Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.534 – 544

RESEARCH ARTICLE

A Generalized Process of Reverse Engineering in Software Protection & Security

Krishan Kumar[1]

Department of Comp. Sci&Engg.

GNDU, Amritsar, Punjab, INDIA

Er.krishankumar4014@gmail.com

Prabhpreet Kaur[2]

Department of Comp. Sci&Engg.

GNDU, Amritsar, Punjab, INDIA

Prabhsince1985@yahoo.co.in

Abstract- Reverse engineering plays major role in current software development and in Research & Development activities whether to develop new product and upgrading it to new version. Two of the major area of the Reverse engineering in current scenario are Software Security and malware analysis where it can be used by people for evil purposes to break the software security, protection schemas, and spread the vulnerabilities. On the Other Side it is used by Security Researches to Understand the Vulnerabilities in protection schemas, key holes in security, so that they can design protection schemas which are difficult to break as well as understand by crackers/Attackers.

Keywords: Reverse Engineering, Software Security, Software Protection, Protection Schemas.

I. Introduction

In Present scenario there are many software protection application are available in the market which don't allow to use all features if the user are not registered to these application. There are two important concepts in security are Confidentiality and data authenticity. Confidentiality is used to provide data secrecy of the message and data authenticity protects the integrity of the message which is being transferred. Software Protection mainly falls between the domains of security, cryptography and engineering among other disciplines[1]. The Software Protection techniques mainly concern in protecting the software from various attacks such as reverse engineering by using the

obfuscation, modification by software tamper resistance, program-based attacks by software diversity , and BORE-Break Once run everywhere-attacks by architectural design[2]. The software protection may appear in many forms, from end-user which is asking to enter the serial key to those invisible watermarks embedded in the software. The goal of software protection scheme is to protect the intellectual property rights of owner.

In the context of Software engineering, the term reverse engineering was defined in 1990 by Chikofsky and cross in their seminal paper[3] *as the process of analyzing a subject system (i) to identify the system's components and their inter-relationship and (ii) create representation of the system in another form or at a higher level of abstraction*.

On the other there are some people who always wants to break these protection schemes. These people are called crackers who destroy the code of application that they want to crack [4]. When the reverse engineering is used as process/art to remove the protection of programs, then it is known as cracking. The cracking is started as long as protection scheme start appearing. The most common software crack is consists of the modification of an application's binary to cause or prevent a execution specific part of the program. This is mostly achieved through the reverse engineering. Reverse engineering is the art of generating a source code from the binary's/executable of software[5]. It is done to understand how program is doing action, to learn about the type of protection scheme and mainly how to bypass the protection schemes which are protecting the software form unregistered user to use its full functionalities. Crackers mostly find the protection scheme which is protecting the software by using the reverse engineering and usually break it using different tools and tricks, So it is very important thing in world of crackers. Reverse engineering integrates several arts like: code breaking , puzzle solving , programming , and logical analysis. Different people use reverse engineering in different ways depending on the different purposes [6].

II. Literature Survey

Reverse engineering is used in various fields from hardware to software. E.Eilam[4] provide the compact information about the Software reverse engineering, reversing process, tools, reversing malware , Piracy and copy protection , and all the information which is necessary for reverse engineering.

In this pavol [6] provide the information regarding the tools used in cracking, basic introduction to cracking , different techniques used for cracking the software, the basic software protection techniques and protection tricks.

Some People think there is not much relation between reverse engineering and software security. In this hardik [7] define the reverse engineering by relating it to the cracking, history of reverse engineering, Various software protection schemas, their detail study and simple logical mechanism behind each protection schemas, tools of trade , case studies illustrating the how crackers breaks these protection schemas like serial fishing, Bypassing of check also called patching , to make a key generator.

As the uses of computer in criminal activities such as bank fraud, identity theft and corporate theft etc increased to high level as compare to mid 1990's. These illegal activities in cyberspace not only affect national security but also threaten citizen's right and privacy, thus having significant political, economic and social implications. In this Christoph et al [8] had studied the affect of illegal activities, their goals and counter measure against cybercrime. They had conducted the field study in work practices of security reverse engineering and identified five processes: analyzing assembly code, documenting findings through different kinds of artifacts, transferring knowledge to other reverse engineers, articulating work, and reporting of findings to stakeholders. One of the major outcome of this study show that still there is not much research in the field of security reverse engineering and tools which are available also have some limitations.

In the Jiutao et al [9] classify the software protection into two types Software and hardware based encryption with key technologies Such as Data encryption technology, Application of cryptography, Antidisassembly, Antidebugging and others techniques for software protection such as Code Obfuscation Software watermarking and Tamper-proofing.

In the Michael et al [10] discussed the technique to protect the software from dynamic tracing or dynamic debugging used by cracker to track the execution of software by using the debuggers. Anti Debugging is used against the dynamic tracing to prevent the use of debugger. Software can detect the debugger from hardware, software and users levels. when the software detect that debugger is running in memory send the WM_Close message to the debugger window to close its forms or terminate the debug process forcibly.

III. Objectives

1. To study of Reversing engineering tool and Techniques.

2. To Study how Reversing Engineering is being used too Crack or break the Software Protection.

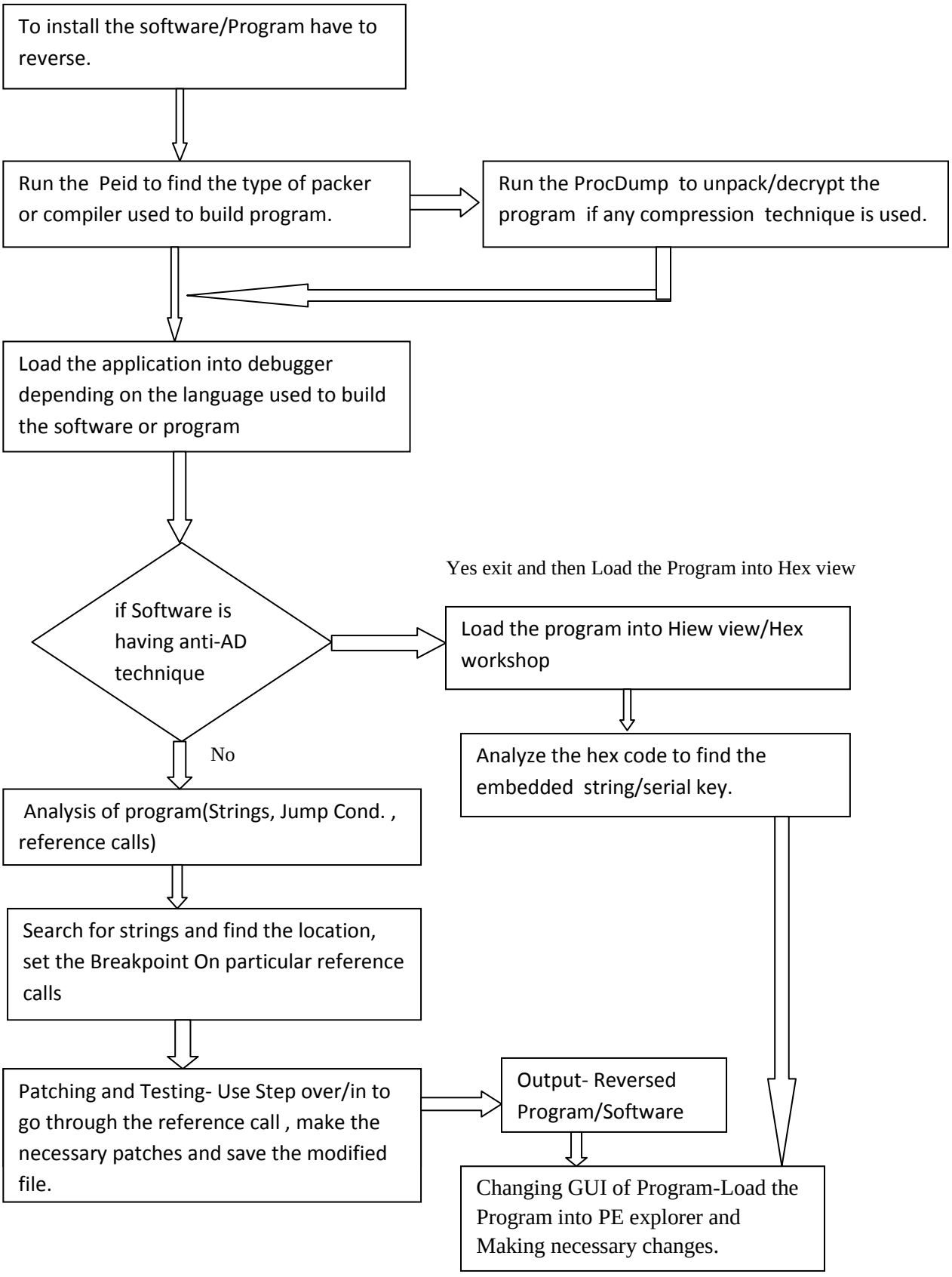
3. To design and implementation of a Cognitive Process Model to reverse the Software Protection.

IV. Cognitive Reverse engineering Process Model to reverse the Software Protection-A Cracker's Workflow

Methodology:- Stepwise execution of Purposed Process model

1: Install the program or software which has to be reversed.
2 : Use the Peid/PE identifier to find the language used to build the program e.g. if a program has been coded in C++, ASM, Delphi or Visual Basic . It also used to identify the type of packer used to pack the file.
Use the ProcDump to dump, unpack/decrypt some protected file without debugger.
3: Use an appropriate debugger depending on the language used to build program/software e.g. smartcheck for Vb, OllyDbg for C/C++, Delphi, Immunity Debugger for Python etc.
If the program/software going to reverse having any anti-debugger technique then there are chances that program/software get freeze or does not loaded into memory.
Use Hiew view/Hex workshop to find the serial of program/software.
4: Detail Analysis of program/Software when it is loaded into debugger and run the program in debugger to find out the possible error strings, reference strings, nag screens and reference calls and finding the location.
5: After finding the location of references strings set the Break point at this location and re- run the program to get into that reference call.
6: Patching and Testing-Use step over/in to go through the call and to find the possible location where patches needed and modify the assembly. save the changes to new copy.
Step in/over is used to get deep into the call and execute each instruction one by one to find out the possible jump and other instructions which causes an error.
Output-Reversed program/Software
7:It is optional only required if GUI of program/software has to be changed. Load the reversed program/software into PE explorer and do the necessary changes into strings , logo ,icons etc.

Output-GUI modified Reversed Program/Software.

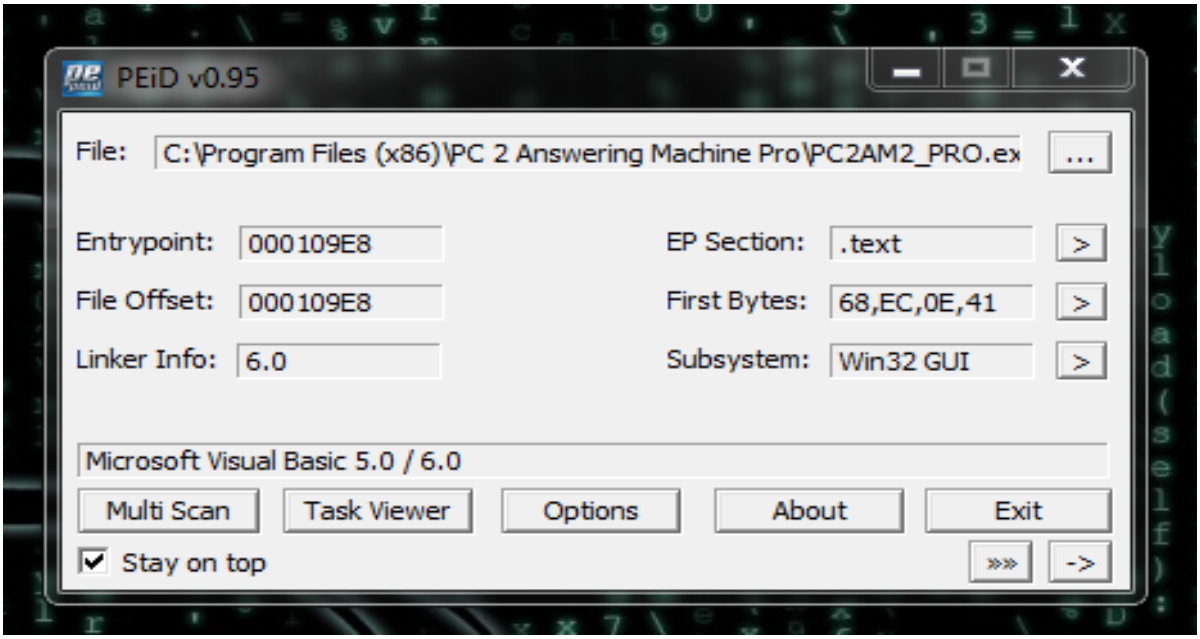


V. Case Study

Software-PC 2 Answering Machine Pro

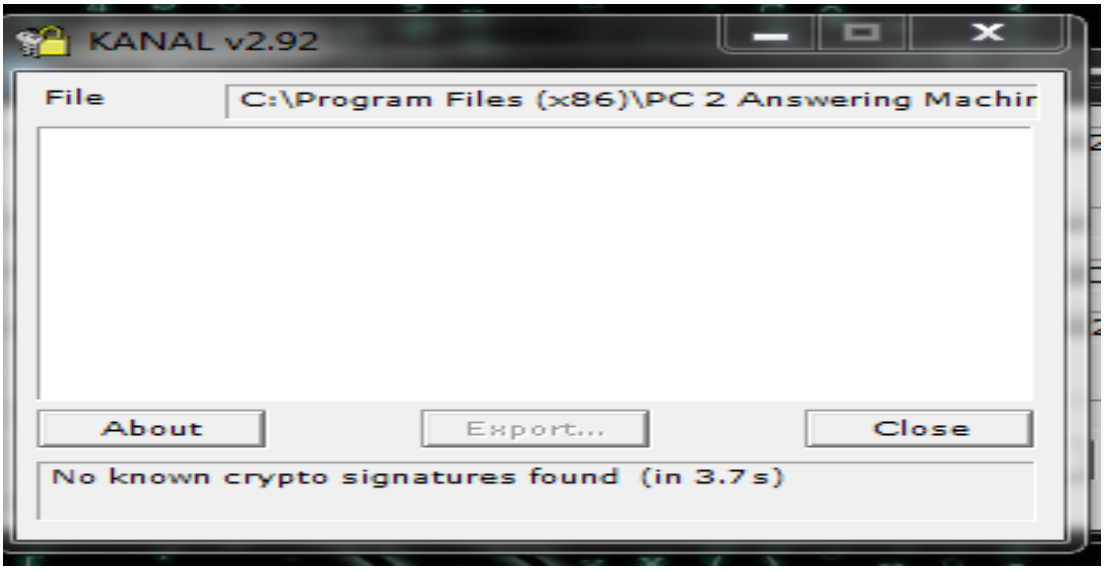
Tools Used-Peid, Ollydbg, VB Decompiler

1. Installing PC 2 Answering Machine .
2. Load into the Peid to find the build language of software.



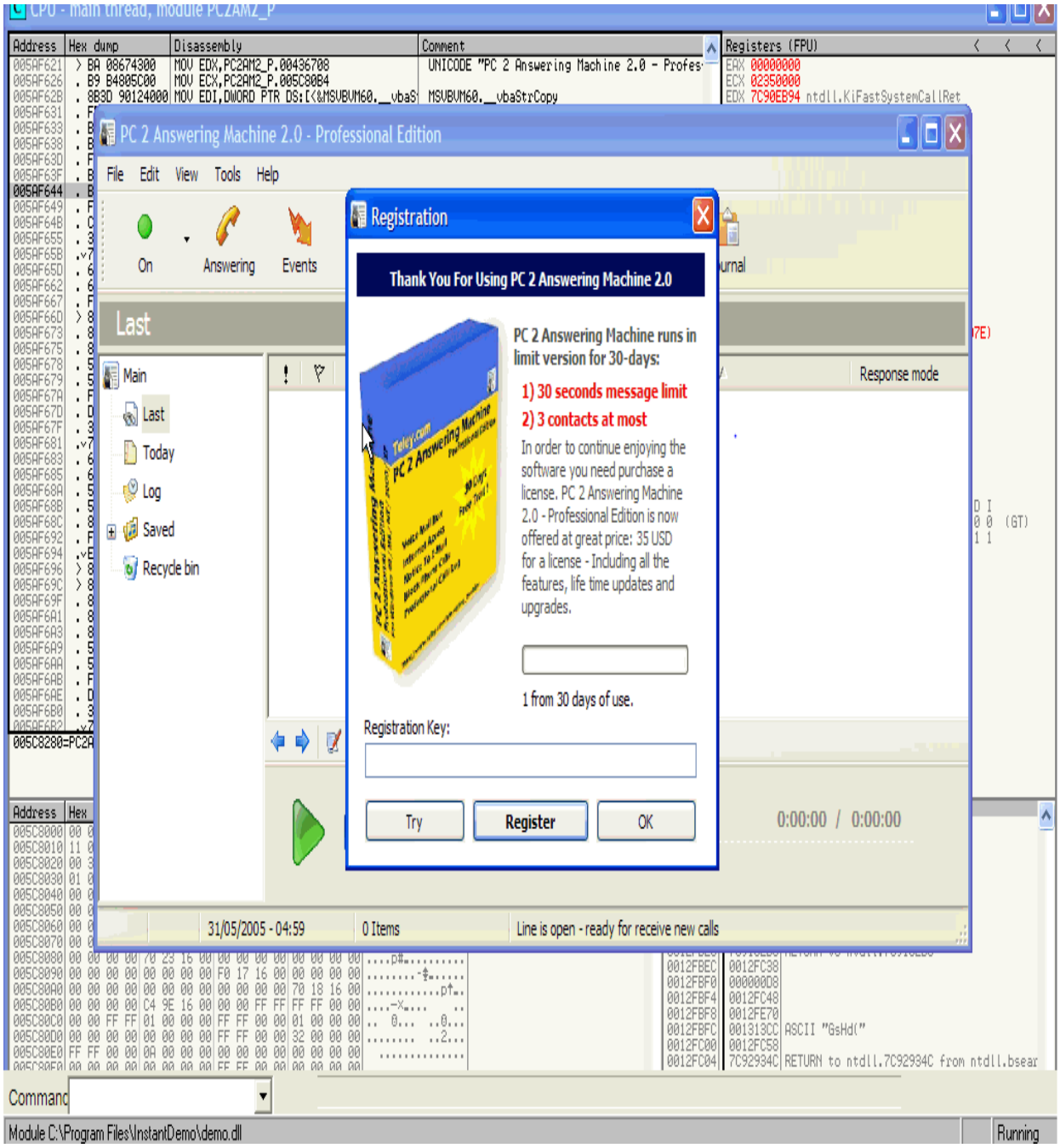
In this case build language is Microsoft Visual basic 5.0.

3. Checking for any protection technique. it does not find any protection technique.

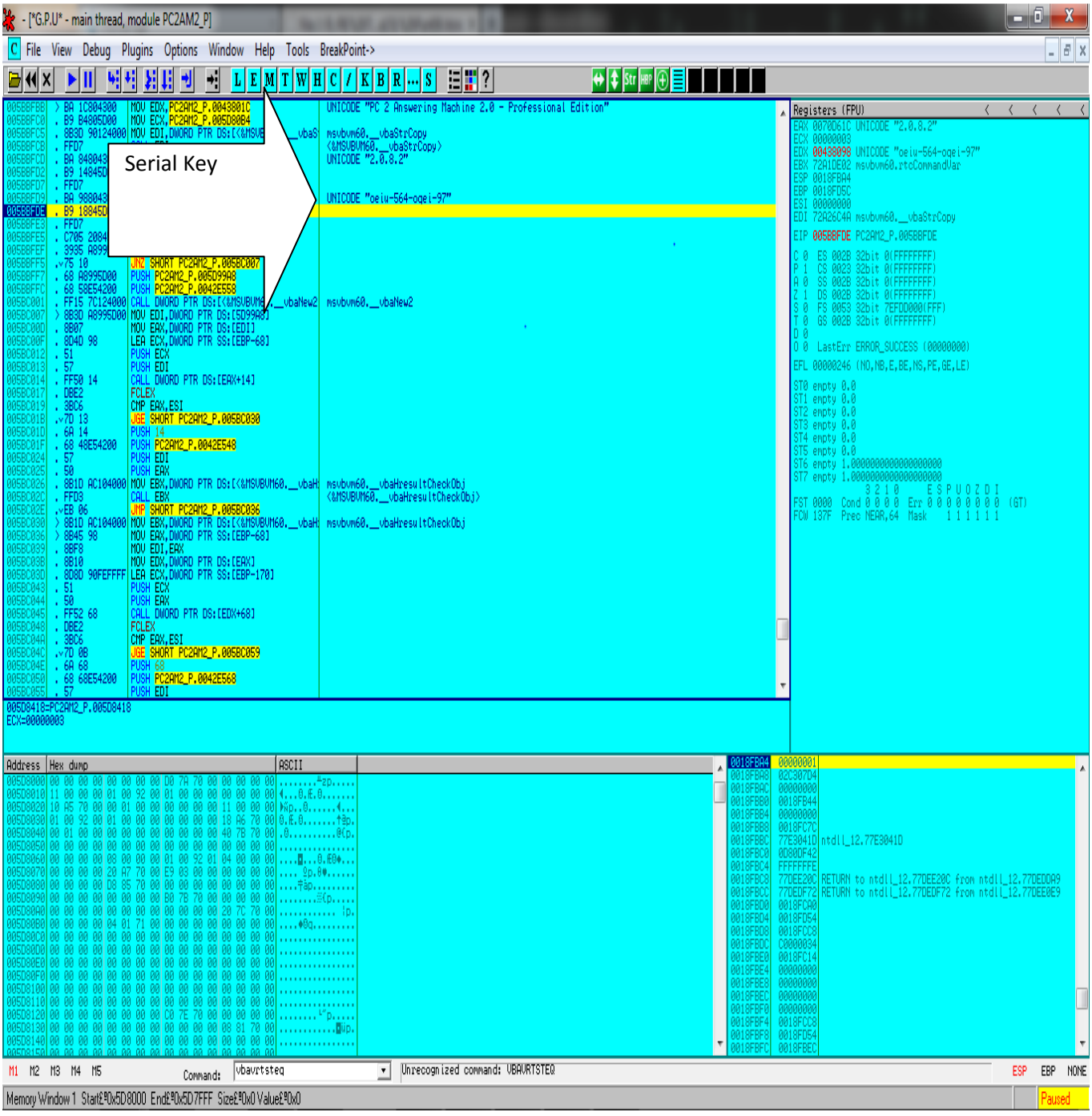


4. Loading it into OllyDbg debugger.

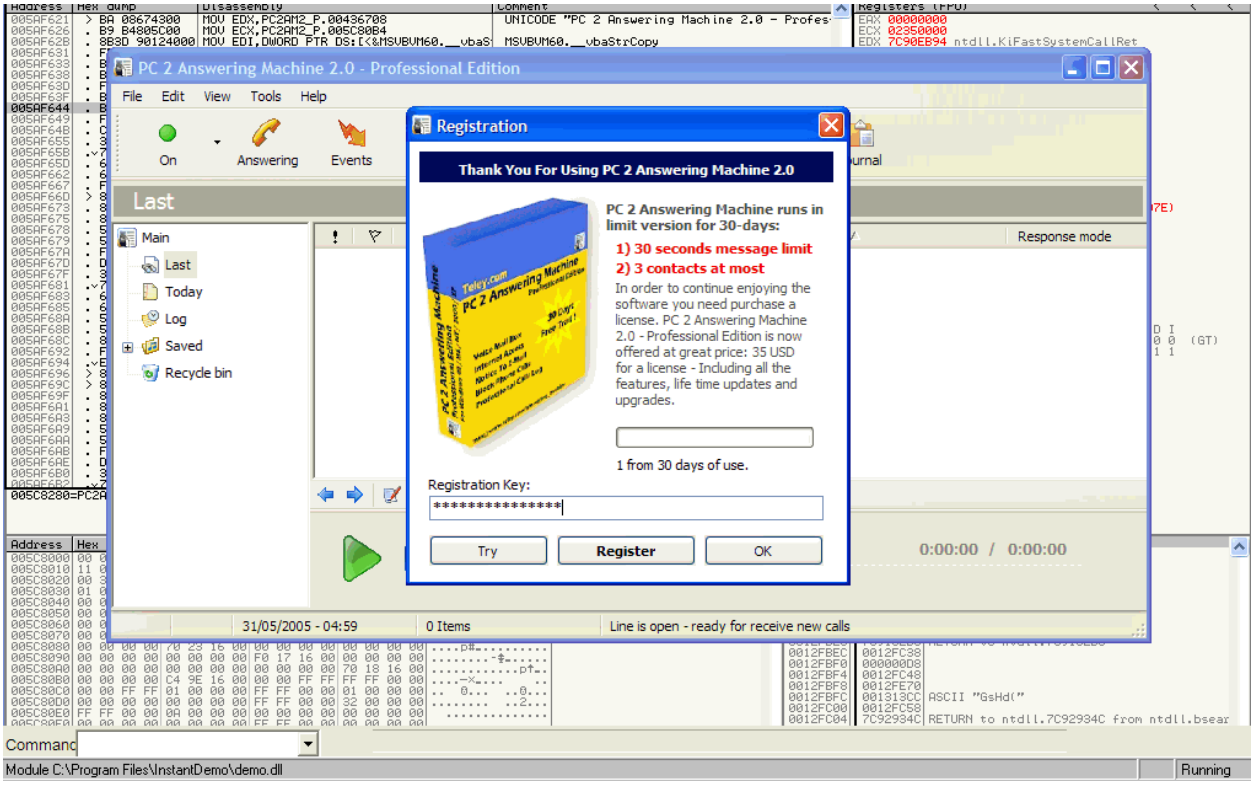
5. When application is run inside ollydbg it show the registration Nag screen show it is 30 days trial Limit

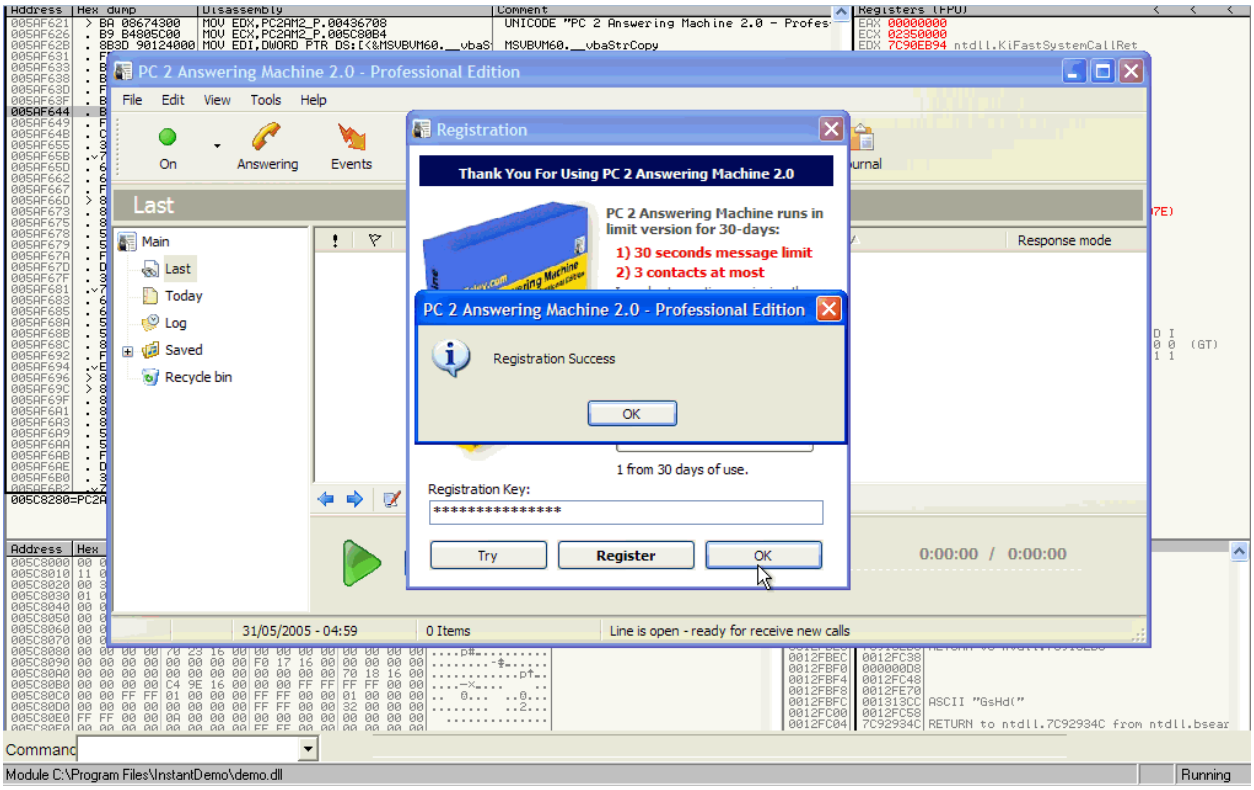


6. After analyzing the different modules of software we find the VBAVARstrTstNe routine which contains the embedded serial key.



7. After adding this serial key into registration window we get the complete version of program and Registration Success Message.





Tools Used in Reverse Engineering

System Monitoring Tools [11] - tools that are used to monitor, explore, sniff and exposing the program being reversed. Most of these tools display information gathered by the operating system about the application and its environment. Because almost all communications between a program and the outside world go through the operating system, the operating system can usually be leveraged to extract such information. System-monitoring tools can monitor networking activity, file accesses, registry access, and so on.

E.g FileMon, TCP View, RegMon, PortMon, Process Explorer.

Disassemblers - disassemblers are programs that take a program's executable binary as input and generate textual files that contain the assembly language code for the entire program or parts of it.

E.g IDA (Interactive Disassembler) by DataRescue (www.datarescue.com), ILDasm-is a disassembler for the Microsoft Intermediate Language (MSIL)

Debugger- One of the most important tool for reverser and malware analyst .A debugger is program that allows software developers to observe their program while it is running. The two most basic features in a debugger are the ability to set breakpoints and ability to trace through code. Breakpoints allow users to select certain function or code line anywhere in the program and instruct the debugger to pause program execution once that line is leached. When the debugger reaches that breakpoints ,the debugger stops and displays the current state of the program. E.g OllyDbg –User mode debugger written by Oleh Yuschuk(http://home.t-online.de /home/Ollydbg.)

WinDbg-WinDbg is a free Debugger by microsoft as a part of the Debugging tools for windows package www.microsoft.com/whdc/devtools/debugging/default.mspx breakpoints and ability to trace through code. Breakpoints allow users to select certain function or code line anywhere in the program and instruct the debugger to pause program execution once that line is leached. When the debugger reaches that breakpoints ,the debugger stops and displays the current state of the program.

E.g OllyDbg –User mode debugger written by Oleh Yuschuk(http://home.t-online.de /home/Ollydbg.)

WinDbg-WinDbg is a free Debugger by microsoft as a part of the Debugging tools for windows package www.microsoft.com/whdc/devtools/debugging/default.mspx

IDA PRO- One of the most widely used debugger by Reverse engineer and Malware analyst. (www.datarescue.com)

Decompilers- Decompilers are next step up from disassemblers. A decompiler takes an executable binary file and attempts to produce readable high-level language code form it. The idea is to try and reverse the compilation process, to obtain the original source file or something to it.

E.g VB Decompiler, Smart Check by Numega

VI. Conclusions

To support the Reverse Engineering process, tool support which is available is not adequate, also there is lack of process model that support Reverse engineering Processes, there is not necessary learning skills are given to software developer to learn the RE tools and techniques, as a result whatever the he software security and protection techniques, mechanism they developed mostly creaked or break by crackers. There is also lack of research in area of work practices of RE in software security and protection. So purposed model will help the software developer in understanding the reverse engineering and tools used in reverse engineering. Reverse engineering is very complex

and vast field. It is not easy like learning the programming language. It need both hardware as well as programming language knowledge.

References

[1] .A Thorough Investigation on Software Protection Techniques against Various Attacks N. Sasirekha and Dr.M. Hemalatha, Bonfring International Journal of Software Engineering and Soft Computing, Vol. 2, No. 3, September 2012

[2].T. Ogiso, U. Sakabe, M. Soshi and A. Miyaji, "Software Tamper Resistance Based on the Difficulty of Interprocedural Analysis", 3rd Workshop on Information Security Applications (WISA 2002), Korea, 2002

[3] E. Chikofsky and J. I. Cross. Reverse engineering and design recovery: A taxonomy. *IEEE Software*, 7(1):13–17,Jan 1990.

[4]**E. Eilam, "Reversing: Secrets of Reverse Engineering", Wiley Publishing, Inc., 2005.**

[5] Software Protection and its Annihilation- Alfred K M LO ,May 2002

[6] "Crackproof Your Software—The Best Ways to Protect Your Software Against Crackers Pavol Cerven ,2002"

[7] Software Security and Reverse Engineeringhttp://www.infosecwriters.com/text_resources /pdf/software_security_and_reverse_engnieering

[8] Christoph Treude, Fernando Figueira Filho, Margaret-Anne Storey, Martin Salois, An Exploratory Study of Software Reverse Engineering in a Security Context, 2011 18th Working Conference on Reverse Engineering, IEEE.

[9] Tang Jiutao, Lin Guoyuan ,Research of Software Protection, 2010 International Conference on Educational and Network Technology (ICENT 2010)

[10] M. N. Gagnon, S. Taylor, A. K. Ghosh, "Software Protection through Anti-Debugging," IEEE Security & Privacy, vol 5, 2007, pp. 82-84

[11]system monitoring tools ,www.sysinternals.com, written by Mark Russinovich

[12] Preetinder Singh, Richa Gupta,Reverse Engineering A Swiftly Growing Technology in Software World, International Journal of Recent Trends in Engineering, Vol 2, No. 4, November 2009

