SETTING UP ID-CARD SUPPORT FOR IIS WEB SERVER

Document information	
Time of creation	21.01.2019
Contracting authority	RIA
Author	Urmas Vanem, OctoX
Version	25.03/1

Version information		
Date	**Version**	**Changes/Notes**
21.01.2019	19.01/1	Public version, based on 18.12 software.
12.02.2019	19.02/1	Added OCSP configuration options. Edited by: Urmas Vanem
01.10.2019	19.10/1	Added information about the status and future availability of Windows Server (IIS) fixes by version. See the last paragraph of the introduction. Edited by: Urmas Vanem
18.10.2019	19.10/2	The described update for Windows Server 2016 KB4516061 that addresses an issue with Chrome-IIS. Edited by: Urmas Vanem
08.11.2019	19.11/1	The described update for Windows Server 2019 KB4520062 that resolves the Chrome-IIS issue. Edited by: Urmas Vanem
14.11.2019	19.11/2	The Windows Server 1903 (SAC) update described KB4524570 that resolves an issue with Chrome-IIS. Edited by: Urmas Vanem
12.12.2019	19.12/1	Added recommendations for securing IIS. Edited by: Urmas Vanem
14.12.2020	20.12/1	Added security settings to restrict access to undesirable CAs. Edited by: Urmas Vanem

17.12.2020	20.12/2	Added some security recommendations to the chapter "Restricting access to unnecessary CAs". Edited by: Urmas Vanem
03.03.2021	21.03/1	Removed an outdated IIS and Google Chrome authentication issue and clarified the information. See the last paragraph of the introduction. Edited by: Kristjan Vaikla
30.04.2021	21.04/1	Removed support for expired ESTEID-SK 2011 certificates. Edited by: Urmas Vanem
14.12.2021	21.12/1	Changed the Windows platform to Server 2022, added a third-party certificate request procedure based on the ECDSA algorithm, improved TLS and *Cipher* recommendations. Edited by: Urmas Vanem
18.01.2022	22.01/1	Added information related to Windows Server 2022 and TLS 1.3 protocol, incl. Procedure for configuring *the in-handshake* authentication method to enable authentication with a certificate using the TLS 1.3 protocol. Edited by: Urmas Vanem
18.12.2023	23.12/1	Removed ESTEID-SK 2015 chain. Edited by: Urmas Vanem
28.02.2025	25.02/1	Thales test card attached. Edited by: Urmas Vanem

Introduction

This manual describes the configuration of the IIS web server for the use of two-way SSL, where the user's certificate is a certificate issued to the Estonian ID-card (hereinafter: the term "Estonian ID-card" refers to a citizen's ID-card, residence permit card, digi-ID and e-resident's digital ID). As a new thing, we have added information about the configuration of the Thales test card to this manual.

Windows Server 2022 and Windows 11 operating systems have been used to create this guide. The sample guide supports certificates from SK ID Solutions AS (hereinafter SK) "EE-GovCA2018" and Zetes "Test EEGovCA2025" chains. To ensure the recognition of certificates, the client computer must have the necessary software installed:

- ID-software for Idemia cards (we recommend using the latest version of ID software, which can be found on the id.ee website).
- ID-software for Thales test cards is available here.

The server certificate of the sample guide has been issued from the OctoX test environment.

When using IIS, you can apply different authentication methods. This document looks at the introduction of a certificate requirement for IIS anonymous authentication – i.e. after the certificate validity has been checked, the user is allowed to access the website with pre-defined user (IUSR) rights.

At the moment, the tests have been successfully carried out with the following web browsers (latest versions):

1) Microsoft Edge
2) Mozilla Firefox
3) Google Chrome

Configuring One-Way SSL/TLS

Windows Server Certificate Configuration

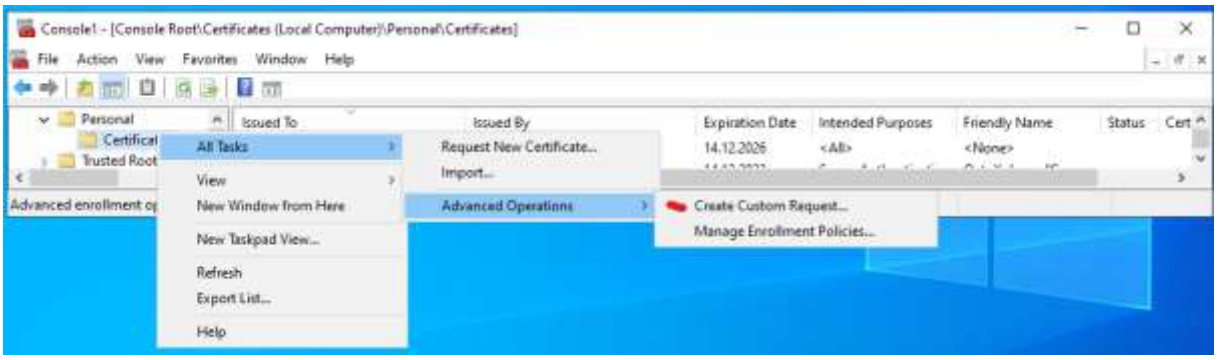
In order to provide a secure web service, the IIS server must be assigned a TLS certificate - in this example, a certificate issued from the OctoX test environment is used. Both client and the web server must trust this certificate.

If you have a domain environment and *an enterprise certificate* authority (CA), it is also reasonable to ask for a server certificate from the domain CA. However, if there is no such option or if you need a certificate that is more widely trusted, you need to create a private key and a request file (CSR) for the certificate and have a certificate created by a well-known CA based on the latter.

Obtaining a server certificate

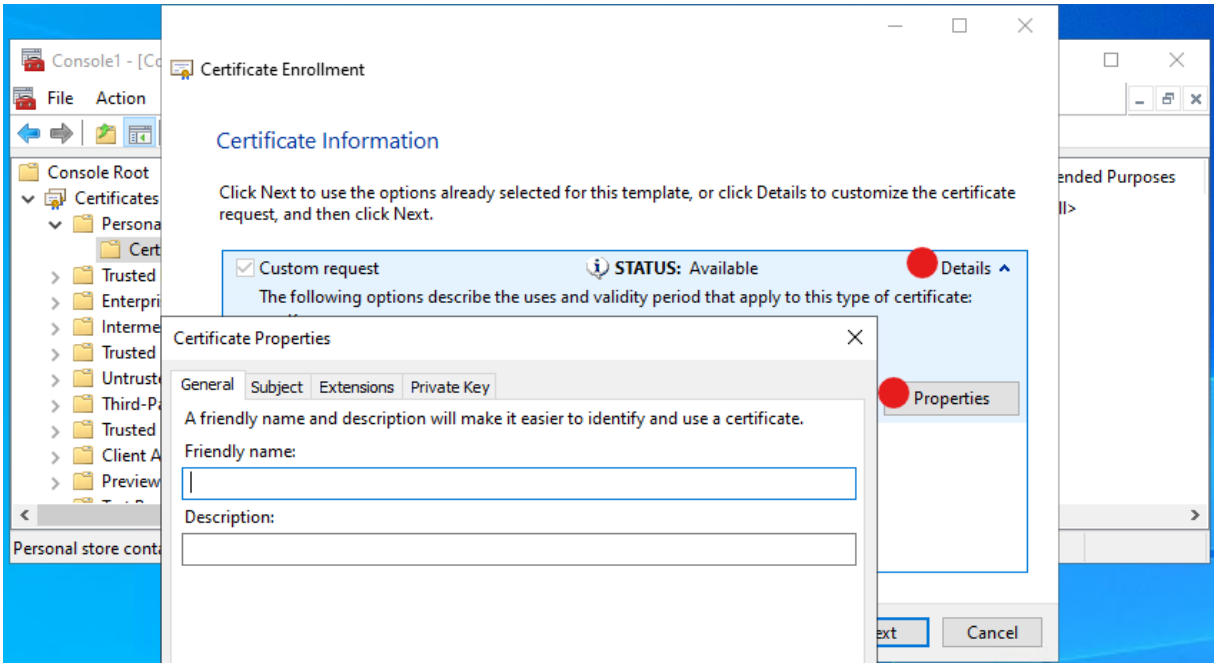
Because the certificate request file that is created from the IIS Management Console is quite limited, you should use the Certificate Management Console to create a server certificate request file instead. (I'll note here again to clarify that it makes more sense to use the instructions below to create an IIS server that uses a certificate issued by a public CA. If our CA solution certificate is enough for us, we can use a locally issued certificate with *Server Authentication written in the EKU*.)

1. Run mmc.exe on the IIS server and add the local computer certificates view to it. Create a custom query:



Picture 1 – Create a custom query

2. Click Next three times , *open Details, Properties.* The certificate request properties window opens:



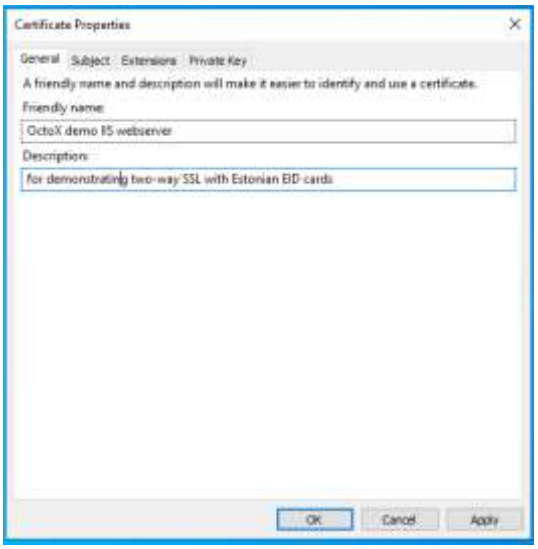
Picture 2 – certificate inquiry properties window

3. Next, you can specify the exact properties of the query file that you want to see later in the web server certificate.

If you need to create similar query files more often, it is recommended to familiarize yourself with *the PowerShell* options to automate the activity.

Tab General

Here, if desired, you can specify the call for the certificate and a brief description. These fields are not the substantive parts of the certificate and these explanations are relevant for later easier understanding.



Picture 3 - general information about the certificate

Tab Subject

In the *Subject* window, you can describe the subject as usual. If you want to use different SAN DNS names or use *something other than FQDN in the case of a common name, then one or more DNS aliases must also be described here.*

Picture 4 - Sample Subject Configuration

Tab Extensions

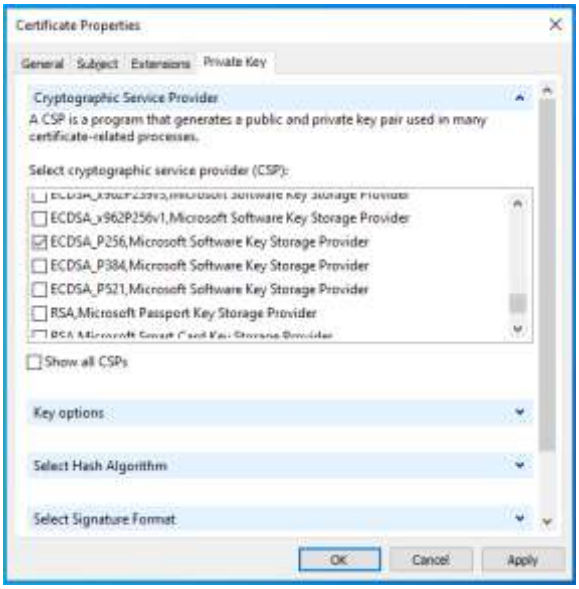
In the *Extensions window*, you can set the following properties:

1. *Key Usage*:
    a. *Digital signature*;
    b. *Key encipherment*.
2. *Extended Key Usage*:
    a. *Server Authentication*.

Picture 5 - Assigning extensions

Tab Private Key

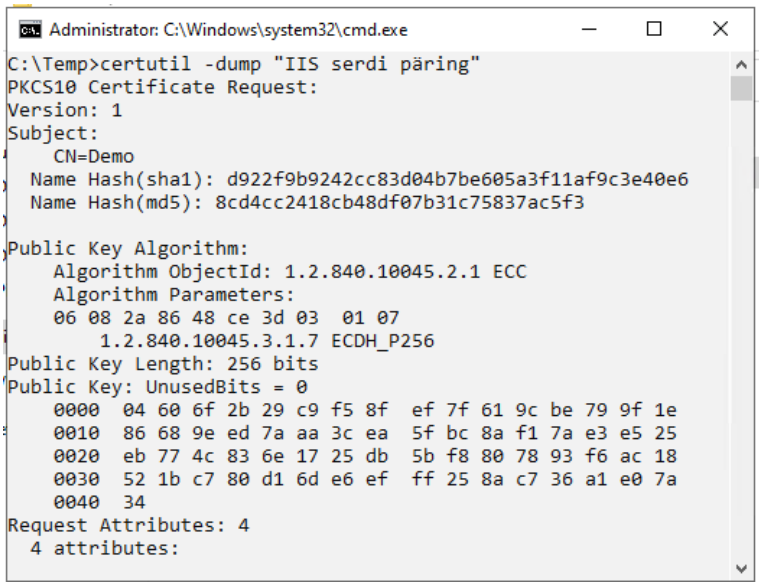
Here you need to select the CSP or certificate key algorithm. The example configuration uses an algorithm ECDSA_P256, so you need to select ECDSA_P256 from the list and remove the RSA at the beginning of the list.



Picture 6 - Choosing a CSP

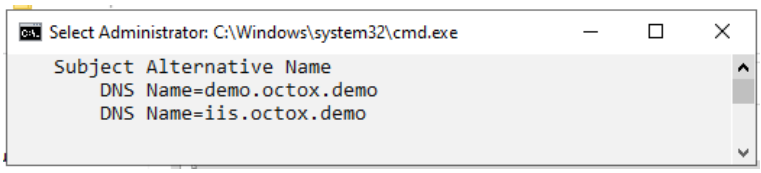
Click *OK* and *Next*, specify the folder and name, and save the certificate request file in Base64 format.

The properties of the newly created certificate request file can be checked with the command "certutil -dump Name_Of_CRS".



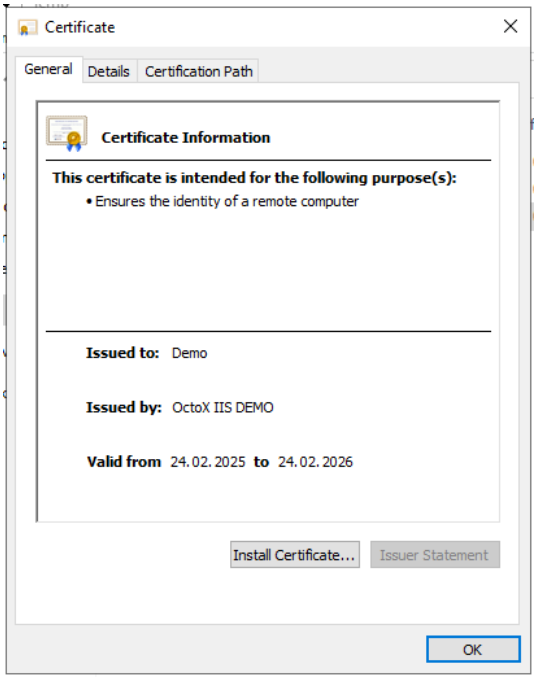
Picture 7 - the content of the query file

Make sure that the DNS alternative names are also present in the query file:



Picture 8 - DNS aliases in the query file

Now you need to forward the certificate request file to a CA server and ask for the certificate to be generated based on it. The result is as follows:

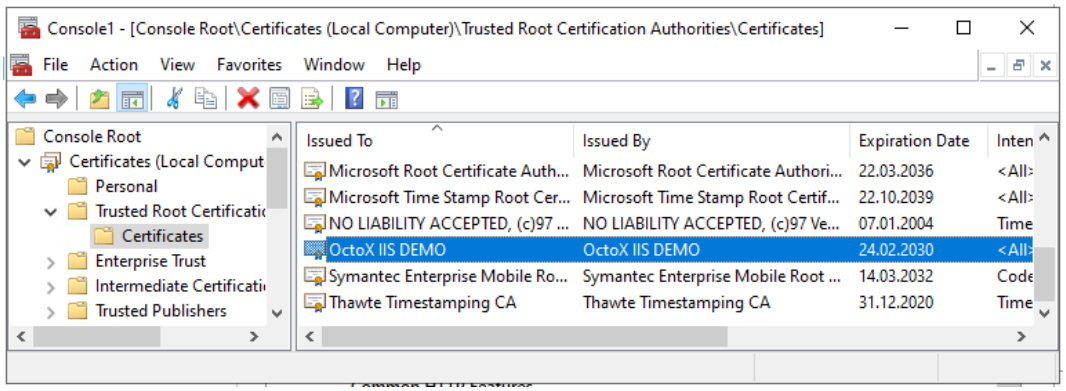


Picture 9 – IIS server certificate

Certificate Installation

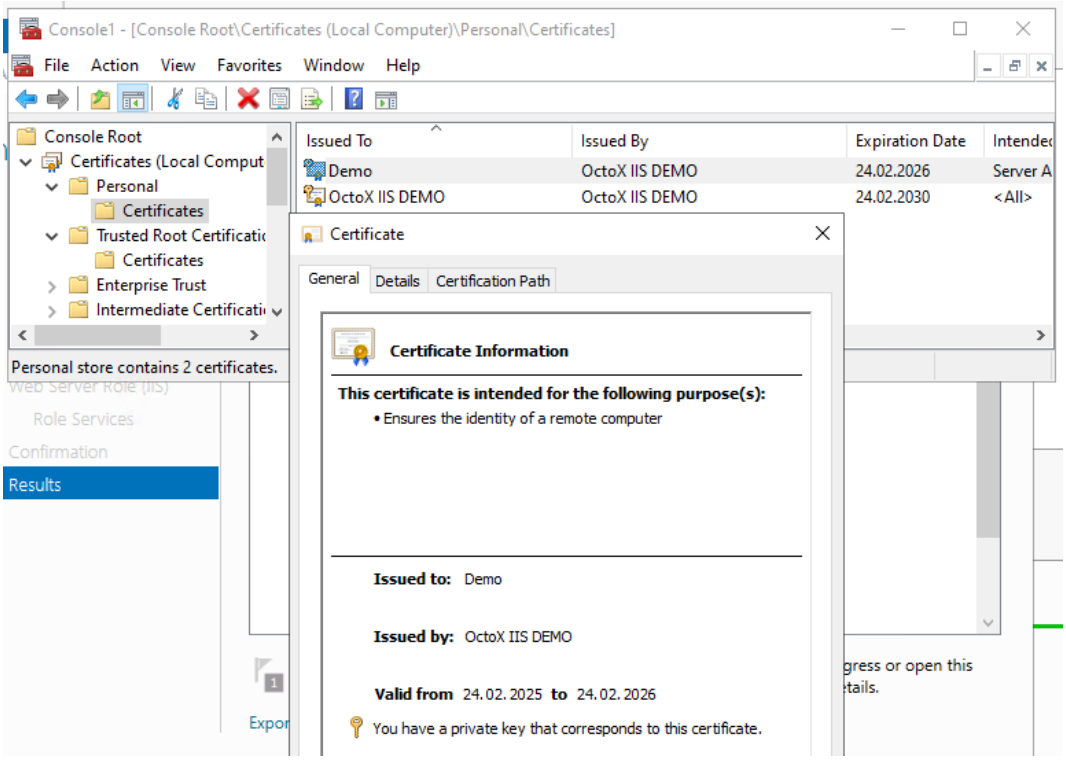
The IIS server must trust the "OctoX IIS Demo" certificate, which is the issuer of the IIS service certificate for the server. To do this, it is necessary to check the presence of this certificate in the container of trusted root certificates. If the issuer's CA certificate is not there, it must be attached.[1]

[1] If the certificate has been issued by a mid-level CA, it must be added to the container of the intermediate CAs. And the root CA certificate that issued the intermediate CA certificate must be added to the container of trusted root certificates in the absence of it.



Picture 10 - IIS Server Trust CAs that issued the certificate to him

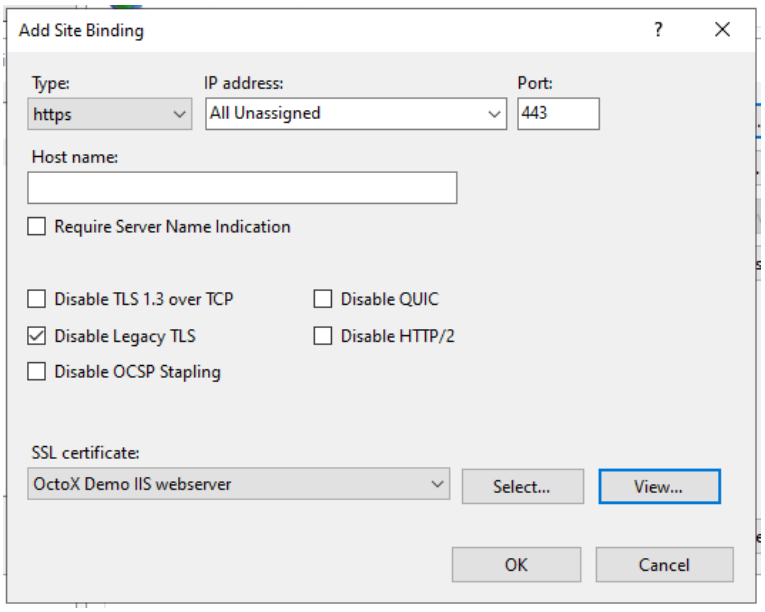
The IIS server certificate must be installed in the personal container of the local computer on the IIS server:



Picture 11 - when opening the certificate, it can be seen that the IIS server can also use its private key as expected

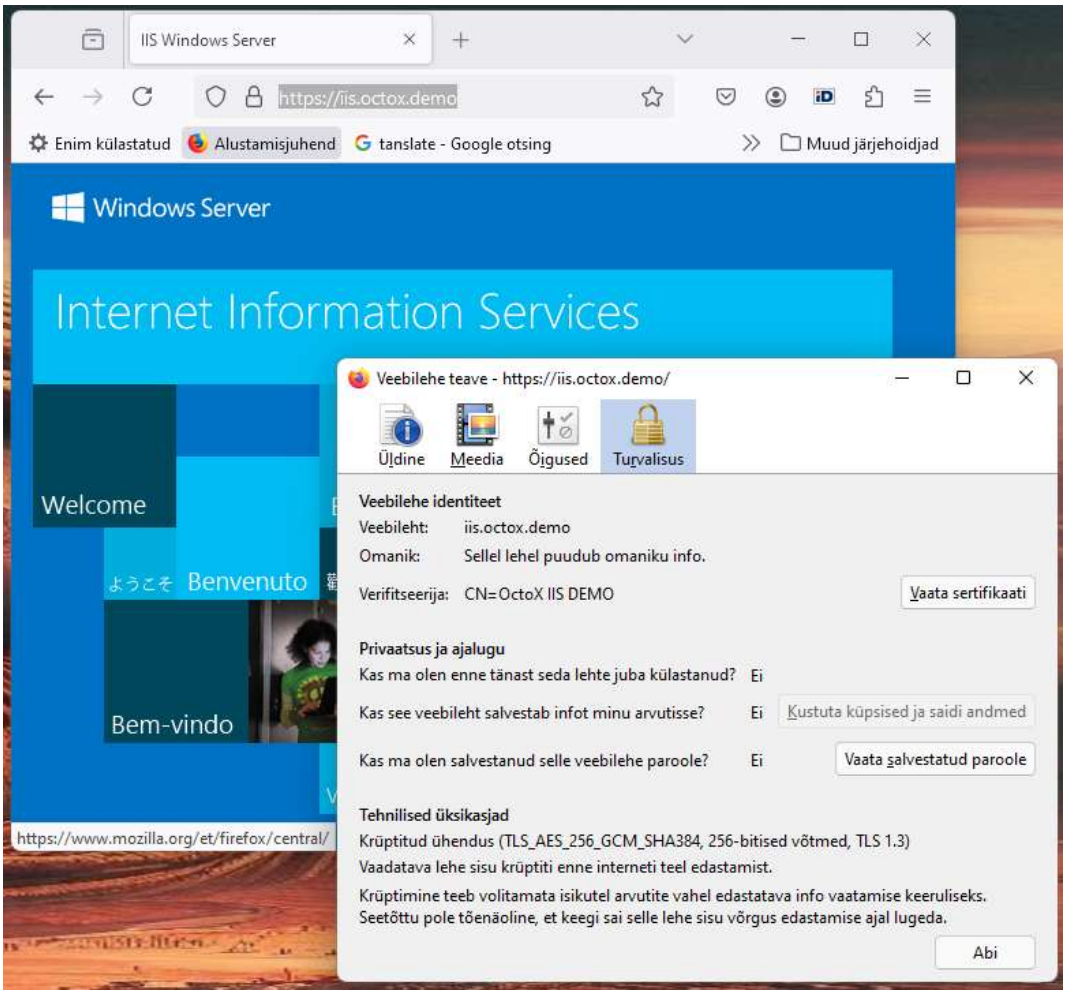
Create one-way SSL configuration

In order to establish one-sided SSL, the website must have a described SSL port (default 443) and it must be linked to the desired certificate. The use of old SSL/TLS protocols (older than 1.2) must also be disabled immediately.



Picture 12 - port 443 is allowed on the website and the certificate used is "OctoX IIS Demo", old TLS protocols must be disabled

After confirming the settings, one-sided SSL works.



Picture 13 - one-sided SSL works with TLS 1.3 protocol, the web browser is Firefox

The Firefox web browser used to demonstrate one-sided SSL also shows the following in the additional information windows:

1. The freshly installed certificate iis.octox.demo is used;
2. The TLS 1.3 protocol is used.

Restricting HTTP access

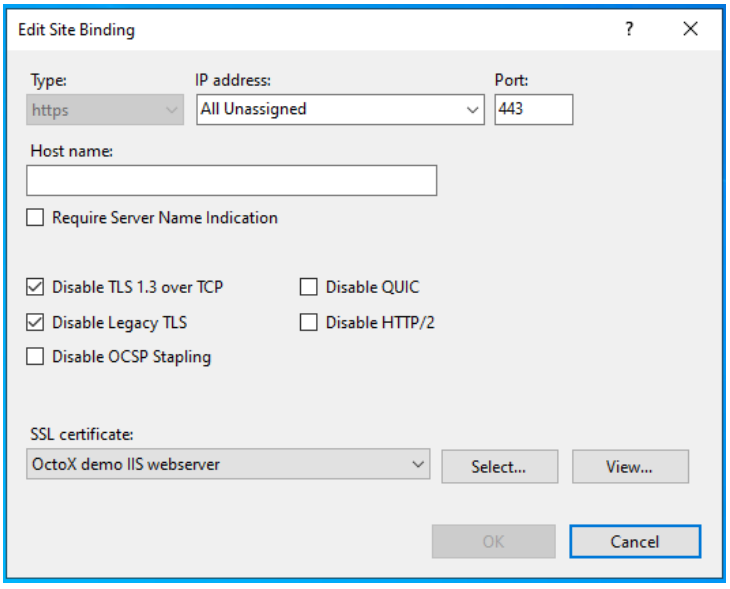
To disable HTTP access, port 80 must be removed from the list of associated protocols and the corresponding access must also be denied from the firewall. Alternatively, HTTP traffic can be automatically redirected to the HTTPS website to avoid the problem where users type the website address into the browser themselves but do not add HTTPS:// specification to it.

Configuring Two-Way SSL, or Requiring Certificate Authentication

Pre-setup

Please note that IIS 10/Schannel (as of 18.01.2022, which was adapted for Windows Server 2022) uses the protocol for authentication with a certificate using TLS 1.3 by default *Post-handshake* authentication method. Since it is not supported by the most common web browsers[2], this solution does not work in practice. If TLS 1.3 is turned on, the server does not send a certificate request to the user by default and disconnects the connection. In order for certificate authentication to work, the use of TLS 1.3 must be disabled. Alternatively, you can turn on *in-handshake* authentication method, see. Chapter "*Enable* the in-handshake authentication method".

TLS protocol version 1.3 can be disabled from the IIS HTTPS connection page by checking the box "Disable TLS 1.3 over TCP":



Picture 14 – To enable authentication with a certificate, the TLS 1.3 protocol must be disabled

Setting up Estonian eID certificates on an IIS server

To enable two-way SSL, authentication with a certificate must be required by the IIS server. By default, the server allows the use of all certificates that are trusted by the server and have the *client*

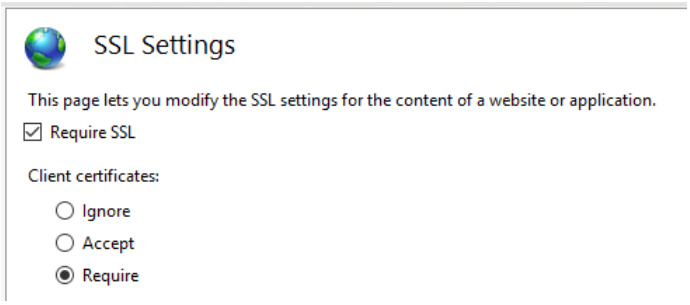
[2] As far as is known, the Firefox web browser does support it, but it is not enabled by default on this web browser either.

authentication extension *described in the EKU.* In order to function correctly, the server must be able to create the entire certificate chain from the user certificate to the root certificate. This means that in addition to the existence of root level certificates, the IIS server also needs *to have intermediate* certificates.

For the example configuration, the certificates must be published on the IIS server as follows:

1) Idemia cards chain certificates:
    a. To the container of trusted root certificates: EE-GovCA2018
    b. Intermediate Certificate Container[3]: ESTEID2018
2) Thales cards chain certificates:
    a. To the container of trusted root certificates: Test EE-GovCA2025
    b. Intermediate Certificate Container: Test ESTEID2025

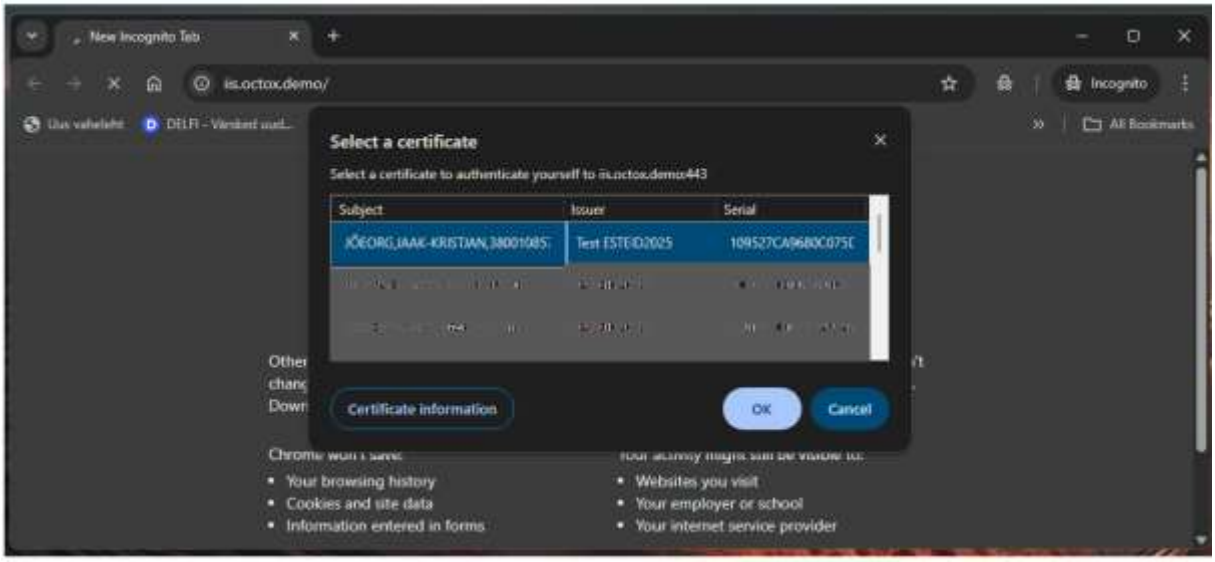
Under the SSL properties of the website, the use of the SSL protocol and user certificates must be required:



Picture 15 - SSL and certificate requirement

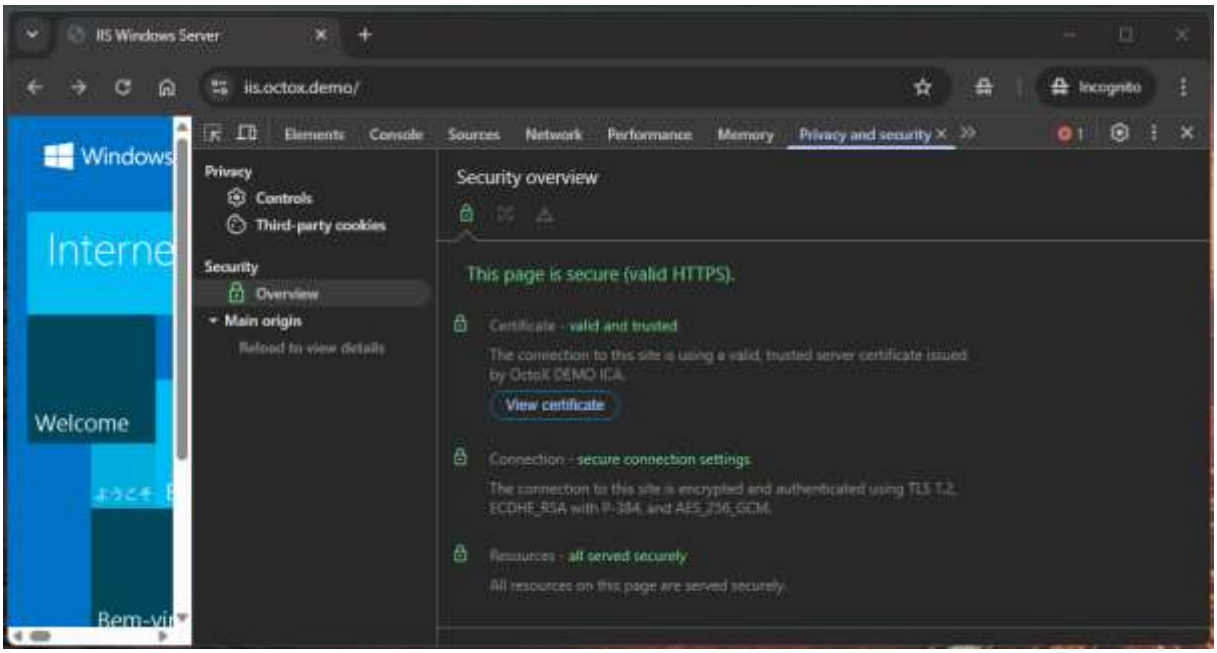
The created configuration allows access to the website via port 443, where the user is required to provide a certificate. When turning to the website, you are allowed to choose the desired certificate accepted by the server:

[3] In the case of the use of the organisation cards issued by SK, the EID-SK 2016 (https://www.sk.ee/upload/files/EID-SK_2016.der.crt) certificates must also be set up among the intermediate level certificateshttps://www.sk.ee/upload/files/EID-SK_2016.der.crt.



Picture 16 - Requesting a certificate in the Firefox browser when accessing a website

After entering the PIN-code, the status of the certificate is checked on the web server, and if everything is in order, the user is allowed to access the website.



Picture 17 - authentication was successful using the TLS 1.2 protocol

Alternatively, instead of requiring a certificate from IIS, you can also use a simple *Accept certificate* by the IIS server – this allows you to access the server with a username and password in addition to the certificate, or without authentication at all.

Enable the in-handshake authentication method

If you want to use the TLS 1.3 protocol and use authentication with a certificate, you can use the *-in-handshake* authentication method. In this method, the server also prompts the user *for a certificate immediately when sending* Server Hello.

To enable the In-handshake authentication method, you need to do the following:

1) Document the configuration of an existing certificate with the command "netsh http show sslcert". It is important to note down *the Certificate Hash* and *Application ID*:

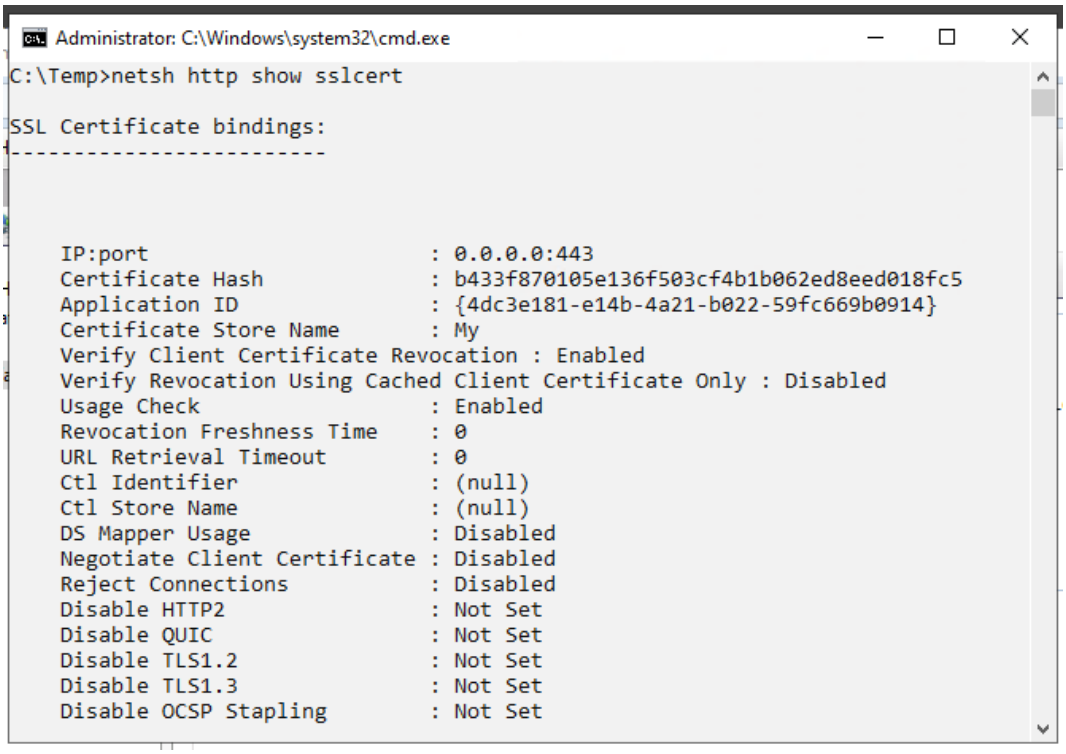


Figure 18 - By default, the "negotiate client certificate" setting is disabled

2) Remove certificate association with port 443 with the command "netsh http del sslcert 0.0.0.0:443":

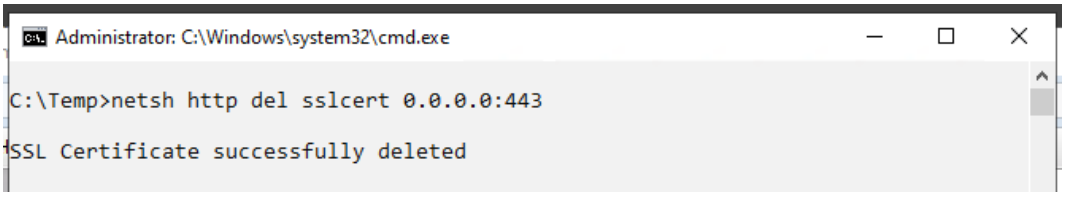


Image 19 - Removing the certificate from port 443

3) Re-enable the certificate while also enabling *the in-handshake* authentication method with the command "netsh http add sslcert ipport=0.0.0.0:443

certhash=b433f870105e136f503cf4b1b062ed8eed018fc5 appid={4dc3e181-e14b-4a21-b022-59fc669b0914} certstorename=MY **clientcertnegotiation=Enable**":

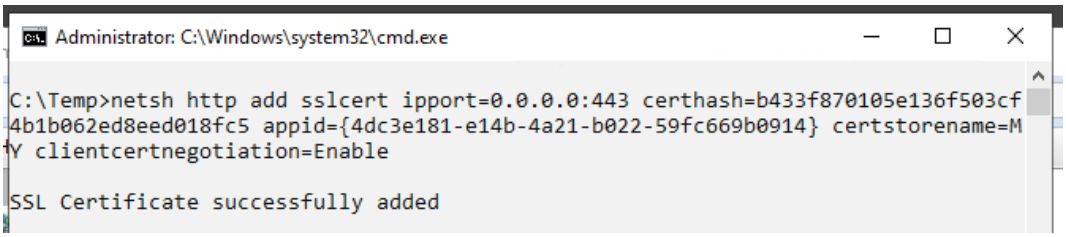


Figure 20 - Enabling clientcertnegotiation

Looking at the certificate information again, you can see that *the Negotiate Client Certificate* is now enabled:

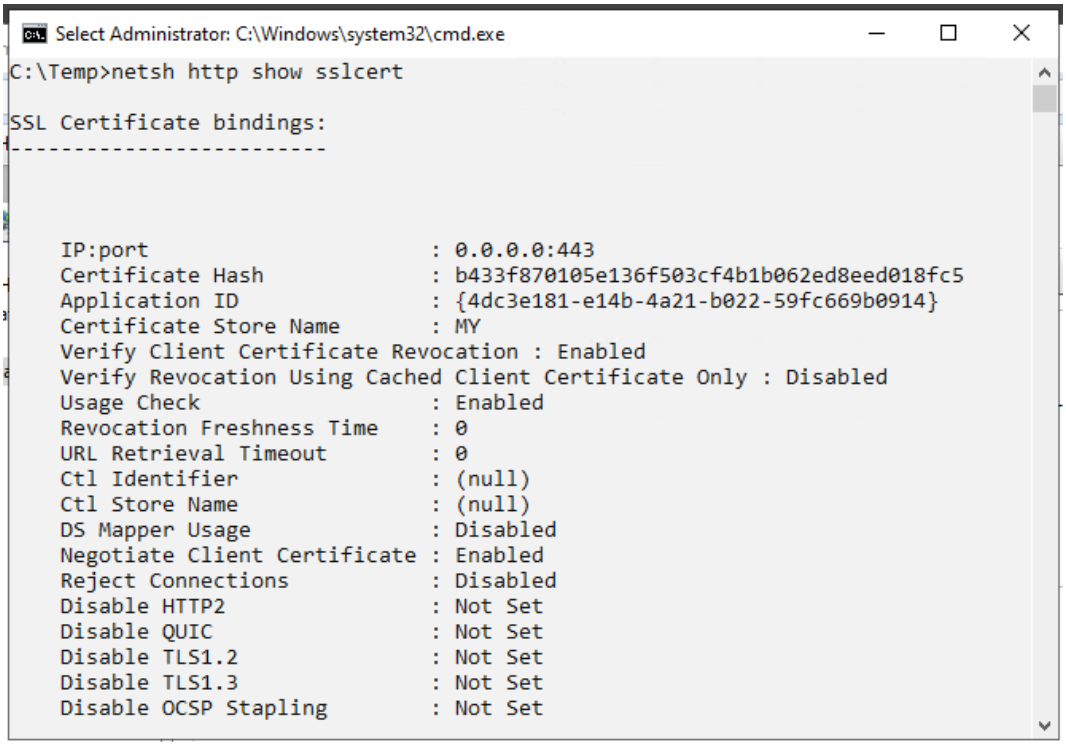


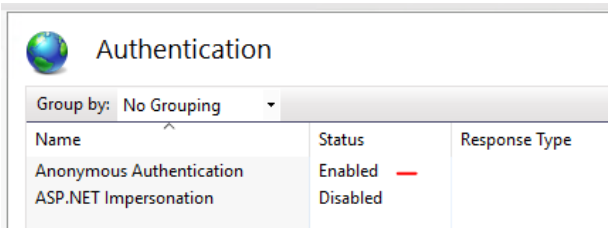
Figure 21 – In-handshake authentication method is now turned on

Note:

Since *session renegotiation* is prohibited in TLS 1.3, it must be taken into account that authentication must take place on the "first page". If an unauthenticated one-sided SSL connection has been established with the user's certificate and you want to access a protected resource on the same page by authenticating with the user's certificate, it will fail because TLS 1.3 does not support such an approach. If necessary, this problem of "landing" will have to be solved in one way or another.

Authentication

In this example, only anonymous authentication is allowed:



Picture 22 - anonymous authentication, the user can access the website with user rights (IUSR)

Possible additional configurations

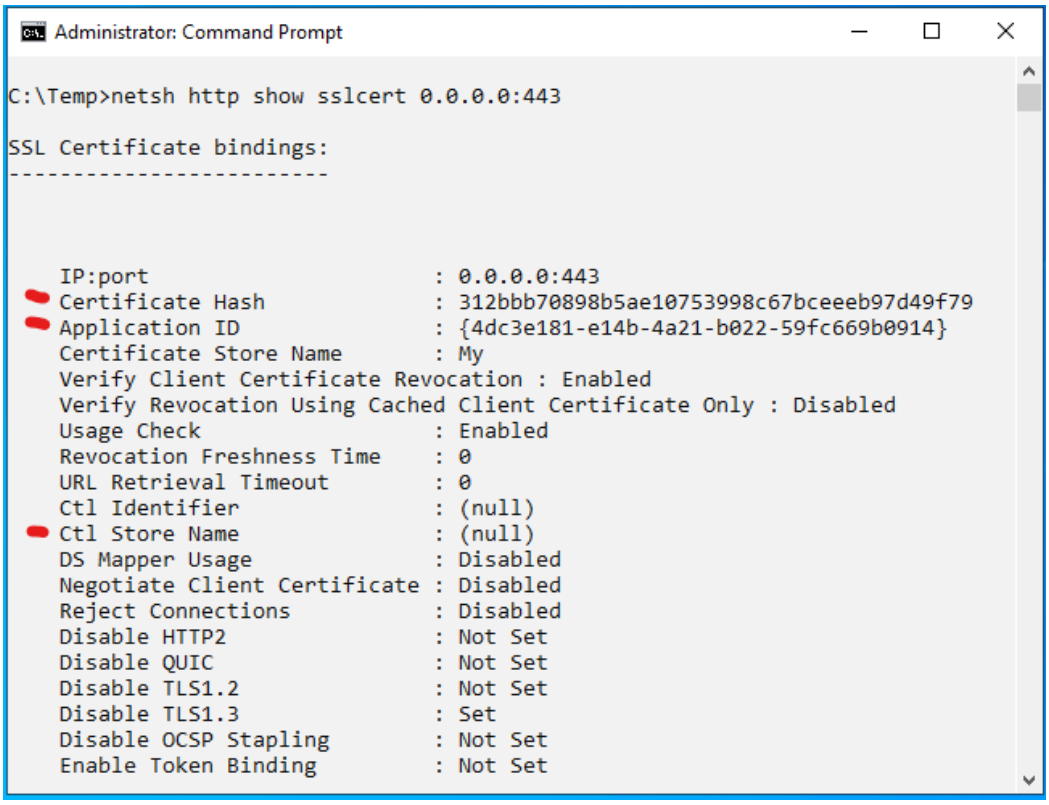
The purpose of this document is not to provide precise instructions for optimal configuration or security of websites, but to introduce the configuration for the use of bilateral SSL with Estonian ID cards. However, it is important to consider the following.

Filter the certificates that are displayed to the user

The default configuration does not limit the choice of certificates displayed to users, which means that when authenticating to a web server, the user is shown all the certificates available to the user that have user authentication written under the properties of the ECU. However, IIS can provide the user with a list of allowed authentication centers and thus display only the certificates of supported chains to the user.

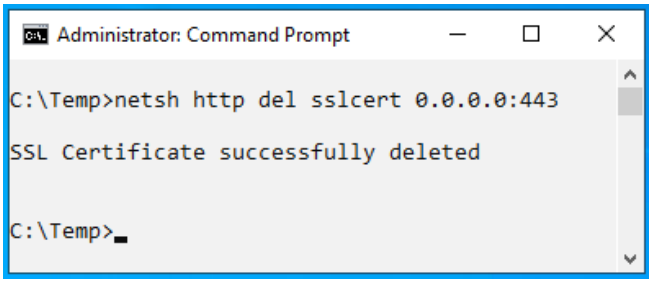
If the goal is to show the user only certificates that come from the chains of specific root servers, "EE-GovCA2018" or "Test EE-GovCA2025", then the following steps must be taken:

1) Display the active IIS certificate information with the command "netsh http show sslcert 0.0.0.0:443":



Picture 23 - Default associated certificate properties

2) Unlink this certificate with the command "netsh http del sslcert 0.0.0.0:443":



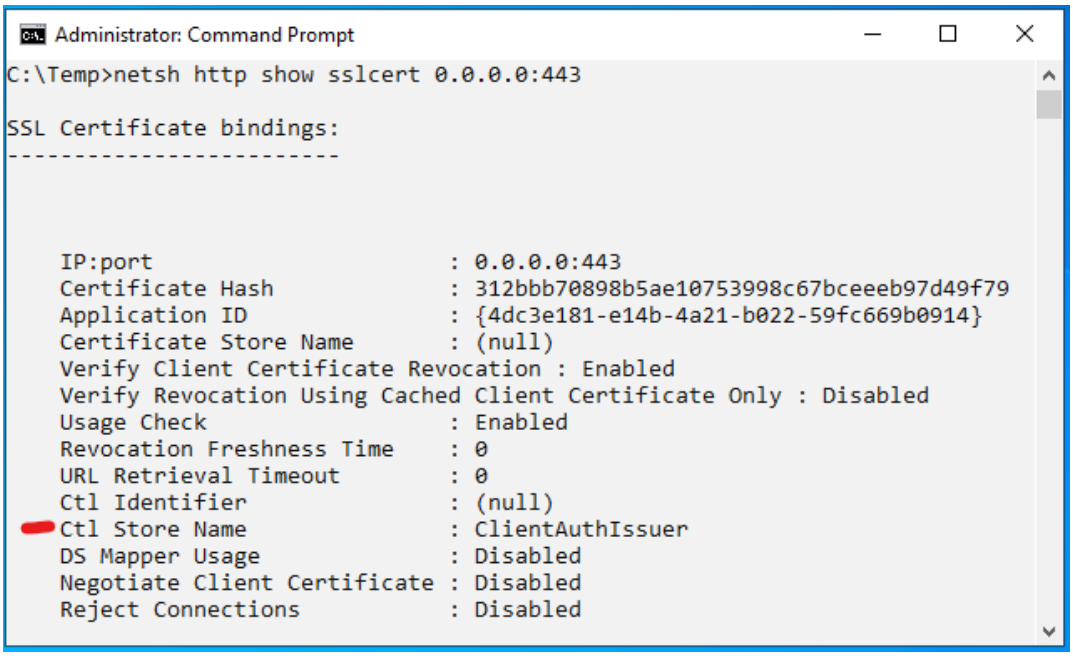
Picture 24 - Certificate removal

3) Add the certificate again and set the "Client Authentication Issuers" *folder to filter the certificates*. The command is "netsh http add sslcert ipport=0.0.0.0:443 certhash=1e75c77c696aa4d49686bb1ef73ac3b07fdff38a appid={4dc3e181-e14b-4a21-b022-59fc669b0914} sslctlstorename=ClientAuthIssuer":

Picture 25 - Adding a certificate with new features

Certhash and *Aid* The values can be obtained from the initial certificate statement, see. „Picture 23 - Default associated certificate properties“.

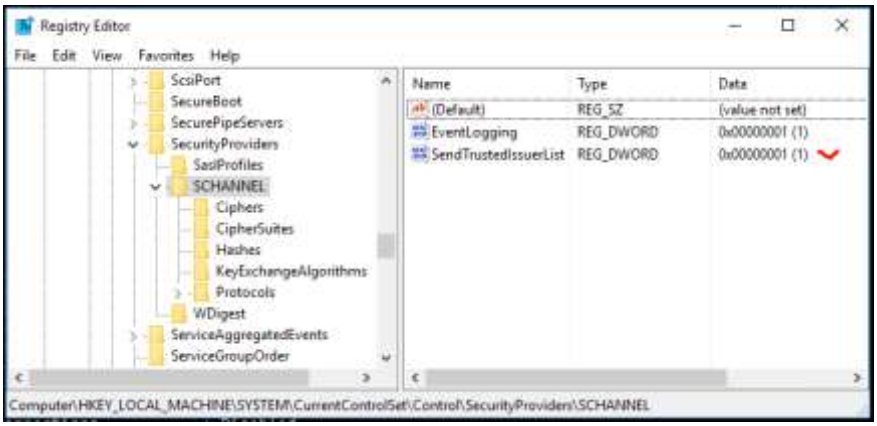
4) Check that the "*CTL Store Name*" is on the new certificate statement *ClientAuthIssuer*:



Picture 26 - Features of the re-linked certificate

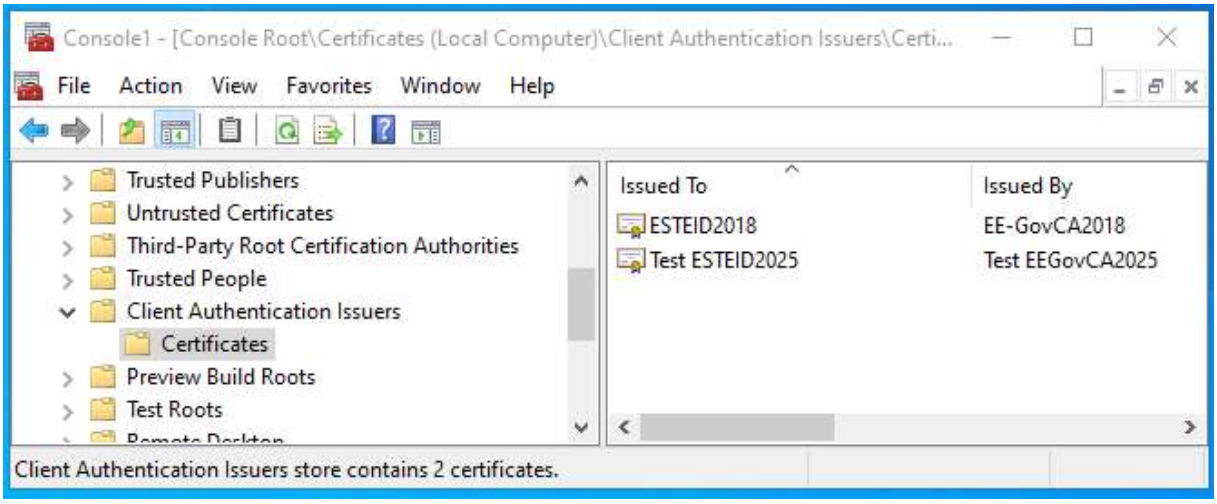
If you wish, you can also see from the IIS configuration that the SSL certificate is correctly bound to port 443 again.

5) Enable filtering of certificates from the IIS server registry by adding the setting "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\SendTrustedIssuerList=1":



Picture 27 - enable certificate filtering in the registry

6) Add intermediate certificates to the IIS Server Certificate Container "*Client Authentication Issuers":*



Picture 28 – in this example, both Idemia and Thales chain certificates have been added

7) If necessary, restart the IIS service or server and check the operation of the desired solution.

User certificate status check against OCSP service

With the help of the OCSP (*Online Certificate Status Protocol*) service, the user's certificate status can be checked in real time. When each user is authenticated, the web server sends a request to the OCSP service, which returns the certificate status information.

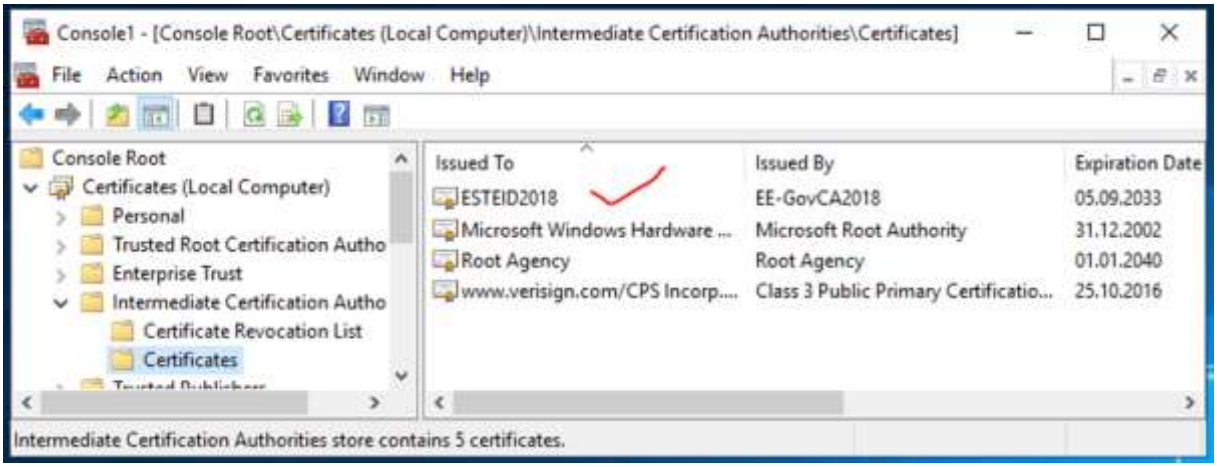
Guaranteed (paid) OCSP service

In order to check the status of the certificate using the guaranteed OCSP service provided by SK, a contract must first be entered into with SK. After that, the subscriber is allowed access to the OCSP

service (with an address http://ocsp.sk.ee) either on the basis of a certificate or an IP address. **This functionality can only be used with the Idemia cards**.

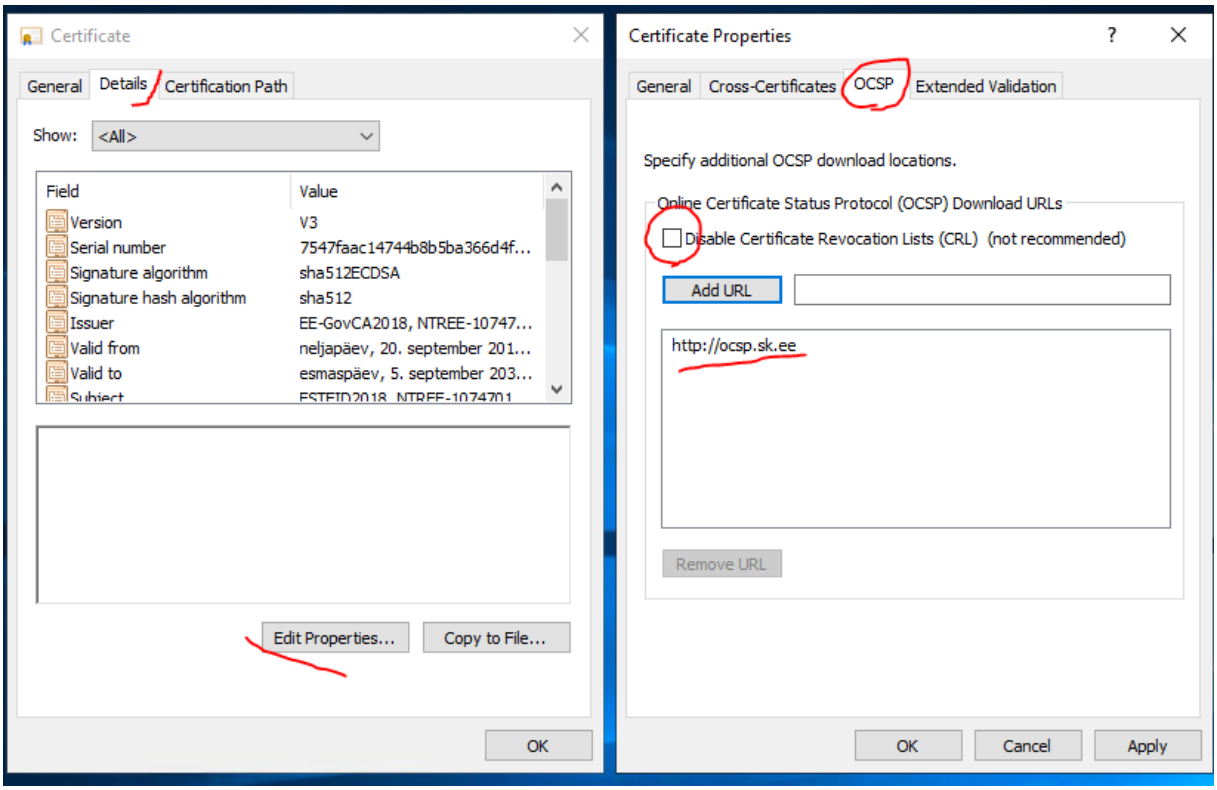
Configuration

In order to use the guaranteed OCSP service on a web server, the intermediate level certificates in the server's operating system must be modified to check the status of the clients' certificates. On the web server, the certificates of the Estonian eID mid-level CAs are published in the container "Intermediate Certification Centres".



Picture 29 - Certificate placement on the IIS server

To change the OCSP properties of certificates, open the certificate, select the *Details* page, click on the "*Edit Properties" button...*" and select the OCSP page. The OCSP service address http://ocsp.sk.ee must be added to the list of OCSP URLs.

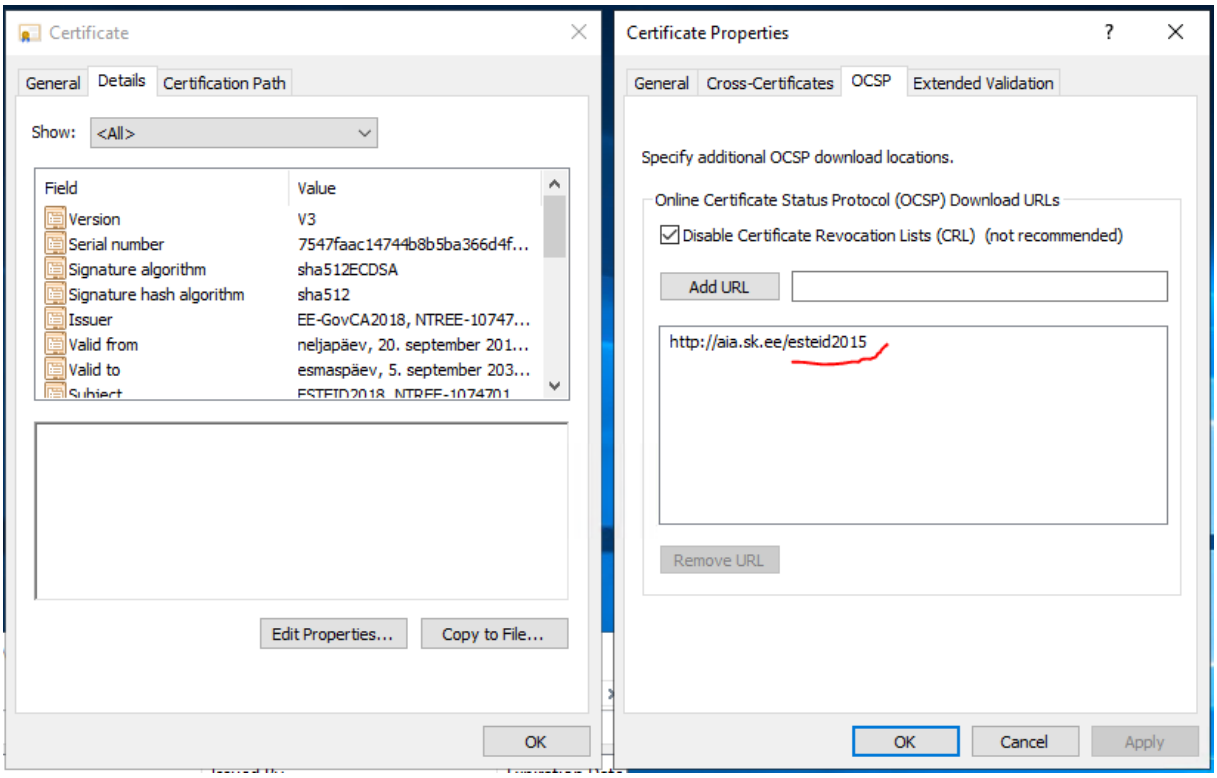


Picture 30 - Determining the location of the OCSP service based on an intermediate level certificate

In the image above, the status check of the user's certificates is directed to the guaranteed OCSP service address (http://ocsp.sk.ee) in the case of certificates issued under ESTEID2018 CA http://ocsp.sk.ee/. If you want to completely disable CRL verification on ID cards (ESTEID2018 cards issued from the CA chain, the CRL address is no longer described in the certificate, so there is no need for it in this configuration), it can be activated with the checkmark "*Disable Certificate Revocation Lists (CRL) (not recommended)*".

Free access (free) AIA OCSP service

In addition to the guaranteed (paid) OCSP service, trust service providers also offer free access AIA OCSP service, in which case the status of certificates is checked against a slightly simpler OCSP service. In the case of certificates issued under ESTEID2018 CA, the AIA OCSP address is already listed in the certificate (http://aia.sk.ee/esteid2018), so you don't need to configure anything separately here. However, if desired, AIA OCSP control can also be established centrally:

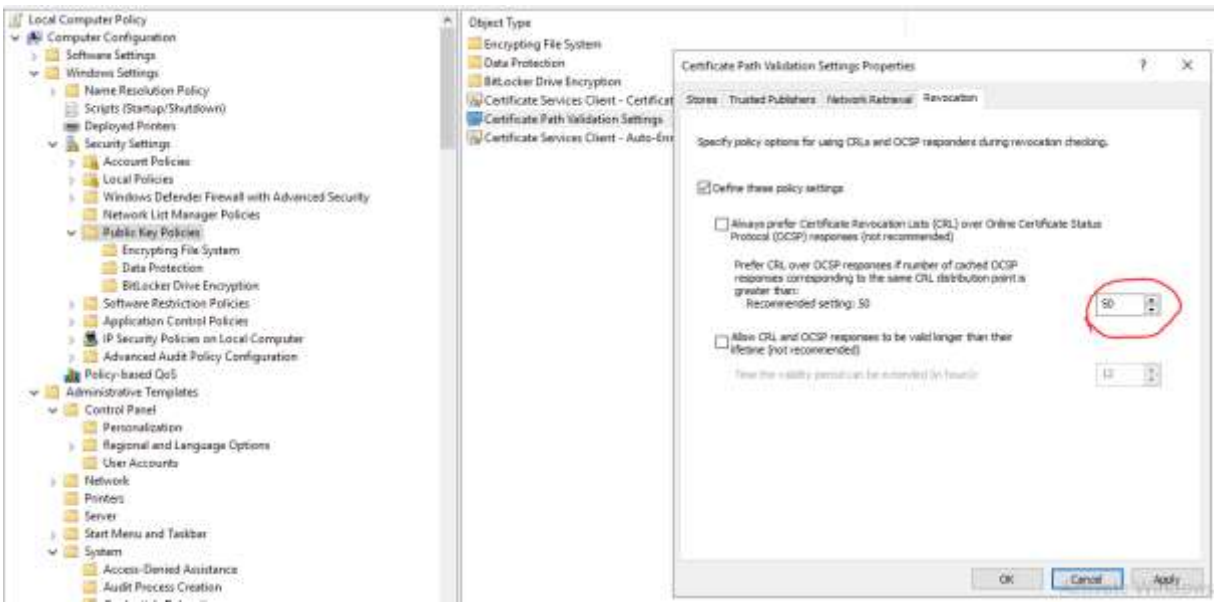


Picture 31 - Configuring the AIA OCSP address

The OCSP path is also described in the certificates of Thales cards, so we do not need to do any additional activities to use it. However, today there is also a CRL path in these certificates. We can abandon the use of CRL by using the central OCSP assignment described above, along with disabling CRL.

Comments:

- All ID-card certificates in use today use the AIA OCSP service with the address as a validity check, the CRL address is not described in them.
- In the case of Windows Server, by default, the OCSP-based certificate validity check is reverted to CRL-based verification if the number of OCSP requests in the cache exceeds the limit of 50. For this example configuration, this is not important as CRL is not used. For other configurations, this number can be changed by creating a registry value HKEY_LOCAL_MACHINE/Software/Policies/Microsoft/SystemCertificates/ChainEngine/Config /CryptnetCachedOcspSwitchToCrlCount and assigning a new value to it. See also OCSP *magic count* or *magic number*. Perhaps an easier way to change this property is *windows policy*.



Picture 32 - changing the magic OCSP number

Recommended IIS security settings

SSL/TLS

IIS version 10 on server 2022 uses all TLS protocol versions, 1.0-1.3[4], by default. Older SSL protocols are not used by default.

Nowadays, TLS versions 1.0 and 1.1 should definitely not be used anymore. For two-way authentication to work, TLS version 1.2 must be enabled and TLS version 1.3 temporarily disabled (read more on page 14: Requiring two-way SSL, certificate authentication - Preconfiguration). If authentication with a certificate is not important, only TLS version 1.3 should be allowed.

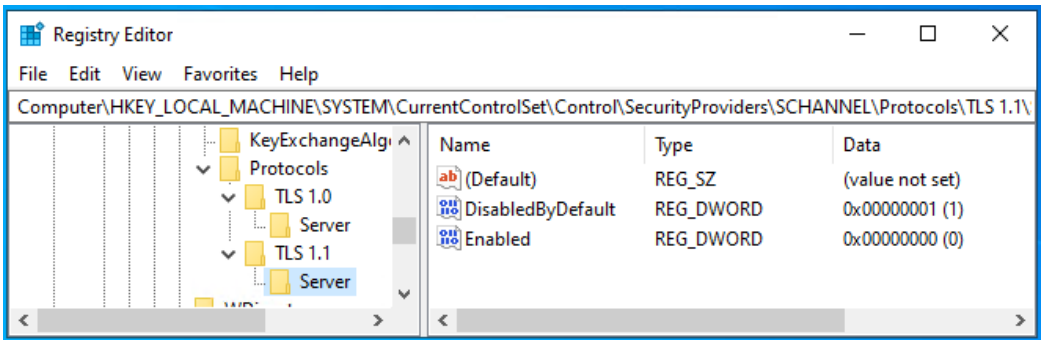
More information on the recommendations for the use of the TLS protocol can be found in the Cryptographic algorithms life cycle reports commissioned by RIA at https://www.id.ee/en/article/cryptographic-algorithms-life-cycle-reports-2/.

Disabling old TLS protocols can be done directly in the registry in addition to disabling them through the IIS configuration. If you want to disable TLS versions 1.0 and 1.1, you need to add the following configuration to the registry[5]:

[4] https://docs.microsoft.com/en-us/windows/win32/secauthn/protocols-in-tls-ssl--schannel-ssp-?redirectedfrom=MSDN.

[5] By default, these values do not exist.

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\[6]:
  - TLS 1.0\Server
    - Enabled DWORD:0
    - DisabledByDefault = DWORD:1
  - TLS 1.1\Server
    - Enabled DWORD:0
    - DisabledByDefault = DWORD:1



Picture 33 - TLS versions 1.0 and 1.1 Disabling in the registry

Of course, it is also possible to distribute the above registry configuration using central policies.

Cipher *suites*

Windows Server comes with a number of cipher suites by default. All of these can be viewed, for example, in PowerShell with the Get-TLSCipherSuite command[7].

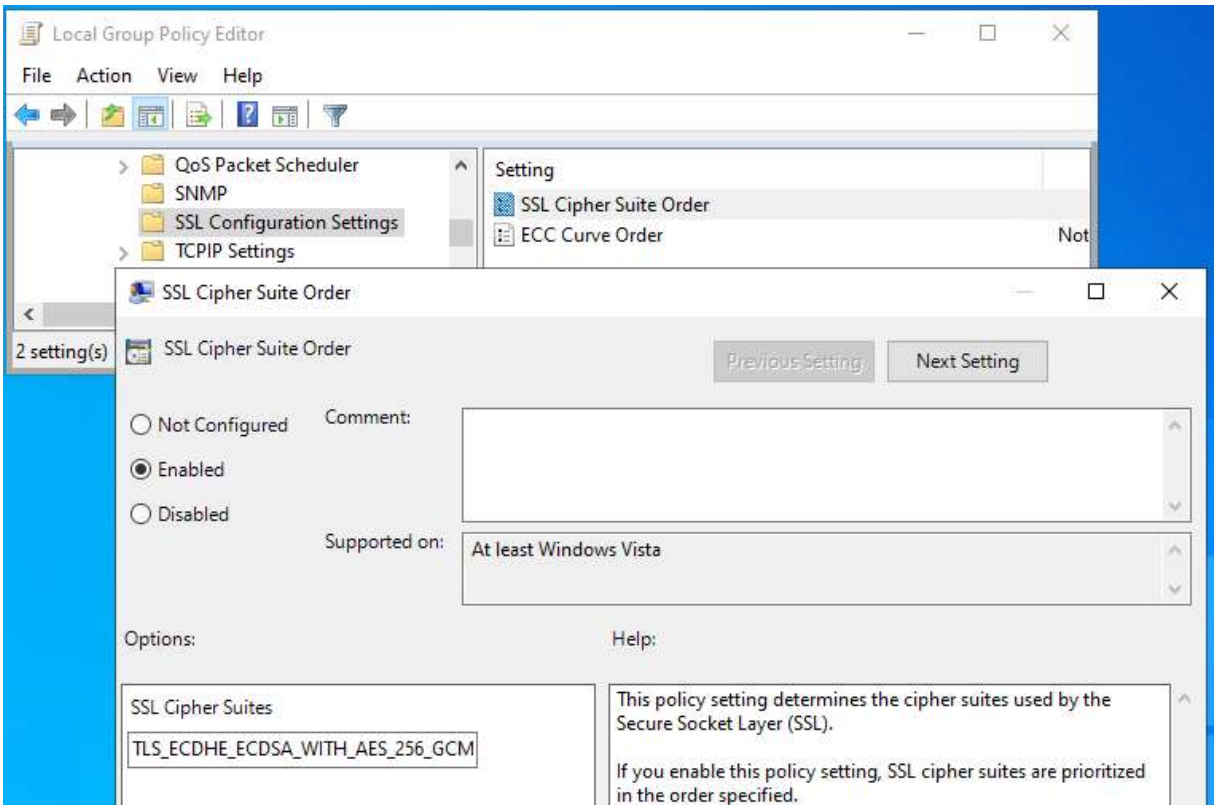
It is not possible to give a definite recommendation for the use of different cipher kits without knowing the terms and conditions presented on the website. However, it is important to remove unsafe cipher kits from the list. Before proceeding with the configuration, we strongly recommend that you familiarize yourself with the recommendations of the cryptographic algorithms life cycle reports commissioned by RIA at https://www.id.ee/en/article/cryptographic-algorithms-life-cycle-reports-2/. It is reasonable to describe the specific cipher suites that are allowed to use TLS 1.2.

Therefore, if you want to determine the cipher suites you can use yourself, it makes sense to use local or central policies to do so. To use only the ECDHE-ECDSA-AES256-GCM-SHA384 and ECDHE-RSA-AES256-GCM-SHA384 cipher sets, the "Computer Configuration/Administrative

[6] It is also possible to configure the user's share in the SSL/TLS protocols view. While this instruction is only about server-side setup, this does not mean that configuring the user part is not recommended, it always depends on the specific situation.

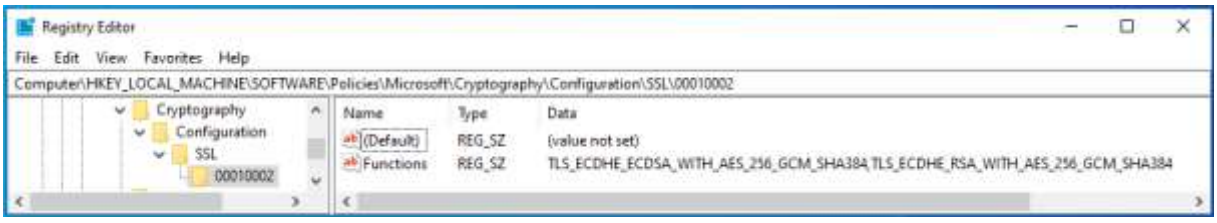
[7] https://docs.microsoft.com/en-us/windows/win32/secauthn/cipher-suites-in-schannel

Templates/Network/SSL Configuration Settings: SSL Cipher Suite Order" setting must be modified. Cipher kits must be separated by a comma.[8]



Picture 34 – Specifying specific cipher suites with a central policy

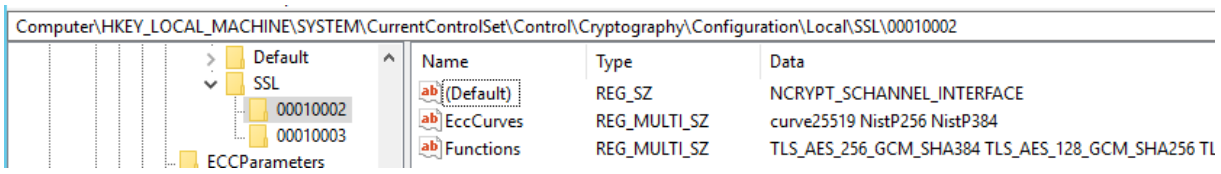
The configuration specified in the previous point is written to the registry:



Picture 35 - Configuration defined by the policy

By default, cipher kits are described in the location described in the following image:

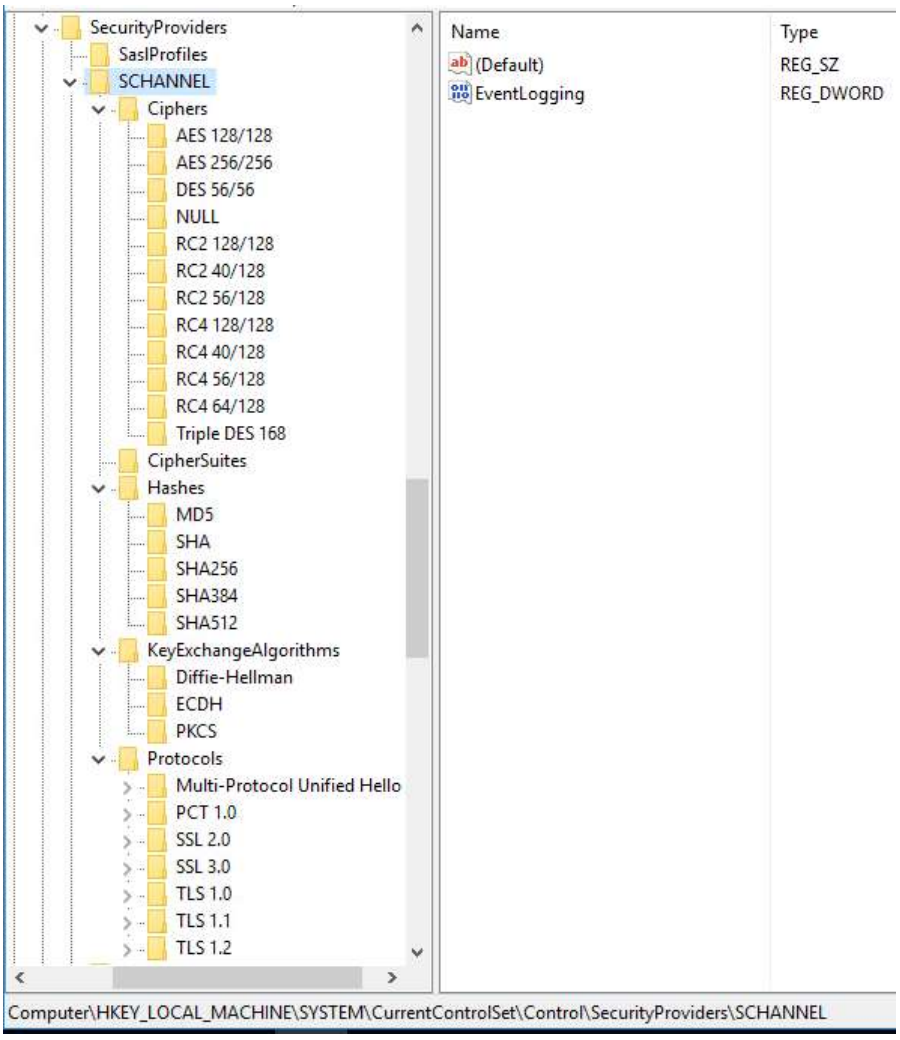
[8] NB! TLS 1.3 does not work with these specific configurations. Rather, the use of these specifications may make sense if you do not want to use TLS 1.3, for example, when enabling authentication with a certificate.



Picture 36 - Default cipher kit configuration

Other configurable Schannel features

The default location *for Schannel-configurable properties is in the registry:* HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL. Here it is possible to enable or disable different components and, if necessary, overwrite the default configuration settings.



Picture 37 - Configurable features of Schannel

Other options

In addition to setting up TLS and cipher suites, it is recommended to pay attention to the security of the IIS server in the following points:

- Keep the operating system up to date.
- Disable the presentation of server information.
- Disable HTTP requests.
- Disable the option to browse files (*directory listing*).
- Use non-system and non-administrator accounts.
- Enable HSTS.
- …

The above is a sample list of ways to make IIS more secure. More detailed recommendations can be found on the Internet: https://www.google.com/search?q=how+to+secure+IIS+server.