Contents lists available at **IESM**

Industrial Engineering and Strategic Management

Journal homepage: https://www.iesmj.com/

The Essence and Challenges of Reverse Engineering

Shafagat Mahmudova [1,*]

1. Department of Scientific-Theoretical Problems of Software Engineering and Intelligent Software Systems, Institute of Information Technology, Baku, Azerbaijan

Corresponding author: *shafagat@gmail.com*

https://doi.org/10.22115/iesm.2024.454814.1024

ARTICLE INFO

Article history:
Received: 29 April 2024
Revised: 16 May 2024
Accepted: 24 September 2024

Keywords:
Reverse engineering;
Software;
Digital models; Disassembling;
Converting the binary.

ABSTRACT

This paper studies the essence, challenges, etc. of reverse engineering. It explores how the program works with the help of reverse engineering, what data it uses, where and what it sends, as well as what vulnerabilities it has. Through reverse engineering, it is possible to understand how a program works, what data it uses, where and what it sends, as well as the program's weaknesses and how it reacts in the event of a crash. All this will help to improve the product and defeat competitors. The process of reverse engineering may seem simple; however, it is an illusion. Even experienced reverse engineers can analyze the same program for months. And it is not a fact that it will be possible to completely decrypt it. The paper also highlights the jobs reverse engineering is currently used for, and describes the procedure for using the disassembler, providing examples of the areas where reverse engineering is beneficial. It presents the exact ways in which reverse engineering occurs.

1. Introduction

The reverse engineering often refers to a code. When it comes to the field of information technology, this profession requires a thorough study of the software needed to comprehend how these programs work.

How to cite this article: Mahmudova Sh. The essence and challenges of reverse engineering. Ind. Eng. Strateg. Manage. 2023;3(1): 1–6. https://doi.org/10.22115/iesm.2024.454814.1024

With the help of reverse engineering, it is possible to understand how the program works, what data it uses, where and what it sends, as well as what weaknesses it has and how it reacts to emergency situations. All this will help to further improve the software product and defeat competitors [1].

Reverse engineering is currently used for the following tasks [2]:

1. Company ceases to exist and supports the product to optimize the functionality of the application in case when it is a manufacturer;

2. Analysis of viruses, trojans, as well as isolation of their signatures and creation of a protection environment;

3. Decryption of files to improve interoperability of formats.

4. Training.

As a result of reverse engineering, digital models of even complex software products can be quickly and easily obtained [3].

Reverse engineering of software is performed using the following methods:

1. Analysis of data sharing is most common in the reverse engineering of data sharing protocols, which, accordingly, is implemented using a computer network.

2. Disassembling the machine code of the program to achieve its listing in assembler language. This method works with any computer program; however, a non-expert spends a lot of time for it.

3. Decompiling a machine code or bytecode of the program to generate a initial code in some high-level programming language.

Disassembler is a translator that translates machine code into object files or library modules from assembler language into program text.

Two types of disassembler programs are distinguished [3]:

1. Automatic;

2. Interactive.

If the condition is considered in the paradigm of development, the solution to the problem is not so easy and consists of the followings:

- The programmer writes the code in a language he/she understands - for example, Java, Python or C++;
- He/she then passes the file to a compiler, which converts the code from human-readable to binary so that a computer can read it;
- The user downloads the compiled program - for example, in the EXE format mentioned above - and executes it on the computer.

A binary file, or executable as it is also called, is a collection of ones and zeros (machine code). A common user will not be able to read it and therefore will not understand the structure of the program.

In such situations, reverse engineering is offered in order to avoid getting into a dead end. The word “reverse” refers to the development process that goes in the opposite direction, that is, the binary file is converted back to a human-readable format. It looks like as follows:

- developer can obtain a binary file that only the computer can understand;
- then runs this file through a special program and converts it into code in a familiar programming language;
- then programmer studies this file and tries to understand the internal structure of the program.

The process of reverse engineering may seem very simple. Even qualified experts can analyze a program for years. It is not even known whether he/she will complete the work.

As a result of reverse engineering, digital models of even complex products can be obtained quickly and easily [4].

2. Literature review

Many studies have been implemented in recent years in the field of reverse engineering as there are numerous legislative consequences, for example the US Digital Millennium Copyright Act (DMCA). Therefore, some researchers attempted to create better communication between the legal and computing professions. By making it more difficult to share research results, such legislation hinders the exchange of ideas, leading to less innovation and uncertainty and people can get less profit from this [5].

Reverse engineering is the process of deconstructing software, machines, aircraft, architectural structures, and other products to extract design material from them. This prosses usually encompasses analyzing specific components of greater products. The process of reverse engineering allows determining the design and recreation process of any element. when buying an element to replace it, companies frequently apply this approach towards an original equipment manufacturer (OEM). The process of reverse engineering refers to working backwards from the original design process. Still, the companies do not know everything about the engineering methods of the product for its creation. Consequently, they face a problem of getting a working knowledge through disassembling the product piece by piece or layer by layer [6].

This field of engineering is a revolution by obtaining knowledge and information from the previous design, and it is of great importance for the industries of complex product systems (CoPS). But unfortunately, the studies are still not sufficient for the enlargement of this innovation. [7] presents a model in this regard and solves hidden and unknown interactions between reverse and forward engineering. The authors argue on reverse engineering in CoPS projects, which may improve designing and manufacturing processes of a new product. Initially, they develop an analytical framework through revising available literature in the field of reverse

engineering and innovation process models. Later, they develop a ten-step process model by a chronological multiple case study of three CoPS in the aviation industry and examining reverse and forward engineering in cross-case comparison. The model includes several stages, as data evaluation and verification, technical data generation, design verification, design implementation, concept, development, production, utilization, support, and retirement. It is a complex system of dynamic, iterative, concurrent, recursive, and time-interdependent processes. It can ensure a systematic, pragmatic, and detailed framework for improved organization of the new process in CoPS as designed and manufactured based on reverse engineering [7].

This paper studies the essence, challenges, etc. of reverse engineering. It explores how the program works with the help of reverse engineering, what data it uses, where and what it sends, as well as what vulnerabilities it has.

3. About software operation through reverse engineering

Through reverse engineering, it is possible to understand how a program works, what data it uses, where and what it sends, as well as the program's weaknesses and how it reacts in the event of a crash. All this will help to improve the product and defeat competitors [4].

The process of reverse engineering may seem simple; however, it is an illusion. Even experienced reverse engineers can analyze the same program for months. And it is not a fact that it will be possible to completely decrypt it. Figure 1 presents the code of the program using reverse engineering [4]. The program code using reverses engineering is shown in Fiqure 1.

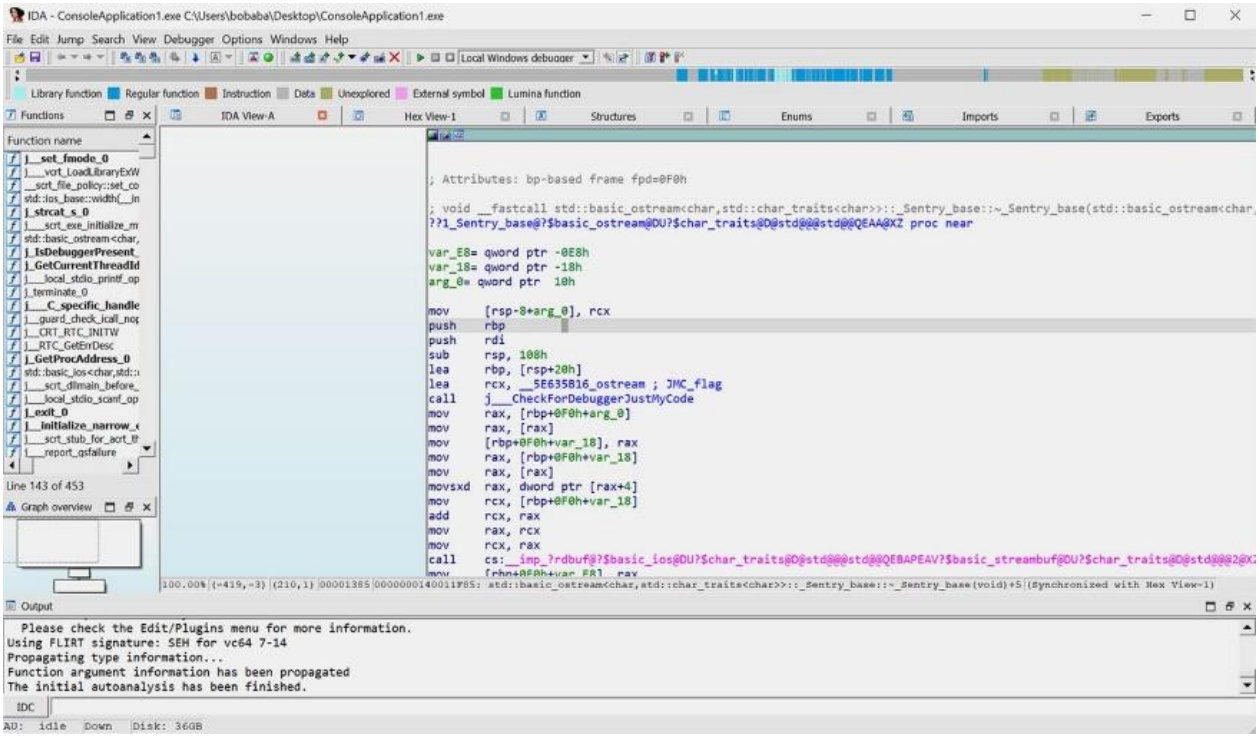

Fig. 1. The program code using reverses engineering.

4. Areas where reverse engineering is beneficial

Below are some examples of areas where reverse engineering is beneficial.

- Virus analysis and vulnerability scanning. Cybersecurity experts have been using reverse engineering techniques to analyze malware for a log time. For example, they can download viruses specifically, infect their computers with those viruses and learn how they work so that they can then develop defenses.
- Development of modules for games. In game development, reverse engineering enables creating modifications for games and changing the game, i.e., new features and add-ons are introduced. For example, modules for the games The Elder Scrolls V: Skyrim and Grand Theft Auto 5 have been generated in this way.
- Study of old technologies. The original documentation for the application can be lost, or the program could be written for older computers that are no longer in production. Therefore, developers must recreate the software code from scratch.

As mentioned above, reverse engineering is the science of converting machine code into human-readable language. But how exactly does reverse engineering work? There are three ways to do this.

1. Converting the binary file into code in a high-level language. There are special programs for this, that is disassemblers. They understand how machine code works and can easily translate it into a programming language that is convenient for human being, for example, C++.

The only disadvantage of using a disassembler is that the finished code often does not contain variable names and comments, therefore, after the procedure, the developer will have to figure out what individual blocks of code are responsible for.

2. Converting the binary file into code in an assembly language. Assembler is a slightly higher-level language than machine code, but much lower than JavaScript and Python. Any processor can run Assembler; thus, this is the easiest way to decrypt a binary file for a computer.

3 Finding out what information the program sends over the Internet or inside the computer. Typically, programs share data over the Internet or within the operating system. Therefore, reverse engineers use special network analyzers to understand how the software works.

While almost any manufactured product can be reversed, experiences are controversial when it comes to software. Developers often go through the process of understanding and improving their custom software base in order to write new code and debug software, whether in their own initial base or a colleague’s initial base.

Many people in the computer security community believe that the text used to compile a security test exception makes it too difficult to work because any results from reverse engineering cannot be made available to others. For example, if a developer analyzes the virus code and determines how to bypass its method, this information cannot be disclosed [8].

5. Conclusion

Today, software has become so multifaceted and interconnected that developers often don't recognize all the specifics and results of what happens in an application. It takes a lot of time. All program management ways and all groups of user options should be tried. Reverse engineering is an essential set of techniques and tools for understanding what software actually is. Formally, it is "the process of analyzing a subject system to identify the components of the system and their relationships and to create representations of the system in a different form or at a higher level." It enables visualizing the software structure, the way it works, and the functions that define its behavior. The use of automated tools for analysis and software testing provides intelligent ways to understand the complexity of software and discover its actuality. Reverse engineering is particularly beneficial when analyzing modern software for various purposes.

Funding

This article is not funding by any organization.

Conflicts of Interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

Authors' contribution statement

The paper also highlights the jobs reverse engineering is currently used for, and describes the procedure for using the disassembler, providing examples of the areas where reverse engineering is beneficial. It presents the exact ways in which reverse engineering occurs.

References

[1] Ayoshin IT. Reverse engineering of a software product using IDA PRO. Curr. Probl. Aviat. Astronaut. 3. Proceeding VII Int. Sci. Pract. Conf., Russia: Reshetnev Siberian State University of Science and Technology; n.d., p. 809–945.

[2] Kaspersky C, Rocco E. The art of disassembly. St. Petersburg: BHV-Petersburg; 2008.

[3] Dang B, Gazet A, Bachaalany E. Practical reverse engineering: x86, x64, ARM, Windows kernel, reversing tools, and obfuscation. John Wiley & Sons; 2014.

[4] Reverse Engineering n.d.

[5] Cifuentes C. Reverse engineering and the computing profession. Computer (Long Beach Calif) 2001;34:167–8.

[6] Hess B. What Is Reverse Engineering and How Does It Work. Link Available Https//Astromachineworks Com/What-Is-Reverse-Engineering 2019.

[7] Rozesara M, Ghazinoori S, Manteghi M, Tabatabaeian SH. A reverse engineering-based model for innovation process in complex product systems: Multiple case studies in the aviation industry. J Eng Technol Manag 2023;69:101765. https://doi.org/10.1016/j.jengtecman.2023.101765.

[8] Eilam E. Reversing: Secrets of Reverse Engineering. Indianapolis, Indiana: Wiley Publishing, Inc.; 2005.