Smuxi - Bug # 640: Validation of certificates always fail

Status:	New	**Priority:**	Normal
Author:	Mirco Bauer	**Category:**	Engine
Created:	11/01/2011	**Assigned to:**	Mirco Bauer
Updated:	01/09/2017	**Due date:**	
Complexity:	High		
Found in Version:			
Subject:	Validation of certificates always fail		

Description:

When connecting to IRC or XMPP servers the certificate validation always fails even when importing their CA and the certificate itself into Mono's certificate storage using the certmgr utility:

<pre>
openssl x509 -in /etc/ssl/certs/Equifax_Secure_CA.pem -out Equifax_Secure_CA.crt -outform der
certmgr -add -c CA Equifax_Secure_CA.crt
</pre>

<pre>
certmgr -list -c CA
Mono Certificate Manager - version 2.6.7.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

Self-signed X.509 v3 Certificate
  Serial Number: CFF4DE35
  Issuer Name: C=US, O=Equifax, OU=Equifax Secure Certificate Authority
  Subject Name: C=US, O=Equifax, OU=Equifax Secure Certificate Authority
  Valid From: 8/22/1998 6:41:51 PM
  Valid Until: 8/22/2018 6:41:51 PM
  Unique Hash: FFA3AC0084DA1673B5A031EBB2156B3E8FBBF6D8
</pre>

<pre>
2011-11-01 12:11:07,831 [-289690768] ERROR Smuxi.Engine.XmppProtocolManager - OnError(): Exception
System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x0026f] in /tmp/buildd/mono-2.6.7/mcs/class/Mono.Security/Mono.Security.Protocol.Tls.Handshake.Client/TlsServerCertificate.cs:323
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00054] in /tmp/buildd/mono-2.6.7/mcs/class/Mono.Security/Mono.Security.Protocol.Tls.Handshake.Client/TlsServerCertificate.cs:105
  at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00037] in /tmp/buildd/mono-2.6.7/mcs/class/Mono.Security/Mono.Security.Protocol.Tls.Handshake/HandshakeMessage.cs:105
  at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
  at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00039] in /tmp/buildd/mono-2.6.7/mcs/class/Mono.Security/Mono.Security.Protocol.Tls/ClientRecordProtocol.cs:81
  at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult)

[0x00127] in /tmp/buildd/mono-2.6.7/mcs/class/Mono.Security/Mono.Security.Protocol.Tls/RecordProtocol.cs:397
--- End of inner exception stack trace ---
at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x0002a] in /tmp/buildd/mono-2.6.7/mcs/class/Mono.Security/Mono.Security.Protocol.Tls/SslStreamBase.cs:102
</pre>

History

11/01/2011 11:20 AM - Mirco Bauer

Certificates can be checked using the Mono tlstest tool found here:
https://raw.githubusercontent.com/mono/mono/23860cbfbe456cbae321fbb0a8139f41ad0caa74/mcs/class/Mono.Security/Test/tools/tlstest/tlstest.cs

Latest version that does not work on Mono < 3.8
https://github.com/mono/mono/blob/master/mcs/class/Mono.Security/Test/tools/tlstest/tlstest.cs

11/01/2011 11:27 AM - Mirco Bauer

<pre>
wget https://raw.github.com/mono/mono/master/mcs/class/Mono.Security/Test/tools/tlstest/tlstest.cs
gmcs tlstest.cs -r:Mono.Security
</pre>

<pre>
certmgr --ssl https://talk.google.com
Mono Certificate Manager - version 2.6.7.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

X.509 Certificate v3
Issued from: C=US, O=Equifax, OU=Equifax Secure Certificate Authority
Issued to: C=US, S=California, L=Mountain View, O=Google Inc., CN=talk.google.com
Valid from: 4/11/2007 7:20:16 PM
Valid until: 4/10/2012 7:20:16 PM
This certificate is already in the AddressBook store.

No certificate were added to the stores.
</pre>

<pre>
./tlstest.exe --tls https://talk.google.com

https://talk.google.com
[Subject]
CN=talk.google.com, O=Google Inc., L=Mountain View, S=California, C=US

[Issuer]
OU=Equifax Secure Certificate Authority, O=Equifax, C=US

[Not Before]
4/11/2007 7:20:16 PM

[Not After]
4/10/2012 7:20:16 PM

[Thumbprint]
953FBE4D549B7E700EC14782C68CD09F9B512BCE

Valid From: 4/11/2007 7:20:16 PM
Valid Until: 4/10/2012 7:20:16 PM

Error #-2146762486: CERT_E_CHAINING 0x800B010A
</pre>

11/05/2011 05:32 PM - Mirco Bauer
On Mono 2.10.5 the same issue happens:

<pre>
meebey@redhorse:~$ openssl x509 -in /etc/ssl/certs/Equifax_Secure_CA.pem -out Equifax_Secure_CA.crt -outform der
meebey@redhorse:~$ certmgr -add -c CA Equifax_Secure_CA.crt
Mono Certificate Manager - version 2.10.5.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

1 certificate(s) added to store CA.
</pre>

<pre>
meebey@redhorse:~/tmp$ wget https://raw.github.com/mono/mono/master/mcs/class/Mono.Security/Test/tools/tlstest/tlstest.cs
--2011-11-05 17:51:01-- https://raw.github.com/mono/mono/master/mcs/class/Mono.Security/Test/tools/tlstest/tlstest.cs
Resolving raw.github.com (raw.github.com)... 207.97.227.243
Connecting to raw.github.com (raw.github.com)|207.97.227.243|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9475 (9.3K) [text/plain]
Saving to: `tlstest.cs'

100%[==>] 9,475
--.-K/s in 0s

2011-11-05 17:51:07 (93.5 MB/s) - `tlstest.cs' saved [9475/9475]

meebey@redhorse:~/tmp$ gmcs tlstest.cs -r:Mono.Security
tlstest.cs(172,37): warning CS0618: `System.Net.ServicePointManager.CertificatePolicy' is obsolete: `Use ServerCertificateValidationCallback instead'
tlstest.cs(201,40): warning CS0618: `System.Net.Dns.Resolve(string)' is obsolete: `Use GetHostEntry instead'
Compilation succeeded - 2 warning(s)
meebey@redhorse:~/tmp$ certmgr --ssl https://talk.google.com
Mono Certificate Manager - version 2.10.5.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

X.509 Certificate v3
Issued from: C=US, O=Equifax, OU=Equifax Secure Certificate Authority

```
   Issued to:   C=US, S=California, L=Mountain View, O=Google Inc., CN=talk.google.com
   Valid from:  4/11/2007 5:20:16 PM
   Valid until: 4/10/2012 5:20:16 PM
Import this certificate into the AddressBook store ?y

1 certificate added to the stores.
meebey@redhorse:~/tmp$ certmgr --ssl https://talk.google.com
Mono Certificate Manager - version 2.10.5.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.


X.509 Certificate v3
   Issued from: C=US, O=Equifax, OU=Equifax Secure Certificate Authority
   Issued to:   C=US, S=California, L=Mountain View, O=Google Inc., CN=talk.google.com
   Valid from:  4/11/2007 5:20:16 PM
   Valid until: 4/10/2012 5:20:16 PM
This certificate is already in the AddressBook store.

No certificate were added to the stores.
meebey@redhorse:~/tmp$ certmgr -list -c CA
Mono Certificate Manager - version 2.10.5.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

Self-signed X.509 v3 Certificate
  Serial Number: CFF4DE35
  Issuer Name:   C=US, O=Equifax, OU=Equifax Secure Certificate Authority
  Subject Name:  C=US, O=Equifax, OU=Equifax Secure Certificate Authority
  Valid From:    8/22/1998 4:41:51 PM
  Valid Until:   8/22/2018 4:41:51 PM
  Unique Hash:   FFA3AC0084DA1673B5A031EBB2156B3E8FBBF6D8

meebey@redhorse:~/tmp$ certmgr -list -c My
Mono Certificate Manager - version 2.10.5.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

meebey@redhorse:~/tmp$ ./tlstest.exe --tls https://talk.google.com

https://talk.google.com
[Subject]
  CN=talk.google.com, O=Google Inc., L=Mountain View, S=California, C=US

[Issuer]
  OU=Equifax Secure Certificate Authority, O=Equifax, C=US

[Not Before]
  4/11/2007 7:20:16 PM

[Not After]
  4/10/2012 7:20:16 PM
```

[Thumbprint]
953FBE4D549B7E700EC14782C68CD09F9B512BCE

Valid From: 4/11/2007 7:20:16 PM
Valid Until: 4/10/2012 7:20:16 PM

Error #-2146762486: CERT_E_CHAINING 0x800B010A
</pre>

01/14/2013 06:11 PM - Mirco Bauer
Here some useful SSL debugging commands:

<pre>
gnutls-cli -V irc.oftc.net --port 6697 --crlf --x509cafile /etc/ssl/certs/ca-certificates.crt
</pre>

<pre>
openssl s_client -showcerts -host irc.oftc.net -port 6697 -CApath /etc/ssl/certs
</pre>

09/16/2014 09:46 PM - Infinity Zero
Firstly, one should use @-c Trust@ instead of @-c CA@. The former is what `mozroots` does. Not sure why but it works.

Secondly, it seems that there is something wrong with how smuxi uses the SSL library. A basic simple program works:

<pre>
using System;
using System.Net.Sockets;
using System.Net.Security;

public class TlsTest {

 public static void Main (string[] args)
 {
  Console.WriteLine("checking " + args[0]);
  var tcpClient = new TcpClient (args[0], int.Parse(args[1]));
  var ssl = new SslStream (tcpClient.GetStream ());
  ssl.AuthenticateAsClient (args[0]);
  Console.WriteLine("success");
 }
}
</pre>

<pre>
with mozroots certs imported into Trust
$ ./lol.exe irc.freenode.net 6697
checking irc.freenode.net
success

$ rm -rf ~/.config/.mono/certs/*
$ ./lol.exe irc.freenode.net 6697
checking irc.freenode.net

Unhandled Exception:
System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
--- End of inner exception stack trace ---
at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
[ERROR] FATAL UNHANDLED EXCEPTION: System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
--- End of inner exception stack trace ---
at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
1
</pre>

09/17/2014 09:19 AM - Mirco Bauer

- Category set to Engine
- Assigned to set to Mirco Bauer
- Target version set to 0.11.1
- Complexity set to High

Your tests deliver important information. I agree that the current situation seems to be that Smuxi is doing something that breaks the default cert validation for some reason. I will look into that.

09/19/2014 11:45 AM - Mirco Bauer

<pre>
18:00:16 <directhex> directhex@marceline:/tmp$ mono hello.exe
18:00:17 <directhex> numcerts:0
18:00:17 <directhex> directhex@marceline:/tmp$ sudo mkdir /usr/share/.mono/keypairs/
18:00:17 <directhex> directhex@marceline:/tmp$ mono hello.exe
18:00:17 <directhex> numcerts:140
18:01:20 <directhex> *enumerating* certs requires that a keypairs/ folder exists in the parent folder of the cert store, and *enumerating* certs will try to create that folder during the process
18:01:32 <directhex> if mkdir fails, 0 certs returned
</pre>

<pre>
18:03:20 <directhex> meebey: Trust is definitely the right store, not CA, by the way
</pre>

09/19/2014 12:01 PM - Mirco Bauer

The machine CA/Trust store of Mono is in /usr/share/.mono/certs/
The user CA/Trust store of Mono is in ~/.config/.mono/certs/

09/19/2014 12:07 PM - Mirco Bauer

<pre>
meebey@redhorse:~$ certmgr -ssl https://irc.freenode.net:6697
Mono Certificate Manager - version 3.2.8.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

X.509 Certificate v3
  Issued from: C=US, S=UT, L=Salt Lake City, O=The USERTRUST Network, OU=http://www.usertrust.com, CN=UTN-USERFirst-Hardware
  Issued to: C=FR, O=GANDI SAS, CN=Gandi Standard SSL CA
  Valid from: 10/23/2008 12:00:00 AM
  Valid until: 5/30/2020 10:48:38 AM
  *** WARNING: Certificate signature is INVALID ***
Import this certificate into the CA store ?y

</pre>

<pre>
meebey@redhorse:~$ ./lol.exe irc.freenode.net 6697
checking irc.freenode.net

Unhandled Exception:
System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
        at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
  at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
    at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
  --- End of inner exception stack trace ---
  at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
[ERROR] FATAL UNHANDLED EXCEPTION: System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
        at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates)

[0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
  at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
    at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
  --- End of inner exception stack trace ---
  at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
meebey@redhorse:~$ mv .config/.mono/certs/CA/ski-B6A8FFA2A82FD0A6CD4BB168F3E7501031A77921.cer .config/.mono/certs/Trust/
meebey@redhorse:~$ ./lol.exe irc.freenode.net 6697
checking irc.freenode.net
success
</pre>

09/19/2014 12:07 PM - Mirco Bauer

First conclusion: -certmgr --ssl is useless to populate the user store because it always imports into the CA user store instead of Trust user store!- (see amended first conclusion)

09/19/2014 12:25 PM - Mirco Bauer

meebey@redhorse:~$ cat lol-with-callback.cs
<pre>
```
using System;
using System.Net.Sockets;
using System.Net.Security;

public class TlsTest {

    public static void Main (string[] args)
    {
        Console.WriteLine("checking " + args[0]);
        var tcpClient = new TcpClient (args[0], int.Parse(args[1]));
        var ssl = new SslStream (tcpClient.GetStream (), false,
            (sender, certificate, chain, sslPolicyErrors) => {
                Console.WriteLine("sslPolicyErrors: {0}", sslPolicyErrors);
                return sslPolicyErrors == SslPolicyErrors.None;
            });
        ssl.AuthenticateAsClient (args[0]);
        Console.WriteLine("success");
    }
}
```
</pre>
<pre>
meebey@redhorse:~$ ./lol.exe irc.freenode.net 6697
checking irc.freenode.net
success
meebey@redhorse:~$ ./lol-with-callback.exe irc.freenode.net 6697
checking irc.freenode.net
sslPolicyErrors: RemoteCertificateNotAvailable

Unhandled Exception:
System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from

</pre>

server.
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
--- End of inner exception stack trace ---
at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
[ERROR] FATAL UNHANDLED EXCEPTION: System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
--- End of inner exception stack trace ---
at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
meebey@redhorse:~$

</pre>

09/19/2014 12:39 PM - Mirco Bauer

Turns out irc.freenode.net sends the intermediate CA cert but not the root CA cert! See:

<pre>
openssl s_client -connect irc.freenode.net:6697
</pre>

First (amended) conclusion: certmgr --ssl imports the *intermediate CA cert* correctly in the *CA store*, as *Trust* is only for *Root CA certs*

09/19/2014 01:02 PM - Mirco Bauer

<pre>
meebey@redhorse:~$ openssl x509 -in /etc/ssl/certs/UTN_USERFirst_Hardware_Root_CA.pem -out UTN_USERFirst_Hardware_Root_CA.crt -outform der
meebey@redhorse:~$ certmgr -add -c Trust UTN_USERFirst_Hardware_Root_CA.crt
Mono Certificate Manager - version 3.2.8.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

1 certificate(s) added to store Trust.

meebey@redhorse:~$ ./lol.exe irc.freenode.net 6697
checking irc.freenode.net
success

```
meebey@redhorse:~$ ./lol-with-callback.exe irc.freenode.net 6697
checking irc.freenode.net
sslPolicyErrors: RemoteCertificateNotAvailable

Unhandled Exception:
System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
        at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
  at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
    at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
  --- End of inner exception stack trace ---
  at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
[ERROR] FATAL UNHANDLED EXCEPTION: System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
        at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
  at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
    at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
  --- End of inner exception stack trace ---
  at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0

meebey@redhorse:~$ ls .config/.mono/certs/{CA,Trust} -l
.config/.mono/certs/CA:
total 0

.config/.mono/certs/Trust:
total 4
-rw-r--r-- 1 meebey meebey 1144 2014-09-19 12:42 ski-A1725F261B289843955D0737D585969D4BD2C345.cer

meebey@redhorse:~$ openssl s_client -connect irc.freenode.net:6697 -CAfile /etc/ssl/certs/UTN_USERFirst_Hardware_Root_CA.pem
CONNECTED(00000003)
depth=2 C = US, ST = UT, L = Salt Lake City, O = The USERTRUST Network, OU = http://www.usertrust.com, CN = UTN-USERFirst-Hardware
verify return:1
depth=1 C = FR, O = GANDI SAS, CN = Gandi Standard SSL CA
verify return:1
depth=0 OU = Domain Control Validated, OU = Gandi Standard Wildcard SSL, CN = *.freenode.net
verify return:1
---
...
</pre>
```

<pre>
meebey@redhorse:~$ certmgr -ssl https://irc.freenode.net:6697
Mono Certificate Manager - version 3.2.8.0
Manage X.509 certificates and CRL from stores.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

X.509 Certificate v3
  Issued from: C=US, S=UT, L=Salt Lake City, O=The USERTRUST Network, OU=http://www.usertrust.com, CN=UTN-USERFirst-Hardware
  Issued to: C=FR, O=GANDI SAS, CN=Gandi Standard SSL CA
  Valid from: 10/23/2008 12:00:00 AM
  Valid until: 5/30/2020 10:48:38 AM
  *** WARNING: Certificate signature is INVALID ***
Import this certificate into the CA store ?y

X.509 Certificate v3
  Issued from: C=FR, O=GANDI SAS, CN=Gandi Standard SSL CA
  Issued to: OU=Domain Control Validated, OU=Gandi Standard Wildcard SSL, CN=*.freenode.net
  Valid from: 1/13/2014 12:00:00 AM
  Valid until: 1/14/2015 11:59:59 PM
Import this certificate into the AddressBook store ?n

1 certificate added to the stores.
meebey@redhorse:~$ ls .config/.mono/certs/{CA,Trust} -l
.config/.mono/certs/CA:
total 4
-rw-r--r-- 1 meebey meebey 1191 2014-09-19 12:48 ski-B6A8FFA2A82FD0A6CD4BB168F3E7501031A77921.cer

.config/.mono/certs/Trust:
total 4
-rw-r--r-- 1 meebey meebey 1144 2014-09-19 12:42 ski-A1725F261B289843955D0737D585969D4BD2C345.cer
meebey@redhorse:~$ ./lol.exe irc.freenode.net 6697
checking irc.freenode.net
success
meebey@redhorse:~$ ./lol-with-callback.exe irc.freenode.net 6697
checking irc.freenode.net
sslPolicyErrors: None
success
</pre>

Second conclusion: certificate validation in Mono works if the intermediate CA was imported _and_ the root CA but not when just the root CA was added to the user Trust store.

09/19/2014 01:58 PM - Mirco Bauer

Third conclusion: Smuxi can connect to irc.freenode.net with certificate validation enabled *if* the root CA was imported into the user Trust store *and* the intermediate CA was imported into the user CA store. This is probably a Mono bug.

11/12/2014 09:09 PM - Mirco Bauer

https://github.com/mono/mono/pull/1290

01/14/2015 11:29 AM - Mirco Bauer

Certificates stored in Mono's user cert store can be checked like this:

<pre>
openssl x509 -inform der -in ~/.config/.mono/certs/CA/ski-B6A8FFA2A82FD0A6CD4BB168F3E7501031A77921.cer -text -noout
</pre>

01/14/2015 12:14 PM - Mirco Bauer

- File lol-with-callback.cs added

01/14/2015 12:18 PM - Mirco Bauer

Fourth conclusion: Mono's default certificate validator is different than != SslPolicyErrors.None as a connect without a validation callback (return sslPolicyErrors == SslPolicyErrors.None) works but providing that minimal wrapper does not!

<pre>
meebey@redhorse:~$ ./lol-with-callback.exe irc.freenode.net 6697
checking irc.freenode.net
sslPolicyErrors: RemoteCertificateNotAvailable

Unhandled Exception:
System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
	at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
  at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
    at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
  --- End of inner exception stack trace ---
  at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
[ERROR] FATAL UNHANDLED EXCEPTION: System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
	at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
  at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
    at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
  at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
  --- End of inner exception stack trace ---
  at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
meebey@redhorse:~$ ./lol.exe irc.freenode.net 6697
checking irc.freenode.net
success
meebey@redhorse:~$

</pre>

01/14/2015 12:32 PM - Mirco Bauer

mono --trace=N:System.Net,N:Mono.Security.X509,N:Mono.Security.Tls lol-with-callback.exe irc.freenode.net 6697

<pre>
...
[0x7f9843fff700: 4.69910 4] ENTER: Mono.Security.X509.X509Extension:Decode ()(this:0x7f985a483148[Mono.Security.X509.X509Extension lol-with-callback.exe],)
[0x7f9843fff700: 4.69910 4] LEAVE: Mono.Security.X509.X509Extension:Decode ()
[0x7f9843fff700: 4.69911 3] LEAVE: Mono.Security.X509.X509Extension:.ctor (Mono.Security.ASN1)
[0x7f9843fff700: 4.69912 2] LEAVE: Mono.Security.X509.X509ExtensionCollection:.ctor (Mono.Security.ASN1)
[0x7f9843fff700: 4.69912 1] LEAVE: Mono.Security.X509.X509Certificate:Parse (byte[])
[0x7f9843fff700: 4.69913 0] LEAVE: Mono.Security.X509.X509Certificate:.ctor (byte[])
[0x7f9843fff700: 4.69914 0] ENTER: Mono.Security.X509.X509Certificate:get_RSA ()(this:0x7f9843a7a9b8[Mono.Security.X509.X509Certificate lol-with-callback.exe],)
[0x7f9843fff700:] EXCEPTION handling: System.ArgumentException: Offset and length were out of bounds for the array or count is greater than the number of elements from index to the end of the source collection.
[0x7f9843fff700: 4.69984 1] ENTER: Mono.Security.X509.X509StoreManager:get_LocalMachine ()()
[0x7f9843fff700: 4.69985 1] LEAVE: Mono.Security.X509.X509StoreManager:get_LocalMachine ()[Mono.Security.X509.X509Stores:0x7f98438c4430]
[0x7f9843fff700: 4.69986 1] ENTER: Mono.Security.X509.X509Stores:Open (string,bool)(this:0x7f98438c4430[Mono.Security.X509.X509Stores lol-with-callback.exe], [STRING:0x7f985c1a2568:CA], 0,)
[0x7f9843fff700: 4.69989 1] LEAVE: Mono.Security.X509.X509Stores:Open (string,bool)[OBJECT:(nil)]
[0x7f9843fff700:] EXCEPTION handling: System.Security.Cryptography.CryptographicException: Store CA doesn't exists.
[0x7f9843fff700: 4.70016 1] ENTER: Mono.Security.X509.X509StoreManager:get_CurrentUser ()()
[0x7f9843fff700: 4.70017 1] LEAVE: Mono.Security.X509.X509StoreManager:get_CurrentUser ()[Mono.Security.X509.X509Stores:0x7f9843cff240]
[0x7f9843fff700: 4.70018 1] ENTER: Mono.Security.X509.X509Stores:Open (string,bool)(this:0x7f9843cff240[Mono.Security.X509.X509Stores lol-with-callback.exe], [STRING:0x7f985c1a2568:CA], 0,)
[0x7f9843fff700: 4.70020 2] ENTER: Mono.Security.X509.X509Store:.ctor (string,bool)(this:0x7f985a484498[Mono.Security.X509.X509Store lol-with-callback.exe], [STRING:0x7f985a484438:/home/meebey/.config/.mono/certs/CA], 1,)
[0x7f9843fff700: 4.70021 2] LEAVE: Mono.Security.X509.X509Store:.ctor (string,bool)
[0x7f9843fff700: 4.70022 1] LEAVE: Mono.Security.X509.X509Stores:Open (string,bool)[Mono.Security.X509.X509Store:0x7f985a484498]
[0x7f9843fff700: 4.70022 1] ENTER: Mono.Security.X509.X509Store:get_Certificates ()(this:0x7f985a484498[Mono.Security.X509.X509Store lol-with-callback.exe],)
...

</pre>

<pre>
meebey@redhorse:~$ sudo mkdir -p /usr/share/.mono/certs/CA
[sudo] password for meebey:
meebey@redhorse:~$ mono --trace=none lol-with-callback.exe irc.freenode.net 6697
checking irc.freenode.net
[0x7f8ad4859700:] EXCEPTION handling: System.ArgumentException: Offset and length were out of bounds for the array or count is greater than the number of elements from index to the end of the source collection.
sslPolicyErrors: RemoteCertificateNotAvailable
[0x7f8ad4859700:] EXCEPTION handling: Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
[0x7f8ad4a5a700:] EXCEPTION handling: Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
[0x7f8ad4a5a700:] EXCEPTION handling: System.IO.IOException: The authentication or decryption has failed.
[0x7f8ad8a11780:] EXCEPTION handling: System.IO.IOException: The authentication or decryption has failed.

Unhandled Exception:
System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
    at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0

at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
--- End of inner exception stack trace ---
at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
[ERROR] FATAL UNHANDLED EXCEPTION: System.IO.IOException: The authentication or decryption has failed. ---> Mono.Security.Protocol.Tls.TlsException: Invalid certificate received from server.
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.LocalValidation (Mono.Security.Protocol.Tls.ClientContext context, AlertDescription description) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.validateCertificates (Mono.Security.X509.X509CertificateCollection certificates) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.Client.TlsServerCertificate.ProcessAsTls1 () [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.Handshake.HandshakeMessage.Process () [0x00000] in <filename unknown>:0
at (wrapper remoting-invoke-with-check) Mono.Security.Protocol.Tls.Handshake.HandshakeMessage:Process ()
at Mono.Security.Protocol.Tls.ClientRecordProtocol.ProcessHandshakeMessage (Mono.Security.Protocol.Tls.TlsStream handMsg) [0x00000] in <filename unknown>:0
at Mono.Security.Protocol.Tls.RecordProtocol.InternalReceiveRecordCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
--- End of inner exception stack trace ---
at Mono.Security.Protocol.Tls.SslStreamBase.AsyncHandshakeCallback (IAsyncResult asyncResult) [0x00000] in <filename unknown>:0
</pre>

Creating /usr/share/.mono/certs/CA gets rid of the handled exception but it does not fix the validation issue:

<pre>
[0x7f8d49bdb700: 1.06881 4] ENTER: Mono.Security.X509.X509Extension:Decode ()(this:0x7f8d4bc81c20[Mono.Security.X509.X509Extension lol-with-callback.exe],)
[0x7f8d49bdb700: 1.06882 4] LEAVE: Mono.Security.X509.X509Extension:Decode ()
[0x7f8d49bdb700: 1.06882 3] LEAVE: Mono.Security.X509.X509Extension:.ctor (Mono.Security.ASN1)
[0x7f8d49bdb700: 1.06883 2] LEAVE: Mono.Security.X509.X509ExtensionCollection:.ctor (Mono.Security.ASN1)
[0x7f8d49bdb700: 1.06883 1] LEAVE: Mono.Security.X509.X509Certificate:Parse (byte[])
[0x7f8d49bdb700: 1.06884 0] LEAVE: Mono.Security.X509.X509Certificate:.ctor (byte[])
[0x7f8d49bdb700: 1.06884 0] ENTER: Mono.Security.X509.X509Certificate:get_RSA ()(this:0x7f8d495ca9b8[Mono.Security.X509.X509Certificate lol-with-callback.exe],)
[0x7f8d49bdb700:] EXCEPTION handling: System.ArgumentException: Offset and length were out of bounds for the array or count is greater than the number of elements from index to the end of the source collection.
[0x7f8d49bdb700: 1.06954 1] ENTER: Mono.Security.X509.X509StoreManager:get_LocalMachine ()()
[0x7f8d49bdb700: 1.06955 1] LEAVE: Mono.Security.X509.X509StoreManager:get_LocalMachine ()[Mono.Security.X509.X509Stores:0x7f8d49420f30]
[0x7f8d49bdb700: 1.06956 1] ENTER: Mono.Security.X509.X509Stores:Open (string,bool)(this:0x7f8d49420f30[Mono.Security.X509.X509Stores lol-with-callback.exe], [STRING:0x7f8d4dd12518:CA], 0,)
[0x7f8d49bdb700: 1.06958 2] ENTER: Mono.Security.X509.X509Store:.ctor (string,bool)(this:0x7f8d4bc82d98[Mono.Security.X509.X509Store lol-with-callback.exe], [STRING:0x7f8d4bc82d48:/usr/share/.mono/certs/CA], 1,)
[0x7f8d49bdb700: 1.06959 2] LEAVE: Mono.Security.X509.X509Store:.ctor (string,bool)
[0x7f8d49bdb700: 1.06959 1] LEAVE: Mono.Security.X509.X509Stores:Open (string,bool)[Mono.Security.X509.X509Store:0x7f8d4bc82d98]
[0x7f8d49bdb700: 1.06960 1] ENTER: Mono.Security.X509.X509Store:get_Certificates ()(this:0x7f8d4bc82d98[Mono.Security.X509.X509Store lol-with-callback.exe],)
[0x7f8d49bdb700: 1.06961 2] ENTER: Mono.Security.X509.X509Store:BuildCertificatesCollection (string)(this:0x7f8d4bc82d98[Mono.Security.X509.X509Store lol-with-callback.exe], [STRING:0x7f8d4bc82d48:/usr/share/.mono/certs/CA],)

[0x7f8d49bdb700: 1.06961 3] ENTER: Mono.Security.X509.X509Store:CheckStore (string,bool)(this:0x7f8d4bc82d98[Mono.Security.X509.X509Store lol-with-callback.exe], [STRING:0x7f8d4bc82d48:/usr/share/.mono/certs/CA], 0,)
[0x7f8d49bdb700: 1.06962 3] LEAVE: Mono.Security.X509.X509Store:CheckStore (string,bool)TRUE:1
[0x7f8d49bdb700: 1.06966 2] LEAVE: Mono.Security.X509.X509Store:BuildCertificatesCollection (string)[Mono.Security.X509.X509CertificateCollection:0x7f8d4bc82dd0]
[0x7f8d49bdb700: 1.06966 1] LEAVE: Mono.Security.X509.X509Store:get_Certificates ()[Mono.Security.X509.X509CertificateCollection:0x7f8d4bc82dd0]

</pre>

01/14/2015 04:42 PM - Mirco Bauer

- File lol.cs added

02/18/2016 06:45 PM - Mirco Bauer

- Target version deleted (0.11.1)

01/09/2017 09:22 AM - Mirco Bauer

As a short term workaround, you can connect to SSL/TLS enabled servers using stunnel: https://smuxi.im/faq/usage/stunnel/
stunnel acts as a socket proxy between Smuxi and the IRCd and does the SSL/TLS handling. Smuxi connects to stunnel using a plaintext socket on localhost which talks to stunnel which talks to the IRC server using SSL/TLS

Files

lol-with-callback.cs	635 Bytes	01/14/2015	Mirco Bauer
lol.cs	410 Bytes	01/14/2015	Mirco Bauer