Windows PowerShell用スクリプト形式編

改版履歴			
版数	日付	内容	担当
V.1.0	2015/4/1	初版	NII
V.2.0	2018/2/26	動作環境の変更に伴う修正	NII
V.2.1	2018/8/21	タイムスタンプ利用手順の追加	NII
V.2.2	2020/6/4	中間CA証明書のURLとリポジトリのURLの変更	NII
V.2.3	2020/12/22	中間CA証明書のURL変更	NII
V.2.4	2020/12/24	鍵長の変更	NII
V.2.5	2021/5/31	コード署名用証明書の中間CA証明書を修正	NII
V.2.6	2023/5/19	コード署名用証明書のCSR作成に関する手順を削除	NII

目次

1. コード署名用証明書の利用
1-1. 前提条件
1-2. PKCS#12ファイルの作成
1-2-1. 事前準備
1-2-2. PKCS#12ファイルの作成
1-3. 署名
1-4. コード署名確認作業

1. コード署名用証明書の利用

1-1. 前提条件

OpenSSLコード署名用証明書を使用する場合の前提条件について記載します。適宜、コード署名用証明書をインストールする利用管理者様の環境により、読み替えをお願いします。
（本マニュアルではWindows PowerShell 5.0での実行例を記載しております。）
コマンドプロンプト上で実行するコマンドは、「＞」にて示しています。

前提条件
OpenSSLがインストールされていること

1-2. PKCS#12ファイルの作成

本章ではPKCS#12ファイルの作成方法について記述します。

1-2-1. 事前準備

事前準備として、「ルートCA証明書」、「中間CA証明書」、「コード署名用証明書」を取得してください。

事前準備

1. 「証明書の申請から取得まで」で受領したコード署名用証明書を任意の名前で任意の場所に保存してください。
2. 「ルートCA証明書」と「中間CA証明書」を準備し、この2つを連結させます。下記URLより、リポジトリへアクセスしてください。

   「中間CA証明書」を下記リポジトリより取得してください。
   セコムパスポート for Member 2.0 PUB リポジトリ：
   https://repo1.secomtrust.net/spcpp/pfm20pub/index.html

   【2021年5月31日00:00以前の発行証明書が対象】
   リポジトリ内にある「証明書の種類」より中間CA証明書を取得してください。

   https://repo1.secomtrust.net/spcpp/pfm20pub/codecag2/CODECAG2.cer

   次に、「ルートCA証明書」を下記リポジトより取得してください。
   Security Communication RootCA2 リポジトリ：
   https://repository.secomtrust.net/SC-Root2/index.html

   Security Communication RootCA2 証明書：
   https://repository.secomtrust.net/SC-Root2/SCRoot2ca.cer

   【2021年5月31日00:00以後の発行証明書が対象】

   リポジトリ内にある「証明書の種類」より中間CA証明書を取得してください。

   https://repo1.secomtrust.net/spcpp/pfm20pub/codecag2/CODECAG2SCROOTCA3.cer

   次に、「ルートCA証明書」を下記リポジトより取得してください。
   Security Communication RootCA3 リポジトリ：
   https://repository.secomtrust.net/SC-Root3/index.html

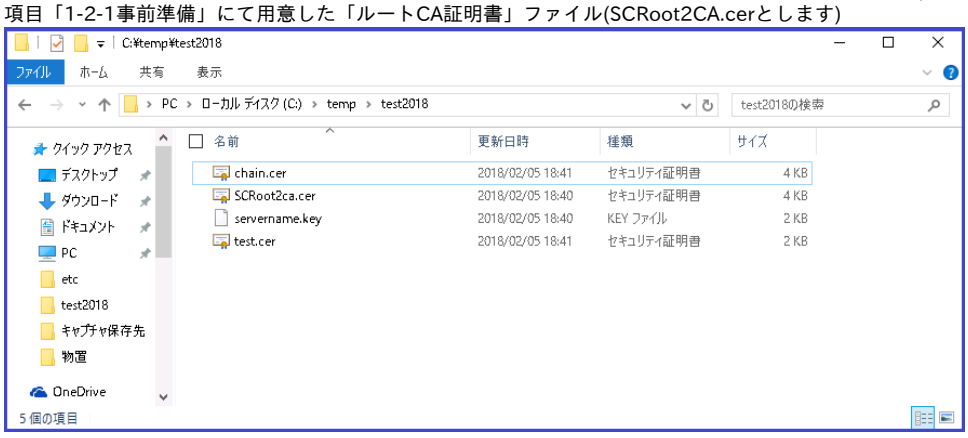
   Security Communication RootCA3 証明書：
   https://repository.secomtrust.net/SC-Root3/SCRoot3ca.cer

1-2-2. PKCS#12ファイルの作成

本項目ではWindowsOS上で任意のフォルダにPKCS#12ファイルを作成する方法を記述します。
以下は、例としてWindows10上での作成方法を記載します。

PKCS#12ファイルの作成

1. 任意のフォルダ(ここではC:\temp\test2018とします)にて以下の4つのファイルを用意してください。
   a. 項目「鍵ペアの生成」にて作成した鍵ペアのファイル(servername.keyとします)
   b. 項目「証明書の申請から取得まで」にて取得したコード署名用証明書(test.cerとします)
   c. 項目「1-2-1事前準備」にて用意した「ルートCA証明書」と 「中間CA証明書」を連結させたファイル(chain.cerとします)
   d. 項目「1-2-1事前準備」にて用意した「ルートCA証明書」ファイル(SCRoot2CA.cerとします)



2. CAfile に指定する証明書をDER形式からPEM形式に変換します。

・Security Communication RootCA2の場合
openssl x509 -inform der -in SCRoot2ca.cer -outform pem -out SCRoot2ca.cer

・中間CA証明書（2021年5月31日00:00以前の発行証明書が対象）の場合
openssl x509 -inform der -in CODECAG2.cer -outform pem -out CODECAG2.cer

3. コマンドプロンプト上にて上記で取得した「ルートCA証明書」と「中間CA証明書」を以下のコマンドにより、連結させてください。中間CA証明書の下部にルートCA証明書が併記されるファイルとなります。

＞ type (中間CA証明書のパス) (ルートCA証明書のパス) > (出力するファイル名)

4. 連結したファイルがPEM形式になっていることを確認してください。
例）PEM形式の証明書

----BEGIN CERTIFICATE----
MIIEcTCCA1mgAwIBAgIIasWHLdnQB2owDQYJKoZIhvcNAQELBQAwbzELMAkGA1UE
BhMCSlAxFDASBgNVBAcMC0FjYWRlbWUtb3BzMSowKAYDVQQKDCFOYXRpb25hbCBJ
bnN0aXR1dGUgb2YgSW5mb3JtYXRpY3MxHjAcBgNVBAMMFU5JSSBPcGVyYXRpbmcg
Q0EgLSBHMjAeFw0xNTAzMTIwMTA4MDJaFw0xNzA0MTEwMTA4MDJaMHAxCzAJBgNV
（中略）
LmeW0e/xkkxwdmKv5y5txLlFcp53AZl/vjn3BHp42PFkkTISEmAUiCtQ2A25QDRR
RG33IaC8E8Tl/SnOA8h95XQtGWm47PrIjXyYtIe0rFousbpIoW8MZw4gDXVQ3485
XEftqwwIMcLNxttJ6i6f9XVyPMRhHy9rdDPseHiXayxcBxJMuw==
----END CERTIFICATE----

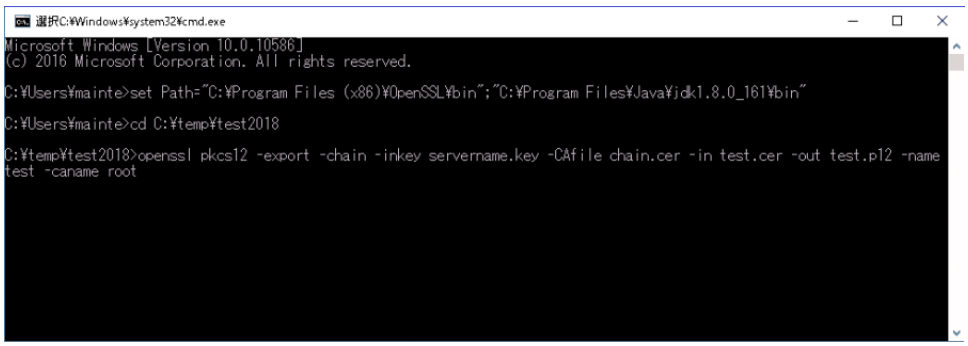
5. コマンドプロンプトを開き、ファイルのある任意のフォルダ(ここではC:\temp\test2018)へ移動します。

＞ set Path=**(OpenSSLインストールディレクトリ)**\bin
※OpenSSLインストールディレクトリをプログラムを探すディレクトリに指定します

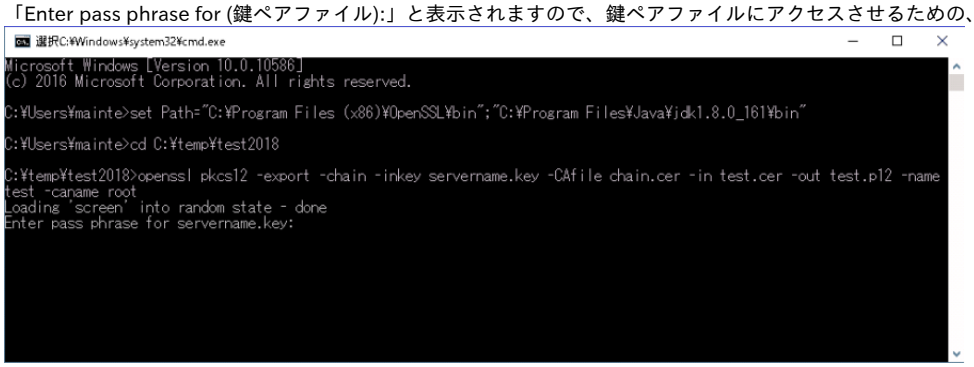
＞ cd **(作業ディレクトリ) ←作業ディレクトリ**

6. 移動後、以下のコマンドを実行しPKCS#12ファイルを作成してください。

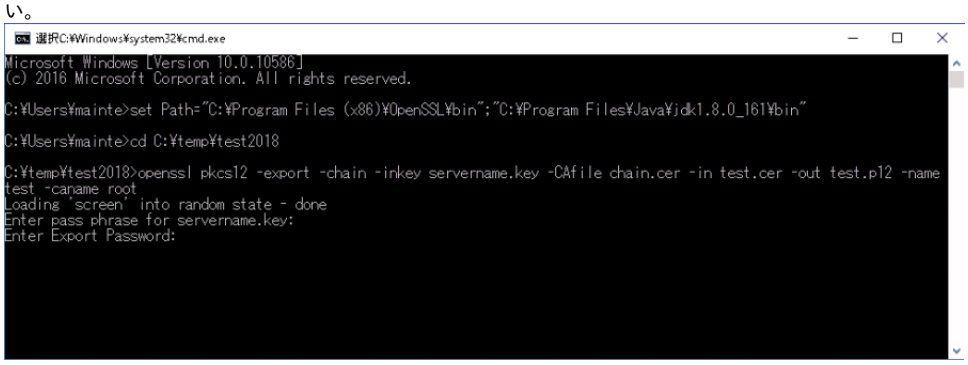
＞ openssl pkcs12 -export -chain -inkey (鍵ペアのファイル名) -CAfile (ルートCA証明書と中間CA証明書を連結させたファイル) -in (コード署名用の証明書ファイル名) -out (PKCS#12形式で出力するファイル名) -name (コード署名用証明書のエイリアス名) -caname (ルートCA証明書と中間CA証明書のエイリアス名)



7. 「Enter pass phrase for (鍵ペアファイル):」と表示されますので、鍵ペアファイルにアクセスさせるための、アクセスPINを入力してください。



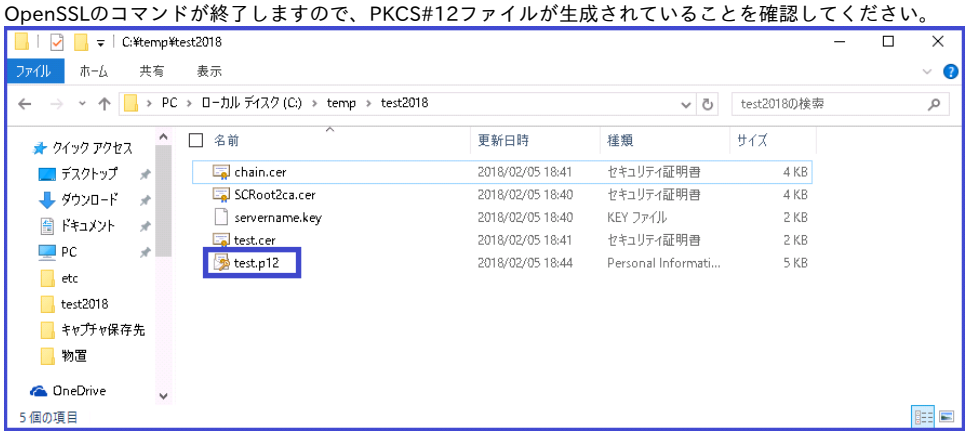
8. 「Enter Export Password:」と表示されますので、PKCS#12形式のファイルを保護するためのアクセスPINとして任意の文字列を入力してください。



9. 「Verifying - Enter Export Password:」と表示されますので、確認のため、同じアクセスPINを再度入力してください。



10. OpenSSLのコマンドが終了しますので、PKCS#12ファイルが生成されていることを確認してください。

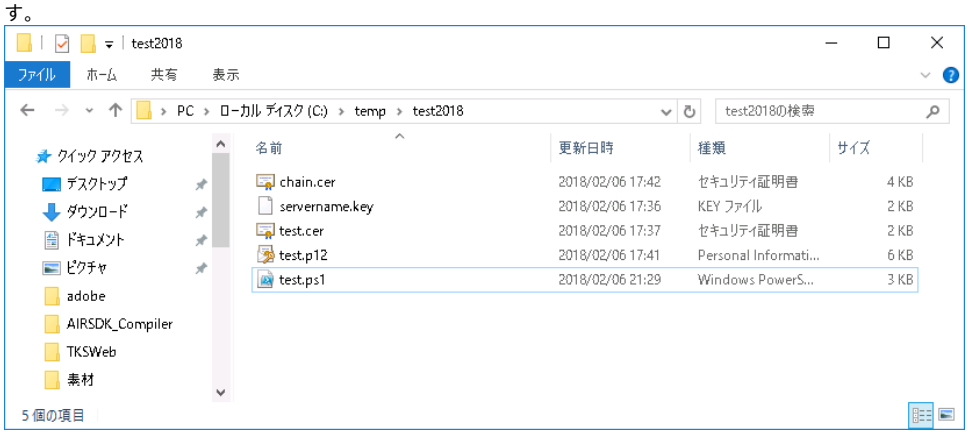


1-3. 署名

本章では、Windows PowerShell用スクリプト形式のファイルにWindowsOS上でデジタル署名をする方法について記述します。
以下は、例としてWindows10上での作成方法を記載します。
Windows PowerShell用スクリプト形式のファイルへの署名はファイル生成時にWindows Powershellを利用して署名します。
タイムスタンプ利用の是非により手順が異なりますので、用途に応じてご参照ください。

署名作業（併せてタイムスタンプを付与しない場合）

1. 同一フォルダ上に署名するWindows PowerShell用スクリプト形式のファイル(test.ps1)と項目1-2-2にて 生成したPKCS#12ファイルを置きます。

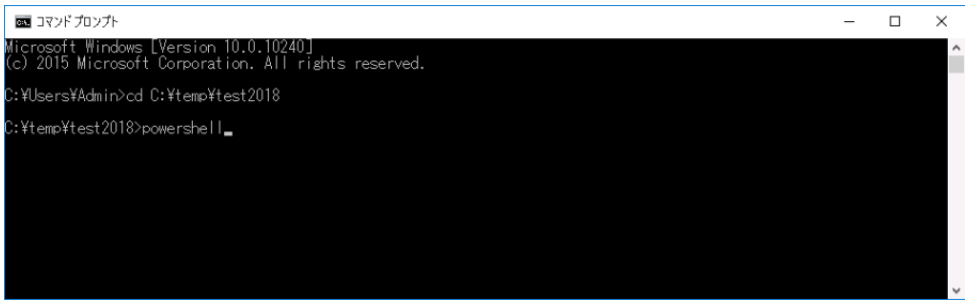


2. コマンドプロンプトを実行し、署名対象ファイルのあるフォルダへ移動します。

```
＞cd (作業ディレクトリ) ←作業ディレクトリ
```

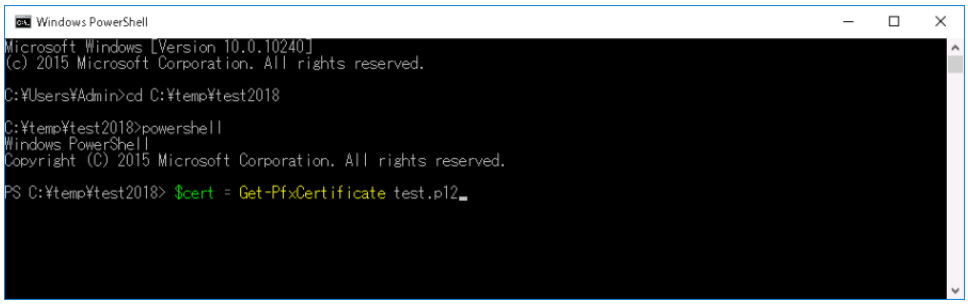
3. フォルダ移動後、PowerShellを起動するため以下のコマンドを実行してください。

```
＞powershell
```

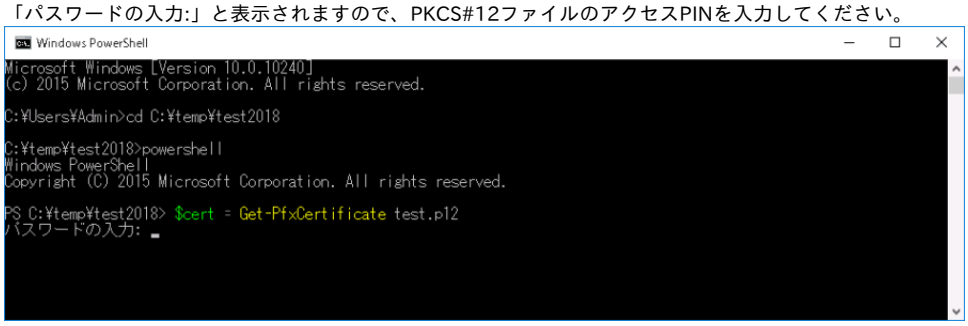


4. PowerShell起動後、「$cert」にコード署名用証明書の情報を読み込むため、以下のコマンドを実行してください。

```
> $cert = Get-PfxCertificate test.p12
```

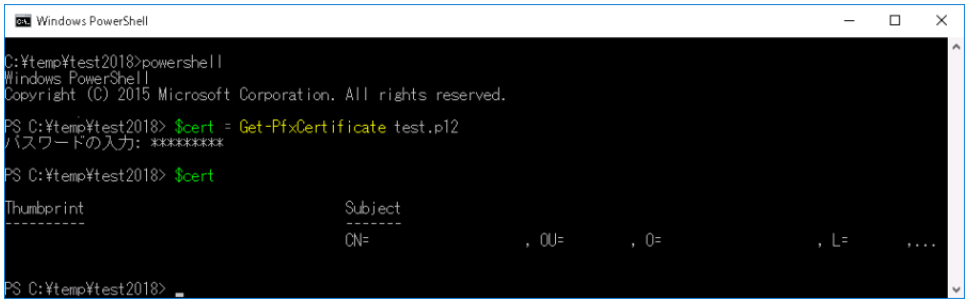


5. 「パスワードの入力:」と表示されますので、PKCS#12ファイルのアクセスPINを入力してください。



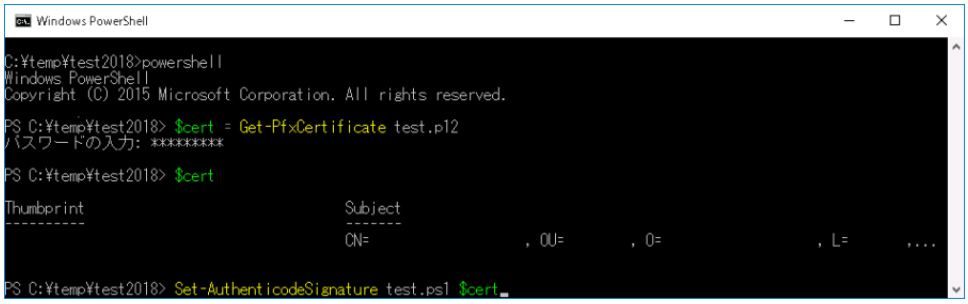
6. コード署名用証明書の情報を確認するため、以下のコマンドを実行してください。

```
> $cert
```

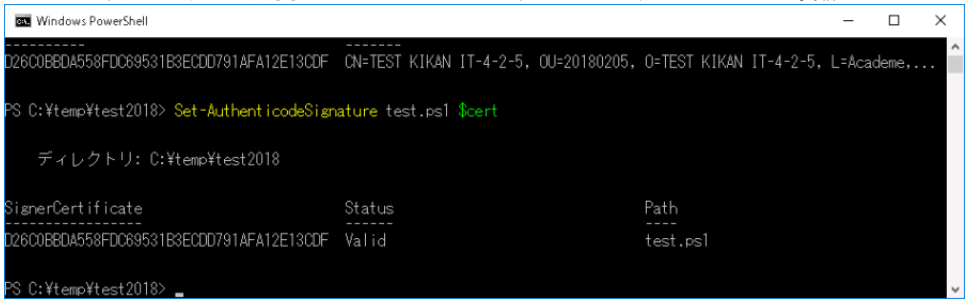


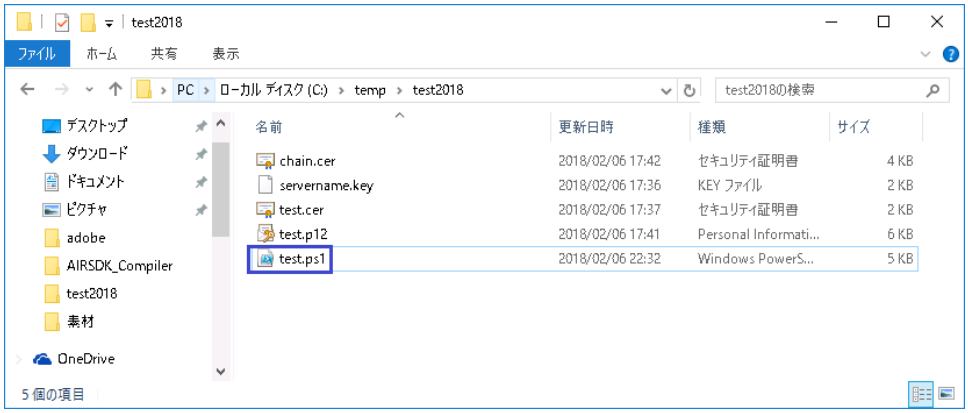
7. コード署名用証明書の情報を確認後、Windows PowerShell用スクリプト形式のファイルへの署名を以下のコマンドを実行してください。

```
> Set-AuthenticodeSignature （署名したいWindows PowerShell用スクリプト形式のファイル名） $cert
```



8. コマンドが終了しますので、対象のWindows PowerShell用スクリプト形式のファイルが更新されていることを確認してください。





署名作業（併せてタイムスタンプを付与する場合）
以下と同様の手順となりますので、ご参照ください。 「コード署名用証明書利用マニュアル」-「Windows用(.exe,.cab,.dll)形式編」-「1-3. 署名」-「署名作業(併せてタイムスタンプを付与する場合)」

1-4. コード署名確認作業

本章ではデジタル署名したWindows PowerShell用スクリプト形式のファイルのコード署名確認作業について記述します。

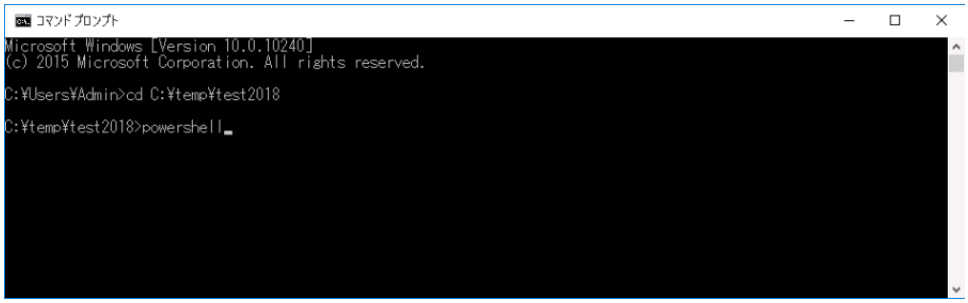
署名確認作業（併せてタイムスタンプを付与しない場合）

1. コマンドプロンプトを実行し、署名対象ファイルのあるフォルダへ移動します。

   ＞ cd **(作業ディレクトリ) ←作業ディレクトリ**

2. フォルダ移動後、PowerShellを起動するため以下のコマンドを実行してください。

   ＞ powershell



3. PowerShell起動後、「&"」の後にWindows PowerShell用スクリプト形式のパス名を書き、 その後「"」で閉じてから実行してください。

   ＞ &"(署名したWindows PowerShell用スクリプト形式のファイル名)"

   ①署名が正しく検証されると、 正常に実行されます。

   ②署名が正しく検証されないと、 次のように表示されます。

   ファイル （作業ディレクトリ）(署名したWindows PowerShell用スクリプト形式のファイル名)を読み込めません。ファイル （作業ディレクトリ）(署名したWindows PowerShell用スクリプト形式のファイル名) の内容は改ざんされている可能性があります。ファイルのハッシュが、 デジタル署名に保存されているハッシュと一致しません。このスクリプトはシステムで実行されません。詳細については、「get-help about_signing」と入力してヘルプを参照してください。
   発生場所 行:1 文字:2
   + & <<<< "(署名したWindows PowerShell用スクリプト形式のファイル名)"
   + CategoryInfo : NotSpecified: (:) []、PSSecurityExc
   eption
   + FullyQualifiedErrorId : RuntimeException
   |

署名確認作業（併せてタイムスタンプを付与した場合）

以下と同様の手順となりますので、ご参照ください。
「コード署名用証明書利用マニュアル」-「Windows用(.exe,.cab,.dll)形式編」-「1-4-2. GUI操作によるコード署名確認」-「署名確認作業（併せてタイムスタンプを付与した場合）」