BIOINFO Systems Engineering
Volume 1, Issue 1, 2011, pp-01-05
Available online at: http://www.bioinfo.in/contents.php?id=347

The Next Step In Reverse Engineering

Kadam P.A.[1] Kurundkar G.D.,[2] Naik N.A.,[3] Khamitkar S.D.[4]

1.Department of computer science V.I.P.Road Nanded.
2Dept. Computer Science S.G.B.College Purna(Jn.),Dist: Parbhani.
3. Dept of Computer Science Nanded .
4Head Dept. of Computer Science .S.R.T.M.University Nanded

Abstract :

In computer word reverse engineering a very important topic..In today's computer system there are many changes like Performance, capacity of ram, hardisk and many changes in devises ,security of computer systems and human interaction between client and server , simultaneous multiple operation between them etc reverse engineering is done mostly on old systems whose documentation is not up to date. reverse engineering is a process reconstituting the system, keeping in view of its current functionality, in order to its understandability, hence easing its maintenance. Since it is easy to change the design of a system then its implementation, it is reasonable to change the recovered design

In this paper, we try to find the scope of the reverse engineering, this paper mainly highlight about reverse engineering process using that we are able to understand hardware, software, operating system, requirements, documentation and maintain our database ,using that we know about internal operations of program.

Keywords: Reverse engineering, scrutiny, code, reengineeraring

I. INTRODUCTION

Reverse Engineering (RE) is the process of Understanding device ,aim of that device, working of device ,its structure ,how data flow, operation and analyzing its workings in detail, used in maintenance or to try to make a new device or program that does the same thing without copying anything from the original(4) Software reverse engineering is done for different reasons such as, to study how the program performs certain operations, to improve the performance of a program, to fix a bug, to identify program written for use with one microprocessor for use with another.(7)

Reverse engineering is used on whole computer system means hardware and software reverse engineering involves taking apart a device to see how it works whereas software reverse engineering is the process of recovering or reconstructing functional and technical specifications of a software system at a high level of abstraction.. Reverse engineering should aim at recovering architecturally significant views of the system, which can help keep track of the evolution of software architecture. Software reverse engineering is done for various reasons such as, to study how the program performs certain operations, to improve the performance of a program, to fix a bug, to identify program written for use with one microprocessor for use with another. The reason for performing reverse engineering is to maintain legacy code. Therefore, it should not be focused on program understanding but on system maintenance instead.

This should be done in a way that frees us from reverse engineering a system again and again because of modifications made to its code over time.(8)

Reverse engineering consists of

1. scrutiny of the product
2. creation of an in between level product explanation
3. Human examination of the product description to produce a plan
4. creation of a new product using the plan.

The plan is made as conceptual and functional as possible by the reverse engineers, and is then handed over to a fresh design team who have no other contact with the old product, or the team who analyzed it. Several different approaches have been proposed by the research

community. In practice, reverse engineering still presents several challenges: scalability of the techniques, usability of tools, visualization, dealing with multiple perspectives, level of abstraction, and so on.

WHY NEED REVERSE ENGINEERING?

The problem of redesign an existing system in a different programming language has been around for years and three general approaches have emerged:

1. Manually modify the existing system.
2. Use an automatic language translator.
3. Redesign the system

In the first approach, manually modify the existing software system means manually translating from the source language to the target language. There is flexibility in terms of translating the system and changing the system structure. However, there are several disadvantages as well. This approach is time consuming. The number of lines of code that can be translated manually per programmer per day is small. Finally there is possibility of Humans make mistakes. There is no means to guarantee that the rewritten system is functionally equivalent to the original. The second approach, automatic translation, relies on the use of a tool that accepts software written in the source programming language and generates new source code written in the target language. This approach generates new code quickly. However, This approach has several disadvantages. The source language may not yield itself to simple translation into the chosen target language. Some automated translator tools only do the easy part of the translation and leave difficult portions for a human. The third approach is to redesign the system. This approach starts with the requirements for the current system and builds a completely new system in the target language. This new system is required to be functionally equivalent to the original system even though it is not derived from it. Of the three approaches, this approach has the greatest chance of producing the best possible new system. Redesigning the system in the new language provides the most power and the greatest flexibility in terms of creating the end product. The resultant system may have significantly lower maintenance costs than systems generated by the other approaches. However, this approach also has several disadvantages. It is more difficult than doing an initial design, because of the requirement to emulate the existing systeminterfaces. This approach has the highest initial cost. It is equivalent to building a new system. The most serious disadvantage is that for many systems it is not possible to redesign from the system requirements, since the requirements may not exist. For many older systems the only accurate statement of the system's capabilities and functionality is often the source code itself. There often is no valid requirement specification for the system. If there is no requirement specification for a system, reverse engineering the system can produce a reconstructed design that captures the functionality of the system.. Thus re-engineering based on a reverse engineering process offers many of the advantages of the redesign approach. This approach is always feasible if the source code for a system exists.

Need of reverse engineering

Reverse-engineering is used for several purposes: as a learning tool; as a way to make new, compatible products that are cheaper than what's currently on the market; for making software interoperate more effectively or to bridge data between different operating systems or databases; and to uncover the undocumented features of commercial products etc. The major motivations behind usage of this technology are as following:

Correcting : Sometimes software need to be updated or corrected as according to the current need, at those times it is required specially in case of no or insufficient documentation.

Misplaced documentation : It is possible that documentation of a system has been misplaced

Product analysis : To examine how a product works, what components it consists of, estimate costs, etc. Internal function of product: understand the internal operation of that product

Competition : understand what your competitor is actually doing versus what they say they are doing

Learning : It may be used for learning from others' mistakes. So that same mistakes that others have already

made and corrected, are not made by the analyzer.

CODE REVERSE ENGINEERING :

Using reverse engineering we understand how code is execute and flow of that code etc. in this paper we presently focus on both forward and reverse engineering at code level. Using reverse engineering we can get source code but sometime in poorly documented software we can not get easily source code. if there is source code available then we understand each and every components also we understand data flow between each and every model. using that we think new idea behind that software and lastly we develop software which are better from previous Knowledge of software architecture from multiple user perspective is needed to make large scale, structural changes, and the capability to perform architecture reconstruction is becoming increasingly important. Thus, reverse engineering techniques are used as an attempt to regain useful understanding.

HARDWARE REVERSE ENGINEERING

Using that we understand how device works. if a processor manufacturer wants to see how a other' company processor works, they can purchase other' company processor, disassemble it, and then make a processor similar to it. Computer vision has been widely used to scan PCBs for quality control and inspection purposes, and based on this, there are a number of machine vision for analyzing and reverse engineering PCBs. However, this process is illegal in many countries., hardware reverse engineering requires a great deal of expertise and is quite expensive.

IV. SOFTWARE REVERSE ENGINEERING TOOLS

Reverse Engineering is on improving human understanding about how this information is processed, data reverse engineering tack the question of what information is stored and how this information can be used in a different context... But there are some common tools that are use Debugger - A programmer use debugger to find bugs in their program. Debugger is only tool by which we can trace/break a function or code live. There are many debuggers available in the market. We all know how to debug any program, first we put a breakpoint on the required statement and then we run the program. When this instruction is near to be executed the program stops and we can see values! This thing is directly related with cracking. Dissembler — As an executable file is in binary format so a normal user cannot understand the instruction in this file. Also any exe or executable is generally in PE format (which is a standard format for exe file, decided by the committee of software companies like MICROSOFT, IBM, and AT&T. For more about exe search any virus related site or /simply search your favorite search engines.) Hence a cracker first disassemble the program .now a Disassembler converts the binary file in its equitant assembly language instruction's most of program is written in high level language hence size of the disassembly goes in millions (or even larger) of lines and hence it is not possible for any cracker to understand this code. And hence cracker generally looking for strings in this disassembly such as; -"your 30 day trial period has expired." Or "the serial no you entered is not valid!!!" Etc. Then they trace the assembly code some lines and simply reverse the jumps. (For example one to jump) so that control did not come on this string and go to the statement such as "thanks for registration!!!"(We will see later how this can be done but currently this info is enough for you..) Now there are many dissembler available. But two of them, which are most commonly used, are WIN32DASM and IDA .IDA is a powerful debugger then WIN32DASM and used for advanced cracking. But WIN32DASM is most widely used debugger by newcomer and intermediate crackers. This debugger allows you to disassemble any file which is in PE format, we can save disassembly .it can tell us which function is imported, which function is exported, we can execute jump, call, find string data reference and dialog reference easily and many more facilities it provides like we can executes the exe file, step in to it, step over and blah, blah.

3) Hex Editor : A hex editor allows us to change the contents of any file in hex format. It displays the contents of the file in hex format. We can simply have to change the value at memory location which we find using softice. Now

there are a lot of hex editor available such as ultredit, biew, hiew and a lot (I think many c, c++ programmers has developed it). But the most popular among these is HIEW. Which stands for "Hacker's vIEW". This little program offers a lot of facilities such as editing in hex or ASCII format, searching any string in hex or ASCII format. There is another good facility which makes it different from others is that, it offers you to write the assembly code and it can automatically convert this code in to equitant hex format. This is helpful for the crackers who don't know equitant hex value of assembly instruction. (For example: - if we have to change the jump to nope at any memory location then after pressing F7 key then we can only write nope and it will automatically convert it to its hexequilant which is 90.) There are other hex editors also but it is the most widely used.

4) Unpacker/PE Editor : Sometimes programmers used file compressor such as UPX,ASPACK to minimize the size of the program. This is called a file packer. Now what apacker do is using any algorithm he reduce he size of the file and append it code in to the exe file and at run time, first the code of the unpacker is executed and after that it decompress or unpack the program in memory. Since the program we have to crack is unpacked in the memory only hence a cracker cannot simply disassembles and patch the program. User can only patch it runtime. Therefore to un-pack the exe file permanently we use unpacked. Which unpack the exe file and we can store this unpack file to the disk. If a program is using a packer then its exe header will changed. There are various techniques available to manually unpack the exe by modifying the exe header but those are high level techniques and don't want to discuss them here because I think most of the newsiest find difficult to understand it. The most widely used unpacker is procdump. This software has ability to unpack different kind of packer stand-alone. It also allows changing or viewing the header of exe files

5) File Analyzers : To identify which packer is used to pack file cracker uses this kind of programs. By using this, a cracker can know which compiler or packer is used to protect the shareware. This software simply works on signature byte. With the help of this you can find what compiler or in which language the program has written. There are many this kind of program are available such as file inspector, File Info etc.

6) Registry monitor : Some program uses registry keys to store their registration information. Hence, 'Registry Monitor' is a software which works in background and traps all the registry access by the all process, which is currently running.

7) File monitor : some program also uses key file or they have there security algorithm in different file and hence file monitor is use to see which application is using what file.Bypassing the protection

VII. COPYRIGHT ISSUES

It is widely accepted that copyright does not protect "ideas", but only "their expression". Reproduction or translation of the whole or a substantial part of a copyright work will constitute an infringement of copyright . Reverse engineering or the copying or duplicating a program may constitute a copyright violation. In some cases, the licensed use of software specially prohibits reverse engineering. Reverse software systems which is done for the purposes of interoperability is mostly believed to be legal, though patent owners often contest this and attempt to stifle any reverse engineering of their products for any reason. In Europe, special codes of protection exist for computer programs, semiconductor topographies, and databases. Each of these contains special definitions of infringement which are binding across the EU, and which (for computer programs and semi-conductor products) mirror those created in the US. A given act of reverse engineering may involve several of these provisions; if so, it needs to be clear of infringement under each different head of copyright work. With copyright infringement, both the creation of the intermediate copy of the original design documents (which takes place after analysis of the product) and the ultimate products created from it may be infringements of copyright, as we will see from the cases.

VIII. REVERSE ENGINEERING DRAWBACK

Reverse engineering is beneficial in many cases, but it also has some drawbacks attached with it. The main

aim of reverse engineering to study about software A major drawback is Cracking. Cracking is a process using that we know the internal code and there inter functionality between modal. Reverse engineering is used to understand how a program flow, this prosess is bypass your software security . Reverse engineering is used by a cracker to understand the protection scheme and tobreak it, so it's a very important thing in the whole world of the cracking.

Nowadays there are several cracking groups specialized in reverse web scripts. There is nothing of new in this because the web pages are written in java or CGI scripts or something else. So, they can be considered as small programs. The web cracker usually reverses the protection schemes of web pages creating cracked passwords, which are distributed on the web. Thus, it indirectly promotes cyber crimes. For avoiding such things steps have been taken such as some copyrights do not allow reverse engineering and some reverse engineering tools manufacturers have put restriction on their products. But still crackers have found alternatives for these things.

IX. CONCLUSION

Using Reverse engineering, we are able to re-engineer old software, to make it more modular, re-useable, accessible or reliable.Thus, Reverse engineering is process using that we understand which are the newer technology is built up .using reverce engineering process we able to full study about that software which are the challenges in our field with out knowing the whole software we can not work on the software .In this way, the recovered design is updated according to the needs and a new detailed design generated, which, can further be implemented using the modern development techniques and programming languages.

REFERENCES

[1] Anil Panghal, Pawan Kumar, Sharda anghal (2009) "Reverse Engineering" Proceeding of 2nd National Conference on "Recent trends and Advancements in Computing", Sirsa, February.

[2] Muller H, Jahnke J, Smith D, (2000) Reverse Engineering: a roadmap, Proceeding of the 22nd International Conference on Software Engineering, ACM Press, New York. Ref.

[3] K. Wong, Reverse Engineering Notebook. PhD. Thesis, Department of Computer Science, University of Victoria, October 1999.

[4] Jenkin | Reverse Engineering http://www.jenkins.eu/articles/reverse-engineering.asp>

[5] Software Security and Reverse Engineering http://www.infosecwriters.com/text_resources/pdf/software_security_and_reverse_engineering.pdf> © 2009

[6] Reverse Engineering of Computer-Based Control Systems Lonnie R. Welch, Guohui Yu, Binoy Ravindran, Franz Kurfess and Jorge Henriques

[7] Reverse Engineering A Swiftly Growing Technology in Software World A.Preet Inder Singh B. Richa GuptaGuru Nanak Dev University, Amritsar, India

[8] Reverse Engineering is Reverse Forward Engineering Ira D. Baxter Michael Mehlich

[9] Reverse Engineering – Roadmap to Effectivesoftware Design -A. Hexadecimal Dumprer.