routing, geographic routing, and URL-based routing.

Azure Front Door also provides security features such as SSL/TLS termination, DDoS protection, and web application firewall (WAF).

You can use Azure Front Door to improve the performance and availability of your applications by caching content at edge locations and reducing the load on the origin servers.

Azure Front Door can also be integrated with Azure CDN to further improve the delivery of static and dynamic content, such as images, videos, and large files.

To implement Azure Front Door, you need to create a Front Door instance and configure it with one or more backend pools of resources, such as web applications or APIs.

You can also configure Azure Front Door with health probes to ensure that it routes traffic only to healthy backend resources.

Azure Front Door can be integrated with Azure Monitor to monitor its performance and generate alerts for specific conditions or errors.

Azure Front Door provides a unified control plane to manage traffic and security policies across multiple applications and regions.

Plan and implement a Web Application Firewall (WAF)

A Web Application Firewall (WAF) is a security solution that monitors and filters HTTP traffic between a web application and the Internet to protect against various types of attacks, such as SQL injection, cross-site scripting (XSS), and remote file inclusion (RFI).

Azure provides two types of WAF solutions: Azure Web Application Firewall (WAF) and Azure Front Door WAF.

Azure Web Application Firewall (WAF) is a layer-7 WAF that can be deployed as a standalone solution or as an add-on to Azure Application Gateway or Azure Front Door.

Azure Web Application Firewall (WAF) is based on the open-source ModSecurity engine and provides a set of pre-configured rules and policies that can be customized to meet the specific security requirements of your web applications.

Azure Front Door WAF is a network-level WAF that is integrated with Azure Front Door and provides similar security features as Azure Web Application Firewall (WAF).

To implement Azure Web Application Firewall (WAF), you need to create a WAF policy and associate it with a WAF-enabled resource, such as Azure Application Gateway or Azure Front Door.

A WAF policy is a collection of rules and settings that define how the WAF inspects and filters HTTP traffic.

Azure Web Application Firewall (WAF) provides various monitoring and reporting features, such as Azure Monitor, Azure Log Analytics, and Azure Security Center, that can be used to monitor the WAF's performance, generate alerts, and investigate security incidents.

Azure Web Application Firewall (WAF) also provides integration with Azure Security Center to provide advanced threat protection and security recommendations for your web applications.

Recommend when to use Azure DDoS Protection Standard

Azure DDoS Protection Standard is a managed DDoS protection service that provides defense against DDoS attacks for Azure resources. It is recommended to use Azure DDoS Protection Standard if you are hosting mission-critical or high-availability workloads on Azure and want to protect them from DDoS attacks that can cause service downtime or performance degradation.

Azure DDoS Protection Standard is suitable for a wide range of Azure resources, such as Virtual Machines, Virtual Machine Scale Sets, Load Balancers, Application Gateways, and Azure Kubernetes Service (AKS) clusters.

Azure DDoS Protection Standard is particularly recommended for internet-facing workloads that are exposed to the public Internet and are at a higher risk of being targeted by DDoS attacks.

Azure DDoS Protection Standard provides various features that help protect your resources against DDoS attacks, such as traffic monitoring, traffic analytics, and traffic scrubbing. It also provides automatic mitigation of DDoS attacks without any manual intervention.

Azure DDoS Protection Standard is a cost-effective solution compared to traditional on-premises DDoS protection solutions that require significant upfront investment and ongoing maintenance costs.

Azure DDoS Protection Standard provides integration with Azure Monitor and Azure Security Center to provide comprehensive monitoring and security insights into your protected resources.

Azure DDoS Protection Standard provides a Service Level Agreement (SLA) of 99.99% for protection against DDoS attacks.

Secure compute, storage, and databases (20–25%)

Plan and implement advanced security for compute

Plan and implement remote access to public endpoints, including Azure Bastion and JIT

Azure Bastion is a fully managed PaaS service that provides secure and seamless RDP/SSH connectivity to Azure Virtual Machines over the public Internet. It is recommended to use Azure Bastion instead of exposing RDP/SSH ports to the public Internet, which can increase the attack surface and security risks of your Virtual Machines.

Azure Bastion provides a secure and browser-based Remote Desktop Protocol (RDP) and Secure Shell (SSH) access to your Virtual Machines through the Azure Portal. It eliminates the need to use VPN or jump hosts for secure remote access to your Virtual Machines.

Just-in-Time (JIT) VM access is a feature that allows you to restrict access to Azure Virtual Machines by opening the RDP/SSH ports only when needed for a specific time period. JIT VM access helps reduce the attack surface of your Virtual Machines and minimize the exposure to potential threats. JIT VM access is integrated with Azure Security Center and can be enabled for Virtual Machines in a few clicks.

JIT VM access provides an approval workflow that requires users to request access to Virtual Machines and obtain approval from authorized personnel before the RDP/SSH ports are opened. JIT VM access also provides audit logs that allow you to monitor and track all access requests and activities.

When planning and implementing remote access to public endpoints, you should follow the principle of least privilege and limit access to only authorized users and roles. You should also use strong passwords and multi-factor authentication to protect your accounts and credentials.

Azure Bastion and JIT VM access can be integrated with Azure AD for authentication and authorization. This allows you to leverage the benefits of Azure AD, such as conditional access policies, identity protection, and role-based access control (RBAC).

Configure network isolation for Azure Kubernetes Service (AKS)

Azure Kubernetes Service (AKS) is a fully managed Kubernetes service that allows you to deploy and manage containerized applications at scale. AKS provides built-in integration with Azure networking features, such as Virtual Network (VNet) and Network Security Groups (NSGs), to help you secure and isolate your Kubernetes clusters.

By default, AKS clusters are deployed into a new or existing VNet, which provides network isolation and traffic filtering for the nodes and pods in the cluster. AKS also creates a subnet within the VNet for the nodes and a second subnet for the Kubernetes services.

AKS supports two modes of network isolation: Basic and Advanced. Basic network isolation is the default mode and provides network isolation between nodes and services within the cluster, but not between multiple AKS clusters. Advanced network isolation provides network isolation between multiple AKS clusters and allows you to peer VNets across different regions or subscriptions.

To configure network isolation for AKS, you can use the following Azure networking features:

- Virtual Network (VNet): AKS clusters are deployed into a VNet, which provides network isolation and traffic filtering for the nodes and pods in the cluster. You can create a new VNet or use an existing VNet to deploy your AKS cluster.
- Network Security Groups (NSGs): AKS uses NSGs to control inbound and outbound traffic to the nodes and pods in the cluster. You can create custom NSGs and associate them with the AKS subnets to control the traffic flow between the nodes and services.
- Azure Firewall: You can deploy Azure Firewall to provide centralized network security and traffic filtering for your AKS clusters. Azure Firewall can be deployed in a hub-and-spoke architecture to provide secure communication between different VNets and AKS clusters.
- VNet peering: You can use VNet peering to connect VNets in different regions or subscriptions and enable communication between different AKS clusters. VNet peering allows you to extend the VNet address space and enable cross-premises connectivity for your AKS clusters.

To configure Advanced network isolation for AKS, you can use the following Azure networking features:

- Azure Private Link: Azure Private Link allows you to expose your AKS services over a private endpoint within your VNet, instead of a public IP address. Private Link provides secure and private communication between AKS services and clients within the same VNet or peered VNets.
- Azure Private DNS: Azure Private DNS allows you to configure a private domain name system (DNS) zone within your VNet and map AKS services to a private DNS name. Private DNS provides name resolution and DNS caching for AKS services within the same VNet or peered VNets.
- VNet peering: VNet peering allows you to connect VNets in different regions or subscriptions and enable communication between different AKS clusters. VNet peering enables private communication between AKS clusters over the Microsoft backbone network.

Secure and monitor AKS

Use Azure RBAC to control access to AKS resources.

Use Azure Private Link to securely connect to AKS API server and Azure Container Registry.

Use Azure Policy to enforce compliance and security standards.

Use Azure Security Center to monitor and detect threats.

Use Kubernetes RBAC to control access to Kubernetes resources.

Use Kubernetes Network Policies to restrict network traffic between pods.

Use Kubernetes Secrets to store sensitive information.

Use Azure Monitor to monitor AKS clusters, applications, and infrastructure.

Use Log Analytics to analyze and troubleshoot issues in AKS.

Use container image scanning tools to detect vulnerabilities in container images.

Use Azure Key Vault to securely store and manage secrets and keys used by AKS applications.

Use Azure Active Directory for authentication and authorization of AKS cluster access.

Use Azure AD Pod Identity to manage identity of pods running in AKS.

Configure authentication for AKS

Use Azure Active Directory (Azure AD) to authenticate users and applications.

Configure Azure AD integration with AKS using the Azure CLI or Azure portal.

Use role-based access control (RBAC) to control access to Kubernetes resources.

Create Kubernetes service accounts for applications and assign them to specific roles or clusters.

Use Kubernetes secrets to store and manage sensitive information like API keys and credentials.

Use Azure AD Pod Identity to assign identities to pods running in AKS.

Use Kubernetes Authentication and Authorization plugins to control access to Kubernetes resources.

Use Azure AD groups to manage access to AKS clusters and resources.

Use Azure AD Application Roles to assign roles to applications and manage access to AKS resources.

Use Azure AD Conditional Access policies to enforce access policies and security controls.

Configure security monitoring for Azure Container Instances (ACIs)

Use Azure Security Center to monitor ACI containers for vulnerabilities and threats.

Enable container monitoring in Azure Monitor to collect and analyze logs and metrics from ACI containers.

Use Azure Log Analytics to monitor and troubleshoot issues in ACI containers.

Enable Azure Network Watcher to monitor network traffic to and from ACI containers.

Use Azure Application Insights to monitor application performance and errors in ACI containers.

Use container image scanning tools to detect vulnerabilities in container images used by ACI.

Use Azure Key Vault to securely store and manage secrets and keys used by ACI applications.

Use Azure AD Pod Identity to manage identity of pods running in ACI.

Use Azure AD authentication and authorization to control access to ACI resources.

Use RBAC to control access to ACI resources.

Enable auditing of ACI containers using Azure Policy or Azure Security Center.

Configure security monitoring for Azure Container Apps (ACAs)

Use Azure Security Center to monitor ACA containers for vulnerabilities and threats.

Enable container monitoring in Azure Monitor to collect and analyze logs and metrics from ACA containers.

Use Azure Log Analytics to monitor and troubleshoot issues in ACA containers.

Enable Azure Network Watcher to monitor network traffic to and from ACA containers.

Use Azure Application Insights to monitor application performance and errors in ACA containers.

Use container image scanning tools to detect vulnerabilities in container images used by ACA.

Use Azure Key Vault to securely store and manage secrets and keys used by ACA applications.

Use Azure AD Pod Identity to manage identity of pods running in ACA.

Use Azure AD authentication and authorization to control access to ACA resources.

Use RBAC to control access to ACA resources.

Enable auditing of ACA containers using Azure Policy or Azure Security Center.

Manage access to Azure Container Registry (ACR)

Use Azure AD to authenticate users and services to ACR.

Use ACR's built-in roles (such as ACR Administrator, ACR Contributor, and ACR Reader) to grant access to ACR resources.

Use Azure role-based access control (RBAC) to grant granular access to ACR resources, such as repositories and tags.

Use Azure AD groups to manage access to ACR resources for multiple users and services.

Use ACR webhooks to integrate with external services and trigger actions based on events in ACR, such as new image pushes or repository deletions.

Use ACR geo-replication to replicate ACR images across multiple regions for improved availability and performance.

Use Azure Private Link to securely access ACR over a private endpoint within your virtual network.

Use ACR's built-in policies to enforce compliance and security requirements on container images stored in ACR.

Use Azure Policy to enforce organizational policies and compliance requirements on ACR resources.

Configure disk encryption, including Azure Disk Encryption (ADE), encryption as host, and confidential disk encryption

Azure Disk Encryption (ADE) is a built-in disk encryption solution for Azure Virtual Machines (VMs) that uses the BitLocker feature of Windows and the dm-crypt feature of Linux to encrypt data at rest.

ADE uses Azure Key Vault to manage and store encryption keys.

Encryption as host is a new feature in Azure that allows customers to bring their own encryption keys (BYOK) for disk encryption on Azure VMs. This feature uses Azure Disk Encryption (ADE) to perform the actual encryption of the disks, but the encryption keys are managed by the customer.

Confidential disk encryption is a new feature in Azure that provides encryption of data at rest using hardware-based Trusted Platform Module (TPM) technology. This feature is available for Azure VMs and Azure Managed Disks, and uses virtualization-based security (VBS) to protect encryption keys and data.

Confidential disk encryption requires VMs to be deployed on Azure Confidential Computing (ACC) infrastructure.

To configure disk encryption using ADE, you need to create an Azure Key Vault and configure a key and a key policy for ADE to use.

You also need to enable encryption on the VM's OS disk or data disks, either during VM creation or by using Azure PowerShell or Azure CLI.

Encryption as host requires you to bring your own encryption keys and upload them to Azure Key Vault.

You also need to configure Azure Disk Encryption (ADE) to use the customer-managed keys for disk encryption.

Confidential disk encryption requires you to deploy VMs on ACC infrastructure and configure the disks to use the confidential computing technology.

Recommend security configurations for Azure API Management

Use Azure AD for authentication: Azure API Management allows you to integrate with Azure Active Directory (Azure AD) to authenticate users and authorize access to your APIs. This provides a secure and scalable way to manage access to your APIs and can be integrated with other Azure services for advanced security features like conditional access and multi-factor authentication.

Use HTTPS for secure communication: Use HTTPS to encrypt the communication between clients and your API Management instance. You can use a custom domain name and a certificate from a trusted certificate authority (CA) to secure your API Management instance.

Implement rate limiting: Implementing rate limiting can help prevent abuse and protect your APIs from excessive traffic or denial-of-service (DoS) attacks. You can configure rate limits based on IP address, user, or subscription.

Use IP restrictions: Use IP restrictions to allow access to your APIs only from trusted IP addresses. This can help prevent unauthorized access and protect your APIs from attacks like cross-site request forgery (CSRF) and SQL injection.

Implement OAuth2.0 for authorization: Azure API Management supports OAuth2.0 authorization flows for granting access to your APIs. This can help you control access to your APIs and manage user permissions.

Use API keys: Use API keys to authenticate clients and track usage of your APIs. You can create multiple keys for each API and revoke them as needed to control access.

Implement logging and monitoring: Use Azure Monitor and API Management logs to monitor usage and detect suspicious activity. You can also use Application Insights to monitor performance and troubleshoot issues.

Plan and implement security for storage

Configure access control for storage accounts

Use Azure Role-Based Access Control (RBAC): Azure RBAC allows you to assign roles to users, groups, and applications to control access to Azure resources. You can assign roles at the subscription, resource group, or resource level.

Grant permissions: Use RBAC to grant permissions to users or groups to access specific storage accounts. You can assign roles such as Storage Account Contributor, Storage Account Owner, or Storage Account Reader to control access to the storage account.

Use shared access signatures (SAS): Use SAS to grant temporary access to specific resources or containers within a storage account. You can set permissions and expiration times for SAS tokens.

Use virtual networks: Use virtual networks to restrict access to storage accounts to specific IP addresses or ranges. You can configure firewall and virtual network settings to allow access from specific virtual networks and subnets.

Use service endpoints: Use service endpoints to allow access to storage accounts only from within a virtual network. This can help secure your storage accounts by limiting access to only trusted networks.

Use network rules: Use network rules to allow or deny access to storage accounts based on IP address ranges. You can configure network rules to allow access from specific IP addresses or ranges and block all other traffic.

Monitor access: Use Azure Monitor to monitor access to storage accounts and detect suspicious activity. You can enable logging and auditing to track changes and access to your storage accounts.

Manage life cycle for storage account access keys

Access keys are used to authenticate requests to Azure storage accounts.

Storage accounts have two access keys, which are used for redundancy and to allow for key rotation.

Azure recommends using Azure Active Directory (Azure AD) for authentication to storage accounts whenever possible.

Access keys should be rotated regularly for security purposes.

Azure provides the option to regenerate access keys manually or automatically.

To regenerate access keys automatically, you can configure a key rotation policy for your storage account. This policy specifies the interval at which keys are regenerated.

Azure also provides the option to manage access keys programmatically using the Azure SDK or PowerShell.

Best practices for managing access keys include storing them securely, limiting access to them, and monitoring access to them for suspicious activity.

Select and configure an appropriate method for access to Azure Files

Azure Files is a fully managed file share solution in the cloud.

There are several methods for accessing Azure Files, including SMB (Server Message Block), NFS (Network File System), and REST API.

SMB is the most commonly used method for accessing Azure Files, and is compatible with Windows and Linux operating systems.

NFS is primarily used by Linux operating systems and is supported in preview mode in Azure.

REST API can be used for programmatic access to Azure Files, and is supported by a wide range of programming languages and platforms.

Access to Azure Files can be configured using shared access signatures (SAS), Azure AD authentication, or network-based access control.

SAS provides a secure way to grant limited access to Azure Files, and can be used to restrict access to specific files or folders.

Azure AD authentication allows users to access Azure Files using their Azure AD credentials, which can be more secure than using SAS.

Network-based access control can be used to restrict access to Azure Files based on IP address ranges or virtual network service endpoints.

When selecting a method for access to Azure Files, consider factors such as security, compatibility with your operating system and applications, and ease of configuration and management.

Select and configure an appropriate method for access to Azure Blob Storage

Azure Portal: You can use the Azure portal to upload and download files from Azure Blob Storage. You can also use it to manage your blob storage accounts and containers.

Azure Storage Explorer: Azure Storage Explorer is a free, standalone application that allows you to easily work with Azure Blob Storage. It is available for Windows, macOS, and Linux.

Azure PowerShell: You can use Azure PowerShell to create, manage, and delete Azure Blob Storage containers and blobs. PowerShell scripts can also be used to automate tasks.

Azure CLI: Azure CLI is a command-line interface that you can use to manage Azure resources, including Azure Blob Storage. It is available on Windows, macOS, and Linux.

REST API: The Azure Blob Storage REST API provides a way to programmatically access Blob Storage. This allows you to create, manage, and delete blobs and containers programmatically from your applications.

SDKs: Microsoft provides SDKs for a variety of programming languages, including .NET, Java, Python, and Node.js. These SDKs allow you to interact with Azure Blob Storage programmatically and integrate it into your applications.

Select and configure an appropriate method for access to Azure Tables

Azure Storage Explorer: Azure Storage Explorer is a free, standalone app from Microsoft that enables you to work with Azure Storage data on Windows, macOS, and Linux. It provides a graphical user interface to browse, manage, and access Azure Table data.

Azure Portal: You can access and manage Azure Tables via the Azure Portal. Navigate to the storage account that contains the table and select the table you want to access.

Azure Storage REST API: You can use the Azure Storage REST API to programmatically access Azure Tables. This method requires you to write code to send HTTP requests and receive responses that contain table data.

Azure Storage Client Libraries: You can use the Azure Storage Client Libraries to access Azure Tables programmatically. The client libraries provide a set of abstractions that simplify working with Azure Tables, and they support multiple programming languages and platforms, including .NET, Java, Python, and Node.js.

Select and configure an appropriate method for access to Azure Queues

Azure Queues allow applications to asynchronously communicate between components by passing messages or files between components. Azure Queues supports two types of authentication to access the queue service:

- **Shared Key Authorization:** This is a simple and commonly used method that requires the storage account name and account key to authenticate the request.
- **Shared Access Signatures (SAS):** SAS provides a more granular control over access to the queue service, allowing you to specify access permissions, start and expiry times for the permission, and the IP address range for the permission.

To access Azure Queues, you can use the following methods:

- **Azure Portal:** You can use the Azure portal to manage your queues and perform operations such as creating a new queue, adding messages to a queue, and monitoring the status of your queues.

- **Azure Storage SDK:** You can use the Azure Storage SDK for .NET, Java, Python, Node.js, Ruby, and other programming languages to access queues programmatically.
- **REST API:** You can use the REST API to access queues programmatically. You can use any programming language that supports HTTP requests to interact with the API.
- **Azure Storage Explorer:** Azure Storage Explorer is a free, cross-platform tool that allows you to access and manage your queues, as well as other Azure storage resources, from a graphical user interface (GUI).

Select and configure appropriate methods for protecting against data security threats, including soft delete, backups, versioning, and immutable storage

Soft delete: Enables recovery of data that was deleted accidentally or maliciously. When soft delete is enabled, deleted objects are retained for a specific period before they are permanently deleted.

Backups: Azure Storage provides several backup options to help protect against data loss. These include snapshots, incremental backups, and geo-redundant storage.

Versioning: Azure Blob Storage provides versioning support, which allows you to preserve and restore previous versions of blobs. This feature helps protect against accidental or malicious overwrites or deletions.

Immutable storage: Azure Blob Storage provides immutable storage, which allows you to store data in a write-once-read-many (WORM) state. This helps protect against accidental or malicious modification or deletion of data.

Configure Bring your own key (BYOK)

Bring your own key (BYOK) is a feature in Azure that allows customers to bring their own encryption keys to encrypt and decrypt their data.

BYOK is available for several Azure services, including Azure Storage, Azure Disk Encryption, and Azure Virtual Machines.

With BYOK, customers can use their own keys to protect their data in Azure, instead of relying on keys managed by Azure.

BYOK requires the use of Azure Key Vault, a cloud-based service for storing and managing cryptographic keys, secrets, and certificates.

To configure BYOK, customers first need to create a key vault in their Azure subscription and then generate a key or upload an existing key to the key vault.

Once the key is in the key vault, customers can configure their Azure services to use the key for encryption and decryption.

BYOK provides customers with greater control over their data encryption keys, which can help them meet regulatory and compliance requirements. However, it also requires additional management overhead and increases the risk of key loss or compromise.

Enable double encryption at the Azure Storage infrastructure level

Generate or obtain a customer-managed key (CMK) from Azure Key Vault.

Configure your storage account to use a CMK as the key for encrypting blob and file data.

Configure your storage account to require encryption for all incoming requests.

Ensure that all data is encrypted before it is stored in the storage account.

Monitor and manage access to the CMK and storage account to ensure proper security.

Plan and implement security for Azure SQL Database and Azure SQL Managed Instance

Enable database authentication by using Microsoft Azure AD

Azure AD is a cloud-based identity and access management service that allows you to manage user identities and access to resources in the cloud.

Azure AD can be used to authenticate users and applications to a variety of Azure services, including Azure SQL Database, Azure Synapse Analytics, and Azure Data Lake Storage Gen2.

By using Azure AD authentication, you can simplify user management, reduce the risk of password-based attacks, and enable single sign-on across Azure services.

To enable Azure AD authentication for an Azure SQL Database or Azure Synapse Analytics instance, you need to create an Azure AD server-level or database-level principal and grant it appropriate permissions.

You can create an Azure AD principal by using the Azure portal, Azure PowerShell, or Azure CLI.

Once you have created an Azure AD principal, you can use it to connect to your Azure SQL Database or Azure Synapse Analytics instance using Azure AD authentication.

To enable Azure AD authentication for Azure Data Lake Storage Gen2, you need to create an Azure AD application and grant it appropriate permissions.

You can create an Azure AD application by using the Azure portal or Azure PowerShell.

Once you have created an Azure AD application, you can use it to authenticate requests to your Azure Data Lake Storage Gen2 account by using OAuth 2.0.

Enable database auditing

Azure provides the capability to audit database activities such as queries, login events, and changes to schema or data.

Auditing can be enabled on the server or database level and requires an Azure Storage account to store audit logs.

To enable auditing for a server or database, navigate to the Azure portal and select the server or database, then click on "Auditing" in the menu.

Select the events to audit and configure the audit logs storage settings.

Auditing can also be configured using Azure PowerShell, Azure CLI, or ARM templates.

Audit logs can be viewed and analyzed using Azure Monitor or third-party tools.

Identify use cases for the Microsoft Purview governance portal

Microsoft Purview governance portal is used to manage and govern your organization's data. Some of the use cases for Microsoft Purview governance portal include:

Data discovery and classification - Purview helps you discover and classify your organization's data assets, and provides a unified view of your data landscape.

Data cataloging - Purview allows you to catalog your data assets, making it easy for users to find, understand, and use the data they need.

Data lineage and impact analysis - Purview provides a detailed view of the lineage of your data, helping you understand how it is transformed and used across your organization.

Data privacy and compliance - Purview helps you comply with data privacy regulations and policies by providing a centralized platform to manage data access and permissions.

Data analytics and insights - Purview integrates with other Azure services to provide powerful analytics and insights on your data assets.

Implement data classification of sensitive information by using the Microsoft Purview governance portal

To implement data classification of sensitive information by using the Microsoft Purview governance portal, you can follow these steps:

- Connect your data sources to Purview: Start by connecting your data sources to the Purview governance portal. This can be done using a range of connectors available in Purview, including for Azure Data Services, on-premises data sources, and SaaS applications.
- Discover and scan your data: Use the Purview data discovery and scanning capabilities to identify sensitive data across your data sources. You can create scans based on predefined templates, or define your own custom scans.
- Classify your data: Once your scans are complete, use the Purview classification capabilities to assign sensitivity labels to your data. Purview provides built-in classifiers for common data types such as credit card numbers, social security numbers, and personally identifiable information (PII). You can also define your own custom classifiers to identify additional sensitive data types.
- Monitor and manage data access: Use Purview's data access management capabilities to control who can access your sensitive data. You can define access policies based on user roles, and set up alerts to notify you when there are unauthorized access attempts.
- Maintain compliance: Use Purview's compliance reporting capabilities to monitor and report on your data governance and compliance efforts. Purview provides prebuilt compliance reports for common regulatory standards such as GDPR and CCPA.

Plan and implement dynamic masking

Dynamic masking is available in Azure SQL Database and SQL Server on-premises.

Dynamic masking can be used to mask sensitive data based on various criteria, such as user or role, without altering the data in the database.

Dynamic masking can mask data in various formats, such as full or partial masking, format preserving masking, or random masking.

You can implement dynamic masking using T-SQL statements or by using the Azure portal or Azure PowerShell.

Dynamic masking can be implemented on column level or row level.

When implementing dynamic masking, you should carefully consider the sensitivity of the data, the access requirements, and the applicable regulations or compliance standards.

Dynamic masking should be tested thoroughly to ensure that it is working as expected and not causing any performance issues or unintended consequences.

Implement Transparent Database Encryption (TDE)

To implement TDE in Azure SQL Database, you need to follow these steps:

1. Create a server-level firewall rule to allow traffic from Azure services. This is necessary because the encryption process is performed by an Azure service.
2. Create a server certificate or use an existing one.
3. Create a database encryption key (DEK) in the master database.
4. Encrypt the DEK with the server certificate.
5. Enable TDE for the database.

Once TDE is enabled, the database files are encrypted at rest. When a user or application connects to the database, the data is automatically decrypted, so there is no impact on the application.

It's important to note that TDE only encrypts the database files at rest. If you need to encrypt data in transit, you will need to use other encryption methods, such as SSL or TLS.

Recommend when to use Azure SQL Database Always Encrypted

Azure SQL Database Always Encrypted should be used when there is a requirement to protect sensitive data such as personally identifiable information (PII) or financial information from unauthorized access or data breaches. With Always Encrypted, sensitive data is encrypted at rest, in transit, and in use, providing end-to-end encryption and minimizing the risk of data breaches. The encryption keys are managed outside of the database, and only authorized parties have access to the keys, making it more difficult for attackers to gain access to the sensitive data. Always Encrypted is particularly useful for scenarios where third-party applications or services need to access the database, but the database owner wants to ensure that sensitive data is protected even if the third-party system is compromised.

Manage security operations (25–30%)

Plan, implement, and manage governance for security

Create, assign, and interpret security policies and initiatives in Azure Policy

Azure Policy is a service in Azure that allows you to create, assign, and manage policies that enforce rules and effects over resources in Azure.

A policy definition is a JSON file that describes the policy and its associated rules and effects. You can create policy definitions using the Azure Policy service, the Azure portal, Azure PowerShell, Azure CLI, or Azure Resource Manager templates.

A policy effect is the action that is taken when a resource violates a policy rule. There are two types of effects: deny and audit. A deny effect blocks the resource from being created or modified, while an audit effect logs the violation but allows the resource to be created or modified.

A policy initiative is a collection of policy definitions that are grouped together for ease of management. You can create custom initiatives or use built-in initiatives provided by Azure.

Initiatives are assigned to a scope, which can be a management group, subscription, or resource group. Policies within the initiative are then enforced at that scope.

You can interpret the results of policy enforcement using the Azure Policy Compliance dashboard, which provides an overview of the compliance status for your resources and policies.

Azure Policy integrates with Azure Security Center, allowing you to use policies to assess and remediate security issues in your environment.

Configure security settings by using Azure Blueprint

Azure Blueprint is a service in Azure that allows you to define a repeatable set of Azure resources that implements and adheres to an organization's standards, patterns, and requirements.

Azure Blueprint provides a framework for creating and managing compostable artifacts such as policies, role assignments, resource templates, and resource groups.

Blueprint artifacts can be versioned and published to different environments, such as development, test, or production, to ensure consistency across multiple deployments.

You can use Azure Blueprint to configure security settings by defining a set of security policies, such as network security groups, virtual network rules, encryption keys, and RBAC permissions, which are applied consistently across all Azure resources in a subscription or resource group.

When creating a blueprint, you can specify the security requirements for your environment and include the appropriate policies to enforce those requirements.

Once a blueprint is published, it can be assigned to a management group, subscription, or resource group to apply the defined policies to the resources within that scope.

You can monitor compliance and track changes to blueprint assignments using Azure Policy and Azure Resource Graph.

Azure Blueprint integrates with Azure DevOps and other CI/CD tools, allowing you to automate the deployment of blueprint artifacts and enforce security policies as part of your pipeline.

Deploy secure infrastructures by using a landing zone

A landing zone is a cloud infrastructure environment that is designed to provide a secure, scalable, and compliant foundation for hosting business workloads in the cloud.

A landing zone typically includes a set of core Azure services, such as virtual networks, security groups, storage accounts, and monitoring tools, that are deployed and configured according to best practices and security standards.

A landing zone provides a consistent and repeatable way to deploy Azure resources and enforce security policies across multiple subscriptions, resource groups, and regions.

You can deploy a landing zone using Azure native tools, such as Azure Resource Manager templates, Azure Blueprints, and Azure Policy, or third-party solutions, such as Terraform and Ansible.

A landing zone can be customized to meet the specific requirements of an organization by adding or removing services, configuring settings, and defining policies.

A landing zone can also be integrated with other Azure services, such as Azure Security Center, Azure Sentinel, and Azure Active Directory, to provide additional layers of security and compliance.

A landing zone can be updated and maintained over time using automated pipelines and configuration management tools, such as Azure DevOps, GitHub Actions, and PowerShell DSC.

A landing zone can be extended to other cloud platforms and on-premises environments by using hybrid cloud solutions, such as Azure Arc and Azure Stack.

Create and configure an Azure Key Vault

Azure Key Vault is a cloud service that allows you to store and manage cryptographic keys, certificates, and secrets, such as connection strings and passwords, in a secure and centralized location.

To create a new Azure Key Vault, you can use the Azure portal, Azure CLI, Azure PowerShell, or Azure Resource Manager templates.

When creating a new Azure Key Vault, you must specify a unique name, a resource group, a location, and an access policy that defines who can manage and access the key vault resources.

Azure Key Vault supports multiple types of keys, including RSA, EC, and symmetric keys. You can generate new keys in Azure Key Vault or import existing keys from other sources.

Azure Key Vault also supports managing digital certificates and secrets, such as passwords and connection strings, as well as storing and retrieving large data blobs.

Access to Azure Key Vault resources can be controlled using Azure RBAC, Azure AD authentication, and firewall rules. You can also enable Azure Virtual Network service endpoints to restrict access to the key vault from specific IP addresses or networks.

Azure Key Vault integrates with other Azure services, such as Azure App Service, Azure VMs, and Azure Functions, to provide secure access to keys, secrets, and certificates from within those services.

Azure Key Vault also supports backup and restore operations to protect against accidental deletion or corruption of key vault resources, as well as logging and auditing to track access and usage of key vault resources.

Recommend when to use a Dedicated HSM

Regulatory compliance: If your organization is subject to regulatory requirements for data protection, such as PCI DSS, HIPAA, or FIPS, you may need to use a Dedicated HSM to meet those requirements. Dedicated HSMs are designed to meet the highest levels of security and compliance standards and are certified by third-party auditors.

High-value assets: If you are managing highly sensitive data or assets, such as cryptographic keys for financial transactions, intellectual property, or national security, you may want to use a Dedicated HSM to protect against theft or unauthorized access. Dedicated HSMs are designed to provide tamper-resistant hardware protection for cryptographic keys and other secrets.

Scale and performance: If you need to manage a large number of cryptographic keys or require high-performance cryptographic operations, a Dedicated HSM can provide the necessary scalability and throughput to meet those requirements. Dedicated HSMs are designed to handle thousands or even millions of cryptographic operations per second, depending on the model.

Multi-tenant environments: If you are running a multi-tenant cloud environment, such as a public or private cloud, you may want to use a Dedicated HSM to provide isolation and segregation of cryptographic keys and secrets between tenants.

Dedicated HSMs can provide secure partitions and key management services to ensure that keys are not shared or exposed between tenants.

Hybrid cloud scenarios: If you need to manage cryptographic keys and secrets across multiple cloud and on-premises environments, you may want to use a Dedicated HSM to provide a consistent and secure key management solution. Dedicated HSMs can be integrated with other cloud services and on-premises applications to provide a unified key management experience.

Configure access to Key Vault, including vault access policies and Azure Role Based Access Control

Access to Azure Key Vault resources can be controlled using two main methods: Azure RBAC and Vault Access Policies.

Azure RBAC is used to grant permissions at the subscription, resource group, or resource level. You can assign roles to users, groups, or applications to control what they can do within Azure Key Vault. For example, you can grant a user the "Key Vault Contributor" role to allow them to manage keys and secrets in a specific key vault.

Vault Access Policies are used to control access to individual keys, secrets, and certificates within a key vault. You can create access policies to grant permissions to specific users, groups, or applications for specific keys, secrets, or certificates. For example, you can create an access policy to grant a specific application the ability to read a secret value from a key vault.

When creating a new access policy, you can specify the permissions (such as get, list, set, delete) that are granted to the principal (user, group, or application). You can also specify whether the principal can manage the key vault itself, or just the specific resource within the key vault.

Azure Key Vault also supports Azure AD authentication, which allows users to authenticate using their Azure AD credentials instead of using a key or certificate. You can configure Azure AD authentication when creating the key vault or by updating the key vault settings later.

Azure Key Vault also supports firewall rules, which allow you to control which IP addresses or ranges can access the key vault over the internet. You can create firewall rules when creating the key vault or by updating the key vault settings later.

When configuring access to Azure Key Vault, it's important to follow the principle of least privilege, which means granting only the necessary permissions to users, groups, or applications to perform their intended tasks. This helps minimize the risk of unauthorized access to sensitive data or resources.

Manage certificates, secrets, and keys

Azure Key Vault can be used to manage and store certificates, secrets, and keys. These can include SSL/TLS certificates, API keys, passwords, and cryptographic keys for data encryption and decryption.

To manage certificates, you can import certificates into Azure Key Vault or generate a new certificate using a certificate signing request (CSR). You can also configure certificate renewal settings to ensure that certificates are automatically renewed before they expire.

To manage secrets, you can store sensitive data such as connection strings, passwords, and API keys in Azure Key Vault. Secrets can be created and updated programmatically using Azure Key Vault APIs or through the Azure portal.

To manage keys, you can create and store cryptographic keys in Azure Key Vault. Keys can be generated using various algorithms such as RSA, AES, and elliptic curve cryptography (ECC). You can also import existing keys into Azure Key Vault or use Azure Key Vault to generate new keys.

Azure Key Vault supports various cryptographic operations such as encrypting and decrypting data using keys, signing and verifying data using keys, and wrapping and unwrapping keys. These operations can be performed programmatically using Azure Key Vault APIs or through the Azure portal.

Azure Key Vault also provides features for key rotation, versioning, and backup and restore. Key rotation allows you to periodically change cryptographic keys to help mitigate the risk of key compromise. Versioning allows you to create multiple versions of a key or certificate to support rolling upgrades and compatibility with legacy applications. Backup and restore allows you to create backups of your keys and certificates to protect against data loss.

When managing certificates, secrets, and keys in Azure Key Vault, it's important to follow best practices for security and compliance, such as using strong encryption algorithms, enforcing access controls, and monitoring key usage and access.

Configure key rotation

Key rotation is the process of regularly changing cryptographic keys to help mitigate the risk of key compromise. It is an important security practice for managing cryptographic keys.

Azure Key Vault supports key rotation for managing keys, certificates, and secrets. You can configure key rotation for a key vault, a specific key or certificate, or a specific secret.

To configure key rotation for a key vault, you can enable the key vault's "Soft Delete" feature, which allows you to recover deleted keys and enables key versioning. You can also configure a retention period for deleted keys to help prevent accidental deletion.

To configure key rotation for a specific key or certificate, you can set a rotation policy that specifies the interval at which the key or certificate should be rotated. The rotation policy can be set to a specific number of days or a specific date and time.

When a key or certificate is rotated, a new version of the key or certificate is generated and the old version is marked as inactive. The new version can then be used for encrypting or decrypting data or for signing and verifying data.

When rotating keys or certificates, it's important to consider the impact on applications that use the keys or certificates. You may need to update applications to use the new keys or certificates and ensure that old keys or certificates are no longer used.

Key rotation should be performed on a regular basis, such as every 90 days, to help mitigate the risk of key compromise. This interval can be adjusted based on the level of risk and the sensitivity of the data being protected.

When configuring key rotation, it's important to follow best practices for security and compliance, such as using strong encryption algorithms, enforcing access controls, and monitoring key usage and access.

Configure backup and recovery of certificates, secrets, and keys

Azure Key Vault provides features for backing up and restoring certificates, secrets, and keys to protect against data loss.

Backing up certificates, secrets, and keys allows you to create a secure copy of the data that can be used to recover the data in case of accidental deletion, corruption, or other types of data loss.

To backup a certificate, secret, or key in Azure Key Vault, you can use the Azure portal, Azure CLI, Azure PowerShell, or Azure Key Vault APIs. The backup can be stored in a secure location such as an Azure Storage account or on-premises storage.

To restore a backed-up certificate, secret, or key, you can use the Azure portal, Azure CLI, Azure PowerShell, or Azure Key Vault APIs. The restored data is returned to the same key vault or to a different key vault.

When restoring a certificate, secret, or key, a new version of the data is created with a new identifier. This allows you to recover the data without overwriting existing data.

It's important to secure the backups of certificates, secrets, and keys to prevent unauthorized access. Backups should be encrypted using strong encryption algorithms and stored in a secure location.

It's also important to regularly test the backup and recovery process to ensure that backups are being created correctly and can be restored in case of a data loss event.

When configuring backup and recovery of certificates, secrets, and keys, it's important to follow best practices for security and compliance, such as using strong encryption algorithms, enforcing access controls, and monitoring backup and recovery activities.

Manage security posture by using Microsoft Defender for Cloud

Identify and remediate security risks by using the Microsoft Defender for Cloud Secure Score and Inventory

Microsoft Defender for Cloud (formerly known as Azure Security Center) provides a Secure Score and Inventory dashboard that helps organizations identify and remediate security risks in their Azure environment.

The Secure Score provides a quantitative assessment of an organization's security posture by measuring how well the organization's security controls align with best practices and industry standards. The score ranges from 0 to 100, with a higher score indicating a more secure environment.

The Secure Score provides recommendations for improving an organization's security posture based on the organization's current configuration and usage of Azure services. The recommendations can be prioritized based on the potential impact and ease of implementation.

The Secure Score also provides a trend over time of the organization's security posture, allowing organizations to track their progress in improving their security posture.

The Inventory dashboard provides a comprehensive view of an organization's assets and their security status. It allows organizations to see all the Azure resources and virtual machines (VMs) in their environment and identify those that are at risk of security threats.

The Inventory dashboard provides detailed information about each asset, including its security configuration, compliance status, and risk level. This information can be used to identify and remediate security risks.

The Inventory dashboard also allows organizations to create and manage security policies to enforce security standards across their environment. Organizations can define policies to automatically enforce security controls and remediate security risks.

Microsoft Defender for Cloud also provides alerts and notifications for security events and threats detected in an organization's environment. These alerts can be used to quickly identify and remediate security risks.

When using Microsoft Defender for Cloud to identify and remediate security risks, it's important to follow best practices for security and compliance, such as keeping systems up to date, using strong authentication and access controls, and monitoring for suspicious activity.

Assess compliance against security frameworks and Microsoft Defender for Cloud

Compliance refers to the adherence to security standards, regulations, and policies. Assessing compliance is an important part of maintaining a secure and compliant environment in Azure.

Microsoft Defender for Cloud provides several compliance assessments against industry standards such as CIS, NIST, PCI DSS, and ISO 27001. These assessments help organizations measure their security posture against best practices and identify areas where they need to improve their security controls.

The compliance assessments in Microsoft Defender for Cloud provide a score for each control, along with a list of recommendations for improving the security posture.

Microsoft Defender for Cloud also provides a regulatory compliance dashboard that helps organizations assess their compliance with regulatory standards such as GDPR, HIPAA, and SOC 2. The dashboard provides a summary of the compliance status, along with detailed information about the compliance requirements and recommendations for remediation.

Microsoft Defender for Cloud provides continuous monitoring of the environment to identify security risks and compliance issues. It provides alerts and notifications for security events and threats, which can be used to quickly remediate issues and maintain compliance.

To assess compliance against security frameworks and Microsoft Defender for Cloud, organizations should regularly review and evaluate their security controls, policies, and procedures. They should also conduct regular vulnerability assessments and penetration testing to identify weaknesses in their environment.

Organizations should also implement a security and compliance management program that includes policies, procedures, and training to ensure that everyone in the organization understands their roles and responsibilities in maintaining a secure and compliant environment.

It's important to follow best practices for security and compliance, such as using strong authentication and access controls, regularly updating systems, and monitoring for suspicious activity. Organizations should also document their security controls and procedures, and regularly review and update them to ensure they remain effective.

Add industry and regulatory standards to Microsoft Defender for Cloud

Microsoft Defender for Cloud allows organizations to add their own industry and regulatory standards to the compliance assessment framework. This feature is called Custom Regulatory Compliance.

Custom Regulatory Compliance allows organizations to define their own compliance requirements and assess their compliance against those requirements.

To add a custom compliance standard, organizations must define the compliance requirements in a YAML file and upload the file to Microsoft Defender for Cloud.

The YAML file should include the compliance requirements, controls, and mappings to Microsoft Defender for Cloud's built-in compliance assessments.

Once the custom compliance standard is added, organizations can use Microsoft Defender for Cloud to assess their compliance against the standard and receive recommendations for remediation.

Custom Regulatory Compliance allows organizations to tailor their compliance assessments to their specific industry or regulatory requirements. It can also help organizations maintain compliance with internal policies and procedures.

When adding custom compliance standards to Microsoft Defender for Cloud, it's important to ensure that the requirements are clear and measurable. The controls should be specific and actionable, and the mappings to Microsoft Defender for Cloud's built-in assessments should be accurate.

It's also important to regularly review and update the custom compliance standards to ensure they remain relevant and effective.

Adding custom compliance standards to Microsoft Defender for Cloud can help organizations maintain a comprehensive view of their compliance posture and identify areas where they need to improve their security controls.

Add custom initiatives to Microsoft Defender for Cloud

Microsoft Defender for Cloud provides a set of built-in security policies and initiatives that organizations can use to monitor and enforce compliance with security best practices.

Custom initiatives in Microsoft Defender for Cloud allow organizations to define their own security policies and initiatives and enforce them across their environment.

Custom initiatives can be created using Azure Policy, which is a service that helps organizations enforce compliance with policies and rules for their resources.

To create a custom initiative, organizations define the policy using Azure Policy's policy definition language (PDL) and then publish it to their environment.

Custom initiatives can be created to enforce specific security controls, such as requiring multi-factor authentication for administrative accounts or ensuring that all virtual machines are encrypted.

Once a custom initiative is published, it can be assigned to resources within the environment. Microsoft Defender for Cloud will then monitor compliance with the initiative and provide recommendations for remediation.

Custom initiatives can help organizations maintain a consistent security posture across their environment and enforce compliance with internal policies and procedures.

When creating custom initiatives, it's important to ensure that the policies are clear and measurable. The policies should be specific and actionable, and the enforcement should be aligned with the organization's security objectives.

It's also important to regularly review and update the custom initiatives to ensure they remain relevant and effective.

Adding custom initiatives to Microsoft Defender for Cloud can help organizations improve their security posture and reduce the risk of security breaches and compliance issues.

Connect hybrid cloud and multi-cloud environments to Microsoft Defender for Cloud

Microsoft Defender for Cloud can be integrated with hybrid cloud and multi-cloud environments to provide a unified view of an organization's security posture.

To connect hybrid cloud and multi-cloud environments to Microsoft Defender for Cloud, organizations can use connectors or agents that are designed to work with specific cloud platforms.

Connectors and agents provide a way to collect security data from cloud resources, such as virtual machines, storage accounts, and databases, and send it to Microsoft Defender for Cloud for analysis and remediation.

Microsoft provides connectors and agents for popular cloud platforms, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure.

The connectors and agents can be configured to collect different types of security data, such as logs, metrics, and events, and send them to Microsoft Defender for Cloud for analysis.

Once the security data is received by Microsoft Defender for Cloud, it can be analyzed and compared to known security threats and vulnerabilities. Microsoft Defender for Cloud can then provide recommendations for remediation to help organizations improve their security posture.

Integrating hybrid cloud and multi-cloud environments with Microsoft Defender for Cloud can help organizations maintain a unified view of their security posture and identify security risks across all of their cloud resources.

When integrating hybrid cloud and multi-cloud environments with Microsoft Defender for Cloud, it's important to ensure that the connectors and agents are properly configured and secured. It's also important to ensure that the security data is being sent to Microsoft Defender for Cloud in a timely and secure manner.

Adding hybrid cloud and multi-cloud environments to Microsoft Defender for Cloud can help organizations improve their security posture and reduce the risk of security breaches and compliance issues.

Identify and monitor external assets by using Microsoft Defender External Attack Surface Management

Microsoft Defender External Attack Surface Management (EASM) is a cloud-based solution that helps organizations identify and monitor their external attack surface.

EASM can scan the internet and other external data sources to identify an organization's assets that are publicly exposed, such as web applications, domain names, and IP addresses.

EASM provides a unified view of an organization's external attack surface, allowing organizations to identify and prioritize vulnerabilities and threats.

Once the external assets are identified, EASM can continuously monitor them for changes and provide alerts when new assets are added or removed or when vulnerabilities are identified.

EASM can also integrate with other Microsoft security solutions, such as Microsoft Defender for Endpoint and Azure Sentinel, to provide a comprehensive view of an organization's security posture.

EASM can be configured to comply with regulations such as General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), and Payment Card Industry Data Security Standard (PCI DSS) by enabling options such as the deletion of collected data.

By identifying and monitoring external assets with EASM, organizations can reduce the risk of security breaches and compliance issues by proactively addressing vulnerabilities and threats in their external attack surface.

To use EASM, organizations need to onboard their external assets to the EASM portal, where they can be managed and monitored.

EASM can also provide recommendations for improving an organization's external attack surface, such as removing unused or unnecessary assets and implementing security controls for critical assets.

EASM can help organizations improve their security posture and reduce the risk of external attacks by providing a centralized view of their external attack surface and identifying potential vulnerabilities and threats.

Configure and manage threat protection by using Microsoft Defender for Cloud

Enable workload protection services in Microsoft Defender for Cloud, including Microsoft Defender for Storage, Databases, Containers, App Service, Key Vault, Resource Manager, and DNS

Microsoft Defender for Cloud provides workload protection services for various cloud resources, such as storage, databases, containers, app service, key vault, resource manager, and DNS.

To enable workload protection services in Microsoft Defender for Cloud, organizations need to onboard the respective resources to the Microsoft Defender for Cloud portal and configure the appropriate policies for each workload.

Microsoft Defender for Storage provides security for Azure Blob Storage by analyzing access patterns, detecting anomalies, and blocking suspicious activities. It can also integrate with Azure Sentinel for advanced threat detection and remediation.

Microsoft Defender for Databases provides security for Azure SQL Database and Azure Database for MySQL by analyzing queries, detecting anomalies, and blocking malicious activities. It can also integrate with Azure Sentinel for advanced threat detection and remediation.

Microsoft Defender for Containers provides security for container workloads running in Azure Kubernetes Service (AKS) by analyzing images, detecting vulnerabilities, and blocking malicious activities. It can also integrate with Azure Security Center for enhanced security insights.

Microsoft Defender for App Service provides security for Azure App Service by analyzing requests, detecting anomalies, and blocking malicious activities. It can also integrate with Azure Sentinel for advanced threat detection and remediation.

Microsoft Defender for Key Vault provides security for Azure Key Vault by monitoring access, detecting anomalies, and blocking suspicious activities. It can also integrate with Azure Sentinel for advanced threat detection and remediation.

Microsoft Defender for Resource Manager provides security for Azure Resource Manager by monitoring activities, detecting anomalies, and blocking malicious activities. It can also integrate with Azure Sentinel for advanced threat detection and remediation.

Microsoft Defender for DNS provides security for Azure DNS by analyzing queries, detecting anomalies, and blocking malicious activities. It can also integrate with Azure Sentinel for advanced threat detection and remediation.

Enabling workload protection services in Microsoft Defender for Cloud can help organizations detect and remediate security threats in their cloud resources, reducing the risk of security breaches and compliance issues.

When enabling workload protection services in Microsoft Defender for Cloud, it's important to configure the appropriate policies and alerts to ensure that suspicious activities are detected and remediated in a timely manner.

Microsoft Defender for Cloud provides a centralized view of an organization's cloud security posture, allowing organizations to identify and prioritize vulnerabilities and threats across their workloads.

Configure Microsoft Defender for Servers

Microsoft Defender for Servers is a security solution that provides advanced threat protection for Windows Servers running on-premises, in the cloud, or in hybrid environments.

To configure Microsoft Defender for Servers, organizations need to install and activate the Microsoft Defender for Servers agent on their Windows Servers.

After installation, the Microsoft Defender for Servers agent can be configured using Group Policy, PowerShell, or the Microsoft Endpoint Manager console.

Organizations can configure Microsoft Defender for Servers to perform various security tasks, such as real-time malware detection and removal, network protection, attack surface reduction, and endpoint detection and response.

Microsoft Defender for Servers can also integrate with Azure Security Center to provide a centralized view of an organization's security posture and to identify and prioritize vulnerabilities and threats across their Windows Servers.

Microsoft Defender for Servers supports various deployment modes, such as standalone, hybrid, and cloud-managed. Organizations can choose the deployment mode that best suits their requirements and environment.

To ensure optimal performance and protection, organizations should regularly update the Microsoft Defender for Servers agent and configure the appropriate policies and exclusions for their Windows Servers.

Microsoft Defender for Servers can help organizations detect and remediate security threats in their Windows Servers, reducing the risk of security breaches and compliance issues.

When configuring Microsoft Defender for Servers, it's important to follow best practices and industry standards to ensure that the security solution is configured correctly and effectively.

Microsoft Defender for Servers provides a comprehensive set of security features and capabilities that can help organizations protect their Windows Servers from advanced threats and attacks.

Configure Microsoft Defender for Azure SQL Database

Microsoft Defender for Azure SQL Database is a security solution that provides advanced threat protection for Azure SQL Database instances.

To configure Microsoft Defender for Azure SQL Database, organizations need to enable the Advanced Threat Protection (ATP) feature for their Azure SQL Database instance. ATP can be enabled through the Azure Portal or Azure PowerShell.

After enabling ATP, organizations can configure various security policies, such as vulnerability assessment, threat detection, and data classification, to protect their Azure SQL Database instance.

Microsoft Defender for Azure SQL Database can help organizations detect and remediate security threats, such as SQL injection attacks, brute-force attacks, and data exfiltration attempts.

Microsoft Defender for Azure SQL Database can also integrate with Azure Security Center to provide a centralized view of an organization's security posture and to identify and prioritize vulnerabilities and threats across their Azure SQL Database instances.

When configuring Microsoft Defender for Azure SQL Database, it's important to follow best practices and industry standards to ensure that the security solution is configured correctly and effectively.

Microsoft Defender for Azure SQL Database provides a comprehensive set of security features and capabilities that can help organizations protect their Azure SQL Database instances from advanced threats and attacks.

Organizations can also leverage other Azure security services, such as Azure Key Vault and Azure Active Directory, to enhance the security of their Azure SQL Database instances and to comply with industry and regulatory requirements.

Regular monitoring, maintenance, and testing of Microsoft Defender for Azure SQL Database can help organizations ensure that their Azure SQL Database instances are protected against evolving security threats and vulnerabilities.

Manage and respond to security alerts in Microsoft Defender for Cloud

Microsoft Defender for Cloud provides real-time threat protection and security insights across an organization's hybrid cloud and multi-cloud environments.

When a security threat is detected by Microsoft Defender for Cloud, an alert is generated and sent to the appropriate security personnel for investigation and response.

To manage and respond to security alerts in Microsoft Defender for Cloud, organizations can use the Microsoft Defender for Cloud portal or integrate with their existing Security Information and Event Management (SIEM) solution.

The Microsoft Defender for Cloud portal provides a unified view of an organization's security posture, including security alerts, security recommendations, and compliance assessments.

Organizations can configure the alert notifications and severity levels based on their security policies and requirements.

When a security alert is generated, organizations should investigate and triage the alert to determine its severity, impact, and root cause. This may involve analyzing the affected resources, reviewing the threat intelligence, and conducting a risk assessment.

Depending on the severity and type of the security threat, organizations may need to take immediate actions, such as blocking the attacker, containing the threat, or remediating the vulnerability.

Microsoft Defender for Cloud provides automated response capabilities that enable organizations to quickly mitigate security threats, such as isolating the affected resource, quarantining the suspicious file, or applying a security patch.

Organizations should also review and analyze the security alerts and response data to identify trends, patterns, and gaps in their security posture. This can help organizations improve their security operations and prevent future security incidents.

Regular monitoring and tuning of security alerts and response processes can help organizations ensure that their security operations are effective and efficient in detecting, mitigating, and responding to security threats.

Configure workflow automation by using Microsoft Defender for Cloud

Microsoft Defender for Cloud provides workflow automation capabilities to help organizations automate and streamline their security operations and incident response processes.

Workflow automation in Microsoft Defender for Cloud is based on the integration with Microsoft Power Automate, which is a cloud-based service that enables organizations to create automated workflows and business processes.

To configure workflow automation in Microsoft Defender for Cloud, organizations can create custom workflows in Microsoft Power Automate and use the Microsoft Defender for Cloud connectors to connect to the security alerts and actions in Microsoft Defender for Cloud.

Organizations can use workflow automation in Microsoft Defender for Cloud to automate various security operations, such as incident response, threat remediation, and compliance management.

For example, organizations can create a workflow that automatically assigns a security incident to a specific security team member, sends an email notification to the security manager, and creates a ticket in the organization's incident management system.

Microsoft Defender for Cloud also provides pre-built workflows, known as playbooks, which organizations can use as a starting point for their own custom workflows. Playbooks are designed to automate common security scenarios, such as ransomware attacks, compromised accounts, and phishing attempts.

Organizations can customize the pre-built playbooks to fit their specific security policies and requirements, and they can also create their own playbooks from scratch.

Workflow automation in Microsoft Defender for Cloud can help organizations improve their security operations by reducing manual efforts, increasing efficiency and accuracy, and enabling faster response to security incidents.

Regular monitoring and testing of workflow automation in Microsoft Defender for Cloud can help organizations ensure that their automated workflows are effective and reliable in detecting and responding to security threats.

Evaluate vulnerability scans from Microsoft Defender for Server

Microsoft Defender for Server provides vulnerability assessment capabilities that help organizations identify and remediate security vulnerabilities in their Windows Server environments.

Vulnerability assessment in Microsoft Defender for Server is based on the integration with the Common Vulnerabilities and Exposures (CVE) database, which is a publicly available repository of security vulnerabilities and exposures.

To perform a vulnerability assessment in Microsoft Defender for Server, organizations can initiate a scan of their Windows Server environment, either manually or on a scheduled basis.

The vulnerability scan in Microsoft Defender for Server checks the installed software and operating system components for known vulnerabilities and exposures, based on the CVE database.

After the vulnerability scan is complete, Microsoft Defender for Server provides a report of the identified vulnerabilities, along with their severity level and recommended remediation actions.

Organizations can use the vulnerability assessment report from Microsoft Defender for Server to prioritize their remediation efforts and address the most critical vulnerabilities first.

Organizations can also customize the vulnerability assessment settings in Microsoft Defender for Server, such as the frequency of the scans, the scope of the assessment, and the severity level thresholds for reporting vulnerabilities.

Microsoft Defender for Server also provides integration with other security tools and services, such as Microsoft Intune, System Center Configuration Manager (SCCM), and Azure Security Center, to provide a more comprehensive and integrated approach to vulnerability management.

Regular monitoring and testing of the vulnerability assessment in Microsoft Defender for Server can help organizations ensure that their Windows Server environments are secure and compliant with their security policies and standards.

Configure and manage security monitoring and automation solutions

Monitor security events by using Azure Monitor

Azure Monitor is a cloud-based monitoring solution that helps organizations collect, analyze, and act on telemetry data from their applications, infrastructure, and services in Azure and on-premises environments.

Azure Monitor provides a unified platform for monitoring and alerting on security events and threats across the entire organization, including Azure resources, hybrid cloud, and multi-cloud environments.

Azure Monitor offers several security-related features, including log analytics, metrics, alerts, and dashboards, that can help organizations detect and respond to security threats and events.

Azure Monitor can collect security-related telemetry data from various sources, such as Azure Security Center, Azure Active Directory, Azure Key Vault, Azure Firewall, and Azure Policy, among others.

Azure Monitor uses Log Analytics to collect, store, and analyze security-related logs and events from different sources. Organizations can use Log Analytics to perform advanced analytics and correlation across different data sources, detect security incidents, and investigate and respond to security threats.

Azure Monitor provides pre-built dashboards and visualizations that can help organizations monitor and analyze their security posture and performance, including security alerts, vulnerabilities, compliance, and threat intelligence.

Organizations can use Azure Monitor Alerts to create custom alerts and notifications based on specific security events and conditions, such as failed login attempts, malware detections, or unauthorized access to resources.

Azure Monitor also integrates with third-party security tools and services, such as SIEM solutions, threat intelligence platforms, and incident response workflows, to provide a more comprehensive and integrated approach to security monitoring and response.

Regular monitoring and testing of the security monitoring capabilities in Azure Monitor can help organizations ensure that their security posture is effective and aligned with their security policies and standards.

Configure data connectors in Microsoft Sentinel

Microsoft Sentinel is a cloud-native security information and event management (SIEM) solution that provides intelligent security analytics and threat intelligence across the entire enterprise, including hybrid and multi-cloud environments.

Sentinel collects and analyzes data from various sources, such as logs, events, alerts, and telemetry, to detect and respond to security threats and incidents.

Sentinel uses data connectors to ingest data from different sources into the SIEM platform. A data connector is a pre-built or custom-built integration that provides a secure and reliable way to bring data into Sentinel.

Sentinel supports a wide range of data connectors for different types of data sources, including Azure services, Microsoft 365, third-party security tools, and custom applications.

Sentinel provides a user-friendly interface to configure and manage data connectors. Users can access the data connectors from the "Data connectors" tab in the Sentinel workspace.

To configure a data connector, users need to provide the required configuration settings, such as credentials, endpoints, and data types, to connect to the data source. Sentinel provides step-by-step instructions and validation checks to ensure that the data connector is configured correctly.

Sentinel supports different types of data connectors, such as API-based connectors, file-based connectors, and streaming connectors, to accommodate different types of data sources and data formats.

Sentinel also provides built-in support for common data enrichment and transformation scenarios, such as parsing, filtering, enriching, and aggregating data, to improve the quality and relevance of the data in the SIEM platform.

Sentinel provides several features and tools to manage and monitor data connectors, such as connector health status, data ingestion metrics, and data mapping and parsing rules.

Regular testing and validation of data connectors can help organizations ensure that their data sources are properly integrated into Sentinel and that the SIEM platform is receiving the relevant and high-quality data needed to detect and respond to security threats and incidents.

Create and customize analytics rules in Microsoft Sentinel

Microsoft Sentinel provides a rich set of built-in analytics rules that use machine learning and other advanced techniques to detect known and unknown security threats and incidents across different data sources and environments.

Analytics rules are pre-configured queries or algorithms that analyze data in real-time or near real-time to identify suspicious activities or patterns that may indicate a security threat or incident.

Sentinel provides a user-friendly interface to create and customize analytics rules to meet specific security requirements and use cases. Users can access the analytics rules from the "Analytics" tab in the Sentinel workspace.

To create an analytics rule, users need to define the query or algorithm that analyzes the data, the data sources and fields that the rule applies to, and the conditions and thresholds that trigger the rule.

Sentinel supports different types of analytics rules, such as simple queries, advanced queries, machine learning-based rules, and custom code-based rules, to accommodate different levels of complexity and flexibility.

Sentinel provides several features and tools to manage and monitor analytics rules, such as rule status, rule performance, and rule history.

Customizing analytics rules can help organizations improve the accuracy and relevance of their security detections and reduce false positives and false negatives. Customization can also help organizations meet their specific security requirements and compliance mandates.

Regular review and tuning of analytics rules can help organizations ensure that their security detections are up-to-date and effective against emerging threats and attack techniques. This can help organizations stay ahead of the threat landscape and minimize the risk of security breaches and incidents.

Evaluate alerts and incidents in Microsoft Sentinel

Microsoft Sentinel provides a unified view of security alerts and incidents across different data sources and environments, enabling security teams to quickly and efficiently investigate and respond to security threats and incidents.

Alerts are generated by analytics rules or other detection mechanisms in Sentinel, indicating potential security threats or suspicious activities that require further investigation.

Incidents are collections of related alerts that represent a security threat or incident that requires a response from the security team.

Sentinel provides a user-friendly interface to view, manage, and investigate alerts and incidents in the "Incidents" tab of the Sentinel workspace.

Users can triage, prioritize, and assign alerts and incidents to different team members based on their severity, impact, and relevance to the organization's security posture and objectives.

Sentinel provides several features and tools to help users investigate and analyze alerts and incidents, such as timeline view, entity explorer, investigation graph, and built-in threat intelligence.

Users can also collaborate and communicate with each other through the incident comments and share their findings and recommendations for remediation and mitigation.

Sentinel supports automation and orchestration of incident response through integration with other Microsoft and third-party services and tools, such as Azure Logic Apps and Power Automate.

Regular evaluation and improvement of incident response processes and procedures can help organizations enhance their ability to detect and respond to security threats and incidents effectively and efficiently. This can help organizations reduce the impact and cost of security breaches and incidents and improve their overall security posture.

Configure automation in Microsoft Sentinel

Microsoft Sentinel provides several features and tools for automating and orchestrating security operations and incident response, such as playbooks, automation rules, and Azure Logic Apps integration.

Playbooks are collections of tasks and actions that can be triggered by alerts or incidents in Sentinel to automate and streamline incident response processes. Playbooks can be created and customized using the Sentinel playbooks designer, which provides a drag-and-drop interface for building workflows and connecting different tasks and actions.

Playbooks can perform a wide range of actions, such as sending notifications, executing scripts, querying external APIs, creating tickets, blocking IP addresses, and more. Playbooks can also be integrated with other Azure services, such as Logic Apps and Azure Functions, to extend their capabilities and automate complex workflows.

Automation rules are a type of rule in Sentinel that can automatically respond to alerts based on predefined conditions and actions. Automation rules can be used to perform tasks such as closing, assigning, or re-analyzing alerts, creating incidents, triggering playbooks, or sending notifications.

Azure Logic Apps is a cloud-based service that allows users to create and run workflows that integrate with other Azure services and external systems. Sentinel can integrate with Logic Apps to enable more complex and advanced automation scenarios, such as enriching alerts with external threat intelligence, triggering remediation actions in external systems, or creating custom workflows that combine Sentinel data with other data sources.

Users can also use PowerShell, REST API, or other scripting and automation tools to integrate with Sentinel and automate security operations and incident response. Sentinel provides a rich set of REST APIs that allow users to query and manipulate Sentinel data programmatically, and PowerShell cmdlets that allow users to automate common tasks in Sentinel.

Regular review and optimization of automation rules, playbooks, and workflows can help organizations enhance their incident response capabilities, improve their efficiency and effectiveness, and reduce the risk of human error and delay in response to security threats and incidents.

Diagrams / SubTopics that may be helpful for exam

Azure BluePrints

Azure Blueprints is a service provided by Microsoft Azure that allows users to create a set of pre-defined resources, policies, and configurations that can be automatically deployed to multiple subscriptions or environments within Azure. Azure Blueprints are essentially a declarative way to orchestrate the deployment of various Azure resources, including virtual machines, storage accounts, networking, and more.

Azure Blueprints can be used to establish a baseline set of configurations, policies, and security controls that can be consistently applied across an organization's Azure environment. This can help organizations ensure that their Azure deployments adhere to industry best practices, security standards, and compliance requirements.

Some of the key features of Azure Blueprints include the ability to:

- Define and version Azure resources and configurations
- Assign Azure Blueprint to multiple subscriptions or environments
- Implement Azure policies and role-based access control (RBAC)
- Automate the deployment of Azure resources and configurations

In summary, Azure Blueprints provide a way for organizations to streamline the deployment of Azure resources and ensure that their Azure environment is consistently configured and secured.

Azure Automation State Configuration

Azure Automation State Configuration is a service provided by Microsoft Azure that allows users to define, manage, and enforce configuration settings on their Azure virtual machines and physical servers. With Azure Automation State Configuration, users can create and manage a desired state configuration (DSC) that specifies the state they want their machines to be in, and then deploy and monitor that configuration across their environment.

Some of the key features of Azure Automation State Configuration include:

- A declarative, PowerShell-based configuration management system

- Integration with Azure Automation to allow for remote configuration management of virtual machines and physical servers

- Support for configuring Windows and Linux machines

- Custom script support to allow for the execution of additional configuration tasks

Azure Automation State Configuration can be used to ensure that an organization's Azure environment is configured according to best practices and compliance requirements, and to maintain the desired state of the environment over time. By defining a desired state configuration, users can easily apply and enforce consistent configuration settings across their environment, reducing the risk of configuration drift and improving overall security and compliance.

In summary, Azure Automation State Configuration provides a way for organizations to manage and enforce configuration settings on their Azure virtual machines and physical servers, and to maintain a desired state configuration across their environment.

Microsoft Graph

Microsoft Graph is a service provided by Microsoft that enables developers to access and interact with data from a variety of Microsoft services, including Office 365, OneDrive, SharePoint, and more, through a single REST API endpoint. With Microsoft

Graph, developers can build applications that integrate with Microsoft services and can access and manipulate data across those services.

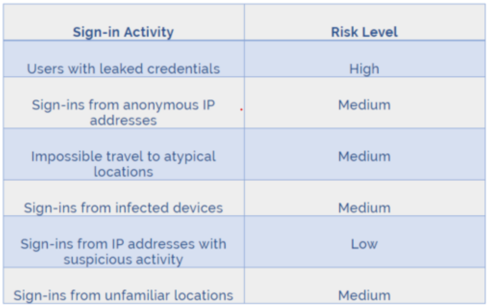
Some of the key features of Microsoft Graph include:

- A single API endpoint for accessing data across multiple Microsoft services
- Rich data models that provide contextual information about data objects
- Support for querying, creating, updating, and deleting data objects
- Authentication and authorization through Azure Active Directory (AAD)

Microsoft Graph can be used to build a wide range of applications, including productivity apps, business intelligence tools, and automation workflows. By providing a unified API endpoint for accessing data across multiple Microsoft services, Microsoft Graph simplifies the development process and makes it easier for developers to build applications that integrate with Microsoft services.

In summary, Microsoft Graph is a service that provides developers with a unified API endpoint for accessing data from multiple Microsoft services, and enables the development of applications that integrate with those services.

Sign-in Activity Risk Level



Sign-in Activity	Risk Level
Users with leaked credentials	High
Sign-ins from anonymous IP addresses	Medium
Impossible travel to atypical locations	Medium
Sign-ins from infected devices	Medium
Sign-ins from IP addresses with suspicious activity	Low
Sign-ins from unfamiliar locations	Medium

Configure Always Encrypted

Source Article by Microsoft https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine?view=sql-server-ver16

This section provides an overview of setting up Always Encrypted. For details and to get started, see Tutorial: Getting started with Always Encrypted.

To set up Always Encrypted in your database, you need to:

Provision cryptographic keys to protect your data. Always Encrypted uses two types of keys:

- Column encryption keys.
- Column master keys.

A column encryption key is used to encrypt data in an encrypted column. A column master key is a key-protecting key that encrypts one or more column encryption keys.

You need to store column master keys in a trusted key store outside of the database system, such as Azure Key Vault, Windows certificate store, or a hardware security module.

Then, you need provision column encryption keys and encrypt each of them with a column master key.

Finally, you need to store the metadata about the keys in your database.

The column master key metadata captures the location of the column master key. The column encryption key metadata contains the encrypted value of the column encryption key. The Database Engine never stores or uses the keys of either type in plaintext.

For more information about managing Always Encrypted keys, see Overview of key management for Always Encrypted

Configure encryption for selected database columns that contain sensitive data to be protected. This can involve creating new tables with encrypted columns or encrypting existing database columns and existing data. When setting up encryption for a column, you specify the information about an encryption algorithm, a column encryption key to protect the data in the column, and an encryption type. Always Encrypted supports two encryption types:

Deterministic encryption always generates the same encrypted value for a given plaintext value. Using deterministic encryption allows point lookups, equality joins, grouping and indexing on encrypted columns. However, it may also allow unauthorized users to guess information about encrypted values by examining patterns in the encrypted column, especially if there's a small set of possible encrypted values, such as True/False, or North/South/East/West region.

Randomized encryption uses a method that encrypts data in a less predictable manner. Randomized encryption is more secure, but prevents searching, grouping, indexing, and joining on encrypted columns.

Use deterministic encryption for columns that will be used as search or grouping parameters. For example, a government ID number. Use randomized encryption for data such as confidential investigation comments, which aren't grouped with other records and aren't used to join tables.

For details on Always Encrypted cryptographic algorithms, see Always Encrypted cryptography.

You can perform the above steps using SQL tools:

SQL Server Management Studio (SSMS)

SQL Server PowerShell

sqlpackage - which automate the setup process

To ensure Always Encrypted keys and protected sensitive data are never revealed in plaintext to the database environment, the Database Engine can't be involved in key provisioning and data encryption, or decryption operations. Therefore, Transact-SQL (T-SQL) doesn't support key provisioning or cryptographic operations. For the same reason, encrypting existing data or re-encrypting it (with a different encryption type or a column encryption key) needs to be performed outside of the database (SQL tools can automate that).

RBAC - Azure page

https://learn.microsoft.com/en-us/azure/role-based-access-control/role-definitions

Review scope of roles and actions. Understand when things like dataactions [] and actions [] apply and what those differences are. The diagrams are useful as a reference to understand things.

Service Principal vs Managed Identity

Azure Service Principal and Managed Identity are both identity options in Azure that provide authentication and authorization capabilities. Here's a brief explanation of the differences between them:

Azure Service Principal:

- Azure Service Principal is an identity created for use by an application or a service.

- It is typically used when you need to grant permissions to an application to access Azure resources.

- Service Principals are commonly used in scenarios where you need non-interactive access, such as automation or deployment scripts.

• Service Principals are created and managed separately from the Azure resources they access.

• They can be assigned roles and permissions to control access to Azure resources.

• Service Principals require a client ID and client secret or a certificate for authentication.

Managed Identity:

• Managed Identity is a feature of Azure Active Directory (Azure AD) that provides automatic management of the identity lifecycle.

• It eliminates the need to manage credentials manually.

• When you enable a Managed Identity on an Azure resource (e.g., virtual machine, Azure Function), a service principal is created automatically and associated with that resource.

• Managed Identities can be used to authenticate and authorize requests to Azure services, eliminating the need for explicit credentials.

• The lifecycle management, including rotation and renewal of credentials, is handled automatically by Azure AD.

• Managed Identities can be used in scenarios where an application needs to access Azure services securely without managing secrets.

In summary, Service Principals are typically used for non-interactive access and require explicit management, while Managed Identities are automatically managed by Azure AD and provide a more streamlined approach to authentication and authorization for Azure resources.

Compare OAuth 2.0 vs SAML vs Kerberos

OAuth 2.0, SAML, and Kerberos are all authentication and authorization protocols used in different contexts and scenarios. Here's a comparison of these protocols:

OAuth 2.0:

Purpose: OAuth 2.0 is primarily designed for delegated authorization, allowing users to grant access to their resources without sharing their credentials directly.

Use cases: OAuth 2.0 is commonly used in scenarios involving third-party applications and APIs, such as social media login, mobile app authentication, and API access.

Token-based: OAuth 2.0 relies on access tokens, which are issued by an authorization server after the user's approval. These tokens are then used by the client to access protected resources.

Authorization: OAuth 2.0 provides a framework for authorization, but it does not define how user identity is verified or authenticated.

Standards: OAuth 2.0 is an industry-standard protocol and has wide support across various platforms and services.

SAML (Security Assertion Markup Language):

Purpose: SAML is an XML-based standard for exchanging authentication and authorization data between an identity provider (IdP) and a service provider (SP).

Use cases: SAML is commonly used in single sign-on (SSO) scenarios, where a user logs in once and gains access to multiple applications or services without re-authentication.

XML-based: SAML messages are based on XML and typically exchanged over HTTP or HTTPS. They contain assertions that provide information about the user's identity and attributes.

Centralized identity management: SAML relies on a centralized identity provider that authenticates the user and generates SAML assertions, which are then consumed by service providers to grant access.

Standards: SAML is an industry-standard protocol widely adopted in enterprise environments, particularly for federated identity management.

Kerberos:

Purpose: Kerberos is a network authentication protocol designed to provide secure authentication over an insecure network. It verifies the identities of clients and servers communicating over an open network.

Use cases: Kerberos is commonly used in enterprise environments, where it provides single sign-on capabilities and secures network communications.

Ticket-based: Kerberos uses tickets to authenticate users and services. A ticket-granting ticket (TGT) is obtained by the client after authentication, which is then used to obtain service tickets for accessing specific services.

Symmetric key cryptography: Kerberos uses symmetric key cryptography to secure the authentication process. It relies on a trusted third-party Key Distribution Center (KDC) that issues and manages the necessary cryptographic keys.

Standards: Kerberos is an industry-standard protocol and is integrated into various operating systems and network services.

In summary, OAuth 2.0 is primarily used for delegated authorization in third-party application scenarios, SAML is commonly used for SSO and federated identity management, and Kerberos provides secure network authentication in enterprise environments. The choice of protocol depends on the specific use case and requirements of the system or application being developed.

Permission Reference

Review the built-in permission for questions in regards to provide permissions based on least privilege.

https://learn.microsoft.com/en-us/azure/active-directory/roles/permissions-reference

What’s the Difference Between MFA Enabled and Enforced?

Microsoft Azure Active Directory uses various terms to show the status of multi-factor authentication (MFA) for each user. These user states are shown in the Azure portal and all start out as disabled.

Enabled: The user has been enrolled in MFA but has not completed the registration process. They will be prompted to complete the registration process the next time they sign in.

Enforced: The user has been enrolled and has completed the MFA registration process. Users are automatically switched from enabled to enforced when they register for Azure AD MFA.

Disabled: This is the default state for a new user that has not been enrolled in MFA.

Keep in mind, regarding the enforced MFA user status, some older non-browser apps, like Office 2010 or earlier, modern authentication protocols won’t work. In order to enable MFA for user accounts in these apps, with Azure AD multi-factor authentication still enabled, app passwords can be used instead of the user’s regular username and password.

RBAC JSON Security – Understanding Actions / DataActions and NotActions / NotDataActions

https://learn.microsoft.com/en-us/azure/role-based-access-control/role-definitions

Within Azure security, the terms "Actions," "dataActions," "notActions," and "notDataActions" are related to role-based access control (RBAC) and are used to define the permissions and access rights for various Azure resources. Let's break down each term:

1. Actions[]: This refers to the list of actions or operations that a user or identity is allowed to perform on a specific Azure resource. These actions can include read, write, delete, execute, or other specific operations associated with that resource. For example, if you have an Azure virtual machine, the actions might include starting, stopping, restarting, or resizing the VM.

2. dataActions[]: This is similar to "Actions[]," but it specifically focuses on data-related operations or actions that a user or identity can perform on a particular Azure resource. Data actions are typically associated with data-centric services like Azure Storage, Azure SQL Database, Azure Cosmos DB, or Azure Data Lake Storage. These actions might include read, write, delete, or modify operations on the data stored within those services.

3. notActions[]: The "notActions[]" property is used to explicitly deny certain actions or operations on an Azure resource. It allows you to restrict specific actions that might be allowed by default for a particular role. By specifying "notActions[]", you can define a list of actions that should be denied to the user or identity associated with that role.

4. notDataActions[]: Similar to "notActions[]," "notDataActions[]" is used to explicitly deny specific data-related actions or operations on an Azure resource. It allows you to further restrict the data-related actions that might be allowed by default. By specifying "notDataActions[]", you can define a list of data actions that should be denied to the user or identity associated with that role.

In summary, "Actions[]" and "dataActions[]" define the allowed actions or operations that can be performed, while "notActions[]" and "notDataActions[]" define the denied actions or operations on Azure resources. These properties are used in RBAC to fine-tune access permissions for users, groups, or service principals within Azure.