International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 9, Issue 9 (January 2014), PP. 04-06

Understanding Contemporary advances in Reverse Engineering Tools

Kadam P.A.[1], Dr. S.D.Khamitkar[2], Dr.S.B.Thorat[3]

[1]*Assit. Professor, Department of Computer Science, Institute of Technology & Management College, Nanded*
[2]*Associate Professor, School of Computational Sciences, SRTM University, Nanded.*
[3]*Director, Institute of Technology & Management College, Nanded.*

Abstract:- Reverse engineering is the process of analyzing software. Reverse engineering is essential for providing security to applications and it has been an active research topic today. Presently there are number of R.E.(Reverse Engineering) tools available in the market. We have observed that every tool has its own speciality and drawbacks. In this paper we have rigorously studied some of the tools.

Keywords:- Reverse Engineering, Software Engineering, Decompilation, Software Testing.

I. INTRODUCTION

Today, in software industries number of operating platforms and languages are available and they are continuously increasing so using R.E. tool it is easy to maintain them. The aim of reverse engineering (RE) is to get different types of information from existing software and use this information for rebuilt the software and maintain that software and better understand about that software. Different freeware R. E. tools are available in the software market, but each and every tool having their limitation. They are designed for reverse engineering software for a specific language. No such fully automatic tool is currently available in the software market which can reverse the software 100%.The concept of Reverse Engineering is used in many fields of IT every day. Reverse engineering is not actively taught as a part of computer science courses. Reverse Engineering has been one of the most promising technologies to understand the legacy system and help them in reengineering it.RE encompasses any activity that is done to determine how a product works, to learn the ideas and technology that were used in developing that product. Using R.E. tools we are able to maintain Software and their problems.

II. CURRENT TRENDS IN REVERSE ENGINEERING

Currently, research on reverse engineering on some of the programming languages such as Java and C/C++, with secondary interest in the .NET area. Some freely available reverse engineering tools are as follows.

1. In java: cavaj,djdec312,djjava,jcavaj and jdgui etc.
2. In .net: codereflect,devevtras,dis#,dotpeek, spices evalution, etc.

Almost all software’s are design in the base that is not open-source and they are protected against the unauthorised users, but as far as per above we can tell all specialised software does reverse engineering in some degree. This is fundamental unanswered argument. Every person in computer industry will face reverse engineering issues. Software Reverse Engineering is done to improve the performance of a program [1]. The concept of Reverse Engineering is used in many fields of IT every day, to name just a few: legacy compatibility, binary code patching, malware analysis, network protocols analysis, debugging or even rapid prototyping. Reverse engineering techniques provide the means for recovering lost information and developing alternative representations of a system, such as generation of structure charts, dataflow diagrams, entity-relationship diagrams, etc [2]. Various types of software research engineering tools are there, still there is great need of reverse engineering tools and processes to evolve with the development environment that stresses components, the Web, and distributed systems [3].

Software reverse engineering use to see how particular software work and with the help of that information we can able to create new or duplicate product. Software reverse engineering use to reversing a program's machine code which is in the format of binary language i.e. 0 and 1 translate it back to back into the source code that it was written in, Software reverse engineering is done to get back the source code of a program if the source code is lost, to learning how the program performs certain operations, to develop the performance of a program, to repair a bug , to identify hateful content in a program such as a virus or to adapt a program written for use with one computer chip for use with another. Reverse engineering for the reason of copying or duplicating programs may stand for a copyright disobedience. In some cases, the permitted use of software purposely prohibits reverse engineering. There are several commercial reverse engineering tools on the market providing different capabilities and supporting specific source code languages [4]

Somebody doing reverse engineering on software may use several tools to disassemble a program. Several techniques have been suggested for supporting reverse engineering and design recovery activities[5] One of the many available tools is a Hexadecimal Dumper, which prints or displays the binary numbers of a program in hexadecimal format. By knowing the small parts that characterize the processor information as well as the instruction lengths, the reverse engineer can classify certain portions of a program to see how they work. Another common tool is the Disassembler. Hardware reverse engineering means to see how particular hardware work for example if a mobile development company want to see how competitors mobile works then they can purchase competitors mobile ,disassemble it and then make a mobile similar to it, however this process is banned in many countries. In general hardware reverse engineering requires a great deal of skill and is quite costly. Various reverse engineering and program comprehension tools and methods have been developed to help the software maintainer or reengineer understand the current structure and behaviour of the subject system.[6]

III. UNDERSTANDING CONTEMPORARY ADVANCES IN R. E. TOOLS

1. As a knowledge tool.
2. As a way to make new well-suited products that are cheaper than what's currently in the market.
3. For making software inter operate more effectively or to link different operating systems or databases.
4. To find out the uncoordinated features of money-making products.
5. Reverse engineering software helps researchers to examine the strength of systems and identify their weaknesses with different aspects and applications of reverse engineering.
6. Criminal cyberspace actions are growing rapidly and many software engineers are using reverse engineering methods to answer the attacks.
7. The security sensitive environment of these tasks, such as the understanding of malware or the decryption of encrypted content, brings unique challenges to reverse engineering, work has to be done offline, files can rarely be shared, time pressure is huge, and there is a require of tool and process support for capturing and sharing the knowledge obtained while trying to understand basic assembly code.
8. Most malware code is distributed in encrypted form at runtime; an unpacker routine transforms this to the original executable form of the code, which is then executed.
9. Recovery of missing information, providing proper system documentation , secondary with maintenance ,recognition of side effects and anomalies, Migration to another hw/sw platform , Facilitating software reuse.

As reverse engineering is a fundamental need to understand the software . As they are often badly designed and have an incomplete, nonexistent, or, even worse, wrong documentation without any design information, this is a challenging task [7]. This kind of inquiry engages individuals in a constructive learning process about the operation of systems and products. The process of taking something apart and revealing the way in which it works is an effective way to learn how to build a technology or make improvements to it. It is important to identify the goals and limitations of the effort before beginning the reverse engineering activity[8]. Reverse engineering consists of the following steps:

1. Watch and review the mechanisms that make the device work.
2. analyze and learning the internal workings of a mechanical device.
3. evaluate the real device to your observations and suggest progress.

By using reverse engineering, a researcher can collect the practical data necessary for the documentation of the operation of a technology or element of a system. When reverse engineering software, researchers are able to observe the strong point of systems and identify their weaknesses in terms of performance, protection, and interoperability. Reverse engineering is a systematic form of program understanding that takes a program and constructs a high-level representation useful for documentation, maintenance, or reuse.[9] The reverse engineering method allows researchers to recognize both how a program works and also what aspects of the program contribute to its not working. The IEEE Standard for Software Maintenance (IEEE Std 1219-1993) defines reverse engineering as “the process of extracting software system information (including documentation) from source code”.

IV. CONCLUSION

In this paper we overview the concept of reverse engineering which begins with the creation and works through the propose process in the reverse way to turn up at a product definition statement. The use of existing R.E. Tools and how they are useful in the process of R.E. The conception of better designs and the interoperability of existing products often begin with reverse engineering. In doing so, it uncovers as much information as possible about the design ideas that were used to produce a particular product. It usually involves rising a set of functional provision for a product, system or piece of equipment, based on an analysis of existing product. There must be need of reverse engineering.

REFERENCES

[1]. MamtaGarg & Manoj Kumar Jindal “Reverse Engineering – Roadmap to Effective software Design by in” International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009. P.No.186-188.

[2]. Spencer Rugaber and Richard Clayton , “The representation problem in reverse engineering”, In Proceedings of the Working Conference in Reverse Engineering, Baltimore, Maryland, pp. 8–16, IEEE Computer Society, 1993.

[3]. K. Bennett and V. Rajlich , “Software maintenance and evolution: A roadmap”, In A. Finkelstein, ed., The Future of Software Engineering, ACM ISBN: 1-58113-253-0, 2000, pp. 73-90

[4]. Berndt Bellay, Harald Gall, “An Evaluation of Reverse Engineering Tool Capabilities” Distributed Systems Group, Technical University of Vienna, A-1040 Vienna, Austria, Europe

[5]. Gerald C. Gannod , Betty H. C. Cheng “A Framework for Classifying and Comparing Software Reverse Engineering and Design Recovery Techniques”

[6]. Paolo Tonella Marco Torchiano Bart Du Bois Tarja Systä “Empirical studies in reverse engineering: state of the art and future trends ” © Springer Science + Business Media, LLC 2007

[7]. Ira D. Baxter Michael Mehlich “Reverse Engineering is Reverse Forward Engineering “ Semantic Designs, Inc. 12636 Research Blvd. C-214 Austin

[8]. Nadim Asif “Software reverse engineering process: Factors, elements and features” International Journal of Library and Information Science Vol. 2(7), pp. 124-136, October 2010

[9]. Asit Kumar Gahalaut, Padmavati Khandnor “ Reverse Engineering: An Essence for Software Re-Engineering and Program Analysis” International Journal of Engineering Science and Technology Vol. 2(06), 2010, 2296-2303