

Blowfish

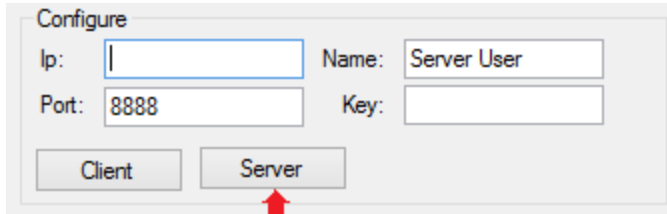
Samuel Lewis

Setup

There is a single executable which can run as either the client or the server.

Server

1. Run the executable
2. Input the desired port
3. Optional: Enter a username
4. Click the server button



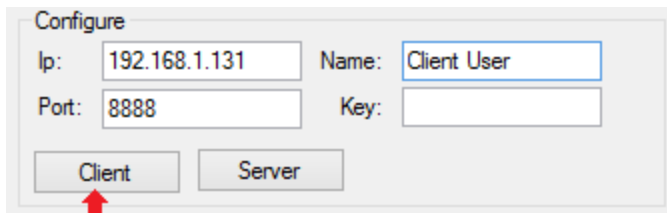
Configure

Ip: Name:

Port: Key:

Client

1. Run the executable
2. Change the port and ip to point at the server
3. Optional: Enter a username
4. Click the client button



Configure

Ip: Name:

Port: Key:

You should now be able to send messages between the two. At any time a blowfish key can be entered (HEX string) both the client and the server must have matching keys to be able to decrypt each others messages. If the key field is left blank no encryption is done.

WireShark

Note: WireShark cannot be used to sniff local traffic in Windows.

1. Open WireShark on either machine
2. Start capture with the selecting the correct network interface
3. For the filter use :
 - a. `(ip.src == <remote ip> && ip.dest == <local ip>) || (ip.src == <local ip> && ip.dst == <remote ip>)`
 - b. ipv6 may be need instead of ip

Interception Test

The message sent between the two chats is a JSON string containing three elements: name of the user, whether the message body was encrypted and the message body.

Example:

```
{
  "encrypted":false,
  "text":"This is the body of the message",
  "user":"Username"
}
```

Encrypted is spelled encrytped in the follow test results due to a spelling error in the software at the time these screenshots were taken.

Clear Text

0000	e0 06 e6 ab 3e bf 00 08	54 a1 53 e7 86 dd 60 00>... T.S....`.
0010	00 00 00 7e 06 80 fe 80	00 00 00 00 00 00 29 17	...~.....
0020	e6 a0 3f e4 11 e6 fe 80	00 00 00 00 00 00 49 e9	..?.....
0030	77 e6 68 5b d6 dc ee 30	22 6c 36 c2 b7 22 12 4b	w.h[...0 "l6..".K
0040	96 bf 50 18 01 02 cd 78	00 00 7b 22 65 6e 63 79	..P....x ..{"ency
0050	72 70 74 65 64 22 3a 66	61 6c 73 65 2c 22 74 65	rpted":f alse,"te
0060	78 74 22 3a 22 54 68 69	73 20 69 73 20 61 6e 20	xt":"Thi s is an
0070	75 6e 65 6e 63 72 79 70	74 65 64 20 6d 65 73 73	unencryp ted mess
0080	61 67 65 20 66 72 6f 6d	20 74 68 65 20 63 6c 69	age from the cli
0090	65 6e 74 20 74 6f 20 74	68 65 20 73 65 72 76 65	ent to t he serve
00a0	72 2e 22 2c 22 75 73 65	72 22 3a 22 43 6c 69 65	r.","use r":"clie
00b0	6e 74 22 7d		nt"} }

The text of the message can be clearly read as "This is an unencrypted message from the client to the server."

Ciphertext

0000	00 08 54 a1 53 e7 e0 06	e6 ab 3e bf 86 dd 60 00	..T.S... ..>...`.
0010	00 00 00 d0 06 80 fe 80	00 00 00 00 00 00 49 e9
0020	77 e6 68 5b d6 dc fe 80	00 00 00 00 00 00 29 17	w.h[....
0030	e6 a0 3f e4 11 e6 22 6c	ee 30 12 4b 96 bf 36 c2	..?..."] .0.K..6.
0040	b7 8c 50 18 00 fe 9f 85	00 00 7b 22 65 6e 63 79	..P..... ..{"ency
0050	72 70 74 65 64 22 3a 74	72 75 65 2c 22 74 65 78	rpted":t rue,"tex
0060	74 22 3a 22 63 33 30 63	36 64 65 36 34 65 36 64	t":"c30c 6de64e6d
0070	32 61 30 65 33 30 35 62	36 66 33 66 61 35 30 64	2a0e305b 6f3fa50d
0080	61 33 36 30 30 65 31 31	39 62 30 39 33 64 66 37	a3600e11 9b093df7
0090	63 65 66 62 36 65 32 32	32 61 65 65 37 65 31 38	cefb6e22 2aee7e18
00a0	66 38 65 31 32 66 39 33	36 34 63 64 63 65 30 64	f8e12f93 64cdce0d
00b0	31 63 65 33 37 66 62 66	30 35 35 32 62 61 31 62	1ce37fbf 0552ba1b
00c0	35 66 63 32 30 31 62 61	34 66 64 66 64 33 64 37	5fc201ba 4fdfd3d7
00d0	61 62 37 66 37 35 36 33	30 34 36 62 63 37 66 31	ab7f7563 046bc7f1
00e0	30 34 37 38 38 35 31 32	64 66 36 63 66 34 38 37	04788512 df6cf487
00f0	31 38 62 35 22 2c 22 75	73 65 72 22 3a 22 53 65	18b5","u ser":"Se
0100	72 76 65 72 22 7d		rver"} }

This plain text was "This is an encrypted message from the server to the client." it was encrypted using blowfish with 'FF' as the key.

Below is a copy of the JSON that the client received showing that it is the same as what WireShark sees:

```
Received:
{"encrypted":true,"text":"c30c6de64e6d2a
0e305b6f3fa50da3600e119b093df7cefb6e
222aee7e18f8e12f9364cdce0d1ce37bf05
52ba1b5fc201ba4dfd3d7ab77563046bc7
f104788512df6cf48718b5","user":"Server"
}
```

Encryption Performance

To measure the time it took to measure encrypting and decrypting the message I used the C# Stopwatch class. The code for both encryption and decryption look approximately like this:

```
var sw = new Stopwatch();
sw.Start();
text = fish.Decrypt_CBC(text);
sw.Stop();
var executionTime = sw.ElapsedTicks;
```

To measure this I sent messages of various sizes and looked at both the encryption and decryption time. The time was very inconsistent between messages of similar lengths but what was consistent was that decryption took longer than encryption. On average decryption took approximately 10-15% longer. However, I don't know if this is because of how Blowfish works or just a limitation of the Blowfish implementation I used.