My NET.MAUI Android app is throwing SSLPeerUnverifiedException: Hostname 192.168.100.2 not verified when sending POST request to a PHP script located on my desktop's IIS localhost, as shown below.

```csharp
await Task.Run(() =>
{
                                        URL urlWebhook = new URL("https://192.168.100.2:443/medservices/ReceiveWebhooks.php");
          HttpsURLConnection? connection = urlWebhook.OpenConnection() as HttpsURLConnection;
   if(connection != null)
   {
      connection.DoInput = true;
      connection.DoOutput = true;
      connection.RequestMethod = "POST";
      connection.ConnectTimeout = 10000;
      connection.ReadTimeout = 10000;
      connection.SetRequestProperty("Content-Type", "application/json");
      connection.SetRequestProperty("Accept", "*/*");
      connection.SetRequestProperty("User-Agent", "SmsGateway/1.0");
                                          BufferedOutputStream os = new BufferedOutputStream(connection.OutputStream);
      os.Write(Encoding.Default.GetBytes(strRequestBody));
      os.Close();
      isSent = connection.ResponseCode == HttpStatus.Ok;
      connection.Disconnect();
   }
});
```

To configure https binding and SSL in IIS, I used the following PowerShell script, which also exports the SSL certificate in pfx and pem formats.

```powershell
$cert_params =
@{
        Type = "Custom"
        DnsName = "192.168.100.2"
        FriendlyName = "IIS Express SSL Server Authentication Certificate"
        CertStoreLocation = "Cert:\LocalMachine\My"
        NotAfter = (Get-Date).AddYears(10)
        KeyAlgorithm = "RSA"
        KeyLength = 2048
        KeyUsageProperty = "All"
        KeyUsage = "KeyEncipherment", "DataEncipherment", "DigitalSignature"
        KeyExportPolicy = "Exportable"
```

```
    HashAlgorithm = "SHA256"
    TextExtension = @("2.5.29.37={text}1.3.6.1.5.5.7.3.1")
}
$cert = New-SelfSignedCertificate @cert_params
$store = New-Object System.Security.Cryptography.X509Certificates.X509Store([System.Security.Cryptography.X509Certificates.StoreName]::Root, [System.Security.Cryptography.X509Certificates.StoreLocation]::LocalMachine)
$store.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::ReadWrite)
$store.Add($cert)
$store.Close()
$export_params =
@{
    Cert = $cert
    FilePath = "C:\Users\pnfst\Documents\IISExpress\iis_ssl_server_certificate.pfx"
    ChainOption = "EndEntityCertOnly"
    Password = ConvertTo-SecureString -String "XXXXXXXX" -Force -AsPlainText
}
Export-PfxCertificate @export_params
$certBase64 = [System.Convert]::ToBase64String($cert.RawData, [System.Base64FormattingOptions]::InsertLineBreaks)
$pem = @"
-----BEGIN CERTIFICATE-----
$certBase64
-----END CERTIFICATE-----
"@
Out-File -FilePath "C:\Users\pnfst\Documents\IISExpress\iis_ssl_server_certificate.cer" -InputObject $pem -Encoding ascii
Import-Module WebAdministration
New-WebBinding -Name "Default Web Site" -IPAddress "192.168.100.2" -Port 443 -HostHeader "" -Protocol "https" -SslFlags 0
$binding = Get-WebBinding -Name "Default Web Site" -Port 443 -Protocol "https"
$binding.AddSslCertificate($cert.Thumbprint, "My")
Set-WebConfiguration -Location "Default Web Site" -Filter "system.webServer/security/access" -Value "Ssl"
Restart-WebItem -PSPath "IIS:\Sites\Default Web Site"
```

I then used created pem to import certificate into the Android user's truststore. I also added this pem file to my app's Platforms\Android\Resources\raw folder, set the build action AndroidResource and created AndroidResource network_security_config.xml in the Platforms\Android\Resources\xml folder with the build action AndroidResource and the following content.

```xml
```

```
<?xml version="1.0" encoding="utf-8" ?>
<network-security-config>
  <domain-config>
    <domain includeSubdomains="true">192.168.100.2</domain>
    <trust-anchors>
      <certificates src="@raw/iis_ssl_server_certificate"/>
    </trust-anchors>
  </domain-config>
</network-security-config>
```
And added link to this network security configuration to my AndroidManifest:
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
  <application
    android:networkSecurityConfig="@xml/network_security_config"
    android:allowBackup="true"
    android:supportsRtl="true"
    android:label="SmsGateway">
  </application>
  <!--
  ...
  -->
  <uses-permission android:name="android.permission.INTERNET" />
</manifest>
```
However, none of these steps resolved the issue.
When I first connect to the previously specified ip-address from the Android web browser, it reports that the connection is not secure and establish the connection after my confirmation. Subsequent connections to this ip-address were established without any prompts.
I also tested this https conntection using openssl s_client, as recommended. It returned "Verify return code: 18 (self-signed certificate)".
[This is result file](/api/attachments/7e00b46b-5f51-48e2-8ec1-ddfea6275190?platform=QnA)
I know that simplest solution of my problem is to change HttpsURLConnection.DefaultHostnameVerifier to my custom verifier that returns true for my IP address. But I want to understand why the SSL configuration isn't working correctly.
How do I resolve this issue?
[My solution repo](https://github.com/pnfstas/SmsGateway)
[My original post on StackOverflow](https://stackoverflow.com/questions/79797621/how-to-resolve-sslpeerunverifiedexception-hostname-not-verified-in-httpsurlconn)