Exam Alert: Implement Azure Security

PREPARING FOR THE EXAM

David Tucker
TECHNICAL ARCHITECT & CTO CONSULTANT
@_davidtucker_ davidtucker.net

Objectives for the Exam

Implement Azure Security

Implement Azure Security

Implement User Authentication and Authorization

Implement Azure Security

Implement User Authentication and Authorization

Implement Secure Cloud Solutions

Implement Azure Security

15-20%

Implement User Authentication and Authorization	Implement Secure Cloud Solutions

Implement User Authentication and Authorization

Implement User Authentication and Authorization

Implement OAuth2 authentication

Implement User Authentication and Authorization

Implement OAuth2 authentication

Create and implement shared access signatures

Implement User Authentication and Authorization

Implement OAuth2 authentication

Create and implement shared access signatures

Register apps and use Azure Active Directory to authenticate users

Implement User Authentication and Authorization

Implement OAuth2 authentication

Create and implement shared access signatures

Register apps and use Azure Active Directory to authenticate users

Control access to resources by using role-based access controls (RBAC)

Implement Secure Cloud Solutions

Implement Secure Cloud Solutions

Secure app configuration data by using the App Configuration and KeyVault API

Implement Secure Cloud Solutions

Secure app configuration data by using the App Configuration and KeyVault API

Manage keys, secrets, and certificates by using the KeyVault API

Implement Secure Cloud Solutions

Secure app configuration data by using the App Configuration and KeyVault API

Manage keys, secrets, and certificates by using the KeyVault API

Implement Managed Identities for Azure resources

Review User Authentication and Authorization

Areas of Focus

Areas of Focus

Azure AD
App Manifests

Areas of Focus

Azure AD
App Manifests

Azure Role-based
Access Control (RBAC)

Areas of Focus

Azure AD
App Manifests

Azure Role-based
Access Control (RBAC)

Azure Storage Shared
Access Signatures (SAS)

Areas of Focus

Azure AD
App Manifests

Azure Role-based
Access Control (RBAC)

Azure Storage Shared
Access Signatures (SAS)

Mutual TLS
Authentication

Azure AD App Manifest

The definition of an application object within the Microsoft Identity platform which includes all configuration for allowed authentication and authorization integrations.

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims": "All",
    "oauth2AllowIdTokenImplicitFlow": true,
    "oauth2AllowImplicitFlow": true,
    "oauth2Permissions": [],
    "oauth2RequirePostResponse": false,
     ...
}
```

App Manifest Attributes to Review

appRoles

App Manifest Attributes to Review

appRoles

groupMembershipClaims

App Manifest Attributes to Review

appRoles

groupMembershipClaims

optionalClaims

App Manifest Attributes to Review

appRoles

groupMembershipClaims

optionalClaims

oauth2AllowImplicitFlow

App Manifest Attributes to Review

App Manifest Attributes to Review

appRoles

groupMembershipClaims

optionalClaims

oauth2AllowImplicitFlow

oauth2Permissions

App Manifest Attributes to Review

appRoles

groupMembershipClaims

optionalClaims

oauth2AllowImplicitFlow

oauth2Permissions

signInAudience

Core Azure RBAC Concepts

Core Azure RBAC Concepts

Security Principal

Core Azure RBAC Concepts

Security Principal

Role Definition

Core Azure RBAC Concepts

Security Principal

Role Definition

Scope

Core Azure RBAC Concepts

Security Principal	Role Definition
Scope	Role Assignments

“A **shared access signature** (SAS) provides secure delegated access to resources in your storage account without compromising the security of your data.”

Microsoft Azure Documentation

Shared Access Signature Types

Shared Access Signature Types

User Delegation

Shared Access Signature Types

User Delegation

Service

Shared Access Signature Types

User Delegation

Service

Account

Azure Storage SAS Forms

Azure Storage SAS Forms

Ad hoc SAS

Azure Storage SAS Forms

Ad hoc SAS

Service SAS (with stored access policy)

SAS Best Practices

SAS Best Practices

Always use HTTPS when creating or distributing an SAS

SAS Best Practices

Always use HTTPS when creating or distributing an SAS

Use user delegation SAS whenever possible

SAS Best Practices

Always use HTTPS when creating or distributing an SAS

Use user delegation SAS whenever possible

Define a stored access policy for a service specific SAS

SAS Best Practices

- Always use HTTPS when creating or distributing an SAS
- Use user delegation SAS whenever possible
- Define a stored access policy for a service specific SAS
- Use near-term expiration on ad hoc, service, or account SAS

SAS Best Practices

- Always use HTTPS when creating or distributing an SAS
- Use user delegation SAS whenever possible
- Define a stored access policy for a service specific SAS
- Use near-term expiration on ad hoc, service, or account SAS
- Follow least-privilege access for resources to be accessed

Azure App Service Mutual TLS Auth

Not supported on free or shared tiers

Azure App Service Mutual TLS Auth

Not supported on free or shared tiers

Certificate is the X-ARR-ClientCert header

Azure App Service Mutual TLS Auth

Azure App Service Mutual TLS Auth

Not supported on free or shared tiers

Certificate is the X-ARR-ClientCert header

Certificate value is Base64 encoded

Azure App Service Mutual TLS Auth

Not supported on free or shared tiers

Certificate is the X-ARR-ClientCert header

Certificate value is Base64 encoded

App code is required to validate certificate

Scenario Understanding

Scenario Understanding

Review different use cases for authentication approaches

Scenario Understanding

Review different use cases for authentication approaches

Understand the order to implement different approaches

Scenario Understanding

Review different use cases for authentication approaches

Understand the order to implement different approaches

Know limits of services and service tiers

Scenario Understanding

- Review different use cases for authentication approaches
- Understand the order to implement different approaches
- Know limits of services and service tiers
- Be able to spot poor security implementations

Review Secure Cloud Solutions

Areas of Focus

Areas of Focus

Managed Identities

Areas of Focus

Managed Identities

Azure Key Vault

Managed Identities

System-assigned	User-assigned

Managed Identities

System-assigned	User-assigned
Widely supported across Azure	

Managed Identities

System-assigned	User-assigned
Widely supported across Azure	
Automatically attached to a single Azure resource	

Managed Identities

System-assigned

Widely supported across Azure

Automatically attached to a single Azure resource

Deleted when attached Azure resource is deleted

User-assigned

Managed Identities

System-assigned

Widely supported across Azure

Automatically attached to a single Azure resource

Deleted when attached Azure resource is deleted

Azure resources can have a single system-assigned identity

User-assigned

Managed Identities

System-assigned

Widely supported across Azure

Automatically attached to a single Azure resource

Deleted when attached Azure resource is deleted

Azure resources can have a single system-assigned identity

User-assigned

Supported by a growing list of services on Azure (with some in preview)

Managed Identities

System-assigned

Widely supported across Azure

Automatically attached to a single Azure resource

Deleted when attached Azure resource is deleted

Azure resources can have a single system-assigned identity

User-assigned

Supported by a growing list of services on Azure (with some in preview)

Created as a standalone Azure resource

Managed Identities

System-assigned

Widely supported across Azure

Automatically attached to a single Azure resource

Deleted when attached Azure resource is deleted

Azure resources can have a single system-assigned identity

User-assigned

Supported by a growing list of services on Azure (with some in preview)

Created as a standalone Azure resource

Must be deleted manually

Managed Identities

System-assigned

Widely supported across Azure

Automatically attached to a single Azure resource

Deleted when attached Azure resource is deleted

Azure resources can have a single system-assigned identity

User-assigned

Supported by a growing list of services on Azure (with some in preview)

Created as a standalone Azure resource

Must be deleted manually

Azure resources can have multiple user-assigned identities

Azure Key Vault Deletion Protection

Azure Key Vault Deletion Protection

Soft-delete

Azure Key Vault Deletion Protection

Soft-delete

Purge Protection

Creating an Azure Key Vault

PowerShell and CLI Commands

```
# Create a Key Vault using PowerShell
New-AzKeyVault -Name ‘Sample-Vault' -ResourceGroupName
'SampleResourceGroup' -Location 'East US’
```

Creating an Azure Key Vault

PowerShell and CLI Commands

```
# Create a Key Vault using PowerShell
New-AzKeyVault -Name ‘Sample-Vault' -ResourceGroupName
'SampleResourceGroup' -Location 'East US’
```

Creating an Azure Key Vault

PowerShell and CLI Commands

```
# Create a Key Vault using PowerShell
New-AzKeyVault -Name ‘Sample-Vault' -ResourceGroupName
'SampleResourceGroup' -Location 'East US’

# Create a Key Vault using Azure CLI
az keyvault create --name "Sample-Vault2" --resource-group
"SampleResourceGroup" --location eastus
```

Creating an Azure Key Vault

PowerShell and CLI Commands

Example Scenarios

Scenario 1

Scenario 1

Sylvia's company is building a prototype for a new internal App Service app

Scenario 1

Sylvia's company is building a prototype for a new internal App Service app

She has created a user-managed identity to control her access to blob storage

Scenario 1

Sylvia's company is building a prototype for a new internal App Service app

She has created a user-managed identity to control her access to blob storage

She also wants to grant access to other Azure resources for her application

Scenario 1

Sylvia's company is building a prototype for a new internal App Service app

She has created a user-managed identity to control her access to blob storage

She also wants to grant access to other Azure resources for her application

Which of the following statements are true about her approach?

True & False Questions

1

2

3

True & False Questions

1 When Sylvia deletes her App Service app, the user-assigned identity will also be deleted.

2

3

True & False Questions

1. When Sylvia deletes her App Service app, the user-assigned identity will also be deleted.

2. When using a user-assigned identity, the app will also have access to the permissions granted by the system-assigned identity.

3.

True & False Questions

1. When Sylvia deletes her App Service app, the user-assigned identity will also be deleted.

2. When using a user-assigned identity, the app will also have access to the permissions granted by the system-assigned identity.

3. Sylvia wants permissions from multiple user-assigned identities. She needs to create a new identity with the combined permissions, since an app can only have a single user-assigned identity.

Scenario 2

Scenario 2

Edward currently has a .NET Core application running as a Function app

Scenario 2

Edward currently has a .NET Core application running as a Function app

He is storing a connection string for Cosmos DB in his application settings

Scenario 2

Edward currently has a .NET Core application running as a Function app

He is storing a connection string for Cosmos DB in his application settings

He wants to avoid redeployments for his Function app

Scenario 2

Edward currently has a .NET Core application running as a Function app

He is storing a connection string for Cosmos DB in his application settings

He wants to avoid redeployments for his Function app

What is the most efficient approach he can take to improve security?

Scenario 3

Scenario 3

Cindy's company is implementing a new App Service app in Node.js

Scenario 3

Cindy's company is implementing a new App Service app in Node.js

The app will leverage Mutual TLS for authentication

Scenario 3

Cindy's company is implementing a new App Service app in Node.js

The app will leverage Mutual TLS for authentication

Cindy is responsible for writing the code to validate the client certificate

Scenario 3

Cindy's company is implementing a new App Service app in Node.js

The app will leverage Mutual TLS for authentication

Cindy is responsible for writing the code to validate the client certificate

How can she access the certificate that the client has used for the request?

Scenario 4

Scenario 4

William is creating an application that will use Azure AD for authentication

Scenario 4

William is creating an application that will use Azure AD for authentication

He wants to allow users from his company's directory to login

Scenario 4

William is creating an application that will use Azure AD for authentication

He wants to allow users from his company's directory to login

He wants to retrieve group membership for groups assigned to the app

Scenario 4

William is creating an application that will use Azure AD for authentication

He wants to allow users from his company's directory to login

He wants to retrieve group membership for groups assigned to the app

How should William configure his app manifest for these requirements?

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims":                    ,
    “oauth2Permissions": [],
    "signInAudience":                           ,
     ...
}
```

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims": [redacted],
    “oauth2Permissions": [],
    "signInAudience": [redacted],
     ...
}
```

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims": [redacted],
    “oauth2Permissions": [],
    "signInAudience": [redacted],
     ...
}
```

Scenario 5

Scenario 5

Oscar’s is creating an application to track customer rebates

Scenario 5

Oscar's is creating an application to track customer rebates

Part of the application is storing the customer submitted receipt images

Scenario 5

Oscar's is creating an application to track customer rebates

Part of the application is storing the customer submitted receipt images

The app currently uses an account SAS that is stored in app configuration

Scenario 5

Oscar's is creating an application to track customer rebates

Part of the application is storing the customer submitted receipt images

The app currently uses an account SAS that is stored in app configuration

How can Oscar ensure the most secure access to storage resources?

Scenario 6

Scenario 6

James’s company processes healthcare data for billing analysis

Scenario 6

James's company processes healthcare data for billing analysis

They have a requirement that all data must be encrypted using managed keys

Scenario 6

James's company processes healthcare data for billing analysis

They have a requirement that all data must be encrypted using managed keys

They require leveraging hardware encryption (HSM) **for key storage**

Scenario 6

James's company processes healthcare data for billing analysis

They have a requirement that all data must be encrypted using managed keys

They require leveraging hardware encryption (HSM) for key storage

James has moved all encryption keys to Azure Key Vault (standard tier)

Scenario 6

James's company processes healthcare data for billing analysis

They have a requirement that all data must be encrypted using managed keys

They require leveraging hardware encryption (HSM) for key storage

James has moved all encryption keys to Azure Key Vault (standard tier)

Does his approach meet the criteria?

Scenario Answers

Scenario 1

Sylvia's company is building a prototype for a new internal App Service app

She has created a user-managed identity to control her access to Azure Storage

She also wants to grant access to other Azure resources for her application

Which of the following statements are true about her approach?

Scenario 1

1. When Sylvia deletes her App Service app, the user-assigned identity will also be deleted.

2. When using a user-assigned identity, the app will also have access to the permissions granted by the system-assigned identity.

3. Sylvia wants permissions from multiple user-assigned identities. She needs to create a new identity with the combined permissions, since an app can only have a single user-assigned identity.

Scenario 1

False **1** When Sylvia deletes her App Service app, the user-assigned identity will also be deleted.

2 When using a user-assigned identity, the app will also have access to the permissions granted by the system-assigned identity.

3 Sylvia wants permissions from multiple user-assigned identities. She needs to create a new identity with the combined permissions, since an app can only have a single user-assigned identity.

Scenario 1

False **1**

When Sylvia deletes her App Service app, the user-assigned identity will also be deleted.

False **2**

When using a user-assigned identity, the app will also have access to the permissions granted by the system-assigned identity.

3

Sylvia wants permissions from multiple user-assigned identities. She needs to create a new identity with the combined permissions, since an app can only have a single user-assigned identity.

Scenario 1

False **1**

When Sylvia deletes her App Service app, the user-assigned identity will also be deleted.

False **2**

When using a user-assigned identity, the app will also have access to the permissions granted by the system-assigned identity.

False **3**

Sylvia wants permissions from multiple user-assigned identities. She needs to create a new identity with the combined permissions, since an app can only have a single user-assigned identity.

Scenario 2

Edward currently has a .NET Core application running as a Function app

He is storing a connection string for Cosmos DB in his application settings

He wants to avoid redeployments for his Function app

What is the most efficient approach he can take to improve security?

Scenario 2

Edward currently has a .NET Core application running as a Function app

He is storing a connection string for Cosmos DB in his application settings

He wants to avoid redeployments for his Function app

What is the most efficient approach he can take to improve security?

Solution: Utilize an Azure Key Vault Reference for the Cosmos DB connection

Scenario 3

Cindy's company is implementing a new App Service app in Node.js

The app will leverage Mutual TLS for authentication

Cindy is responsible for writing the code to validate the client certificate

How can she access the certificate that the client has used for the request?

Scenario 3

Cindy's company is implementing a new App Service app in Node.js

The app will leverage Mutual TLS for authentication

Cindy is responsible for writing the code to validate the client certificate

How can she access the certificate that the client has used for the request?

Solution: Access the X-ARR-ClientCert header and decode the Base64 string

Scenario 4

William is creating an application that will use Azure AD for authentication

He wants to allow users from his company's directory to login

He wants to retrieve group membership for groups assigned to the app

How should William configure his app manifest for these requirements?

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims": [redacted],
    “oauth2Permissions": [],
    "signInAudience": [redacted],
     ...
}
```

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims":                    ,
    “oauth2Permissions": [],
    "signInAudience":                          ,
     ...
}
```

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims": "ApplicationGroup" ,
    “oauth2Permissions": [],
    "signInAudience":                          ,
     ...
}
```

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims": "ApplicationGroup" ,
    “oauth2Permissions": [],
    "signInAudience":                          ,
     ...
}
```

App Manifest

```
{
    "id": "058477a1-5d5f-45e7-bc71-66c059a58eff",
    "name": "SampleSPA",
     ...
    "allowPublicClient": true,
    "groupMembershipClaims": "ApplicationGroup" ,
    “oauth2Permissions": [],
    "signInAudience": "AzureADMyOrg"            ,
     ...
}
```

Scenario 5

Oscar's is creating an application to track customer rebates

Part of the application is storing the customer submitted receipt images

The app currently uses an account SAS that is stored in app configuration

How can Oscar ensure the most secure access to storage resources?

Scenario 5

Oscar's is creating an application to track customer rebates

Part of the application is storing the customer submitted receipt images

The app currently uses an account SAS that is stored in app configuration

How can Oscar ensure the most secure access to storage resources?

Solution: Utilize a user-delegation SAS, which uses Azure AD credentials

Scenario 6

James's company processes healthcare data for billing analysis

They have a requirement that all data must be encrypted using managed keys

They require leveraging hardware encryption (HSM) **for key storage**

James has moved all encryption keys to Azure Key Vault (standard tier)

Does his approach meet the criteria?

Scenario 6

James's company processes healthcare data for billing analysis

They have a requirement that all data must be encrypted using managed keys

They require leveraging hardware encryption (HSM) **for key storage**

James has moved all encryption keys to Azure Key Vault (standard tier)

Does his approach meet the criteria?

Solution: No. He will need to utilize the Premium Tier for Azure Key Vault