



## Reverse Engineering a Myth in Software Industry

*Neelanjan Manna*

Metapyxl

### ABSTRACT

Reverse engineering, also referred to as back engineering or backward engineering, is a process or method used to attempt to understand through deductive reasoning how a previously created device, process, system, or piece of software accomplishes a task with little to no understanding of how it does so. In essence, it is the act of examining a system's inner workings in order to copy or improve it. The information gathered through reverse engineering can assist with learning how something works, re-purposing obsolete objects, doing security assessments, and other tasks, depending on the system under review and the technologies used.

### Introduction

The term "subject system" refers to the finished piece of software development. In 1990, the Institute of Electrical and Electronics Engineers (IEEE) defined (software) reverse engineering (SRE) as "the process of analysing a subject system to identify the system's components and their interrelationships and to create representations of the system in another form or at a higher level of abstraction." Reverse engineering is a process of investigation only; it does not involve re-engineering or restructuring, which would involve changing the software system in question. Reverse engineering can be carried out at any point in the product cycle, not just at the functional conclusion.

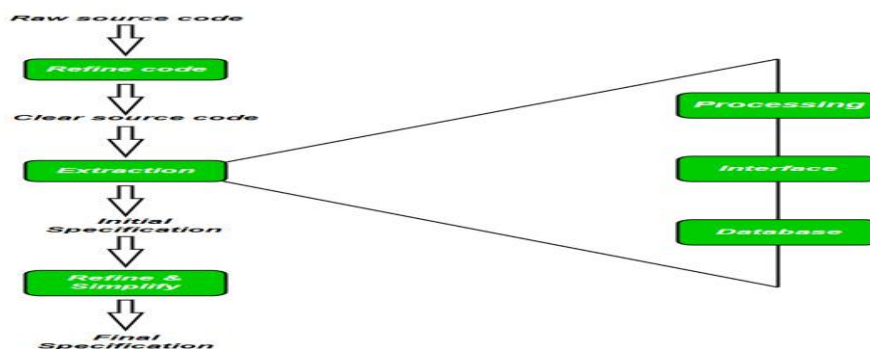
### Methodology of Reverse Engineering

Recovering a product's design, requirements, and functionality from an analysis of its code is known as software reverse engineering. It creates a programme database and uses this to produce information.

Reverse engineering is used to create the necessary documentation for a legacy system and to make maintenance tasks easier by making a system easier to comprehend.

Goals for reverse engineering

- Adapt to complexity
- recuperate lost data.
- Find negative effects.
- higher abstraction synthesis.
- Encourage reuse.



## Tools for Reverse Engineering

Reverse engineering needs to be aided by automated tools because it would take a lot of time and labour to complete manually. The following are some tools:

- A graphical navigator for software and web repositories, as well as a set of tools for reverse engineering, are called CIAO and CIA.
- Rigi: A tool for understanding software visually.
- Bunch is a tool for software clustering and modularization.
- GEN++: An application generator to facilitate the creation of C++ language analysis tools.
- Software Bookshelf resources for extracting and displaying software architecture.

### *How to totally eradicate reverse engineering in software industry*

Its time to fight fire with fire so we must use crypters . A crypter is a sort of software that can alter, obfuscate, and encrypt malware to make it more difficult for security systems to recognise it. Cybercriminals utilise it to construct malware that can avoid security tools by posing as a useful programme until it is installed.

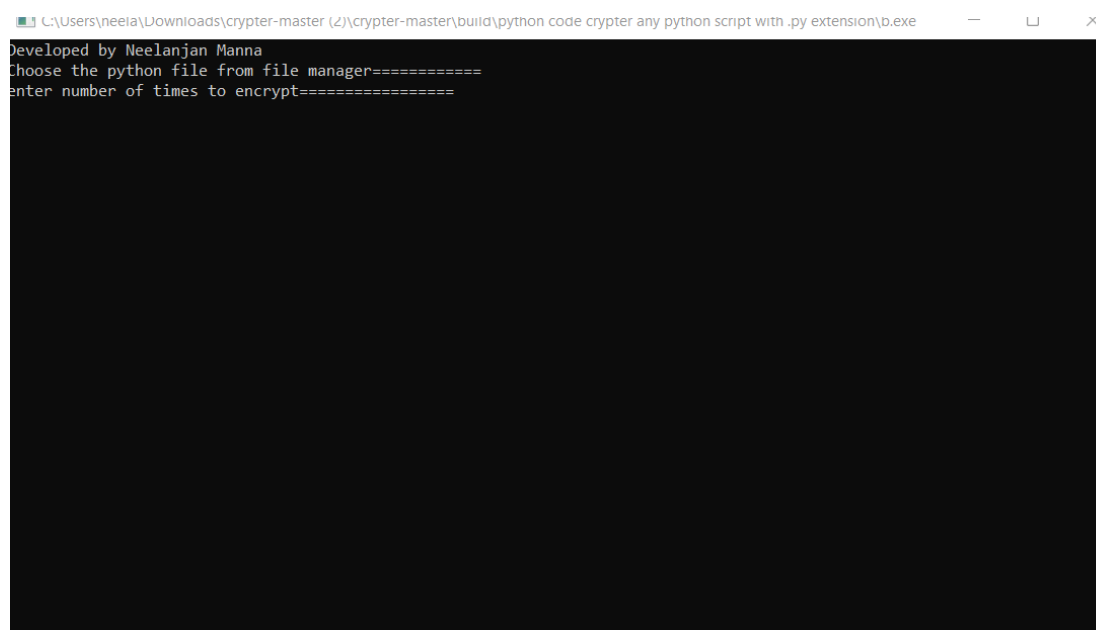
### *Varieties of crypters*

A crypter is a programme that may be used to both encrypt and decrypt harmful code. Static/statistical or polymorphic crypters can be identified based on the kind of stub they employ.

Each encrypted file is made unique by static/statistical crypters using various stubs. Since each client has its own stub, malevolent actors will find it simpler to modify or, in hacking lingo, "clean" a stub after it has been discovered by security tools.

Obfuscation is what crypters refer to as their most basic approach. Obfuscation: Malware's Best Friend is a more thorough study on that. In scripts like javascripts and vbscripts, obfuscation is also frequently utilised. However, most of the time it's not that difficult to go around or decipher these. More sophisticated techniques employ true encryption. Most crypters provide users a variety of other options in addition to file encryption, making it as difficult for security companies to find the concealed executable as feasible. For certain packers, the same is true.

### *Development by an independent researcher*



```
C:\Users\neela\Downloads\crypter-master (2)\crypter-master\build\python code crypter any python script with .py extension\b.exe
Developed by Neelanjan Manna
Choose the python file from file manager=====
enter number of times to encrypt=====
```

---

## Conclusion

This technique inflates a python script up to 700 Mb in its nascent form and ensures algorithm logic cannot be reverse engineered .

## References

---

1. Fares, M.B., Jagannath, S. and Lashuel, H.A., 2021. Reverse engineering Lewy bodies: how far have we come and how far can we go?. *Nature Reviews Neuroscience*, 22(2), pp.111-131.
2. Saiga, K., Ullah, A.S. and Kubo, A., 2021. A Sustainable Reverse Engineering Process. *Procedia CIRP*, 98, pp.517-522.