

Forum:[Hack Forums](#)

Thread: [Remote \(server side\) code with netseal validation.](#)

BahNahNah - January 22, 2017, 8:47 am

I wanted to make a project to display some usages of [SyncIO](#), so here it is.

Download: [Github](#)

All remote function calls are validated server side by the netseal api to check that the user making the call has a valid licence.

Setting up the remote code
[\[Remote code lib example\]](#)

Make a new class library

Include SIORemoteCode.dll

Write the code

The RemoteApp Attribute must be attached to classes containing remote functions. This attribute takes 2 parameters.

- 1) App ID: The netseal Application ID (Same one used to call Broker.Initialize)
- 2) Remote API Token: can be found in the netseal licence panel under remote.

Each function that you want to be made available for remote calling mustBe public

Be static

Have the RemoteFunction Attribute

The RemoteFunction Attribute takes 1 required argument, and one optional.

- 1) Function Name (Required): This is the name that will be used to call this function.
- 2) Description (Optional): Displayed on main app for your use only, has no effect on behavior.

Put the dll in the "Remote" folder found in the same directory as SyncIONetSealRemoteCode.exe. If "Remote" directory is not found, run SyncIONetSealRemoteCode.exe to generate then close.

Click settings -> Network and add any ports you wish to listen on.

Close the ports window and the status label should indicate that the application is listening.

Setting up the application

This application will have both SyncIO.dll and SIORemoteCode.dll as dependencies when using

this.

[\[Application example\]](#)

Create a new instance of RemoteCode supplying the AppID and netseal username in the constructor

Subscribe to the OnConnect event

call Init supplying the IP and Port of the running server

The OnConnect event will be raised when connection is completed. On this event is where the GetFunction method should be called.

Code [RemoteCode].GetFunction<return type>(function name)
This will return a SecureFunction<return type>
there are 2 methods that can be called from a SecureFunction

Call: Non-blocking remote function call. Returns the Call Id of the requested function call (Subscribe to the ReturnCallback event to get the remote function return).

CallWait: Calls the remote function and blocks the current thread until the return value is retrieved.

If the function does not exist or throws an exception, the default value for the return type is returned.

I'll update this thread with more info at some point.

gigajew - January 22, 2017, 9:13 am

Great Job! I'm a fan of your coding

Barret - January 22, 2017, 9:39 am

Been scrolling into it, looks tight lad

Malum™ - January 22, 2017, 10:06 am

Great example to get people to use SyncIO. I'm loving it. Seriously, keep up the good work!

debug - January 22, 2017, 3:45 pm

Can't wait to take a look at this when I get some free time. I tell people all the time Net Seal + Server sided code/variables is the way to go.

Well done once again.

BahNahNah - January 22, 2017, 3:50 pm

(Jan 22, 2017 04:45 PM)debug Wrote: [\[CITE\]](#) Can't wait to take a look at this when I get some free time. I tell people all the time Net Seal + Server sided code/variables is the way to go.

Well done once again.

Be warned finished it at 6am so there might be some unexceptionably stupid code lying around.

WinA - January 29, 2017, 3:36 am

Really nice work there OP! Will you be continuing on SyncIO?

Aristocles - February 1, 2017, 3:31 am

Hello BahNahNah,

First, thread rated 5/5, can't understand why is there so few replies.

I would like to know if there is a limited amount of parameter allowed or is it only defined by the function itself ?

<https://github.com/BahNahNah/SyncIONSRemoteCode/blob/master/ExampleApplication/Program.cs#L50>

Then, do you think it's possible to grab already-coded functions via dnlib/cecil in a main .exe file, extract them and inject them in the remote code library, adding the correct attribute according to user's netseal ID ? I'd like to do that but before I'd like to know if its theoretically possible.

Again, thank you :)

BahNahNah - February 1, 2017, 3:46 pm

(Feb 1, 2017 04:31 AM).NETGuard Wrote: [\[CITE\]](#) Hello BahNahNah,

First, thread rated 5/5, can't understand why is there so few replies.

I would like to know if there is a limited amount of parameter allowed or is it only defined by the function itself ?

<https://github.com/BahNahNah/SyncIONSRemoteCode/blob/master/ExampleApplication/Program.cs#L50>

Then, do you think it's possible to grab already-coded functions via dnlib/cecil in a main .exe file, extract them and inject them in the remote code library, adding the correct attribute according to user's netseal ID ? I'd like to do that but before I'd like to know if its theoretically possible.

Again, thank you :)

When calling the remote function, it is up to the dev to make sure that the correct number of parameters are sent through as it is not checked client-side (it is determined by the function registered)

So no, there is no limit, but the parameter amounts and types must be the same as the function that was registered.

And yes, it would be possible to add the attributes with dnlib/cecil.

Cecilia - February 2, 2017, 12:53 am

As always, quality work from Bah.

A lovely edition to Net Seal security.

(Feb 1, 2017 04:31 AM).NETGuard Wrote: [\[CITE\]](#) Hello BahNahNah,

First, thread rated 5/5, can't understand why is there so few replies.

I would like to know if there is a limited amount of parameter allowed or is it only defined by the function itself ?

<https://github.com/BahNahNah/SyncIONSRemoteCode/blob/master/ExampleApplication/Program.cs#L50>

Then, do you think it's possible to grab already-coded functions via dnlib/cecil in a main .exe file, extract them and inject them in the remote code library, adding the correct attribute according to user's netseal ID ? I'd like to do that but before I'd like to know if its theoretically possible.

Again, thank you :)

Are you talking about a Net Seal auto-implementer?

I would like to see something like it, especially with NETGuard.