YOU SECRETS SAFE IN AZURE KEY VAULT?

Protecting a multi-tenant application

MVP
Microsoft®
Most Valuable Professional

STEPHAN VAN ROOIJ

Microsoft MVP Security

Software architect

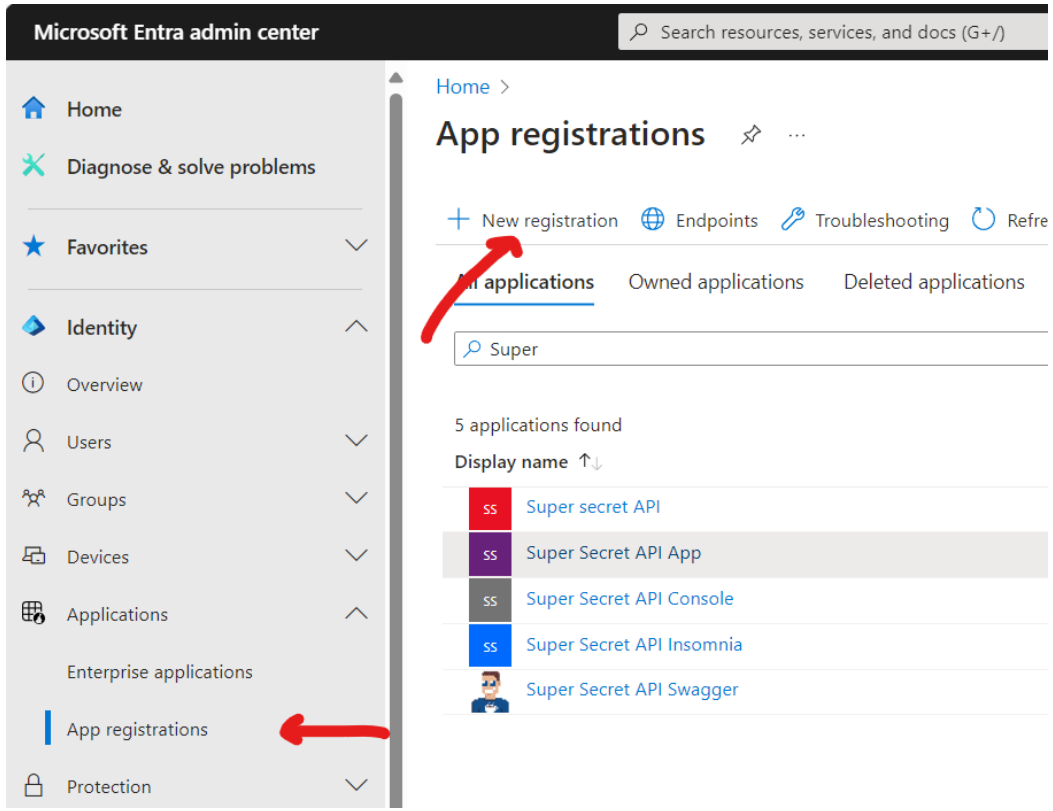
Open-source developer

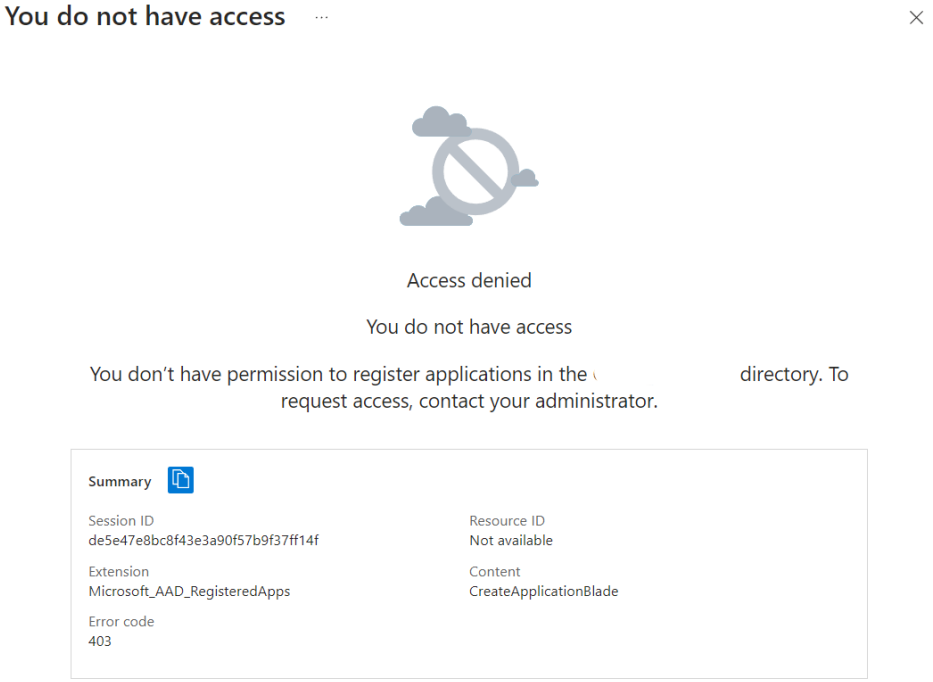
The Netherlands

OBJECTIVE

Protect our multi-tenant application and limit the exposure time in case of a breach

CAN EVERYBODY CREATE APPLICATIONS IN YOUR TENANT?





CREATE MULTI-TENANT APP

Register an application

* Name

The user-facing display name for this application (this can be changed later).

Graph reader multi-tenant

Supported account types

Who can use this application or access this API?

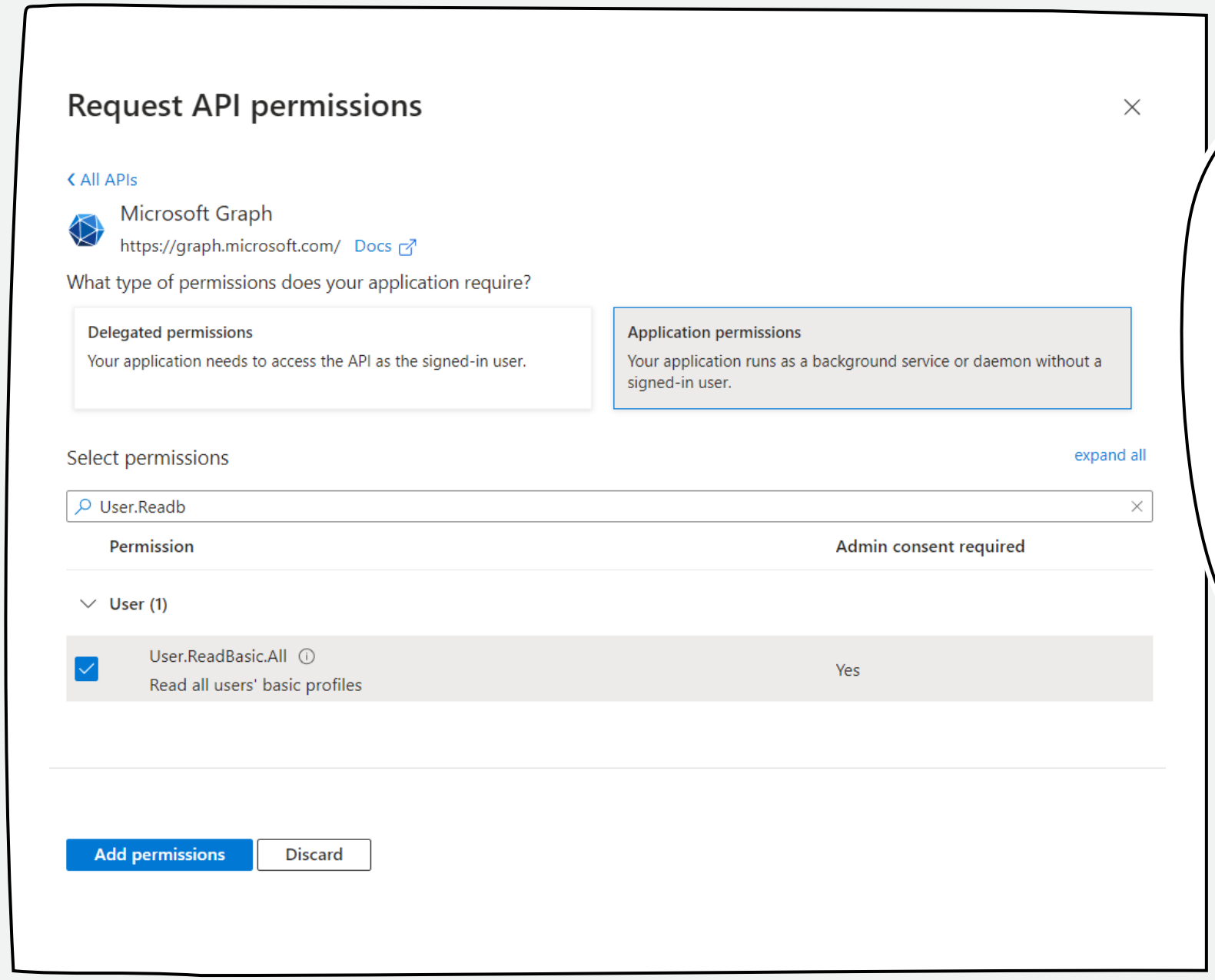
- ○ Accounts in this organizational directory only (Coding Stephan only - Single tenant)
- ◉ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- ○ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- ○ Personal Microsoft accounts only

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Select a platform | e.g. https://example.com/auth



APPLICATION PERMISSIONS ->

User.ReadBasic.All

ADMIN CONSENT NOT GRANTED, (YET)!

Configured permissions

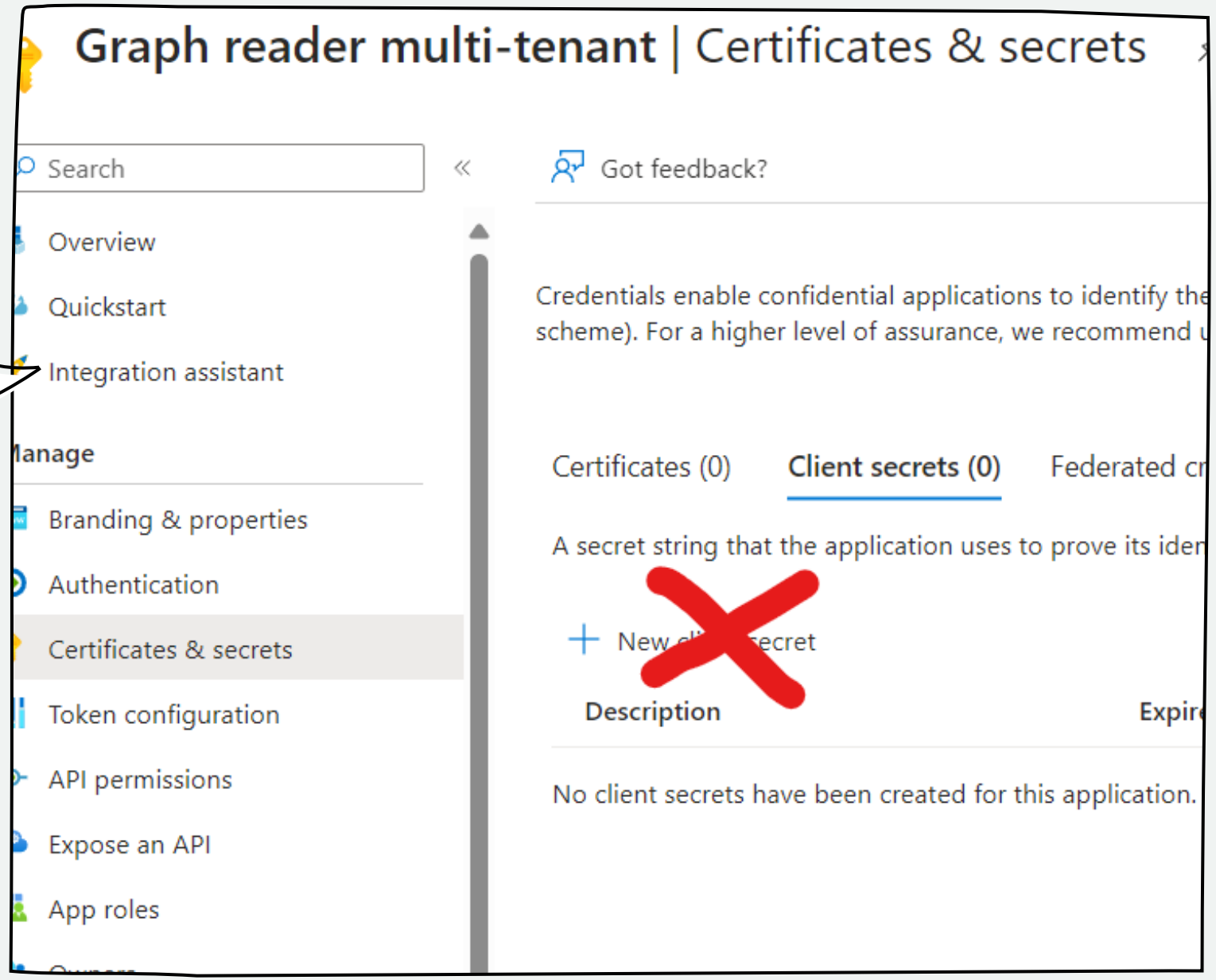
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

+ Add a permission ✓ Grant admin consent for Coding Stephan

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (2)				
User.Read	Delegated	Sign in and read user profile	No	
User.ReadBasic.All	Application	Read all users' basic profiles	Yes	⚠ Not granted for Coding ...

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try Enterprise applications.

CLIENT SECRET?



CREATE CERTIFICATE IN KEY VAULT

Create a certificate

Method of Certificate Creation	Generate
Certificate Name *	test-client-credentials
Type of Certificate Authority (CA)	Self-signed certificate
Subject *	CN=graph-reader.invalid
DNS Names	0 DNS names
Validity Period (in months) *	12
Content Type	PKCS #12 / PEM
Lifetime Action Type	E-mail all contacts at a given percentage lifeti...
Percentage Lifetime *	80
Advanced Policy Configuration	Not configured
Tags	0 tags

Create Cancel

Home > App registrations > Graph Sample Client

Graph Sample Client | Certificates & secrets

Search

Overview
Quickstart
Integration assistant

Manage

Branding & properties
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners
Roles and administrators
Manifest

Support + Troubleshooting

Troubleshooting
New support request

Got feedback?

Credentials enable confidential applications to identify themse
HTTPS scheme). For a higher level of assurance, we recommer

Application registration certificates, secrets and federated c

Certificates (0) Client secrets (0) Federated crede

Certificates can be used as secrets to prove the application's

Upload certificate

Thumbprint	Description

No certificates have been added for this application.

Upload certificate

Upload a certificate (public ke
"test-client-credentials_64b

Description
Enter a description for t

Add Cancel

CLIENT CERTIFICATE!

DEMO

```
[Function(nameof(LoadCertificateAndAuthenticateToEntra))]
1 reference
public async Task<HttpResponseData> Run(
    [HttpTrigger(AuthorizationLevel.Function, "get", Route = "demo1/call-graph")] HttpRequestData req,
    string? tenantId, // This will be loaded from the url if passed
    FunctionContext executionContext
    )
{
    // Create a certificate client, passing in the key vault uri (from configuration) and the token credential
    var certificateClient = new CertificateClient(_strongConfiguration.KeyVaultUri, _tokenCredential);
    // Download the certificate, passing in the name of the certificate
    var certificate = await certificateClient.DownloadCertificateAsync(new DownloadCertificateOptions(_strongConfiguration.ClientCertificateName), cancellationToken: executionContext.CancellationToken

    // At this point, the certificate is loaded and can be used to authenticate to Entra

    // Option 1: Use Azure.Identity to authenticate to Entra
    var token = await GetTokenUsingClientCertificateCredentialAsync(tenantId ?? _strongConfiguration.TenantId!, _strongConfiguration.ClientId!, certificate.Value, [_strongConfiguration.GraphScope!]

    _logger.LogInformation("Getting users from Graph API with token {token}", TokenModifier.RemoveSignature(token));

    // Use the token to call the Microsoft Graph API
    var httpRequest = new HttpRequestMessage(HttpMethod.Get, "https://graph.microsoft.com/v1.0/users?$select=id,displayName&$top=3");
    httpRequest.Headers.Authorization = new AuthenticationHeaderValue("Bearer", token);
    var response = await _httpClient.SendAsync(httpRequest, executionContext.CancellationToken);
    var responseData = await response.Content.ReadAsByteArrayAsync(executionContext.CancellationToken);

    // Return the graph response
    var result = req.CreateResponse(response.StatusCode);
    await result.WriteBytesAsync(responseData, executionContext.CancellationToken);
    return result;
}

/// <summary>
/// Get a token using Azure.Identity
/// </summary>
/// <param name="tenantId"></param>
/// <param name="clientId"></param>
/// <param name="certificate"></param>
/// <param name="scopes"></param>
/// <param name="cancellationToken"></param>
/// <returns>Access token</returns>
1 reference
private async Task<string> GetTokenUsingClientCertificateCredentialAsync(string tenantId, string clientId, X509Certificate2 certificate, string[] scopes, CancellationToken cancellationToken)
{
    var clientCredential = new ClientCertificateCredential(tenantId, clientId, certificate);
    var result = await clientCredential.GetTokenAsync(new TokenRequestContext(scopes), cancellationToken);

    return result.Token;
}
```

CERTIFICATE "SECURELY" STORED IN KEY VAULT

1

Load certificate

2

Create unsigned JWT

3

Sign JWT with certificate

4

Request token from Entra ID with JWT

ALL GOOD?

Cert stored in **Key Vault** ✅
We get a **token** ✅

ATTACKER DEPLOYS NEW VERSION OF AZURE FUNCTIONS APP

1

Download certificate

2

Post full certificate to external server

3

Return internal server error

DEMO

```
[Function(nameof(LoadCertificateAndPostToExternalServer))]
1 reference
public async Task<HttpResponseData> Run(
    [HttpTrigger(AuthorizationLevel.Function, "get", Route = "demo1/steal-certificate")] HttpRequestData req,
    FunctionContext executionContext
    )
{
    // Create a certificate client, passing in the key vault uri (from configuration) and the token credential
    var certificateClient = new CertificateClient(_strongConfiguration.KeyVaultUri, _tokenCredential);
    // Download the certificate, passing in the name of the certificate
    var certificate = await certificateClient.DownloadCertificateAsync(new DownloadCertificateOptions(_strongConfiguration.Client

    // At this point, the certificate is loaded ... and we can do whatever we want with it

    _logger.LogWarning("We are stealing the certificate by posting it to an external server");
    // By calling ExportCertificatePem and ExportRSAPrivateKeyPem and puting it underneath each other the certContent will look l
    // -----BEGIN CERTIFICATE-----
    // MIIDBzCCAe+gAwIBAgIQJQ6+J5Zz1z5zq2Z
    // -----END CERTIFICATE-----
    // -----BEGIN RSA PRIVATE KEY-----
    // MIIEpAIBAAKCAQEAz0Zz1z5zq2Z
    // -----END RSA PRIVATE
    var certContent = certificate.Value.ExportCertificatePem() + "\r\n" + certificate.Value.GetRSAPrivateKey()!.ExportRSAPrivateKe

    // An external server might not be reachable, but it just a string at this point.
    // You could also write it to the blog storage (every Azure Function has a storage account)
    // or send it to a queue, or maybe even just return it in the response.

    var result = req.CreateResponse(System.Net.HttpStatusCode.OK);
    // I don't want to show the full (working) certificate, so I'm cutting it off but you get the idea
    await result.WriteStringAsync(certContent.Substring(0, certContent.Length - 400) + "...", executionContext.CancellationToken)
    return result;
}
```

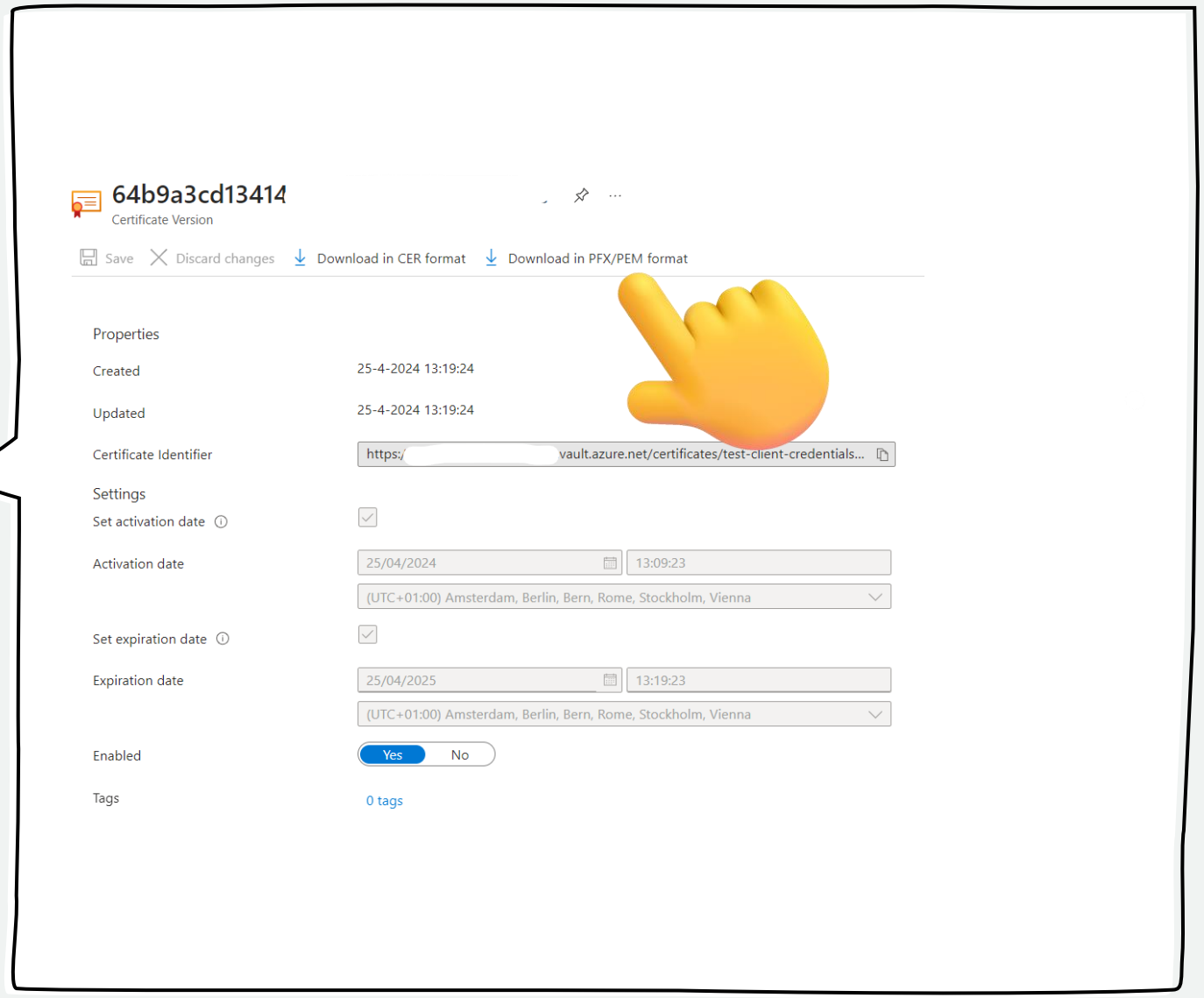
localhost:33640/api/demo1/steal-certificate


```
-----BEGIN CERTIFICATE-----
MIIDRDCCAiygAwIBAgIQYDm6X1wxQ82DQAzF5JOUMjANBgkqhkiG9w0BAQsFADAf
MR0wGwYDVQQDExRncmFwaC1yZWFkZXIuaW52YWxpZDAeFw0yNDA0MjUxMTA5MjNa
Fw0yNTA0MjUxMTE5MjNaMB8xHTAbBgNVBAMTFGdyYXBoLXJlYWRlci5pbnZhbGlk
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2Sq70WYqvUX2U52eV49A
6z4yMfahUdj6q+rO95mWtzoiKLVjpw/DDjUwuNBm+GpV/qrOhthSxtlOBjb+KyEx
GjwEPx1h0aLrnNW1uoYikXQD/eipAqIfo3AY8NfGmxwjV2Llld+YVJuAUNRKmhba
tF26ArHkGEqWhHQuTRq1oDSa/SjP+xhE7+vB1s8FHJfE0Mo5uNoY/DKFW7/Nc3dX
rUf3BR7MpedsXYdJQNIKOocWNdIsNT8KXuVS4biw0fE8heXqZHaVwRz0vD4+K8wP
CKdjHjiVM3Pwxn5PhNR5EB/HUa2k3ySoEp1ZgRqyIw4dd9WjYMxQ5Lg4PRAVC8xT
aQIDAQABo3wwejAOBgNVHQ8BAf8EBAMCBaAwCQYDVR0TBAIwADAdBgNVHSUEFjAU
BggrBgEFBQcDAQYIKwYBBQUHAwIwHwYDVR0jBBgwFoAUsadMCSrsqCxZnBHEFjI+
7hpX7TowHQYDVR0OBBYEFLGnTAkq7KgsWZwRxBYyPu4aV+06MA0GCSqGSIb3DQEB
CwUAA4IBAQBMNtg4fRdPshY8rviP3M5ZaXWkt0Q+mg+SMO4RQUM1nnzv7Tqh1BvB
sUuJfvI4so/2y2CjklSftzgKUwO2Aysfjfv3Ecif+W95DaEcIcELwmDNPfYhCS3c
0PZ0dadcXXD+yaUDyUMFmbOmbl+kWy+3SZN2KufrkUyYPR9ONwG+Dv26stlu+nYp
V7SD5gDD8uk0SMk8fmE35WdytJYcDZu6uLVwQym8EsY3IhG4lWkWvCy6AsasIMFR
k9LkYBRWisD1vxEMYsRGrin8O539FDrOz2Rl4hsOvYmaRgS8c+cwOzxj+4wiwveT
EX7iIi1Y/A6rkSeVuZzANZa/TcFWQlVT
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA2Sq70WYqvUX2U52eV49A6z4yMfahUdj6q+rO95mWtzoiKLVj
```


CERTIFICATE STOLEN EXPOSURE TIME

360 days

ATTACKER GAINS ACCESS TO ACCOUNT WITH KEY VAULT ACCESS



CERTIFICATE STOLEN
EXPOSURE TIME

360 days

WE NEED THE CERTIFICATE TO LOGIN?

Or just a signature?

sign - sign

Reference

Feedback

Service: Key Vault

API Version: 7.4

Creates a signature from a digest using the specified key.
The SIGN operation is applicable to asymmetric and symmetric keys stored in Azure Key Vault since this operation uses the private portion of the key. This operation requires the keys/sign permission.

HTTP

Copy

```
POST {vaultBaseUrl}/keys/{key-name}/{key-version}/sign?api-version=7.4
```

KEY VAULT -> KEYS -> SIGN

CERTIFICATE SECURELY STORED IN KEY VAULT ✅

1

Create unsigned JWT

2

Sign JWT with Key in Azure Key Vault

3

Request token from Entra ID with JWT

INTRODUCING

Smartersoft.Identity.Client.Assertion

You don't have to mess with unsigned tokens

DEMO

```
1 reference
private async Task<string> GetTokenUsingMsalAsync(TokenCredential tokenCredential, Uri keyVaultUri, string tenantId, string clientId, string certificateName,
{
    // The code for the `WithKeyVaultCertificate` extension can be found here https://github.com/Smartersoft/identity-client-assertion/blob/b12318fe7b0d054a65
    // which eventually calls this code: https://github.com/Smartersoft/identity-client-assertion/blob/b12318fe7b0d054a65334d61686dc96a9a29024d/src/Smartersof
    var app = ConfidentialClientApplicationBuilder.Create(clientId)
        //.WithCertificate(certificateName) // this is replaced with the following line
        .WithKeyVaultCertificate(keyVaultUri, certificateName, tokenCredential) // this is the new line
        .WithTenantId(tenantId)
        .Build();
    var result = await app.AcquireTokenForClient(scopes)
        .ExecuteAsync(cancellationToken);

    return result.AccessToken;
}
```

Microsoft Azure

Search resources, services, and docs (G+/)

Home > | Certificates >

Create a certificate

Method of Certificate Creation	Generate
Certificate Name *	test-client-credentials-no-export
Type of Certificate Authority (CA)	Self-signed certificate
Subject *	CN=client-cert-protected.invalid
DNS Names	0 DNS names
Validity Period (in months) *	12
Content Type	PKCS #12 / PEM
Lifetime Action Type	E-mail all contacts at a given percentage lifetime
Percentage Lifetime *	
Advanced Policy Configuration	Not configured
Tags	0 tags

Create Cancel

Advanced Policy Configuration

Create a certificate

Extended Key Usages (EKUs)

1.3.6.1.5.5.7.3.1, 1.3.6.1.5.5.7.3.2

X.509 Key Usage Flags

Digital Signature

Reuse Key on Renewal?

- Yes
- No

Exportable Private Key?

- Yes
- No

Key Type

- RSA
- EC

Key Size

- 2048
- 3072
- 4096

Enable Certificate Transparency?

- Yes
- No

Certificate Type

For example: "OV-SSL".

OK

Microsoft Security USER GROUP

ATTACKER DEPLOYS NEW VERSION OF AZURE FUNCTIONS APP

1

Sign JWT with Key in Azure Key Vault

2

Request token from Entra ID

3

Return token

~~CERTIFICATE~~

TOKEN STOLEN

EXPOSURE TIME

3600 seconds

GENERAL KEY VAULT RECOMMENDATIONS

1. Use Managed identity to access Key Vault
2. Periodically review role assignments
3. Limit network traffic
4. Monitor access
5. Disable Vault Access Policies

🎉 ✅

MORE INVESTIGATION NEEDED

Nuget RSAKeyVaultProvider?

Key Vault backed X509Certificate2?

- Client Certificate in HttpClient
- RSA async support?

SECURE MULTI-TENANT APP BY STEPHAN VAN ROOIJ

Source:
https://github.com/svrooij/SecureMultiTenantApp

Blog: svrooij.io

Twitter: @svrooij

Github: @svrooij