

Forum:[Hack Forums](#)

Thread: [Simple authentication mechanism with .NET Remoting \(it's like LiteCode\)](#)

gigajew - June 24, 2018, 8:08 am

Finally started digging into .NET Remoting and found that it's pretty much like LiteCode. You can easily host an app that requires authentication this way. You could create remote crypters to protect your compilation method or source code, rats and other things with this. It pretty much has endless capabilities. You could probably even license an app with this if you stored licenses in a database and had your server check and verify.

What the project looks like right now:

Server and client shares a library that contains the interface `IAuthenticate` which exposes method `Authenticate(user, password, ref token)`

Server shares an instance of a class that inherits from `IAuthenticate` to verify that users and password exist in the database (which I just made a static class)

Client sends calls to servers `IAuthenticate` instance to check that users exist

Download sample project: <https://files.catbox.moe/tvhkex.zip>

int - June 24, 2018, 8:12 am

Interesting, I was looking at litecode a while ago. Posting here to look at later.

gigajew - June 24, 2018, 8:53 am

For dual protection you could actually authenticate the user first using your secure web api with https

Consider this scenario:

client authenticates with www.domain.com/user/authenticate

Upon successful authentication, the client then receives the necessary library to authenticate with the remoting-server, and the client's remoting-client is started, you could even stream a token with it, so that the user doesn't have to log in twice

The client now communicates with the remoting-server and can use functions from it

0xDEAD10CC - June 24, 2018, 9:41 am

.NET remoting is old, and pretty much dead, and has been for a decade. It was also very slow.

HFDone - June 24, 2018, 11:48 am

LiteCode is over complicated it can be done much simple with few lines

int - June 24, 2018, 12:45 pm

(Jun 24, 2018 11:48 AM)HFDone Wrote: [\[CITE\]](#) LiteCode is over complicated it can be done much simple with few lines! want to see your work on this.

@OP can you upload to github? Just want to look on mobile

HFDone - June 24, 2018, 12:55 pm

simple server client connection client send the method name server call the method and send it back

0xDEAD10CC - June 24, 2018, 4:31 pm

(Jun 24, 2018 12:55 PM)HFDone Wrote: [\[CITE\]](#) simple server client connection client send the method name server call the method and send it back

Yeah that's the high level concept, I haven't looked yet but I'm pretty sure the traffic itself is not just "sent" like you're saying though. IIRC he's created his own security algorithm or something for the transfer of this information as a handshake to validate the callee. This is how I would've done it too anyways, not just for added security on the client permissions aspect, but to prevent the server itself from being easily bypassed.

Closed User - June 25, 2018, 3:19 am

There's literally zero fucking reason to use all those libraries if you're able to:

Write super basic TCP or even copy-paste public code

Basic auth-check for each incoming TCP request(server-wise), aka compare one/more parameters depending on how you lock(hwid, hwid+ip etc..)

Put the sensitive code on your server, end-user program requests your server-> server does action
-> server responds to end-user program with result

This is literally a 2 hour job to write up a base, stop being lazy! :)

gigajew - June 25, 2018, 3:05 pm

Yeah sockets are easy but the serialization part is not as easy

Washi - June 25, 2018, 4:11 pm

(Jun 25, 2018 03:19 AM)Big Benim Wrote: [\[CITE\]](#) There's literally zero fucking reason to use all those libraries if you're able to:

Write super basic TCP or even copy-paste public code

Basic auth-check for each incoming TCP request(server-wise), aka compare one/more parameters depending on how you lock(hwid, hwid+ip etc..)

Put the sensitive code on your server, end-user program requests your server-> server does action
-> server responds to end-user program with result

This is literally a 2 hour job to write up a base, stop being lazy! :)

There is a difference between plain network communication and secure network communication using a proper encryption + authentication protocol, not to mention efficient, yet modular serialisation and deserialisation of objects being transferred between the client and the server.

Closed User - June 26, 2018, 6:11 am

(Jun 25, 2018 04:11 PM)Washi Wrote: [\[CITE\]](#) There is a difference between plain network communication and secure network communication using a proper encryption + authentication protocol, not to mention efficient, yet modular serialisation and deserialisation of objects being transferred between the client and the server.

If you dont know how to encrypt your data or write authentication or even just implement SSL you shouldn't even think about using it in the first place lol.

Serialisation/deserialisation is a kindergardens play with json ?

0xDEAD10CC - June 26, 2018, 12:42 pm

(Jun 26, 2018 06:11 AM)Big Benim Wrote: [\[CITE\]](#) If you dont know how to encrypt your data or write authentication or even just implement SSL you shouldn't even think about using it in the first place lol.

Serialisation/deserialisation is a kindergardens play with json ?

He did say modular, serialization itself is easy, but write something that can correctly interpret any type without knowing what that type might be beforehand, and tell the caller dynamically how to use that object. Still trivial, but is worth a bit more thinking to do efficiently. Easy to hardcode an API into a remote endpoint, try writing something tailored to suit everyone's needs.