**SSL in WinHTTP**

Microsoft Windows HTTP Services (WinHTTP) supports Secure Sockets Layer (SSL) transactions including client certificates. This topic explains concepts involved in an SSL transaction and how they are handled using WinHTTP.

WinHTTP支持SSL事务和客户端认证。这边文章解释了涉及SSL事务的概念和怎样用WinHTTP来处理

**Secure Sockets Layer**

SSL is an established standard for ensuring secure HTTP transactions. SSL provides a mechanism to perform up to 128-bit encryption on all transactions between the client and server. It enables the client to verify that the server belongs to a trusted entity through the use of server certificates. It also enables the server to confirm the identity of the client with client certificates.

SSL是确保安全的HTTP事物建立了标准.SSL为客户端和服务端事务提供了多个128bit加密的机制。SSL可以使客户端和服务器通过服务证书和客户证书进行认证。

Each of these issues—encryption, server identity, and client identity—are negotiated in the SSL handshake that occurs when a client first requests a resource from a Secure Hypertext Transfer Protocol (HTTPS) server. Essentially, the client and server each present a list of required and preferred settings. If a common set of requirements can be agreed upon and met, an SSL connection is established.

有关于加密、服务端认证、客户端认证每一个问题，都在客户端第一次向服务端请求资源是进行的SSL握手期间进行协商。

WinHTTP provides a high level interface for using SSL. While the details of the SSL handshake and transaction are handled internally, WinHTTP enables you to retrieve encryption levels, specify the security protocol, and interact with server and client certificates. The following sections provide details on creating WinHTTP based applications that elect an SSL protocol version, examine server certificates, and select client certificates to send to HTTPS servers.

WinHTTP为使用SSL提供了一个高层接口，SSL的握手和事务细节将在内部处理。WinHTTP可以使你恢复加密数据，指定安全协议

**Server Certificates**

Server certificates are sent from the server to the client so that the client can obtain a public key for the server and ensure that the server has been verified by a certification authority. Certificates can contain different types of data. For example, an X.509 certificate includes the format of the certificate, the serial number of the certificate, the algorithm used to sign the certificate, the name of the certification authority (CA) that issued the certificate, the name and public key of the entity that requests the certificate, and the CA's signature.

服务器证书从服务器发送到客户端以便客户端获取服务器的公共密钥并确定服务器是经过CA认证的。证书可以包含不同类型的数据。比如，x.509证书包括证书的格式、证书的序列号、证书签名的算法、发布改正的CA名称、请求证书的名字和公共密钥和CA的签名。

When using the WinHTTP application programming interface (API), you can retrieve a server certificate by calling [**WinHttpQueryOption**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/winhttpqueryoption.htm) and specifying the [**WINHTTP\_OPTION\_SECURITY\_CERTIFICATE\_STRUCT**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/option_flags.htm#WINHTTP_OPTION_SECURITY_CERTIFICATE_STRUCT) flag. The server certificate is returned in a [**WINHTTP\_CERTIFICATE\_INFO**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/internet_certificate_info.htm) structure. If you prefer to retrieve the certificate context, specify the **[WINHTTP\_OPTION\_SERVER\_CERT\_CONTEXT](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/option_flags.htm" \l "WINHTTP_OPTION_SERVER_CERT_CONTEXT)** flag instead.

通过 WinHttpQueryOption + [**WINHTTP\_OPTION\_SECURITY\_CERTIFICATE\_STRUCT**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/option_flags.htm#WINHTTP_OPTION_SECURITY_CERTIFICATE_STRUCT) or [**WINHTTP\_OPTION\_SERVER\_CERT\_CONTEXT**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/option_flags.htm#WINHTTP_OPTION_SERVER_CERT_CONTEXT) 查询服务器

If a server certificate contains errors, details about the error can be obtained in the status callback function. The [**WINHTTP\_CALLBACK\_STATUS\_SECURE\_FAILURE**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/internet_status_callback_prototype.htm#dwInternetStatus) notification indicates an error with a server certificate. The *lpvStatusInformation* parameter contains one or more detailed error flags. See [**WINHTTP\_STATUS\_CALLBACK**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/internet_status_callback_prototype.htm) for more information.

如果服务器证书包含错误，详细的错误信息可以通过回调函数获取。

**Client Certificates**

During the SSL handshake, the server might require authentication. The client is authenticated by supplying a valid client certificate to the server. WinHTTP enables you to select and send a certificate from a local [*certificate store*](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/glossary.htm#term_certificate_store). The following sections describe the process that provides client certificates when using either the WinHTTP API or the [**WinHttpRequest**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/winhttprequest.htm) object.

在SSL握手期间，服务器可能需要认证。客户端可以通过向服务器端提供一个有效证书来进行认证。WinHTTP可以是你选择一个本地的证书。

**WinHTTP API**

Both [**WinHttpSendRequest**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/winhttpsendrequest.htm) and [**WinHttpReceiveResponse**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/winhttpreceiveresponse.htm) can fail to indicate that a request was unsuccessful because the HTTPS server requires authentication. In these cases, call **GetLastError** to returns ERROR\_WINHTTP\_CLIENT\_AUTH\_CERT\_NEEDED. Upon receiving this error, use the appropriate CryptoAPI functions to find an appropriate certificate. Indicate that this certificate should be sent with the next request by calling [**WinHttpSetOption**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/winhttpsetoption.htm) with the [**WINHTTP\_OPTION\_CLIENT\_CERT\_CONTEXT**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/option_flags.htm#WINHTTP_OPTION_CLIENT_CERT_CONTEXT) flag.

The following code example shows how to open a certificate store and locate a certificate based on subject name after the ERROR\_WINHTTP\_CLIENT\_AUTH\_CERT\_NEEDED error has been returned.

if( !WinHttpReceiveResponse( hRequest, NULL ) )

{

if( GetLastError( ) == ERROR\_WINHTTP\_CLIENT\_AUTH\_CERT\_NEEDED )

{

//MY is the store the certificate is in.

hMyStore = CertOpenSystemStore( 0, TEXT("MY") );

if( hMyStore )

{

pCertContext = CertFindCertificateInStore( hMyStore,

X509\_ASN\_ENCODING | PKCS\_7\_ASN\_ENCODING,

0,

CERT\_FIND\_SUBJECT\_STR,

(LPVOID) szCertName, //Subject string in the certificate.

NULL );

if( pCertContext )

{

WinHttpSetOption( hRequest,

WINHTTP\_OPTION\_CLIENT\_CERT\_CONTEXT,

(LPVOID) pCertContext,

sizeof(CERT\_CONTEXT) );

CertFreeCertificateContext( pCertContext );

}

CertCloseStore( hMyStore, 0 );

// NOTE: Application should now resend the request.

}

}

}

Before resending a request that contains a client certificate, you can determine if the supported level of encryption is acceptable for your application. Call [**WinHttpQueryOption**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/winhttpqueryoption.htm) and specify the [**WINHTTP\_OPTION\_SECURITY\_FLAGS**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/option_flags.htm#WINHTTP_OPTION_SECURITY_FLAGS) flag to determine the level of encryption that is used.

**Issuer List Retrieval for SSL Client Authentication [C++]**

When the WinHttp client application sends a request to a secure HTTP server that requires SSL client authentication, WinHttp returns an ERROR\_WINHTTP\_CLIENT\_AUTH\_CERT\_NEEDED if the application has not supplied a client certificate. For computers running on Windows Server 2008 and Windows Vista, WinHttp enables the application to retrieve the certificate issuer list supplied by the server in the authentication challenge. The Issuer List specifies a list of Certificate Authorities (CAs) that are authorized by the server to issue client certificates. The application filters the issuer list to obtain the required certificate.

The WinHttp client application retrieves the issuer list when **WinHttpSendRequest**, or **WinHttpRecieveResponse** returns ERROR\_WINHTTP\_CLIENT\_AUTH\_CERT\_NEEDED. When this error is returned, the application calls **WinHttpQueryOption** with the WINHTTP\_OPTION\_CLIENT\_CERT\_ISSUER\_LIST option. The *lpBuffer* parameter must be large enough to contain a pointer to the SecPkgContext\_IssuerListInfoEx structure. The following code example shows how to retrieve the issuer list.

#include <schannel.h>

...

SecPkgContext\_IssuerListInfoEx\* pIssuerList = NULL;

DWORD dwBufferSize = sizeof(SecPkgContext\_IssuerListInfoEx\*);

If (WinHttpQueryOption(hRequest,

WINHTTP\_OPTION\_CLIENT\_CERT\_ISSUER\_LIST,

&pIssuerList,

&dwBufferSize) == TRUE)

{

// Use the pIssuerList for cert store filtering.

GlobalFree(pIssuerList); // Free the issuer list when done.

}

The information in the SecPkgContext\_IssuerListInfoEx structure, *cIssuers* and *aIssuers*, can be used to search for the certificate as shown in the code example below. For more information about **CertFindChainInStore**, see the Platform SDK Security documentation.

PCERT\_CONTEXT pClientCert = NULL;

CERT\_CHAIN\_FIND\_BY\_ISSUER\_PARA SrchCriteria;

::ZeroMemory(&SrchCriteria, sizeof(CERT\_CHAIN\_FIND\_BY\_ISSUER\_PARA));

SrchCriteria.cbSize = sizeof(CERT\_CHAIN\_FIND\_BY\_ISSUER\_PARA);

SrchCriteria.cIssuer = pIssuerList->cIssuers;

SrchCriteria.rgIssuer = pIssuerList->aIssuers;

PCCERT\_CHAIN\_CONTEXT pClientCertChain =

CertFindChainInStore(

hClientCertStore,

X509\_ASN\_ENCODING,

CERT\_CHAIN\_FIND\_BY\_ISSUER\_CACHE\_ONLY\_URL\_FLAG |

// Do not perform wire download when building chains.

CERT\_CHAIN\_FIND\_BY\_ISSUER\_CACHE\_ONLY\_FLAG,

// Do not search pCacheEntry->\_ClientCertStore

// for issuer certs.

CERT\_CHAIN\_FIND\_BY\_ISSUER,

&SrchCriteria,

NULL);

if (pClientCertChain)

{

pClientCert =

pClientCertChain->rgpChain[0]->rgpElement[0]->pCertContext;

CertDuplicateCertificateContext(pClientCert);

CertFreeCertificateChain(pClientCertChain);

pClientCertChain = NULL;

}

**Optional Client SSL Certificates**

Starting in Windows Server 2008 and Windows Vista, the WinHttp API supports optional client certificates. When the server requests a client certificate, **WinHttpSendRequest**, or **WinHttpRecieveResponse** returns an ERROR\_WINHTTP\_CLIENT\_AUTH\_CERT\_NEEDED error. If the server requests the certificate, but does not require it, the application can specify this option to indicate that it does not have a certificate. The server can choose another authentication scheme or allow anonymous access to the server. The application specifies the WINHTTP\_NO\_CLIENT\_CERT\_CONTEXT macro in the *lpBuffer* parameter of **WinHttpSetOption** as shown in the following code example.

BOOL fRet = WinHttpSetOption ( hRequest,

WINHTTP\_OPTION\_CLIENT\_CERT\_CONTEXT,

WINHTTP\_NO\_CLIENT\_CERT\_CONTEXT,

0);

If the WINHTTP\_NO\_CLIENT\_CERT\_CONTEXT is set, and the server still requires a client certificate, it may send a 403 HTTP status code. For more information, see the WINHTTP\_OPTION\_CLIENT\_CERT\_ISSUER\_LIST option.

**WinHttpRequest Object**

Use the [**SetClientCertificate**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/iwinhttprequest_setclientcertificate.htm) method of the [**WinHttpRequest**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/winhttprequest.htm) object to select client certificates to send to the server with a request. Select a certificate by specifying a certificate selection string with the [**SetClientCertificate**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/iwinhttprequest_setclientcertificate.htm) method. The certificate selection string consists of the certificate location, certificate store, and subject name delimited by backslashes. The following table lists components for this selection string.

|  |  |  |
| --- | --- | --- |
| **Component** | **Description** | **Possible values** |
| Location | Determines the registry key under which the certificates are stored. | The possible values are "LOCAL\_MACHINE" to indicate that the certificate store is under  **HKEY\_LOCAL\_MACHINE**  and "CURRENT\_USER" to indicate that the certificate store is under the non-impersonated  **HKEY\_CURRENT\_USER.**  This component is case-sensitive. |
| Certificate store | Indicates the name of the certificate store that contains the relevant certificate. | Typical certificate stores are "MY", "Root", and "TrustedPeople". This component is case-sensitive. |
| Subject name | Identifies a certificate within the specified certificate store. The first certificate that contains the string specified for this component is selected. | The subject name can be any string. A blank string indicates that the first certificate in the certificate store should be used. This component is case-insensitive. |

The certificate store name and location are optional components. However, if you specify a certificate store, you must also specify the location of that certificate store. The default location is CURRENT\_USER and the default certificate store is "MY".

The following code example shows how to specify that a certificate with the subject "My Middle-Tier Certificate" should be chosen from the "Personal" certificate store in the registry under

**HKEY\_LOCAL\_MACHINE.**

HttpReq.SetClientCertificate(

"LOCAL\_MACHINE\Personal\My Middle-Tier Certificate")

**Note**  In some languages the backslash is an escape character. Remember to modify the certificate selection string to account for this. For example, in Microsoft JScript, use two adjacent backslashes instead of one.

If you do not specify a certificate and an HTTPS server requires a client certificate, WinHTTP selects the first certificate in the default certificate store. If no certificates exist, an error is raised. If the certificate is not accepted, the server returns a 403 status code to indicate that the request cannot be fulfilled. You can then choose a more appropriate certificate with [**SetClientCertificate**](ms-help://MS.MSDNQTR.v90.chs/winhttp/http/iwinhttprequest_setclientcertificate.htm) and resend the request.

[Send comments about this topic to Microsoft](mailto:wsddocfb@microsoft.com?subject=Documentation%20feedback%20[winhttp\http%5d:%20SSL%20in%20WinHTTP%20%20RELEASE:%20(3/27/2008)&body=%0A%0APRIVACY%20STATEMENT%0A%0AThe%20SDK%20team%20uses%20the%20feedback%20submitted%20to%20improve%20the%20SDK%20documentation.%20We%20do%20not%20use%20your%20e-mail%20address%20for%20any%20other%20purpose.%20We%20will%20remove%20your%20e-mail%20address%20from%20our%20system%20after%20the%20issue%20you%20are%20reporting%20has%20been%20resolved.%20While%20we%20are%20working%20to%20resolve%20this%20issue,%20we%20may%20send%20you%20an%20e-mail%20message%20to%20request%20more%20information%20about%20your%20feedback.%20After%20the%20issues%20have%20been%20addressed,%20we%20may%20send%20you%20an%20e-mail%20message%20to%20let%20you%20know%20that%20your%20feedback%20has%20been%20addressed.%0A%0AFor%20more%20information%20about%20Microsoft's%20privacy%20policy,%20see%20http://privacy.microsoft.com/en-us/default.aspx.)

Build date: 3/27/2008