sign script ps1 UI

# Progress	100

Policy

Execution Policy	Description	Scope of Impact
Restricted	No scripts allowed to run	Default on Windows
AllSigned	Only runs scripts signed by a trusted publisher	Recommended for most users
RemoteSigned	Requires remote scripts (Internet) to be signed	Flexible yet secure option
Unrestricted	No restrictions on script execution	Not recommended due to security risks
Bypass	Ignores all execution policies	Should be used with extreme caution

Methods of signing scripts

You can sign the scripts that you write and the scripts that you get from other sources. Before you sign any script, examine each command to verify that it's safe to run.

For best practices about code signing, see Code-Signing Best Practices.

For more information about how to sign a script file, see Set-AuthenticodeSignature.

The `New-SelfSignedCertificate` cmdlet, introduced in the PKI module in PowerShell 3.0, creates a self-signed certificate that's appropriate for testing. For more information, see the help topic for the `New-SelfSignedCertificate` cmdlet.

To add a digital signature to a script, you must sign it with a code signing certificate. Two types of certificates are suitable for signing a script file:

- Certificates that are created by a certification authority: For a fee, a public certification authority verifies your identity and gives you a code signing certificate. When you purchase your certificate from a reputable certification authority, you are able to share your script with users on other computers that are running Windows because those other computers trust the certification authority.
- Certificates that you create: You can create a self-signed certificate for which your computer is the authority that creates the certificate. This certificate is free of charge and enables you to write, sign, and run scripts on your computer. However, a script signed by a self-signed certificate will not run on other computers.

Commands

Now, you'll want to select the certificate you wish to use for the signing process. You'll want to set the certificate to the *$cert* variable. To do this, you can use the following command, which selects the specific certificate from the list (starting from the top, which you'll count as 0). This command will look like the screenshot below:

```
$cert = (Get-ChildItem -Path Cert:\LocalMachine\My -CodeSigningCert)[4]
```

4. Sign Your PowerShell Script

Okay, now that you've selected the certificate you wish to use and have saved it to your *$cert* variable, you're ready to rock. Sign your PowerShell script using the *Set-AuthenticodeSignature* command, which looks like this:

```
Set-AuthenticodeSignature -FilePath SCRIPT_PATH -Certificate $cert
```

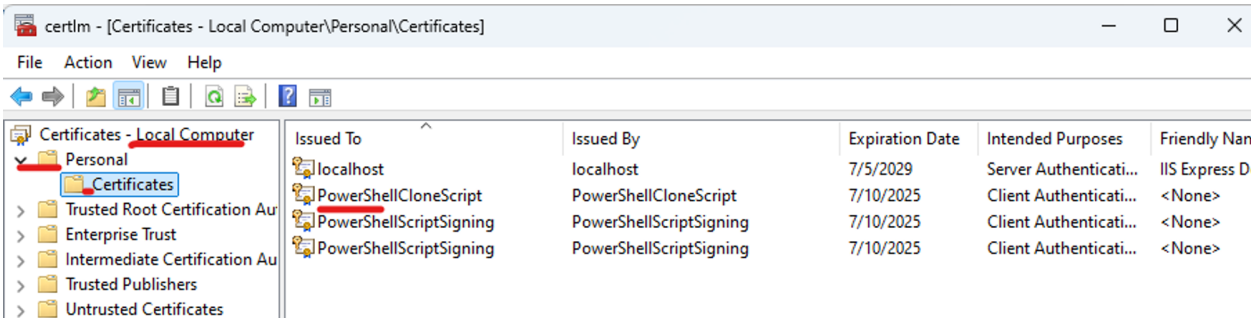
Guide Step by Step

Step 1: Generate a Code Signing Certificate: in an administrator power shell.

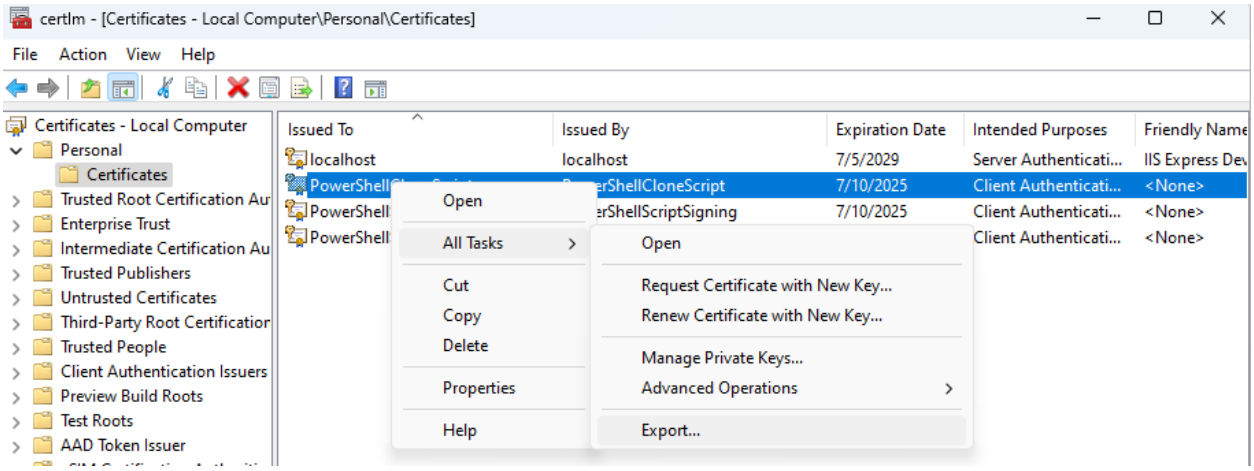
New-SelfSignedCertificate -CertStoreLocation Cert:\CurrentUser\My -Subject "CN=PowerShellScriptSigning" -Type CodeSigningCert -KeySpec Signature

Step 2: Export the Certificate to a File

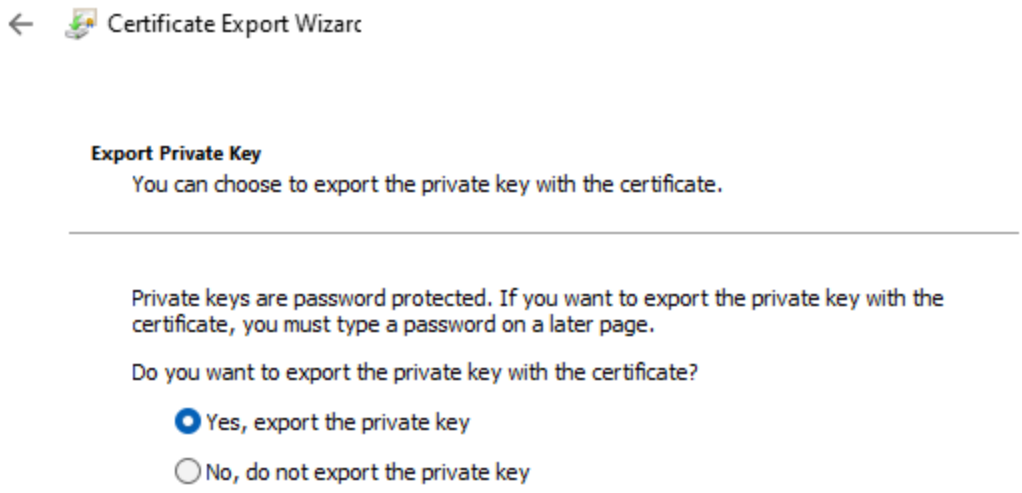
Open the Certificate Management Console with **certmgr.msc**



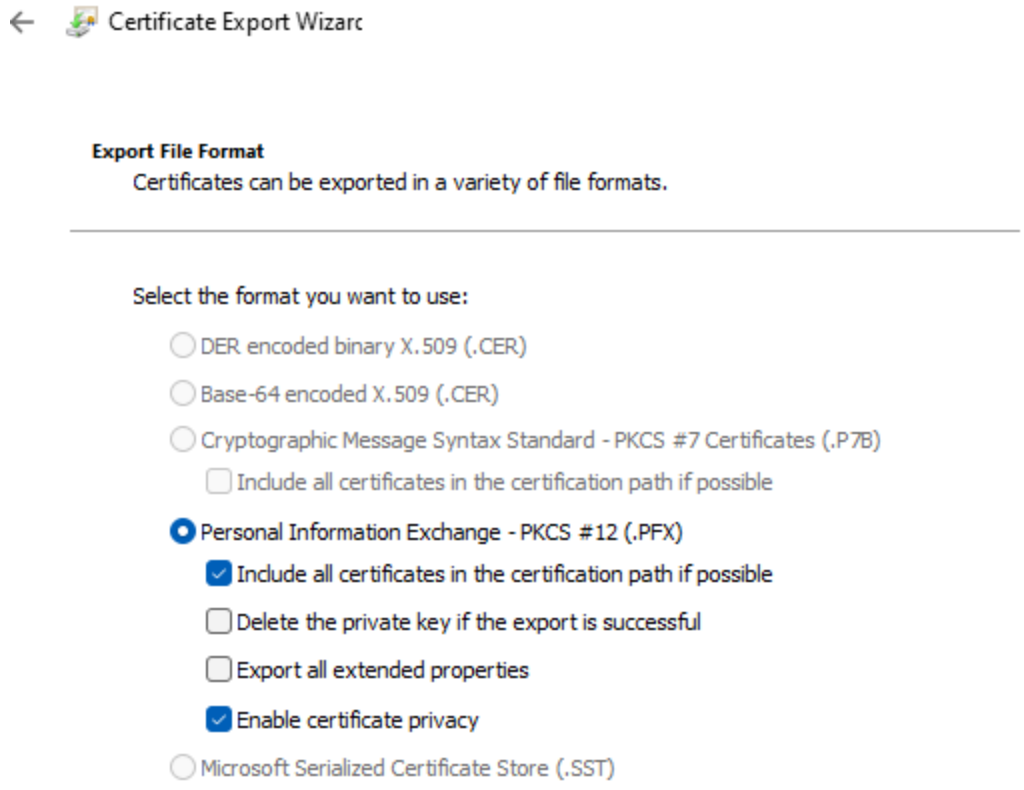
Right-click in our script name then **All Tasks/export.**



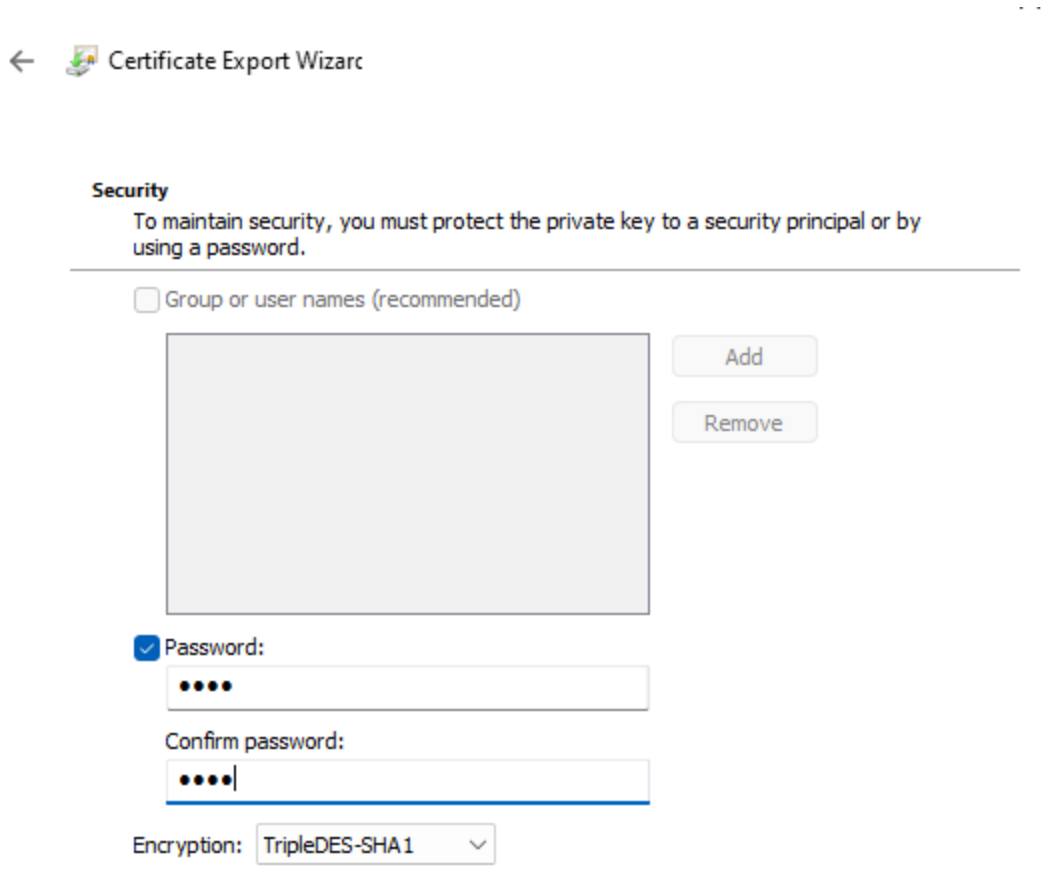
Export with private key.



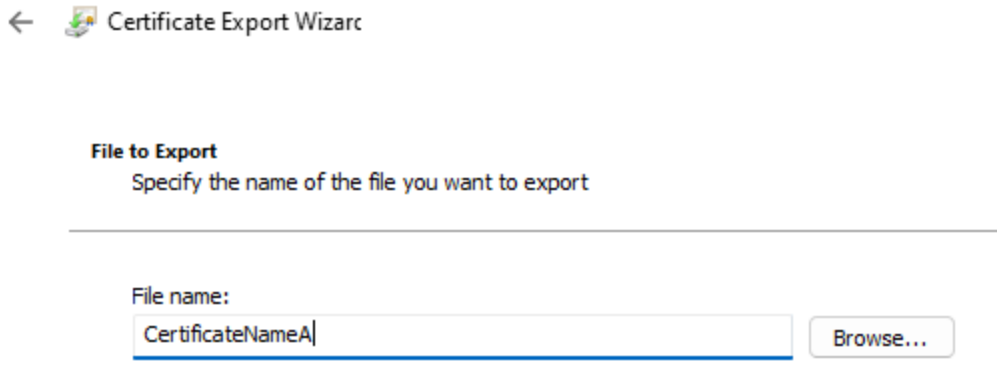
Choose Personal Information Exchange option.



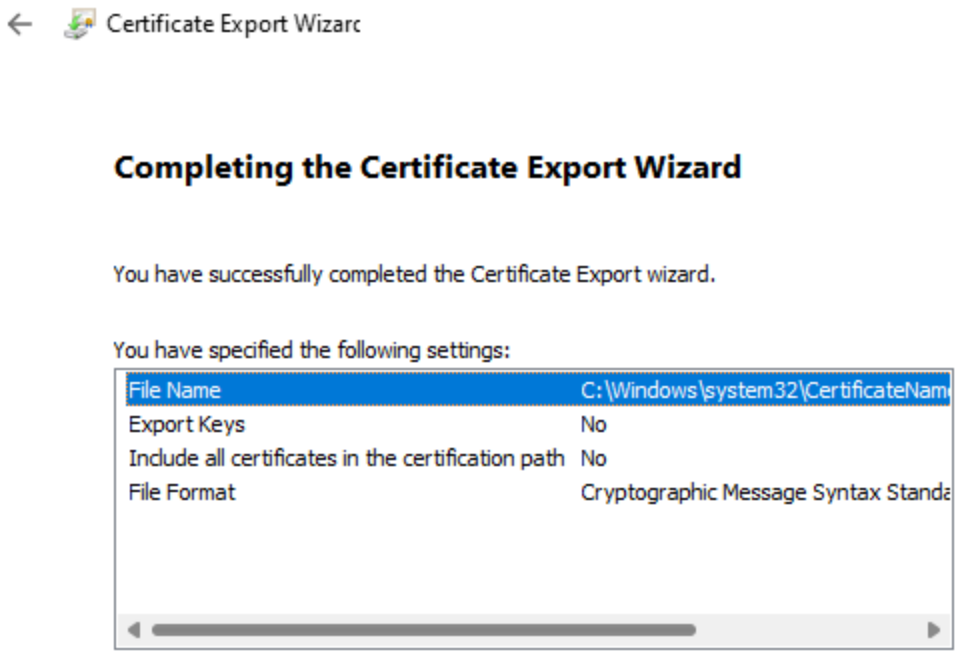
Set a password this password we will be using to import process.



Write the certificate name.



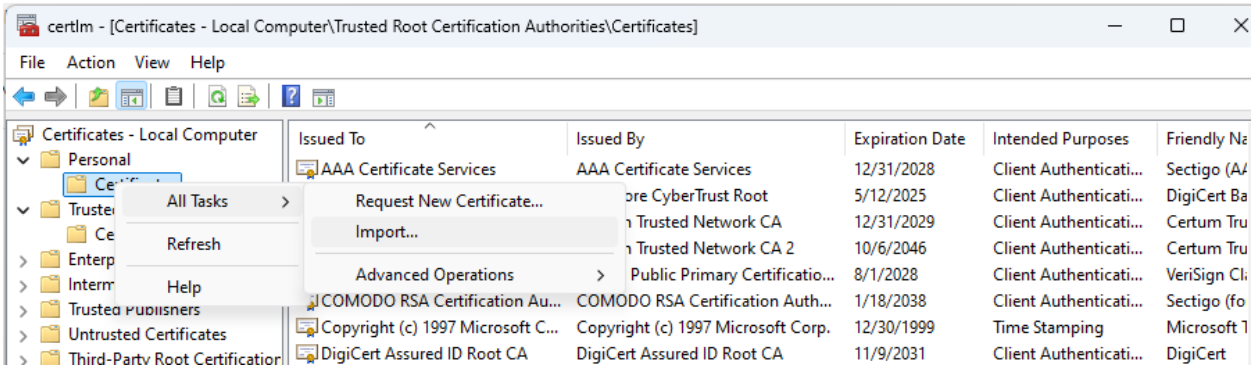
We can see where is stored it.



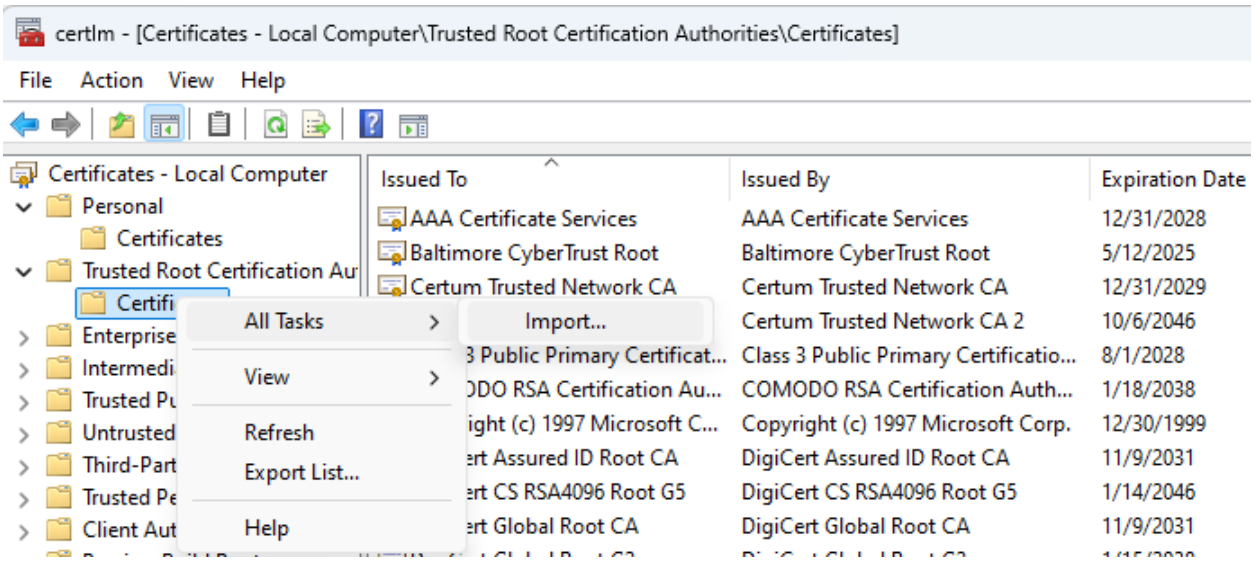
Step 3: Import the certify into the Trusted Root Certification Authorities Store

Open the Certificate Management Console with **certlm.msc**

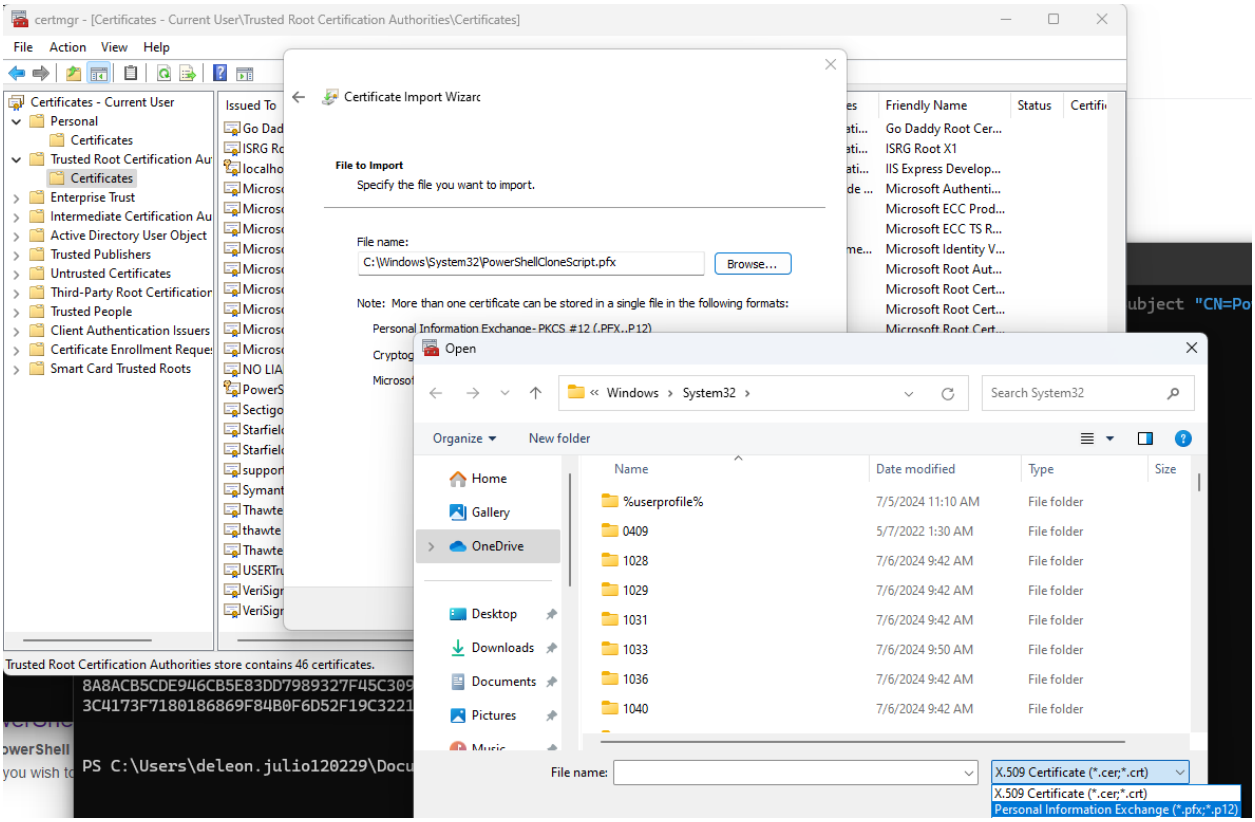
Then we need import our certify exported before in: **Personal/Certificates.**



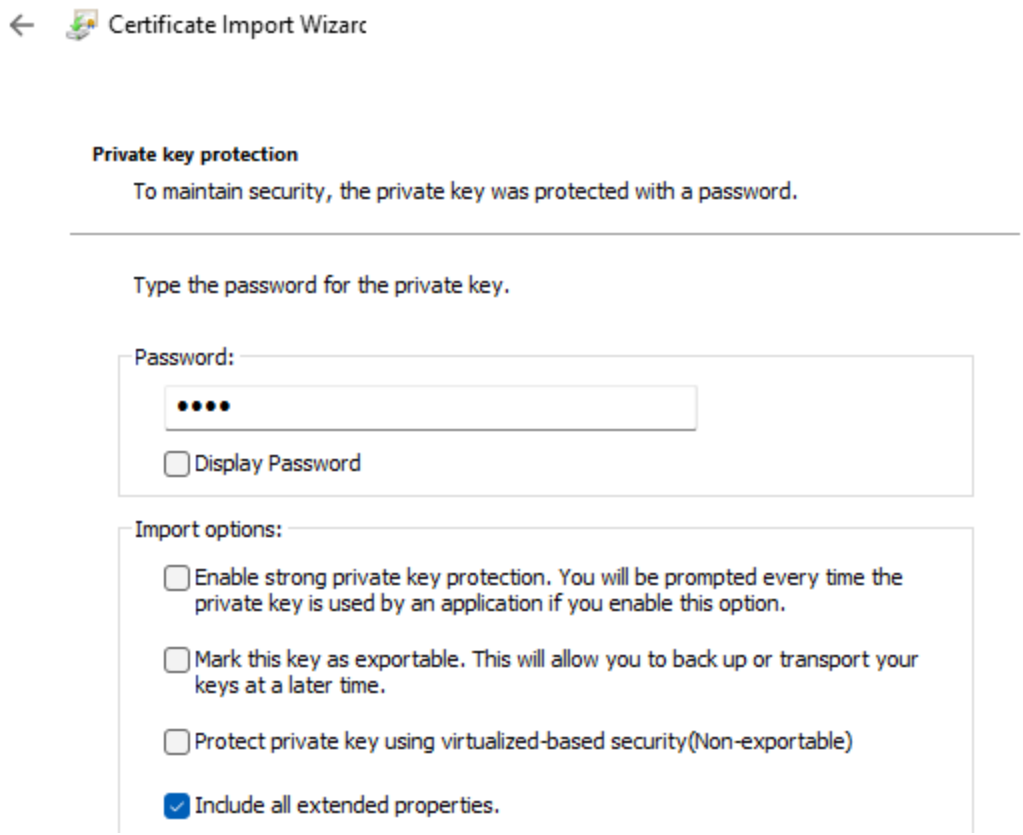
Also we need import our certify exported before in: **Trusted Root Certification Authorities/Certificates folder.**



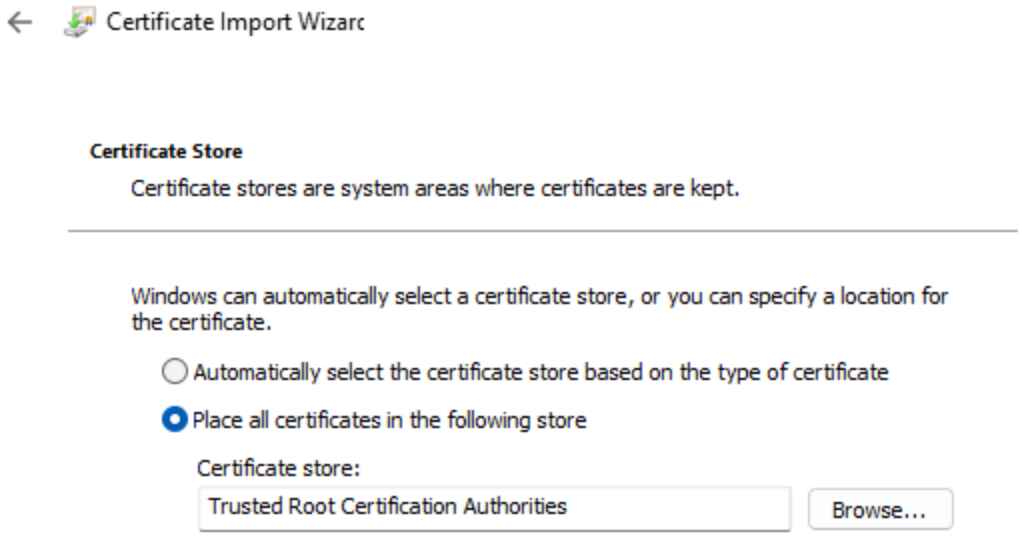
Use Browser option to find your certificate: Choose **Personal Information Exchange** file type, then choose your certificate name.



Write your password



Certificate store: **Trusted Root Certification Authorities**



Step 4: Sign the Script

1. Create reference variable.

   a. command

   $cert = Get-ChildItem -Path Cert:\LocalMachine\My | Where-Object { $_.Subject -eq "CN=PowerShellScriptSigning" }

   b. Command to show certificates

   Get-ChildItem Cert:\LocalMachine\My

2. Sing script

   a. command

   Set-AuthenticodeSignature -FilePath "C:\Users\deleon.julio120229\Documents\clone_script.ps1" -Certificate $cert

Step 5: Configure PowerShell to Allow Only Signed Scripts

Set-ExecutionPolicy AllSigned

Errors

We don’t have certificate.

```
   PSParentPath: Microsoft.PowerShell.Security\Certificate::CurrentUser\My

Thumbprint                                Subject
----------                                -------
8A8ACB5CDE946CB5E83DD7989327F45C309FDB78  CN=d3736902-30d0-44a2-bdb3-e0826b61d0d8


PS C:\Users\deleon.julio120229\Documents> .\clone_script.ps1
.\clone_script.ps1 : File C:\Users\deleon.julio120229\Documents\clone_script.ps1 cannot be loaded. The file
C:\Users\deleon.julio120229\Documents\clone_script.ps1 is not digitally signed. You cannot run this script on the current system.
For more information about running scripts and setting execution policy, see about_Execution_Policies at
https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ .\clone_script.ps1
+ ~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\deleon.julio120229\Documents>
```

We don’t generate import our certificate correctly in: **Personal/Certificates.** .

```
Windows PowerShell
PS C:\Users\deleon.julio120229\Documents> $cert = Get-ChildItem -Path Cert:\LocalMachine\My | Where-Object { $_.Subject -eq "CN=PowerShellScri
ptSigning" }
PS C:\Users\deleon.julio120229\Documents> Set-AuthenticodeSignature -FilePath "C:\Users\deleon.julio120229\Documents\clone_script.ps1" -Certif
icate $cert
Set-AuthenticodeSignature : Cannot bind argument to parameter 'Certificate' because it is null.
At line:1 char:107
+ ... ers\deleon.julio120229\Documents\clone_script.ps1" -Certificate $cert
+                                                                     ~~~~~
    + CategoryInfo          : InvalidData: (:) [Set-AuthenticodeSignature], ParameterBindingValidationException
    + FullyQualifiedErrorId : ParameterArgumentValidationErrorNullNotAllowed,Microsoft.PowerShell.Commands.SetAuthenticodeSignatureCommand

PS C:\Users\deleon.julio120229\Documents>
```

We don’t import the certificate in **Trusted Root Certification Authorities/Certificates folder.**

```
PS C:\Users\deleon.julio120229\Documents> .\clone_script.ps1
.\clone_script.ps1 : File C:\Users\deleon.julio120229\Documents\clone_script.ps1 cannot be loaded. A certificate chain processed, but
terminated in a root certificate which is not trusted by the trust provider.
At line:1 char:1
+ .\clone_script.ps1
+ ~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\deleon.julio120229\Documents> .\clone_script.ps1
```

Reference

https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_signing?view=powershell-7.4