

Enviar una cadena segura (System.Security.SecureString) entre un cliente y un servidor cifrando la comunicación mediante el protocolo TCP/IP usando TLS 1.2 con un certificado X509 autofirmado en PowerShell

Servidor que envía una cadena segura cifrando la comunicación

[crayon-6a9d49073f670609632193/]

Cliente que recibe una cadena segura (en el método AuthenticateAsClient se indica tls12)

[crayon-6a9d49073f677046765931/]

Resultado

```
Windows PowerShell ISE
Sin título1.ps1* X
1 $Certificate = $null
2 $TcpClient = New-Object -TypeName System.Net.Sockets.TcpClient
3
4 $TcpClient.Connect("127.0.0.1", "1234")
5 $TcpStream = $TcpClient.GetStream()
6
7 $SslStream = New-Object System.Net.Security.SslStream $TcpStream, $false, ({ $true } -as [System.Security.Authentication.SslProtocols::Tls12], $false)
8
9 $SslStream.AuthenticateAsClient("127.0.0.1", $null, "tls12", $false)
10 $Certificate = $SslStream.RemoteCertificate
11
12 if ($Certificate) {
13     if ($Certificate -isnot [System.Security.Cryptography.X509Certificates.X509Certificate2]) {
14         $Certificate = New-Object -TypeName System.Security.Cryptography.X509Certificate2 $Certificate
15     }
16     Write-Output $Certificate
17 }
18
19 $Mensaje = New-Object System.IO.StreamReader $SslStream
20
21 $Credenciales = $Mensaje.ReadLine() | ConvertTo-SecureString
22
23 $BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($Credenciales)
24 $PlainPassword = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)
25
26 $SslStream.Dispose()
27 $TcpClient.Dispose()
28
29 Thumbprint      Subject
30 -----
31 E5DB05610E5484B2EFB0341CCC1E77C5622C9466 CN=jesusninoc
32 Juanito
33
34 PS C:\Users\Juan>
```

Completado Lin. 20 Col. 56 95 %

```
Windows PowerShell ISE
Sin título1.ps1* X
1
2
3
4 $listener = [System.Net.Sockets.TcpListener][int]$port
5
6
7 $tcpClient = $listener.AcceptTcpClient()
8 $tcpStream = $tcpClient.GetStream()
9 $sslStream = New-Object System.Net.Security.SslStream $tcpStream, $false, ({ $true } -as [System.Security.Authentication.SslProtocols::Tls12], $false)
10 $sslStream.AuthenticateAsServer($Certificate, $false, [System.Security.Authentication.SslProtocols::Tls12], $false)
11
12 $mensaje = (New-Object System.IO.StreamWriter $sslStream)
13
14 word | ConvertFrom-SecureString
15
16
17
18
19
```

Proporcione valores para los parámetros siguientes:

```
PS C:\Users\Juan> $Certificate = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2 $Certificate
$port = "1234"
$listener = [System.Net.Sockets.TcpListener][int]$port
$listener.start()
$tcpStream = $tcpListener.AcceptTcpClient()
$sslStream = New-Object System.Net.Security.SslStream $tcpStream.GetStream(), $false, ({ $true } -as [System.Security.Authentication.SslProtocols::Tls12], $false)
$mensaje = (New-Object System.IO.StreamWriter $sslStream)
# Enviar password seguro
$var = (Get-Credential).Password | ConvertFrom-SecureString
$mensaje.Write($var)
$mensaje.Dispose()
$tcpListener.Stop()
cmdlet Get-Credential en la posición 1 de la canalización de comandos
Proporcione valores para los parámetros siguientes:
```

Lin. 14 Col. 13 95 %