

Forum:[Hack Forums](#)

Thread: [\[OpenSource\]\[TCP\] Secure Socket Protocol3](#)

DragonHunter - January 18, 2014, 7:07 am

Secure Socket Protocol 3 - a library made in .Net Framework 3.5

Features:

Lightweight TCP/IP IPv4 & IPv6 Networking (1Gbps up/down)

Network Serialization with Protobuf

Builtin encryptions: Hardware Accelerated AES-256, WopEx

Builtin Compressions: QuickLz

Anonimized Authentication (Mazing Handshake) Includes Private Key(s) and Public key

Layer System: Use multiple encryption(s) on-top of each other

Data Integrity Layer (HMAC Support)

SysLogger, A builtin debug logging system for detecting errors and more

Operational Sockets: a "Virtual Socket" inside the real connection also known as "sessions"

Message System: Send serialized messages from/to Server/Client with ease

Set a maximum amount of time a client is able to be connected for

Client & Server are made with Abstract

More writing is coming soon...

This library can be used both in VisualBasic.Net and in C#

There are also examples as server and client, please use different PrivateKeys to keep yourself secure

Github: <https://github.com/AnguisCaptor/SecureSocketProtocol>

PS: I should actually re-do this thread as it looks silly lol

Btw, SSP exists now for +3 years, it came live at juli 18 2012 ?

SubGenius - January 18, 2014, 7:32 am

omg omg omg, I have to try this right now...

Good job DragonHunter.

frontenhack - January 18, 2014, 8:36 am

Great work DragonHunter(as always).

Expect some feedback soon also from me :D

tweety - January 18, 2014, 8:38 am

Bout time you got this back up, glad to see you been working on it lol. Does it come with a example application?

Daedun - January 18, 2014, 5:11 pm

Very happy you got this back up. I used this a lot. I was upset when you took it off codeplex.

Thank you very much :)

Sho?kwave - January 18, 2014, 5:14 pm

Glad to see you have made this project open source again. Incredibly secure.

ALPINE - January 18, 2014, 9:33 pm

Your work is so underrated. You really are intelligent! Every time I see your posts it baffles me! Everything is so intricate and well thought out :) Great job, as always. Feedback will come soonish :)

Ajumma - January 18, 2014, 11:18 pm

I love all your releases, its great, and its not those everyday "rat" or "stealer" programs, they're "unique" and its something I always wanted to try making.

0xBB - January 19, 2014, 1:14 am

(Jan 19, 2014 12:18 AM)iStar Wrote: [CITE](#) I love all your releases, its great, and its not those everyday "rat" or "stealer" programs, they're "unique" and its something I always wanted to try making.

Where do you think HF members reading this will use his lib ? ? . Sadly..

@DerganHunter, Nice one bud. You should release more stuff, not enough of your #swag in this section.

idb. - January 19, 2014, 3:57 am

Absolutely incredible work as per usual. Still the top dog on this site. Excellent job buddy.

DragonHunter - January 19, 2014, 8:48 am

Thanks for the awesome feedback u guys ?

Please report all the bugs/issues (in this thread or in Github) you can find to make it better ?

DragonHunter - January 21, 2014, 12:34 pm

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/00f0567203db9d24f81c96eaac4369b2d010dfde>

Added UDP Support! Keep in mind that the traffic is not yet encrypted and not so many security checks yet
And some other changes

Enjoy ?

DragonHunter - January 24, 2014, 1:39 pm

Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/98b2d8c2de9ad27f2944f51238267f86699e290c>

Added a small Time Synchronisation between server/client, Server takes the lead in having the time sync

Not completely finished yet

DarkPointer - January 25, 2014, 2:32 am

Great job DragonHunter ,, its so inspiring ,, i will test it in my free time ,, i wish you remember me i am Simon we were talking on the auxillium chat program that was made by Banksy ;)

Nextrum - January 25, 2014, 2:54 am

Good to see this back up and running again. You have already made some great changes to it. Keep up the good work! :)

DragonHunter - January 26, 2014, 11:40 am

Good to see people like it ^.^

Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/3333a178cd54bd1d13b8f42f948d15562ef171ed>

Fix for the Deserializing objects which came from a different assembly
Thanks to Banksy for this fix :)

Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/a06f7f92dad0c954cd8d7cc56a997dcd5d3f7a84>

Fixed the stability for the HandShake, it's also more strict now when to receive/send a message
Minor changes to Diffie-Hellman
Channels are working again, this time barely any performance loss!
Some bugfix at receiving packets
And some other minor changes...

LethalNET - January 26, 2014, 12:09 pm

yayyy dragon c: I think I may use this in my project c:

DragonHunter - February 8, 2014, 10:40 am

Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/e6cf73f55bd5f15b44bfe0f0b0405574001363bd>

Updated the key exchange, it's more powerful now
Updated the onException, it will show if it's user or SSP error
Removed some not used code
Updated the UnsafeXor encryption
Minor changes to the RSA class

Note: If you want to feel more secure grab this update!

N3oNight - February 11, 2014, 12:33 am

Grate to see it back again Dragon,
Keep up the grate work..

DragonHunter - February 17, 2014, 2:08 pm

Will commit the changes soon, just making some other security updates

Current changes:

Updated the checksum (data hash) at the messages you're going to send, to make it a little bit harder to modify the messages easily in man-in-the-middle attack
All the handshake messages will get checked with SHA-512bit

Upcoming New features (Both are working but need to check a few things and making sure it's still secure):

SecureStream - Read/Write bytes like you would with e.g. a FileStream or MemoryStream, tested at 150 MegaByte A Second

Requests - Create a request class where messages can be received at and to send back a response, could be handy for a login or a database query you're executing

Banks? - February 17, 2014, 2:10 pm

(Feb 11, 2014 01:33 AM)N3oNight Wrote: [\[CITE\]](#) Grate to see it back again Dragon,

Keep up the grate work..
For the millionth fucking time, it is GREAT.

It is not cheese.

[img]http://pad1.whstatic.com/images/thumb/b/ba/Grate-Cheese-Step-4.jpg/670px-Grate-Cheese-Step-4.jpg[/img]

#Kat - February 17, 2014, 2:23 pm

well jeez so many people using it now yes amazing project good job

DragonHunter - February 23, 2014, 2:12 pm

I'm having the idea in my head for a few days, thinking about how to make it easier for users how to use SSP

Currently how you'd be able to process messages you're receiving is currently by calling the ProcessPayload(...) from IMessage or casting the IMessage to let's say MsgTest

In LiteCode at the other hand you have to share a class at client or server side and at the other side you're able to do GetSharedClass so you can call methods at the remote side

You'd have to create a Interface and at the other side is the SharedClass with the source code

More information about LiteCode: <https://hackforums.net/showthread.php?tid=2655105>

Well it's just a thought, I'll put a poll up at the first post, vote and we'll see :)

DragonHunter - March 8, 2014, 7:51 am

Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/c37ba375807706eaa0d405badb38150aacdce849>

Merged LiteCode and SSP

Fixed quite a few LiteCode bugs regarding to Delegates and Responses

Fixed a key exchange bug

At disconnect you'll get a reason why it disconnected

Only at the channel you're able to use the Messages

And few other bug fixes

DragonHunter - April 2, 2014, 3:16 pm

Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/8ed033e643f758a297380be0fcc72db575f33919>

Merged RootSocket project, BETA

Merged LiteCode project, BETA

LiteCode is now a lot more stable then the original release sometime back

Made some adjustments to the handshake as it's more stable now

RootSocket is optimized for the server to handle a lot of traffic with less cpu usage

Made a few changes to the certificate

Added a few encryption methods

Need todo futher testing whether it's completely stable or not

DragonHunter - June 10, 2014, 1:34 pm

It's been a while since I've posted progress about the project, but no worries I'm still working at it ;)

The last commits for SSP2 as I'm working on a re-design for SSP3

[Key Exchange security update]

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/e6cf73f55bd5f15b44bfe0f0b0405574001363bd>

[LiteCode and Key Exchange fixes]

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/c37ba375807706eaa0d405badb38150aacdce849>

[Handshake fixes, LiteCode, RootSocket and more]

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/8ed033e643f758a297380be0fcc72db575f33919>

[Certificate Visualization]

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/62c56e549fc5a8656e150deaddf939f254cc3a2>

If you're interested in the new re-design, look here,

<https://github.com/AnguisCaptor/SecureSocketProtocol3>

Currently it's estimated to reach 800MBps, so I will be trying to maintain this kind of performance ;) SSP3 is really new so don't use it yet as a project to use, use SSP2 for now if you want to try out my lib ?

DragonHunter - June 28, 2014, 12:18 pm

Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/a4482292b32b65207ad981111a2442c6ebf9dd05>

Replaced the Receiveasync with BeginReceive (ALOT MORE STABLE)

"Random" is now using the server sided seed for randomness, less errors

Updates for LiteCode, some error fixes

Just did this update for people still using SSP2, as I am still working at SSP3

taktaktak - June 30, 2014, 7:57 am

Looks sweet man! Can't wait to use it! +1

DragonHunter - July 13, 2014, 4:38 am

Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/6142f64d990074b820d8a3f3d533e2f1c5b8a19d>

New packet to get a random number provided by the server

LiteCode is now more stable because of the id provided by the server (fixes the most annoying bug in the world, Key already exists in the dictionary)

To accept clients we are now using BeginAccept instead of the Async (more stable)

Rvaveo - July 13, 2014, 9:24 am

(Jul 13, 2014 04:38 AM)DragonHunter Wrote: [\[CITE\]](#) Commit:

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/6142f64d990074b820d8a3f3d533e2f1c5b8a19d>

New packet to get a random number provided by the server

LiteCode is now more stable because of the id provided by the server (fixes the most annoying bug in the world, Key already exists in the dictionary)

To accept clients we are now using BeginAccept instead of the Async (more stable)

Add some documentation in the wiki or source code. It would make life a lot easier. ?

DragonHunter - August 3, 2014, 11:30 am

I wrote a doc for my new Handshake I'm writing, I'm hoping to see people read it and give it a quick

thought to see if it's secure enough ;)

<https://docs.google.com/document/d/1SBUniJcKrHFzGDfBhX3rGU17TDVJi1WF1wQXbiZ1rK0/edit?usp=sharing>

You can also look here at the thread <https://hackforums.net/showthread.php?tid=4375427>

DragonHunter - September 13, 2014, 5:26 am

This project still did not die if you were curious, The last commits:

Improved the performance from ~ 370MBps to ~660MBps (MegaBytes a Second, Not MegaBit)

Fixed quite a few bugs

Created OptimizedPayloadStream which will handle most traffic and make it all nice

Fixed the TestServer Private/Public key locations

My Surface Pro2 can now run SSP3

Fixed a bug when the payload was bigger then 65535

You're now able to create operational sockets

Fixed getBytes in BigInteger thanks to a article on CodeProject

The handshake is having more performance now and even accepts +100MB private/public key files

WopEx bugfix for using the Offset

The new handshake (Maze Handshake) is working and is able to authenticate users, need to finish up the handshake more though but currently it's working

Fixed a bug when sending header data in the message

mundi_86 - September 13, 2014, 6:00 am

great work!!! need to try it :)

thx

Tredude can suck my dick - September 13, 2014, 8:26 am

What is this...? What does it do? I can't figure that out by looking at this thread.

Aeonhack Archivist - September 13, 2014, 9:00 am

(Sep 13, 2014 08:26 AM)Ini Kamoze Wrote: [\[CITE\]](#) What is this...? What does it do? I can't figure that out by looking at this thread.

Hmm....well let's take a look at the title.....Secure Socket Protocol2....I wonder what on Earth that could mean.....maybe that you're a fucking dumbass.

DragonHunter - October 23, 2014, 7:26 am

Currently working on my WopEx encryption and trying to make it more secure by generating new algorithms after every encrypt/decrypt routine is completed

I kept encrypting/decrypting 1337 for 256x and this is the result :P <http://pastebin.com/y9fnWTr6>

Haven't committed in sometime.. oh well will do soon ;)

AngryBirds - October 24, 2014, 4:22 pm

thank you very much dude nice work :)

Nidzgorus - October 25, 2014, 10:17 pm

Very good worl. Nice

Awesome share

Loladmin - October 27, 2014, 12:18 pm

Is SSP3 stable enough to be using in applications or does it need more work?

pr0t0typ3 - October 27, 2014, 4:18 pm

Seems to be a nice project :)

Your work has always been decent ! ?

kenny3514 - October 28, 2014, 4:35 am

Thanks for this thread man it's very usefull :)

DragonHunter - November 1, 2014, 2:41 am

Sorry that I kept u guys waiting so long for the update, just had to make sure it's stable ?

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/f32376c54aa2800a02d1945639ef135cca912094>

SSP3: More Stability, Performance, Security & Bugfixes

Added UserControlS from SSP2 to SSP3

Added new WopEx Instruction (For Loops)

WopEx is having a new mode where it's able to generate a new algorithm after the encrypt / decrypt routine has finished

Fixed failed authentication in Maze Handshake caused by WopEx Encryption Mode

Added Unit Testing

A lot of Updates in the Connection (bugfixes, stability, performance)

umberto94 - November 1, 2014, 12:56 pm

Congratulations, you have been good, keep it up :D

Dangerous Beef - November 9, 2014, 2:38 am

This is awesome, thank you for sharing.

+Bookmark

DragonHunter - November 10, 2014, 9:42 am

SSP3: Performance & Security Update (

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/5a6ad274d3a687c410bdc1f840efff05a5394fd3>)

Updated the performance of WopEx

WopEx is having now a Initial Vector

WopEx will expand the Key/Salt/IV to 2048Bytes

WopEx will use CBC as default (Cipher-Block-Chaining)

Using FastRandom instead of Random, performance improvement

Here you can see the performance improvement from the Unit Tests,

DragonHunter - November 17, 2014, 2:40 pm

Few small updates...

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/fc08510e0eeb76e1c4f25c7dc3d4af6a8aabd1be>

SSPClient is now disposable, a quicker way for making short time connections

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/9822bf0a0f4796667706972899945>

[a81edf439af](#)

Made a small beginning on a SysLogger which will show debug information
Removed MsgConnectionData, no longer needed
Updated ConnectionHeader
At OperationalSocket the onReceiveMessage will finally be triggered

AimBRoT - November 24, 2014, 8:28 am

thank you for this stuff here

<this is not 2 short.>

DragonHunter - December 13, 2014, 3:24 am

It's been a while since I've done a commit, so here it goes again as I've been working on
PirateTunnel futher and a Android Kernel project

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/5f1138e023dac1b472efdd76e91b1cec4067e3fd>

SysLogger - Extra loggings
Certificate validto/from are finally used
Extra packet corruption check
we can now get the Username who is connected
Added OperationalSocket to IMessage

DragonHunter - December 21, 2014, 5:33 am

Just made another security update, you should get it if still using a previous version
When taking a high number of Rounds it will make it harder to bruteforce some stuff
So in this case (High = Better)

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/891b4eca3b934b5ea92934a7aeb832befbed7ce8>

When the user authenticates the server side will now know who authenticated
WopEx is updated and now support Rounds
Maze Handshake is as default using 256 Rounds for WopEx Encryption

Rounds preview: <https://puu.sh/dDA52/12ce98312d.png>

PlayR - December 21, 2014, 5:46 am

GG nice job man.
Thanks for share, nice.

DragonHunter - December 26, 2014, 12:35 pm

Before 2015 I wanted todo this, creating a Algorithm Compiler for my Encryption
This does speed up really well at the moment but it still requires some tweaking

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/4907508318e9e4da4a943d4469035b2ffa458dec>

A beta Algorithm Compiler for WopEx, this is a beta stage of the compiler, more fixes coming soon

Just want to see the compiler ?

<https://github.com/AnguisCaptor/SecureSocketProtocol3/blob/master/SecureSocketProtocol3/Encry>

[ptions/Compiler/AlgorithmCompiler.cs](#)

Encrypting 100MB, with and without the compiler

Samalik - December 26, 2014, 5:58 pm

thanks looks very nice, i'll give it a try.

mechic123 - December 27, 2014, 6:28 am

Amazing work, gonna try this now thanks.

DragonHunter - January 23, 2015, 11:20 am

Some more updates for SSP 3

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/31aecaf7ed05fbf3296ab73464ddabe880fd7095>

Hardware Accelerated AES, can reach upto ~500MBps

Updated the .Net Framework to 3.5

Change between WopEx and AES Encryptions

Change between none or QuickLZ Compression

Updated WopEx's compiler gained a little more performance (Still very Beta)

DragonHunter - January 25, 2015, 5:36 am

I did 2 quick commits for some fixes and I added a small Header Obfuscator for all the packets, just making it a little more "secure"

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/34d40c539f18111983134c2f09de6eda2a02611c>

Quickly fixed the header, error while connecting

Replaced use of Random with FastRandom

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/2be680380cd8fdcdfcbb07bde7554b73679e6efa>

Fixed a small bug for the Obfuscation

Completed the HeaderConfuser

Will probably make a different thread soon...

DragonHunter - February 13, 2015, 12:23 pm

<https://github.com/AnguisCaptor/SecureSocketProtocol3/commit/2c6f898d64e55b7fbc00df95871d128a53f79614>

Added more SysLog messages

Minor changes to Payloadreader

BigInteger can now be serialized by ProtoBuf

Updates to HwAES

User/UserDbInfo can now be serialized by protobuf

Fixed a critical bug where the PublicKey could be always empty (only affects keys which are smaller than 32KB)

Removed the need for Packet Fragments (performance increasment)

You can now choose between WopEx, HwAES to encrypt your traffic (or use them at the same

time)

Compress the traffic using QuickLZ

?

Ajax1337 - February 24, 2015, 9:18 am

Very happy. I used this a lot

Thank you very much :)

DragonHunter - May 14, 2015, 8:11 am

Some more updates...

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/802b4f13bc18a73cb47e09310c6d8197dbf1f054>

Fixed compile, SSPTests is compilable again

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/7999afab202556278fc55f0441d24e5bc8f1a016>

EncryptionTests is Compilable, Fixed compile bug

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/73f851799c2fdcd62737fd10108a38806186e0d6>

Operational Sockets can now disconnect

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/e3d8e6b78a7957353f733065d5bed4e1db020175>

Some LiteCode Patches, Small patches required for LiteCode

More SysLogging

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/bc06dd4cd8797214b1c1d5e7380369e2969f8f1f>

Improved Memory Usage and Security

Removed unused properties from ServerProperties

Fixed a thread-safe issue in CRC32

The buffer used in the Connection is lowered to 8192 bytes and only will increase the usage if required

Ability to change the Maze Handshake Settings

Fixed a issue in the AlgorithmCompiler where it would create too many Modules creating more memory usage

mundi_86 - May 14, 2015, 11:34 pm

great job! thx for the code dude

DragonHunter - December 14, 2015, 7:36 am

A huge list of changes, enjoy :D

LiteCode is probably broken with this update at the moment, will fix that soon

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/db518c2ca5a73b0814df514586b4a277c30bc731>

Updated Protobuf-net to latest version
Removed a few Server/Client properties which are not needed anymore
Syslogger can now show Exceptions
Updated PayloadWriter
Partially removed OptimizedPayloadStream
Fixed an exception in keep-alive bug when disconnected
Fixed MazeHandshake encryption (WopEx was broken)
Created ExtraLayers sample project to implement new Layers
Updated SafeQuickLz
Updated UnsafeQuickLz
Removed "Feature" was not used
Overhaul on the connection class (a huge cleanup)
WopEx now supports MemoryStream (writes the encrypted data to the stream instead of overwriting original)
WopEx: no longer contains a hardcoded IV (This was very bad I have to admit)
WopEx: no longer compiles the instructions when not needed
WopEx: Fixed IV not being used at data less than 8 bytes
WopEx: Fixed an offset issue which could result sometimes in corrupt data
Data Integrity: Added CRC32Layer, HMacLayer, Sha1Layer
LayerSystem: Added AesLayer, QuickLz, WopExLayer,
Connection: Header Obfuscation is currently disabled as it's unstable, will be fixed soon

New:

Layer System (Compression / Encryption Multiple Layers)
Data Integrity Layer (HMAC Support)

Stragic - December 14, 2015, 11:42 pm

Amazing work! Glad to see you updating your thread from the beginning instead of making pointless new threads for every update. ?

TMP_Z3r0D4y - December 16, 2015, 2:40 pm

THIS is what I call a good share.

Try to implement SSL3 with this to test.

SIM-2263652 - December 16, 2015, 5:09 pm

Good work Dragon. And nice to see you still updating this.

LTZkilleur - December 16, 2015, 9:03 pm

Thank you to you the friends it's nice to learn new things

Giratina - January 3, 2016, 2:19 pm

Hi,

Is there a limit to how big the messages are?

Whenever I attempt to send a larger message it just disconnects the client.

I've already tried changing MAX_PACKET_SIZE, but the problem still happens.

How can I solve this? The data I'm trying to send is a byte[] array.

Here's an error message from the server:

Slavoj Žižek - January 4, 2016, 4:26 am

mmmmmmmm, new things - keep up the good work this looks interesting :P

DragonHunter - January 20, 2016, 6:03 am

(Jan 3, 2016 03:19 PM)Giratina Wrote: [\[CITE\]](#) Hi,

Is there a limit to how big the messages are?

Whenever I attempt to send a larger message it just disconnects the client.

I've already tried changing MAX_PACKET_SIZE, but the problem still happens.

How can I solve this? The data I'm trying to send is a byte[] array.

Here's an error message from the server:

Would you mind showing me the code ? send it in a pm if you want

Sending packets bigger than 65KB is never really a good idea, atm the MAX_PACKET_SIZE is at 1MB just for the sake of it

Some more updates and with it I fixed again the LiteCode sample projects

Good to hear you folks like it ?

SSP3 -

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/18d3c1e360924c42046ccd26537c54d119990d14>

A lot of cleanup

Handshake System, implement your own handshakes (SSL, RSA... etc)

LiteCode -

<https://github.com/AnguisCaptor/LiteCode/commit/2bc7ffc98bb7bf05939d7e315047d891ffcd0747>

Fixed the Client / Server sample projects

Slightly updated DynamicClassCreator

(Dec 16, 2015 03:40 PM)TMP_Z3r0D4y Wrote: [\[CITE\]](#) THIS is what I call a good share.

Try to implement SSL3 with this to test. With the new handshake system you could give it a shot implementing SSL3

I already gave it a shot but didn't come really far with SSL, I'll upload what I got

DragonHunter - January 28, 2016, 10:36 am

Some more changes to SSP3, LiteCode is compatible with SSP3 once again

<https://github.com/AnguisCaptor/SecureSocketProtocol/commit/0f2d8b4c4ed3d6619ea2bfbd6d8b82c8d4cfffbd>

Added SerializationAttribute

Add any binary serializer you desire

SendMessage will now return the amount of bytes that was used to send the message

Reduced the Start Buffer Size to 1024 Bytes
Added NetSerializer examples
Added JSon.Net Serializer examples
Added BinaryFormatter Serializer
Editted the client to test the performance of serializers

As an extra, I ran a benchmark to see which serializers were fast

Code ===== Protobuf Performance =====

Packets Send: 2256	147.85MBps	1.18Gbps
Packets Send: 4519	148.31MBps	1.19Gbps
Packets Send: 6757	146.67MBps	1.17Gbps
Packets Send: 9073	151.78MBps	1.21Gbps
Packets Send: 11343	148.76MBps	1.19Gbps
Packets Send: 13608	148.44MBps	1.19Gbps
Packets Send: 15888	149.42MBps	1.2Gbps
Packets Send: 18185	150.53MBps	1.2Gbps
Packets Send: 20453	148.63MBps	1.19Gbps
Packets Send: 22753	150.73MBps	1.21Gbps

===== NetSerializer Performance =====

Packets Send: 1021	66.91MBps	0.54Gbps
Packets Send: 2083	69.6MBps	0.56Gbps
Packets Send: 3139	69.2MBps	0.55Gbps
Packets Send: 4197	69.34MBps	0.55Gbps
Packets Send: 5254	69.27MBps	0.55Gbps
Packets Send: 6308	69.07MBps	0.55Gbps
Packets Send: 7362	69.07MBps	0.55Gbps
Packets Send: 8411	68.75MBps	0.55Gbps
Packets Send: 9467	69.2MBps	0.55Gbps
Packets Send: 10513	68.55MBps	0.55Gbps

===== BinaryFormatter Performance =====

Packets Send: 2052	134.48MBps	1.08Gbps
Packets Send: 4114	135.13MBps	1.08Gbps
Packets Send: 6178	135.26MBps	1.08Gbps
Packets Send: 8251	135.85MBps	1.09Gbps
Packets Send: 10300	134.28MBps	1.07Gbps
Packets Send: 12376	136.05MBps	1.09Gbps
Packets Send: 14453	136.12MBps	1.09Gbps
Packets Send: 16538	136.64MBps	1.09Gbps
Packets Send: 18635	137.43MBps	1.1Gbps
Packets Send: 20733	137.49MBps	1.1Gbps

===== JSON.NET Performance =====

Packets Send: 565	37.03MBps	0.3Gbps
Packets Send: 1310	48.82MBps	0.39Gbps
Packets Send: 2059	49.09MBps	0.39Gbps
Packets Send: 2805	48.89MBps	0.39Gbps
Packets Send: 3551	48.89MBps	0.39Gbps
Packets Send: 4297	48.89MBps	0.39Gbps
Packets Send: 5046	49.09MBps	0.39Gbps
Packets Send: 5789	48.69MBps	0.39Gbps
Packets Send: 6533	48.76MBps	0.39Gbps
Packets Send: 7276	48.69MBps	0.39Gbps

? dont get how u have time ...

adenfr - *January 28, 2016, 3:41 pm*

thanks man i'll see this later