Author: Sukanta Mukherjee
Wednesday, 23 Aug, 2023
Email: sukanta.mukherjee@outlook.com
https://www.linkedin.com/in/sukanta-mukherjee-6b152a46/

Exploring Azure Managed Identity: A Solution for Secure Authentication and Access Management

Introduction

In today's rapidly evolving cloud-centric landscape, managing identities and securely controlling access to resources has become a big concern. Traditional authentication methods such as usernames and passwords are often inadequate for providing a high level of security and flexibility. Azure Managed Identity is a powerful solution provided by Microsoft Azure that addresses these challenges by simplifying identity management and access control while enhancing security. In this blog post, I will discuss the exact problem and followed by solution architecture, technical implementation, challenges, and business benefits of Azure Managed Identity.

Problem Statement

In a cloud environment, applications and services often require access to various Azure resources such as databases, storage accounts, and APIs. Manually managing authentication credentials for each resource introduces several problems:

1. **Security Risk:** Storing and managing credentials increases the risk of accidental exposure or malicious use, potentially leading to data breaches or unauthorized access.

2. **Credential Rotations:** Regularly updating and rotating credentials across various resources is complex and error-prone, leading to potential service disruptions.

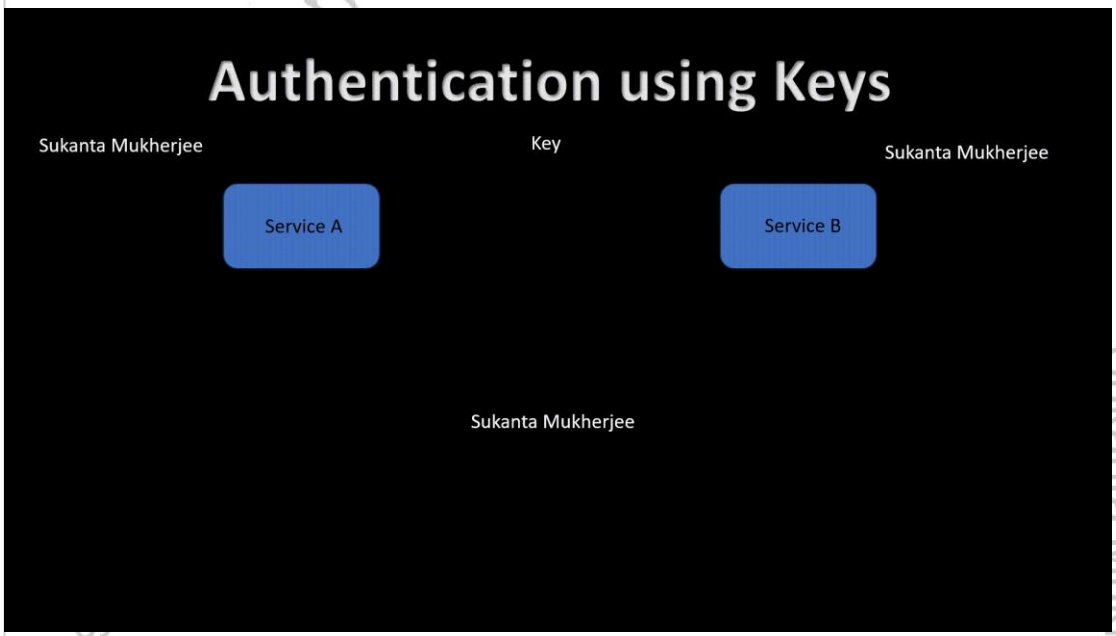
3. **Key Management Overhead:** Managing keys and secrets for authentication adds more operational overhead, effort and resources.

4. **Limited Flexibility:** Traditional authentication methods lack the flexibility needed for dynamic and scalable cloud environments.

Author: Sukanta Mukherjee
Wednesday, 23 Aug, 2023
Email: sukanta.mukherjee@outlook.com
https://www.linkedin.com/in/sukanta-mukherjee-6b152a46/

Working principal of Azure various authentication model

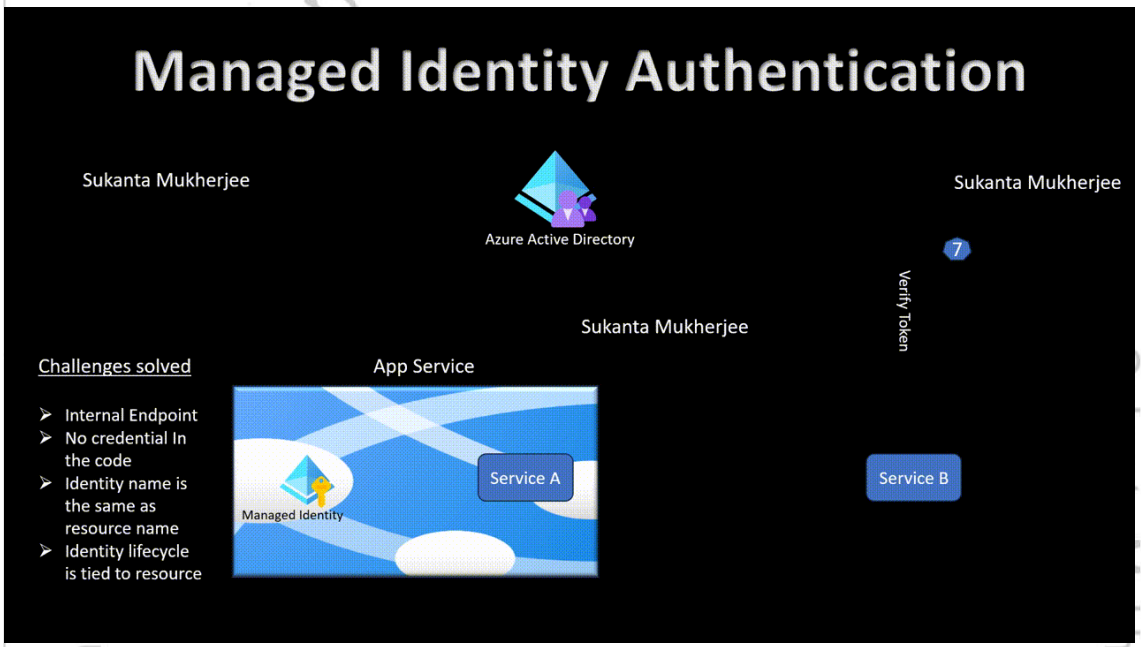
Authentication using keys: challenges and schemetic diagram



Authentication using Azure AD (service principal): challenges and schemetic diagram

Author: Sukanta Mukherjee
Wednesday, 23 Aug, 2023
Email: sukanta.mukherjee@outlook.com
https://www.linkedin.com/in/sukanta-mukherjee-6b152a46/

Authentication using managed identity: challenges and schemetic diagram



Solution/Architecture

Azure Managed Identity provides a solution by enabling applications and services to authenticate themselves without the need for explicit credentials. It eliminates the need to manage secrets or keys by integrating with Azure Active Directory (Azure AD) and creating a unique identity for each Azure resource.

The architecture involves three main components:

1. Azure Resources: These are the services or resources that need to be accessed securely. This could include Azure Virtual Machines, Azure Functions, Azure App Services, Azure Key Vault, and more.

2. Managed Identity: Azure creates a managed identity for each resource. This identity is automatically associated with the Azure AD tenant and the specific resource. It allows the resource to authenticate itself securely without using explicit credentials.

3. Azure AD: Azure AD acts as the identity provider. It authenticates the managed identity and authorizes it to access the specified resources based on the assigned roles and permissions.

Technical Details and Implementation

Currently there are fifty-five Azure services which support Azure managed identity.
Go through the following link for the services list.

Author: Sukanta Mukherjee
Wednesday, 23 Aug, 2023
Email: sukanta.mukherjee@outlook.com
https://www.linkedin.com/in/sukanta-mukherjee-6b152a46/

https://learn.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/managed-identities-status

1. Enabling Managed Identity:

For Azure Virtual Machines: Enable the system-assigned or user-assigned managed identity during VM creation.

For Azure App Services: Enable managed identity in the "Identity" section of the App Service's settings.

For Azure Functions: Similar to App Services, enable managed identity in the Function App's settings.

2. Access Control:

Assign roles and permissions to the managed identity using Azure RBAC (Role-Based Access Control).

Use Azure Policy to enforce access control policies and compliance.

3. Code Integration:

In the application code, utilize Azure SDKs to access resources securely using the managed identity.

No explicit credentials are required; authentication is handled transparently by Azure AD.

Now let's connect Azure managed SQL instance through managed Identity.

To connect to a SQL database, we usually use a connection string that has a username and password. We ensure that the connection string is stored and distributed securely.

Generic connection string for SQL DB connectivity is as follows.

```
"ConnectionStrings": {
    "QuotesDatabase":
"Server=tcp:quotetest.database.windows.net,1433;Database=quotes;User
Id:<UserName>;Password:<YourPasswordHere>"
  }
```

Using Azure AD Token to Connect to SQL

Using the DefaultAzureCredential from Azure Identity SDK Azure AD token can easily be retrieved. SqlConnection uses this token for authentication. Following code example shows the Access Token property of the SqlConnection is the Azure AD token.

Author: Sukanta Mukherjee
Wednesday, 23 Aug, 2023
Email: sukanta.mukherjee@outlook.com
https://www.linkedin.com/in/sukanta-mukherjee-6b152a46/

```
var connectionString =
Configuration.GetConnectionString("QuotesDatabase");
services.AddTransient(a =>
{
    var sqlConnection = new SqlConnection(connectionString);
    var credential = new DefaultAzureCredential();
    var token = credential
        .GetToken(new Azure.Core.TokenRequestContext(
            new[] { "https://database.windows.net/.default" }));
    sqlConnection.AccessToken = token.Token;
    return sqlConnection;
});
```

Setting Up SQL Server For Managed Identity

To manage Azure SQL for AD identities, we need to access SQL under the Azure user context. To do this, we configure the Azure AD user as an SQL administrator. This can be done from the Azure Portal under the Azure Directory Admin option for the database server, as shown below.

Using the SQL AD Admin credentials, you can connect via SQL Server Management Studio or sqlcmd and grant other AD identities access. The below script grants the user 'db_user1, db_user2, and db_admin' access.

```
CREATE USER [<identity-name>] FROM EXTERNAL PROVIDER;
ALTER ROLE db_user1 ADD MEMBER [<identity-name>];
ALTER ROLE db_user2 ADD MEMBER [<identity-name>];
ALTER ROLE db_admin ADD MEMBER [<identity-name>];
GO
```

<identity-name> is the name of the managed identity in Azure AD. For a system-assigned identity, the name is the same as the App Service name. It can also be an Azure AD Group (use the group name in this case). It gives you multiple options on how you want to manage access to the database.

No longer we need any credentials to connect to the SQL database running on Azure. This makes it one less sensitive information to manage for our application.

```
"ConnectionStrings": {
    "QuotesDatabase":
"Server=tcp:quotetest.database.windows.net,1433;Database=quotes"
  }
```

Challenges in Implementing the Solution

While Azure Managed Identity greatly simplifies identity management and access control, there can be challenges during implementation:

Author: Sukanta Mukherjee
Wednesday, 23 Aug, 2023
Email: sukanta.mukherjee@outlook.com
https://www.linkedin.com/in/sukanta-mukherjee-6b152a46/

1. **Resource Compatibility:** Not all Azure services support managed identities, so alternative authentication methods may be needed in some cases.

2. **Legacy Systems:** Integrating managed identities into existing applications or systems might require code modifications and adaptations.

3. **Understanding Roles:** Assigning appropriate roles and permissions requires a deep understanding of Azure RBAC and resource-level access control.

4. **Monitoring and Auditing:** Tracking and auditing managed identity access for compliance and security purposes can be complex.

Business Benefits:

1. **Enhanced Security:** Managed identities eliminates the risk of storing and managing credentials, reducing the threat of records breaches because of leaked or stolen passwords.

2. **Simplified Management**: Automation of identity and get entry to management reduces operational overhead, permitting groups to focus on cost-delivered duties.

3. **Scalability:** Managed identities help dynamic scaling, making them properly-suitable for cutting-edge cloud architectures.

4. C**ompliance and Auditing:** Managed identity get admission to may be monitored and audited for compliance with regulatory necessities.

5. **Developer Productivity:** Developers can aware on writing code rather than dealing with authentication credentials.

Conclusion:

Azure Managed Identity addresses the challenge of securing identity management and access control in the cloud by providing a seamless and secure approach to objects without explicit credentials This solution provides security improvements, simplifies management, and supports the scalability and flexibility required in today's cloud environment. While challenges can arise during implementation, the benefits in terms of security, compliance and operational efficiencies make Azure Managed Identity a key tool for organizations adopting the cloud model.