

Forum:[Hack Forums](#)

Thread: [Do third party modules give you away? \(Python\)](#)

Comrade Loading - January 9, 2019, 6:12 pm

Hello, so I decided to learn Python because of all the good things I hear about it. I heard its good for writing exploitation tools, and other various things that fit under the category of "hacking". But, I have a question based around third party modules that are community made. For example: I made a virus (I don't know, just something that would hurt your computer) and I used third party modules to help perform specific tasks, would an anti-virus declare my program a virus based on the modules used in the program? like the cryptodome module, its used for encrypting files, which could probably be used in Ransomware. Would the Anti-virus pick up on that? or does it go about it another way? the reason I ask, is because I plan on creating my own pen-testing tools, and I'm afraid that modules which were coded by someone else, and put on the internet publicly will ruin my chance to actually to exploit something. If anything I said was confusing, let me know and I will try to elaborate better. Thanks for taking your time to read, I appreciate it.

One more thing (Sort of off topic) I was reading this comment on a YouTube video, and it ended up in an argument between two people (Was funny as hell tbh). They were arguing over programming and hacking, and who was better essentially lol. One of them decides to say something like "If you're so good, write a hacking tool without any modules!". Do you think what he said is completely absurd, and stupid? I personally do, because I don't see how using modules makes you a bad programmer. The argument went on and on, I had to click view replies well over 8 times lol.

Faith Connors - January 9, 2019, 9:45 pm

for the first part, you're writing it in fucking python, you're going to get a detection anyway just because it's all fucken plaintext (and py2exe isn't any better.)

don't write actual malware in python, first of all. script to automate shit (fuzzing, blind sql, other generic automated stuff) should generally only be used on your computer and doing something remotely. so i dont really know what you're really asking here. just add yourself an exception in your firewall/av for the script and runtime?

as for part two, using a library or framework doesn't necessarily make you a worse programmer, unless you're only using the framework or library because you don't know how to do something yourself.

lets say you're writing something in Go. a hello world app or something. it'll look like this:

```
Code package main
import "fmt"
func main() {
    fmt.Println("Hello World!")
}
```

seems simple enough at first glance, but it's importing the whole 'fmt' library. wouldn't your final output be a thousand times more optimized if you only imported the function you needed? better yet, what if you wrote it yourself and made it better?

(note, this is a really bad example because this simply isn't how go does library imports, it only uses the import statement as a reference for where to find shit.)

as a stark contrast, java lets you either import an entire library or package to use all its modules, methods, classes, etc. but you can also import specific parts as well to keep your total output filesize down (because java is JIT compiled)

and as a final contrast, python not only adds all the functions/classes/methods from a library to the code as it's being executed, but the py2exe option packages the entire python runtime with it. once more, just making everything useless.

so although there's nothing inherently 'wrong' with using a library or framework, if you can do it yourself and do it creatively, then you're better off doing that (or writing your own libraries; start learning c.)