

Forum:[Hack Forums](#)

Thread: [packet drop self](#)

TonicBox - June 2, 2021, 2:56 am

Java's not really my forte so apologies if there's an obvious solution to this I'm missing. But I'm looking for a way in Java to packet drop itself. Context is it's a plugin for a game. Goal I'm trying to achieve is to write in a function bound to a hotkey to toggle on/off packet dropping.

Been kinda hard to find much material on how to go about this though so any pointers in the right direction would be most appreciated.

GenericSkid - June 2, 2021, 10:09 pm

I expect that you only want to drop packets that the game is responsible for, and not all of your own network traffic on the local machine. With this in mind, it depends on the game and what messaging protocol they use.

Lets say the game is Minecraft. There's a popular new tech solution being used to write mods called [Mixins](#). What most people will do is write a mixin that allows them to inject code into the networking logic that effectively just calls "return;" You could write them in such a way that it only does this when a key is pressed, or if the packet only contains some sort of specific data. Now, Minecraft has things like [FabricMC](#) which makes this rather complicated task relatively easy. The same thing can be done for any Java application with enough effort, just Minecraft has most of the development cycle already made for you.

Continuing with the Minecraft example, lets say setting up that developer workspace is too much. You can still edit the compiled game classes with [a bytecode editor](#). It would be effectively the same thing as described above, but you'd hard-code that logic into the game by inserting additional bytecode. You'd navigate to the networking class, then insert some logic like:

```
Code void sendPacket(Packet p) {  
    if (isMyKeyDown()) {  
        return;  
    }  
    ...  
}
```

Where "isMyKeyDown()" would be defined somewhere that has access to the keyboard states. Minecraft uses LWJGL 3, so there's callbacks that it registers for handling user inputs. I imagine most other games follow a similar design, so you'd have to hook some logic in there to track if the key you want is pressed. It's a hacky approach but should do the job.

Another way, if the game easily provides the networking libraries it uses in a way that is easy to swap out, you can clone the source of it from GitHub assuming it's something like Netty, then find where packets are sent, and add your keyboard hook in there. Compile to jar, swap out with what the game provides. Another hacky approach but also works.

Having a system level packet hook would be out of scope for Java really, and probably a bit unwieldy to make it automatically know what packets you want dropped.

TonicBox - July 2, 2021, 4:27 pm

(Jun 2, 2021 10:09 PM)GenericSkid Wrote: [\[CITE\]](#) I expect that you only want to drop packets that the game is responsible for, and not all of your own network traffic on the local machine. With this in

mind, it depends on the game and what messaging protocol they use.

Yeah preferably tho currently I use a 3rd party program to drop, so dropping all would work fine too.
Game I use uses mixins aswell, and thats what I'm currently looking at