

Forum:[Hack Forums](#)

Thread: [Forge Client or Normal Client?](#)

I Like Gold - August 12, 2017, 5:56 pm

Hello! Which type of hacked client would you guys prefer to have (Forge or Normal), and why? (Please tell why)

Forge Client or a standard client with your versions folder or whatever the fuck? Thanks for your opinions ??

marcusant - August 12, 2017, 6:33 pm

You could always have a client that both works with and without Forge, like Iridium

I Like Gold - August 12, 2017, 6:35 pm

(Aug 12, 2017 06:33 PM)marcusant Wrote: [\[CITE\]](#) You could always have a client that both works with and without Forge, like Iridium

That sounds pretty cool, but I don't know how Forge works and just started coding with it a bit today.

Do you develop 2 different Iridiums? One for Forge and the other for normal MC?

marcusant - August 12, 2017, 7:29 pm

(Aug 12, 2017 06:35 PM)I Like Gold Wrote: [\[CITE\]](#) That sounds pretty cool, but I don't know how Forge works and just started coding with it a bit today.

Do you develop 2 different Iridiums? One for Forge and the other for normal MC?

Nope. It uses runtime bytecode injection so it works no matter what mods are installed, including optifine and forge. I don't edit Minecraft directly at all.

I Like Gold - August 12, 2017, 7:45 pm

(Aug 12, 2017 07:29 PM)marcusant Wrote: [\[CITE\]](#) Nope. It uses runtime bytecode injection so it works no matter what mods are installed, including optifine and forge. I don't edit Minecraft directly at all.

That sounds like some complex stuff, yet it sounds fun. Are you taking the Java Instrumentation approach?

marcusant - August 12, 2017, 7:52 pm

(Aug 12, 2017 07:45 PM)I Like Gold Wrote: [\[CITE\]](#) That sounds like some complex stuff, yet it sounds fun. Are you taking the Java Instrumentation approach?

Yes. I used to (in Iridium 4) attach to a running instance with the com.sun.tools.attach stuff (so the client could be injected on the fly to an already running game). Here's a video of it:

<https://streamable.com/nzqgqm> . I haven't decided on a way to deploy the payload for Iridium5 yet -- I probably won't use attach again since it didn't really provide much benefit to just loading into the classpath from the beginning.

I Like Gold - August 12, 2017, 8:16 pm

(Aug 12, 2017 07:52 PM)marcusant Wrote: [\[CITE\]](#) Yes. I used to (in Iridium 4) attach to a running instance with the com.sun.tools.attach stuff (so the client could be injected on the fly to an already running game). Here's a video of it: <https://streamable.com/nzqgqm> . I haven't decided on a way to

deploy the payload for Iridium5 yet -- I probably won't use attach again since it didn't really provide much benefit to just loading into the classpath from the beginning.

That inject button thing looks sick! Don't you have to program in strings in order to inject, though?

MadZ - August 12, 2017, 8:38 pm

(Aug 12, 2017 08:16 PM)I Like Gold Wrote: [\[CITE\]](#) That inject button thing looks sick! Don't you have to program in strings in order to inject, though?

You're most likely thinking of [javassist](#), which lets you insert your code as you would normally program it in in the form of strings.

You don't have to use this library though as you can just use [asm](#) (which iirc javassist is based off of, but the difference is that compiler that converts strings to bytecode).

I Like Gold - August 12, 2017, 8:51 pm

(Aug 12, 2017 08:38 PM)MadZ Wrote: [\[CITE\]](#) You're most likely thinking of [javassist](#), which lets you insert your code as you would normally program it in in the form of strings.

You don't have to use this library though as you can just use [asm](#) (which iirc javassist is based off of, but the difference is that compiler that converts strings to bytecode).

Awesome! Thank you! ASM seems harder to use than Javassist (probably because you deal with actual bytecode), so I think I'll try Javassist.

My plan would be to create maybe a Mapper utility class, and place inside of that class a method which converts the MCP Mappings you input (for ease and readability of code), into Notch Mappings for the minecraft jar file you're dealing with. Then you could probably inject classes without naming errors.

However, when you inject your String code (for example, event calling code) into the minecraft class files, does it automatically inject your normal java code along with it?

For example, do I need to attach an agent jar? And.... oh damn this is getting confusing

MadZ - August 12, 2017, 8:58 pm

(Aug 12, 2017 08:51 PM)I Like Gold Wrote: [\[CITE\]](#) However, when you inject your String code (for example, event calling code) into the minecraft class files, does it automatically inject your normal java code along with it?

For example, do I need to attach an agent jar? And.... oh damn this is getting confusing

Although I am very inexperienced with instrumentation and the whole injection thing, the agent jar is what you are injecting into Minecraft; so yes, you need to attach the "agent jar."

Keep in mind the primary feature that makes a jar an "agent jar" is the agentmain method (which is what you're dealing with when injecting into already loaded classes). This will then call your transformers etc, which can all be put into a single jar, along with your standard client related packages.

I can't really explain it that well because I have never taken the time to really learn it that well or even make a client with any of it. I've only started reading about java agents and javassist (as I have little experience with bytecode). Hope someone else can properly answer your question.

Good luck with your project and the same to Marcusant.

I Like Gold - August 12, 2017, 9:08 pm

(Aug 12, 2017 08:58 PM)MadZ Wrote: [\[CITE\]](#) Although I am very inexperienced with instrumentation and the whole injection thing, the agent jar is what you are injecting into Minecraft; so yes, you need to attach the "agent jar."

Keep in mind the primary feature that makes a jar an "agent jar" is the agentmain method (which is what you're dealing with when injecting into already loaded classes). This will then call your transformers etc, which can all be put into a single jar, along with your standard client related packages.

I can't really explain it that well because I have never taken the time to really learn it that well or even make a client with any of it. I've only started reading about java agents and javassist (as I have little experience with bytecode). Hope someone else can properly answer your question.

Good luck with your project and the same to Marcusant.

Thanks a lot, MadZ! You actually made me think a lot better about how it works, and taught me a bit.

So, you're saying that if I create an Agent JAR with all of my class transformers, agentmain method, and minecraft hacked client code packed inside of it, I simply attach the agent jar file to the running Minecraft, and then it will technically be on its classpath?

MadZ - August 12, 2017, 9:25 pm

(Aug 12, 2017 09:08 PM)I Like Gold Wrote: [\[CITE\]](#) Thanks a lot, MadZ! You actually made me think a lot better about how it works, and taught me a bit.

So, you're saying that if I create an Agent JAR with all of my class transformers, agentmain method, and minecraft hacked client code packed inside of it, I simply attach the agent jar file to the running Minecraft, and then it will technically be on its classpath?

Yes. If you have any additional dependencies you can point to them in your manifest Class-Path attribute.

The method used to actually attach the agent can also be in the same jar as the agent (someone correct me if I'm wrong). Just call the method in the entry point/main-class.

Again, I honestly don't know much about all of this, just recalling the things I've read in the past few days.

marcusant - August 12, 2017, 9:25 pm

(Aug 12, 2017 09:08 PM)I Like Gold Wrote: [\[CITE\]](#) Thanks a lot, MadZ! You actually made me think a lot better about how it works, and taught me a bit.

So, you're saying that if I create an Agent JAR with all of my class transformers, agentmain method, and minecraft hacked client code packed inside of it, I simply attach the agent jar file to the running Minecraft, and then it will technically be on its classpath?

If you don't modify MC's classpath on launch you'll probably have to load your stuff into the classpath from the agent (i.e. with URLClassLoader) since IIRC Attach only loads in the specific class list you specify. Basically IIRC I made my attach load a minimal class or two that dealt with loading and executing the others. I don't remember too many specifics since it was a while ago.

MadZ - August 12, 2017, 9:36 pm

(Aug 12, 2017 09:25 PM)marcusant Wrote: [\[CITE\]](#) If you don't modify MC's classpath on launch you'll probably have to load your stuff into the classpath from the agent (i.e. with URLClassLoader) since IIRC Attach only loads in the specific class list you specify. Basically IIRC I made my attach load a

minimal class or two that dealt with loading and executing the others. I don't remember too many specifics since it was a while ago.

Ahh...

<https://github.com/Xyene/ASM-Late-Bind-Agent/blob/master/src/main/java/tk/ivybits/agent/AgentLoader.java#L38> (basic library to attach to jvm proc)

I suppose you would have to pass those additional classes to the resources argument.

Most of the writeups I've read were based on "premain" javaagents, which are loaded with the javaagent argument at launch, which didn't seem to have this requirement. I suppose using agentmain/attaching to already loaded classes you would have to specifically load relevant classes.

marcusant - August 12, 2017, 9:39 pm

(Aug 12, 2017 09:36 PM)MadZ Wrote: [\[CITE\]](#) Ahh...

<https://github.com/Xyene/ASM-Late-Bind-Agent/blob/master/src/main/java/tk/ivybits/agent/AgentLoader.java#L38> (basic library to attach to jvm proc)

I suppose you would have to pass those additional classes to the resources argument.

Most of the writeups I've read were based on "premain" javaagents, which are loaded with the javaagent argument at launch, which didn't seem to have this requirement. I suppose using agentmain/attaching to already loaded classes you would have to specifically load relevant classes.

Yep. Not too bad if you use a classloader from within the few classes you manually specify. Never did any of the premain stuff and didn't use any libs that weren't part of the JDK (although that would probably be smarter and less work).

I Like Gold - August 12, 2017, 9:45 pm

(Aug 12, 2017 09:39 PM)marcusant Wrote: [\[CITE\]](#) Yep. Not too bad if you use a classloader from within the few classes you manually specify. Never did any of the premain stuff and didn't use any libs that weren't part of the JDK (although that would probably be smarter and less work).

Thanks! So, you're saying, rather than manually add it to MC's classpath, I just need to load the class?

i.e. I just need to, for example, in my agentmain method, do something like MyClass.class.getClass(); or some other advanced way to simply get the classloader to know about it? Nothing else?

JasperRed - May 8, 2018, 3:41 am

forge client is much better in my opinion

Extanity - June 11, 2018, 7:39 pm

(Aug 12, 2017 05:56 PM)I Like Gold Wrote: [\[CITE\]](#) Hello! Which type of hacked client would you guys prefer to have (Forge or Normal), and why? (Please tell why)

Forge Client or a standard client with your versions folder or whatever the fuck? Thanks for your opinions ??

Does Irudium have optifine?

DandyViewerHF - June 24, 2018, 1:07 pm

(May 8, 2018 03:41 AM)BabySeal Wrote: [\[CITE\]](#) forge client is much better in my opinion

Dude I created a new account on HF for once in like a year again, and here it is, my old account "I Like Gold." Such a small world

(Jun 11, 2018 07:39 PM)Extanity Wrote: [\[CITE\]](#) Does Irudium have optifine?

Yes, Iridium has optifine. I'm the Gold dude, btw