

Forum:[Hack Forums](#)

Thread: [Making it harder for slaves to read packets](#)

PapaBeans - October 3, 2014, 1:13 pm

NOTE THIS IS A MORE BASIC TUTORIAL! NOTHING IS INVISIBLE

Hello guys,

Blackhatting has been growing greatly over the years. More and more RATs, botnets, etc are being released. Not all of your slaves will be nitwits who have no clue how to trace you, and if they find you they could DDoS you. Well I'm about to teach you how to help hide your packets a bit.

When you use a program like WireShark to read packets (and are ratted), you often see text like "openProc" or "sendMessage." Then from there you can just block there IP address. But it doesn't have to be like this. Lets start from the beginning.

Make an Enum and call it whatever you want, I called it messages.

Now lets modify (or create) your gotMessage() and sendMessage() Subs.

Now when your stub gets a message, wireshark would read "1000." It would take a lot more time to try and find multiple packets with "1000" then OPENPROC in it. Note that I use the "|" to help seperate my messages when I want to send multiple. You do not need to do this.

Note that client.Client..... is a TCPClient that I named client.

I hope that you learned things. I feel like I'm forgetting something, so this may be edited in the near future. Have fun, and happy hacking. ?

TheBarret - October 3, 2014, 1:17 pm

Dunno about nimwits, but i can clearly de-spoof that package, since your not using any form of packet structure except the header (type) the message is clearly showing then.

I would go for a packet that has a specific header with id, length,data, as bytes.
Ontop i would def go with a modified encryption system, that might be ssl even.

Cornflower Blue ???-?— - October 3, 2014, 1:19 pm

Why would you even send packets that look like this "2361|You have been ratted!".
Look into serialization, or send the raw bytes.

PapaBeans - October 3, 2014, 1:20 pm

(Oct 3, 2014 01:17 PM)TheBarret Wrote: [\[CITE\]](#) Dunnow about nimwits, but i can clearly de-spoof that package, since your not using any form of packet structure except the header (type) the message is clearly showing then.

I would go for a packet that has a specific header with id, length,data, as bytes.

Ontop i would def go with a modified encryption system, that might be ssl even.

Thanks for the feedback, how would you go about changing the packet structure? And I have encryption in my client, just didn't want to share on this tutorial. Makes it read more like "j38ahsd81Hs82HsaR".

(Oct 3, 2014 01:19 PM)Cornflower Blue ???-?— Wrote: [\[CITE\]](#) Why would you even send packets that look like this "2361|You have been ratted!".

Look into serialization, or send the raw bytes.

lOl! It doesn't look like that. It would look like "3123|". That is all. Then once the client receives "3123" it would do a certain task.

sanderele1 - October 3, 2014, 1:26 pm

As TheBarret said, the rest of the message could still be read in plaintext, so you should also implement encryption. He did also say that.

Now from experience I can tell you that if you need a quite secure system, use a AES and RSA encryption exchange.

Make the server send it's public key (AES) to the client. Which will then encrypt it's RSA key with the public key. Then it will send the encrypted key back to the server.

Then all you have to do is decrypt the recieved key and boom. Both your client and server can now talk via RSA.

Also a delimiter system (saying like "3123|param1|param2") is quite obsolete, and not very good.

I'd reccomend you constructing a proper packet.

An example of this would be: (Packet Size UInt16 or whatever max packet size you would like)(Encryption Byte, An enumeration that allows for 256 methods of encryption)(Message ID, UInt16. Like what you were doing above, identifying what you are sending)

So with this you can send any byte array you can think of. Even the "|" character. Also this should require less network bandwidth.

Since instead of sending "12345" which might be 5 or 10 bytes (depends on your encoding), it would only be 2.

Now the advantages are something you only gain of you use an UInt16 or whatever number and comvert it to a byte array. Don't send that stuff as a string

Hopefully you understood something :D

Have a nice day.

tweety - October 3, 2014, 1:27 pm

Seems like you need to do a little more research before you actually do anything. The method you're using is from 2004, string splitting.

PapaBeans - October 3, 2014, 1:28 pm

(Oct 3, 2014 01:27 PM)Tweety. Wrote: [\[CITE\]](#) Seems like you need to do a little more research before you actually do anything. The method you're using is from 2004, string splitting.

I'm simply trying to help people out that send messages raw. This isn't the full method I use.

(Oct 3, 2014 01:26 PM)sanderele1 Wrote: [\[CITE\]](#) As TheBarret said, the rest of the message could still be read in plaintext, so you should also implement encryption. He did also say that.

Now from experience I can tell you that if you need a quite secure system, use a AES and RSA encryption exchange.

Make the server send it's public key (AES) to the client. Which will then encrypt it's RSA key with the public key. Then it will send the encrypted key back to the server.

Then all you have to do is decrypt the recieved key and boom. Both your client and server can now talk via RSA.Wat. I made something like this, but it needs a third computer, helps protect my IP address. Ill deffinetally look into this, probably tons easier.

Bonfire - *October 3, 2014, 1:31 pm*

(Oct 3, 2014 01:28 PM)PapaBeans Wrote: [\[CITE\]](#) I'm simply trying to help people out that send messages raw. This isn't the full method I use.

-snip-

Well why not post the full method? Also, anyone sending messages "raw" shouldn't be creating "RATs"

Cornflower Blue ???-? — *October 3, 2014, 2:19 pm*

<http://pastebin.com/Jcywvekk>

That is a example of how I would do it, it's also the same concept as Aeon uses for his NanoCore. Just custom serialization, mine is very basic but should do the job.