Reverse Engineering: Decipher of elementary Software's using rudimentary tools

Romil Gandhi, Deesha Vora, Prof.Vinayak Shinde

Abstract:
Reverse engineering is a process of analyzing, decompiling, debugging, and if required making patches in software for uncovering the limitations.
Reverse engineering:

1) It is a field in computer engineering which helps to decode the applications to solve the determined purpose. It is the last phase after the application is fully developed and tested i.e. when it is in production phase.
2) It can be used by almost all industries for the betterment of software which is self-developed. It can be used to remove the software glitches, to do better with the security aspect and can be used to optimize the performance of software.
3) It is a way where the particular applications executable is open differently to find the source code instead of running normally. It is exactly opposite of the software development cycle which is used to design application.
4) As the name states that going back from the end point to start point in a linear fashion for accessing and uncovering the features of an application.

Some developers/ hackers/ crackers/ reverse engineers use some basic tools to reverse the functionality and security of an application, so it can be used freely and can be distributed to others. This behavior has created threat to the software development industries in the terms of security.
In this paper I will reveal the working of tools for reverse engineering by explaining with two examples.

Keywords:
Reverse engineering, disassembler, obfuscation, Hiew, Ollydbg, HEdit, W32Dasm.

Objectives:
I will show how to reverse the build applications into assembly code using various tools to analyze, crack security, identify bugs, different reversing tools and their uses.

1 INTRODUCTION:

Reverse engineering is re-creating things from the existing, by analyzing the existing code, how it executes, what it performs and then try to produce the same functionality. It is used by competitors for creating applications similar to opposer. Either the competitors are lacking behind or they need to find out the logic, functionalities of an application for developing better one. Some hackers (crackers) use reverse engineering by exploiting the software faults (bug) for strengthening the security of software.
SDLC: Software development life cycle is a standard procedure while developing an application/software. There are multiple model and ways to implement. There are different phases in which SDLC works. The last phase of the SDLC for an application will be the first phase for reverse engineer. After completion of each and every phase the application is launched in the market, then the reverse engineer back-tracks the software, find out all modules and re-creates the application by fixing bugs and put them all together. In this process reverse engineer breaks the security of an application or creates a patch which will have fully functional application which can be used by others. The reverse process SDLC is shown in the figure 1,

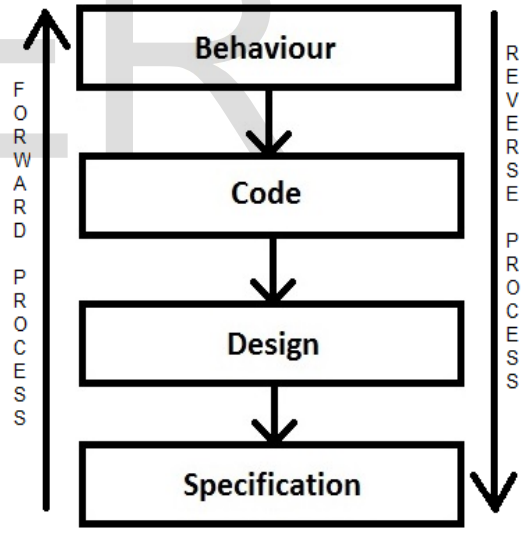

Figure: 1

Ethically, Reverse process can be used by software industries to find the bugs or glitches in software and rebuild before opening in the market. This will help the industry to provide better security to software. If the application is secured then it will be difficult for the reversers to back-track as there will be less chances of a bug in software. So, ethically it can be the advantage for the industry to find out the level of security it develops and makes more profit in the market as it is not easily reversible.
Non-Ethically, the reversers can crack the application code and edit, by removing the limitations, security which will help all to use it freely from internet. This is a real

- *Romil Gandhi is currently pursuing masters degree program in Computer Engineering in Shree L.R Tiwari College of Engineering, India, PH-+91 9819545411 . E-mail: romil.gandhi@gmail.com*
- *Deesha Vora is currently pursuing masters degree program in Computer engineering in Shree L.R Tiwari College of Engineering, India, PH-+91 9819915875. E-mail: voradeesha@gmail.com*
- *Prof.Vinayak Shinde is working with Shree L.R Tiwari college of engineering, Mira Road,India.E-mail:vdshinde@gmail.com*

complication (muddle) for the industry, as it is unethical and illegal. And it is the major disadvantage of reverse engineering.

2 EXAMPLES:

In this paper we will show two examples, how to crack the application, the examples shown here are made for the purpose of education only. The two applications are developed specifically for this purpose of understanding. The applications are login.exe and Keygen.exe.

Requirements: - Tools which we will use are:-

1) Hiew
2) HEdit
3) Ollydbg
4) W32Dasm v8.93

All the above tools used here are for 32-bit processors and 32-bit operating system. Hiew and HEdit are the hex-editors. They convert the executable files into hex-codes. Ollydbg is a debugger which opens executable file in assembly level language. It is a very powerful tool.W32Dasm is windows 32-bit executable disassembler.

We will go first for the simplest one, login.exe. It consists of username 'romil' and some password which we don't know and our job is to find the password for the same. The figure 2 shown is of login.exe with the error message "sorry".

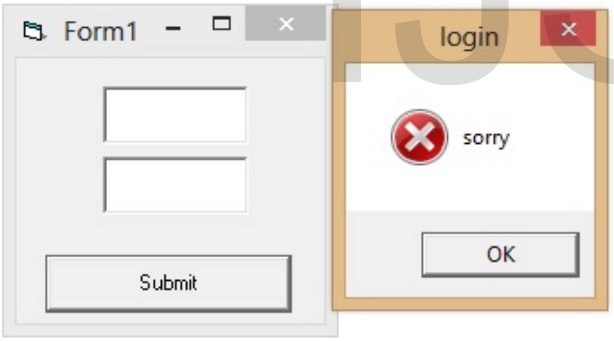

Figure: 2

Now we will open this application in HEdit. Remember the error and the name of the application. As we open the application login.exe in HEdit, the hex code for the login.exe will be displayed. We will search for the keywords like 'username', then try 'login' and we found! That it matches some data inside hex code. The hex-codes are on left-hand side and on right it is the interpretation of hex in English language. You scroll down a little and you see the error message "sorry". There are lots of things written over there, we need to focus on finding 'romil', or 'password', or, etc. Try to insert some words from the values into your password field in application. After some trials you will receive message "Thank u" as shown in figure 3,

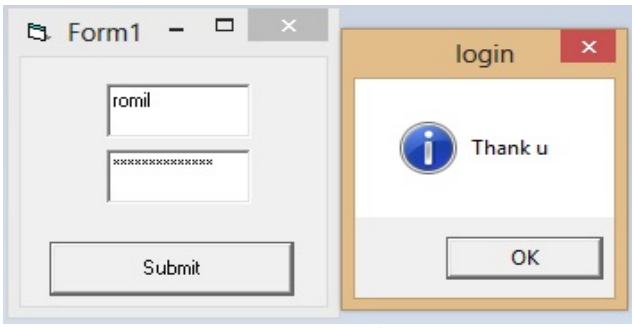

Figure: 3

All the applications are encrypted and so it won't be easy to find the password shown in Example 1. Most of the applications need to be cracked by trial and error method. Now we will look the second Example which is slightly complex than login.exe. Keygen.exe prompts for the serial number and by providing wrong or no serial number it gives "Try again" error message. For finding the error message we use the tool W32Dasm. Hex-editors won't work. We open the Keygen.exe in W32Dasm; we found the error "Try again" on line "226838" as shown in figure 4,

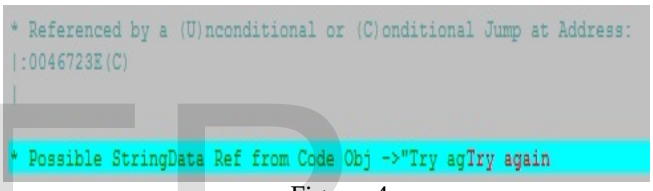

Figure: 4

We see the address ':0046723E (c)', the 'c' in parenthesis states 'condition'. Remember the address and close W32Dasm. Open Keygen.exe application in Hiew. Press 'F4' and select 'decode'. You see assembly level code, and then press 'F5' and type '.46723E' press enter.

You jump to the condition as shown by W32Dasm change the 'jne' condition to 'je' by changing opcode from '75' to '74'. To change the opcode press 'F3'. To save the changes press 'F9' and for the exit press 'F10'.

Run the application, now enter any serial number and it says 'You crack Me' as shown in figure 5,

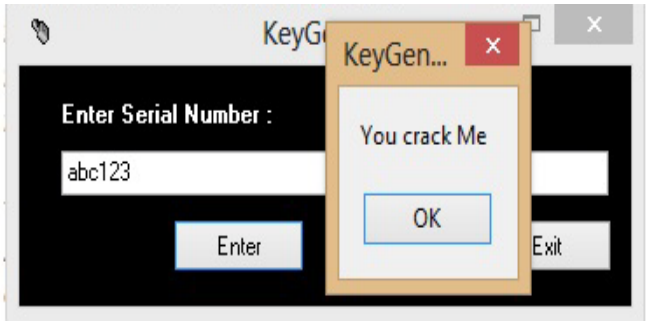

Figure: 5

3 SUGGESTIONS:

There are copious techniques which we can deprive the attack. Always application are assailable. There should be some technique to hedge.

1) Obfuscation of code.
2) Open source Licensed.

3) Runtime checks for debuggers and disassemblers.
4) Casting of code.
5) Dummy calls and dummy functions for confusion.

All the above points help to provide high security but is not and cannot be foolproof. As machine can understand code and so humans can.

4 SUMMARY:

Reverse engineering helps for the betterment of software's security. Reverse engineering ethics are one of the important key points to remember. The examples shown provide overall idea about reversing the application in different way with different tools.

The different techniques for providing security to application by obfuscating code, runtime checks, dummy calls, etc. equip certainty which is crucial to breach.

5 CONCLUSION:

The combination of the techniques provided will provide extremely good security and reversers will have to do further investigation for each and every application. It will provide better security for the application development industry.

REFERENCES:

[1]www.woodmann.com/krobar/beginner/53.htm/collabrative/tools/index.php/w32dasm
[2] www.tuts4you.com/downloads.php?list.17