

Sistema que permite almacenar en un fichero los credenciales de usuario de Windows haciendo una petición a un servidor web mediante el método POST. Después obtener el password en texto plano

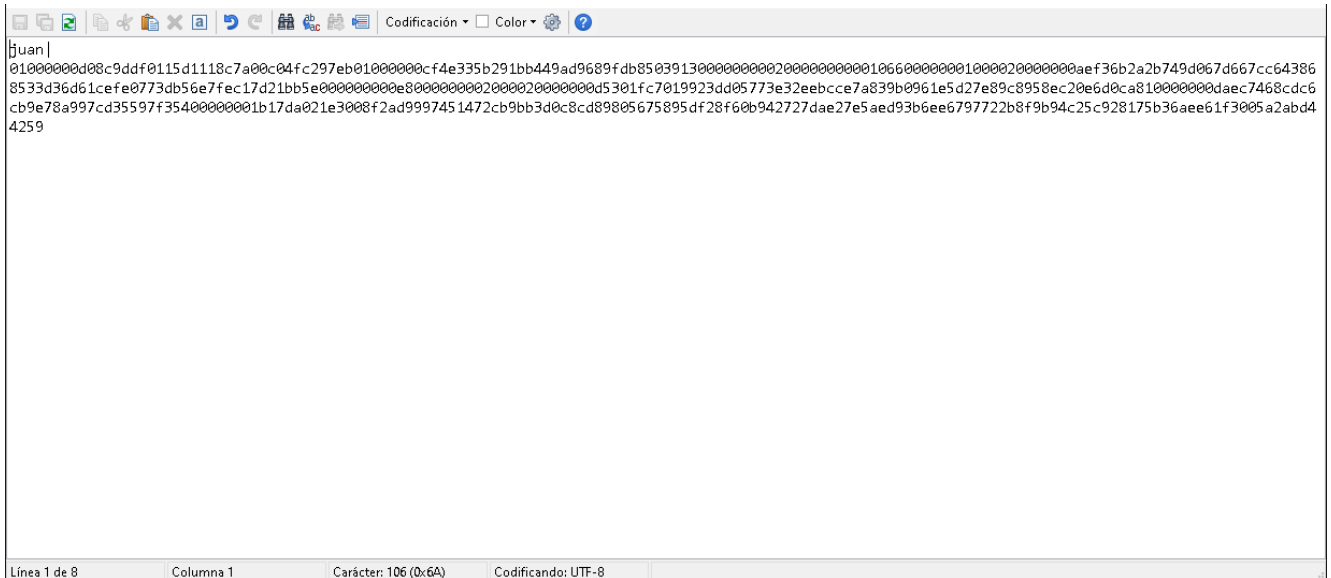
Fichero que almacena los credenciales que llegan por método POST

Cuando llegan los credenciales se almacenan en un fichero cuyo nombre es fichero12.txt, el usuario y el password se separan por el caracter «|».

Script en PowerShell que envía los credenciales

[crayon-6a9d4a667f9b0533714726/]

Ejemplo de fichero con los credenciales que se crea en el servidor



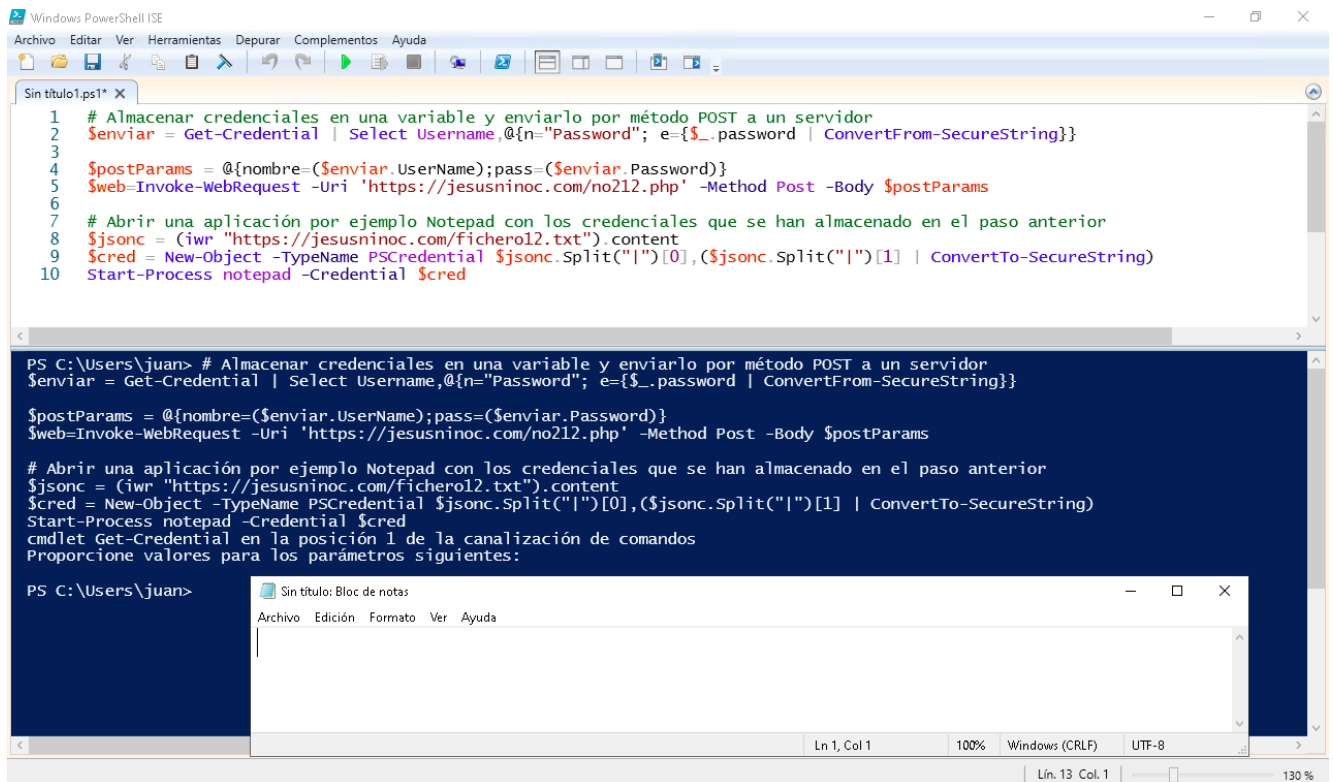
A screenshot of a Notepad window. The title bar shows standard Windows icons and the text 'Codificación' and 'Color'. The text area contains a long string of Base64-encoded data, starting with 'Juan|'. The status bar at the bottom indicates 'Línea 1 de 8', 'Columna 1', 'Carácter: 106 (0x6A)', and 'Codificando: UTF-8'.

```
Juan|
01000000d08c9ddf0115d1118c7a00c04fc297eb01000000cf4e335b291bb449ad9689fdb850391300000000020000000000106600000001000020000000aef36b2a2b749d067d667cc64386
8533d36d61cfe0773db56e7fec17d21bb5e00000000e80000000020000200000000d5301fc7019923dd05773e32eebccce7a839b0961e5d27e89c8958ec20e6d0ca810000000daec7468cdc6
cb9e78a997cd35597f35400000001b17da021e3008f2ad9997451472cb9bb3d0c8cd89805675895df28f60b942727dae27e5aed93b6ee6797722b8f9b94c25c928175b36aee61f3005a2abd4
4259
```

Script en PowerShell que lee los credenciales almacenados en el servidor y comprueba si puede abrir una aplicación con dichos credenciales

[crayon-6a9d4a667f9b7964997710/]

Si los credenciales son correctos no aparece mensaje de error y se abre la aplicación



```
1 # Almacenar credenciales en una variable y enviarlo por método POST a un servidor
2 $enviar = Get-Credential | Select Username,@{n="Password"; e={$_.password | ConvertFrom-SecureString}}
3
4 $postParams = @{nombre=($enviar.UserName);pass=($enviar.Password)}
5 $web=Invoke-WebRequest -Uri 'https://jesusninoc.com/no212.php' -Method Post -Body $postParams
6
7 # Abrir una aplicación por ejemplo Notepad con los credenciales que se han almacenado en el paso anterior
8 $jsonc = (iwr "https://jesusninoc.com/fichero12.txt").content
9 $cred = New-Object -TypeName PScredential $jsonc.Split("|")[0],($jsonc.Split("|")[1] | ConvertTo-SecureString)
10 Start-Process notepad -Credential $cred
```

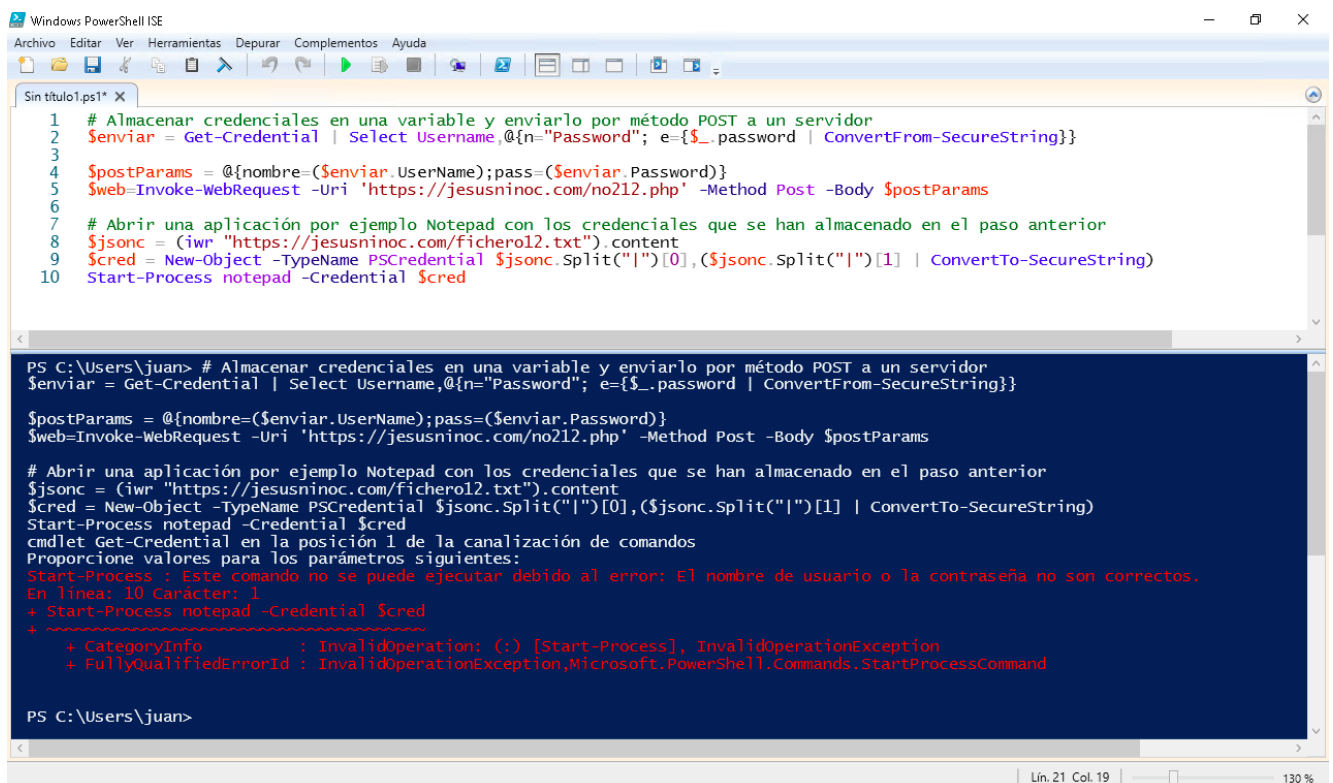
PS C:\Users\juan> # Almacenar credenciales en una variable y enviarlo por método POST a un servidor
\$enviar = Get-Credential | Select Username,@{n="Password"; e={\$_.password | ConvertFrom-SecureString}}

\$postParams = @{nombre=(\$enviar.UserName);pass=(\$enviar.Password)}
\$web=Invoke-WebRequest -Uri 'https://jesusninoc.com/no212.php' -Method Post -Body \$postParams

Abrir una aplicación por ejemplo Notepad con los credenciales que se han almacenado en el paso anterior
\$jsonc = (iwr "https://jesusninoc.com/fichero12.txt").content
\$cred = New-Object -TypeName PScredential \$jsonc.Split("|")[0],(\$jsonc.Split("|")[1] | ConvertTo-SecureString)
Start-Process notepad -Credential \$cred
cmdlet Get-Credential en la posición 1 de la canalización de comandos
Proporcione valores para los parámetros siguientes:

PS C:\Users\juan>

Si los credenciales son incorrectos aparece mensaje de error



```
1 # Almacenar credenciales en una variable y enviarlo por método POST a un servidor
2 $enviar = Get-Credential | Select Username,@{n="Password"; e={$_.password | ConvertFrom-SecureString}}
3
4 $postParams = @{nombre=($enviar.UserName);pass=($enviar.Password)}
5 $web=Invoke-WebRequest -Uri 'https://jesusninoc.com/no212.php' -Method Post -Body $postParams
6
7 # Abrir una aplicación por ejemplo Notepad con los credenciales que se han almacenado en el paso anterior
8 $jsonc = (iwr "https://jesusninoc.com/fichero12.txt").content
9 $cred = New-Object -TypeName PScredential $jsonc.Split("|")[0],($jsonc.Split("|")[1] | ConvertTo-SecureString)
10 Start-Process notepad -Credential $cred
```

PS C:\Users\juan> # Almacenar credenciales en una variable y enviarlo por método POST a un servidor
\$enviar = Get-Credential | Select Username,@{n="Password"; e={\$_.password | ConvertFrom-SecureString}}

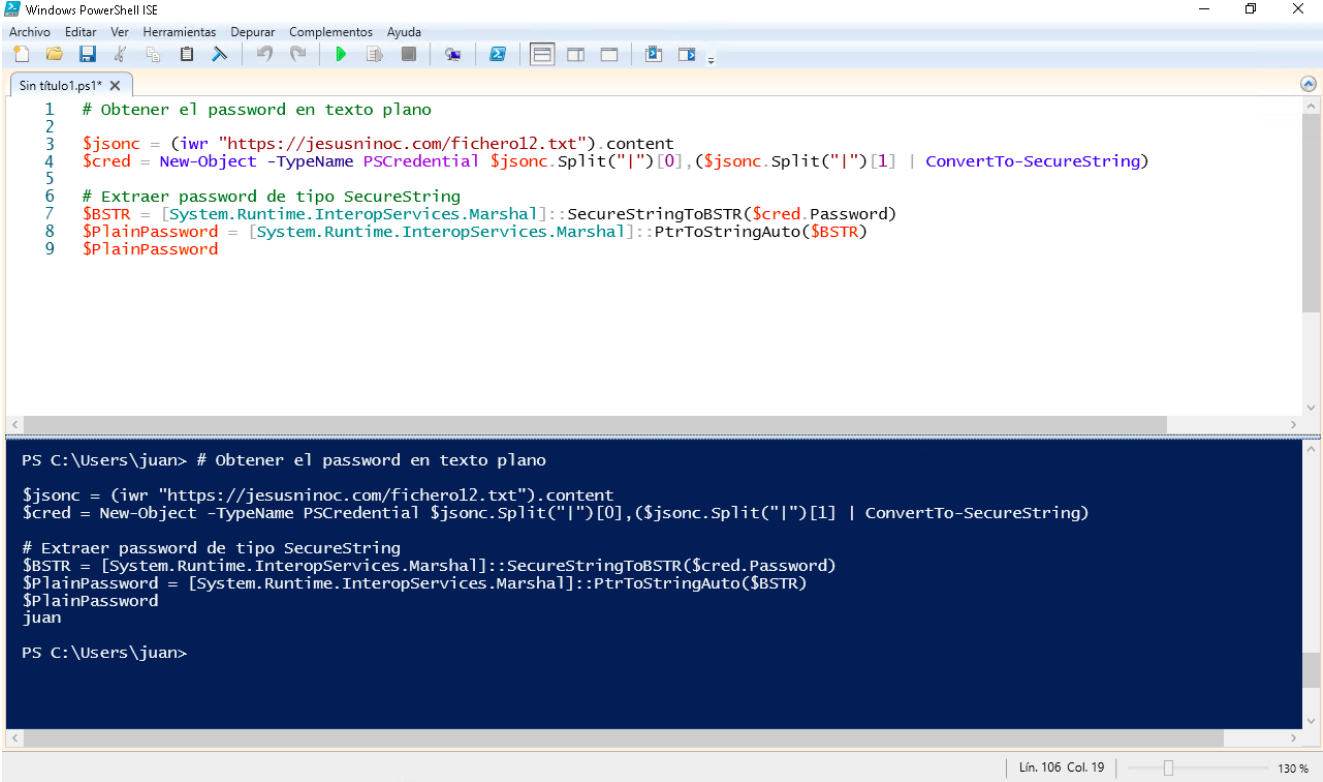
\$postParams = @{nombre=(\$enviar.UserName);pass=(\$enviar.Password)}
\$web=Invoke-WebRequest -Uri 'https://jesusninoc.com/no212.php' -Method Post -Body \$postParams

Abrir una aplicación por ejemplo Notepad con los credenciales que se han almacenado en el paso anterior
\$jsonc = (iwr "https://jesusninoc.com/fichero12.txt").content
\$cred = New-Object -TypeName PScredential \$jsonc.Split("|")[0],(\$jsonc.Split("|")[1] | ConvertTo-SecureString)
Start-Process notepad -Credential \$cred
cmdlet Get-Credential en la posición 1 de la canalización de comandos
Proporcione valores para los parámetros siguientes:
Start-Process : Este comando no se puede ejecutar debido al error: El nombre de usuario o la contraseña no son correctos.
En línea: 10 Carácter: 1
+ Start-Process notepad -Credential \$cred
+ ~~~~~
+ CategoryInfo : InvalidOperation: (:) [Start-Process], InvalidOperationException
+ FullyQualifiedErrorId : InvalidOperationException,Microsoft.PowerShell.Commands.StartProcessCommand

PS C:\Users\juan>

Obtener el password en texto plano

[crayon-6a9d4a667f9b9039774202/]



```
Windows PowerShell ISE
Archivo Editar Ver Herramientas Depurar Complementos Ayuda
Sin título1.ps1* X
1 # Obtener el password en texto plano
2
3 $jsonc = (iwr "https://jesusninoc.com/fichero12.txt").content
4 $cred = New-Object -TypeName PSCredential $jsonc.Split("|")[0],($jsonc.Split("|")[1] | ConvertTo-SecureString)
5
6 # Extraer password de tipo SecureString
7 $BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($cred.Password)
8 $PlainPassword = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)
9 $PlainPassword

PS C:\Users\juan> # Obtener el password en texto plano

$jsonc = (iwr "https://jesusninoc.com/fichero12.txt").content
$cred = New-Object -TypeName PSCredential $jsonc.Split("|")[0],($jsonc.Split("|")[1] | ConvertTo-SecureString)

# Extraer password de tipo SecureString
$BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($cred.Password)
$PlainPassword = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)
$PlainPassword
juan

PS C:\Users\juan>
```

Lin. 106 Col. 19 130 %