

Forum:[Hack Forums](#)

Thread: [Server Side Game Hacking Question?](#)

District - August 19, 2020, 9:42 pm

Imagine yourself playing a standard MMORPG game, and think about the process of trading with another player. You'd have to click them and then click send trade offer or whatever before the game would receive that information and make it happen. I've seen people create cheat programs that allow them to do this kind of thing without the actual interaction, so instead you'd do something like click a button on your cheat and it'd just send a trade offer without you having to do the other actions like clicking the player yourself, and then clicking the trade option. (Maybe not specifically with trading but I'm just using this story to get my question across)

What should I look into if I wanted to do these kinds of things? I know of a game with no real anti-cheat that I want to practice on. I'm a beginner with C# but I want something like this to be the next project that I work towards. I've already made a calculator and a client sided cheat menu for editing basic things like health/ammo addresses etc.

Pawnhub - August 20, 2020, 12:25 am

I would look into how they might be sending this information, Wireshark would provide you the raw TCP/UDP packet information leaving/entering your computer.

If it's through HTTP something like Fiddler could do it. It'd just show you the data sent over HTTP/HTTPS.

Then comes mimicking. So if it's done through HTTP, you'll have that info in Fiddler then you can resend those packets in a script via request's with postData and auth headers but FYI this only works if it's done through UDP as HTTP is a TCP based protocol. You'll have to then go the Wireshark route and mimic those packets and send them to the server.

Lets say you buy an item. Well you've sent some packet or HTTP message saying you bought the item, then the server rewards you with said item.

You just gotta figure out 1. if you can find the packets & 2. if you are able to manipulate them and the server gracefully accepts them. Some games for instance have a middle man in the transaction process which makes these spoof attacks useless. You send a request to server 1. Server 1 then makes a transaction with server 2. Server 2 then sends back a success to server 1 then server 1 says here's your item.

ucsandaasu - August 21, 2020, 12:00 am

its called packet manipulation

District - August 22, 2020, 3:59 pm

(Aug 21, 2020 12:00 AM)ucsandaasu Wrote: [\[CITE\]](#) its called packet manipulation
Any more info? Or possibly direct me to a library etc?

UniveX - August 24, 2020, 3:16 am

(Aug 22, 2020 03:59 PM)District Wrote: [\[CITE\]](#) Any more info? Or possibly direct me to a library etc?
Hey,

basically, for capturing other traffic than HTTP/S, you'll need something like Wireshark to analyze TCP/UDP communication. For capturing this stuff in .NET you can use Pcap.Net.

Usually, in newer applications/games you'll face encrypted traffic, where you need to find out how the traffic is encrypted, usually not so simple with packet analyzing, for most games the recommended way is to hook a function in game that passes the data unencrypted to their own function that later encrypts the traffic.

Then you have three (maybe more) ways to do this;

1) Write a emulator or a packet bot, by emulating all of the packets/requests required.

2) MITM - You will have your own "middle-man" server that works like:

Game sends packet to your MITM -> Your MITM sends it to server, but you can before sending the packet to the server change it and there do your required stuff.

MITM will catch also SERVER -> CLIENT packets, so you can also manipulate packets sent to your client.

3) Hook game functions that take care of sending packets and manipulate those packets in there.

For this you'll need TCPClient or UDP Client in .NET (as for packet manipulation), or an internal (preferably) dll that will interact with the packets within the game itself.

Hopefully this was helpful a bit & have a nice day,
UniveX