SAPIEN

Quick Guide

Digitally signing your VBScript and PowerShell scripts is definitely a scripting best practice and is easier than you think.

Script Signing Made Simple

Writing secure scripts is becoming a more common requirement for many organizations. You can configure both your VBScript and PowerShell environments to only execute scripts that have been digitally signed. Let me show you how easy this can be.

The first thing you need is a digital certificate; more specifically, a code signing certificate of the Microsoft Authenticode type. This certificate must be trusted by all computers that will be executing your scripts. Acquiring such a certificate is beyond the scope of this article, but if you already have an Active Directory based public key infrastructure, this shouldn't be too difficult.

An alternative for testing purposes is to use the command line tool MAKECERT.EXE from the .NET Framework SDK. This tool will create a self-signed code-signing certificate. The major consideration is that the certificate is only trusted on your computer which means signed scripts won't run anywhere else. Still, this is a useful tool to test the script signing process.

First you need to create a self signed root certificate. This command should get the job done:

```
PS C:\> makecert -n
"CN=%COMPUTERNAME% CodeSigning
Root Certificate" -a sha1 -eku
1.3.6.1.5.5.7.3.3 -r -sv
root.pvk root.cer -ss Root -len
1024 -sr localMachine
```

Enter a password when prompted. Then create the code signing certificate with this command:

```
PS C:\> makecert -pe -n
"CN=%computername%\%username%
Code Signing Certificate" -ss MY
-a sha1 -eku 1.3.6.1.5.5.7.3.3 -
iv root.pvk -ic root.cer
```

Again, enter a password when prompted.

The certificates will be created and automatically installed on your computer. Otherwise install your code signing certificate onto your computer by double-clicking on the file. I always let Windows decide where to store it. While you can have as many code signing certify cates installed as you'd like, I'm going to assume you only have one. For most organizations and scripting administrators, that should be sufficient. You should be able to use the same certificate for both VBScript and PowerShell.

To sign either script type, open a file in PrimalScript 2009. Under the Script menu, select "Sign Script". That's it. PrimalScript executes the language appropriate command to sign the script with your default certificate. If you scroll to the end of your script you should see a commented script block. You don't have to do anything else. If your PowerShell execution policy is configured to run digitally signed scripts, this script should execute just fine, assuming the root certificate is trusted on any computer running your script.

Of course, if you modify the script in any way, even adding a single space, the digital signature will be invalid and you'll have to re-sign your script. This can get tedious, especially if you are creating or modifying many scripts a day. For

those situations you want to configure PrimalScript to always sign your scripts.

Open the Options dialog box from the Tools menu. Select Script Security under Script Settings and you should see a dialog box like this.

Since you only have a single code signing certificate, there is very little you need to configure here other than checking the boxes to automatically sign scripts when saving. PrimalScript will use the first code signing certificate it discovers. More than likely this is the only code signing certificate you have installed.

However, you can browse to a specific certificate file for either language. The certificate must be in the PFX file format. Enter the certificate password when prompted. PrimalScript will securely store the associated password

From this point on, every time you modify a script or create a new script and save it, PrimalScript will automatically sign the script with your certificate. You don't even have to think about it.

Jeffery Hicks

Signing your scripts is a best practice, so why aren't you?