Department of Cyber Security Systems and Networks

Amrita School of Engineering, Amritapuri

CYBER FORENICS AND INCIDENT RESPONSE

Report on Tool Demonstration

Binary Reversal Using IDA Pro

By

Hrishikesh N

AM.EN.P2CSN16008

1. Introduction

What is IDA Pro?

The official line is: IDA Pro combines an interactive, programmable, multi-processor disassembler coupled to a local and remote debugger and augmented by a complete plugin programming environment. Quite a mouthful, isn't it? We are aware that the above speaks only to geeks. The “raison d'être” of this small document is to clarify the nature and the purpose of IDA to the non-technical user.

IDA Pro is a disassembler

As a disassembler, IDA Pro explores binary programs, for which source code isn't always available, to create maps of their execution. The real interest of a disassembler is that it shows the instructions that are actually executed by the processor in a symbolic representation called assembly language. If the friendly screen saver you have just installed is spying on your e-banking session or logging your e-mails, a disassembler can reveal it. However, assembly language is hard to make sense of. That's why advanced techniques have been implemented into IDA Pro to make that code more readable, in some cases, quite close to the original source code that produced the binary program. The map of the program's code then can be post processed for further investigation. Some people have used it as the root of a genomic classification of viruses. (digital genome mapping – advanced malware analysis)

IDA Pro is a debugger

But, in real life, things aren't always simple. Hostile code usually does not cooperate with the analyst. Viruses, worms and trojans are often armored and obfuscated. More powerful tools are required. The debugger in IDA Pro complements the static analysis capabilities of the disassembler: by allowing an analyst to single step through the code being investigated, the debugger often bypasses the obfuscation and helps obtain data that the more powerful static

disassembler will be able to process in depth. IDA Pro can be used as a local and as a remote debugger on various platforms, including the ubiquitous 80x86 (typically Windows/Linux) and the ARM platform (typically Windows CE PDAs) and other platforms. Remote debuggers are very useful when one wants to safely dissect potentially harmful programs.

Some IDA debuggers can run the application in a virtual environment: this makes malware analysis even safer.

IDA Pro is interactive

Because no computer can currently beat the human brain when it comes to exploring the unknown, IDA Pro is fully interactive. In sharp contrast with its predecessors, IDA always allows the human analyst to override its decisions or to provide hints. Interactivity culminates in a built-in programming language and open plugin architecture.

IDA Pro is programmable

IDA Pro contains a complete development environment that consists of a very powerful macro-like language that can be used to automate simple to medium complexity tasks. For more advanced tasks, our open plugin architecture puts no limits on what external developers can do to enhance IDA Pro's functionality. One could, for example, extend IDA Pro with a MP3 player and make malware sing. However, we suspect our governmental customers are involved in more serious projects.

2. Why IDA Pro?

Hostile Code analysis

Given the speed and the complexity of today's hostile code, a powerful analysis solution is required. IDA Pro has become such a standard in the field of malware analysis that information about new viruses is often exchanged under the form of “IDA Databases”. The 2001 “CODE RED” incident is typical of what usually happens in the background. When eEye isolated a new worm whose payload targeted the White House's web site, IDA Pro was used to analyse and

understand it: it helped the talented eEye analysts deliver a prompt and accurate warning of the impending attack. 24 hours a day, seven days a week, somewhere in the world, thanks to IDA Pro, anti-virus analysts investigate new virus samples and provide timely solutions.

Vulnerability research

IDA Pro is the ideal tool to investigate why software breaks. While the topic of vulnerability disclosure remains, more than ever, controversial, one cannot ignore the fact that software is unfortunately often vulnerable to outside attacks. It is, without any doubt, a bad idea to let vulnerabilities lurk in essential pieces of software: if they aren't fixed, they could very well be exploited by what we usually call “the bad guys”. The Wisconsin Safety Analyzer is a very interesting project investigating software vulnerability and tamper resistance where IDA Pro plays an important role.

COTS validation

A lot of software is now developed outside the country where it is used. Since programs are incredibly hard to verify, since complete source code audit and rebuilds aren't always practical, tools, such as IDA provide a convenient means to check if a program really does what it claims to do.

Privacy protection

Software is invading our lives at every level. In some cases, such as in what is known as the “Sony Rootkit” story, IDA Pro helps protect your essential rights.

Who are IDA Pro users?

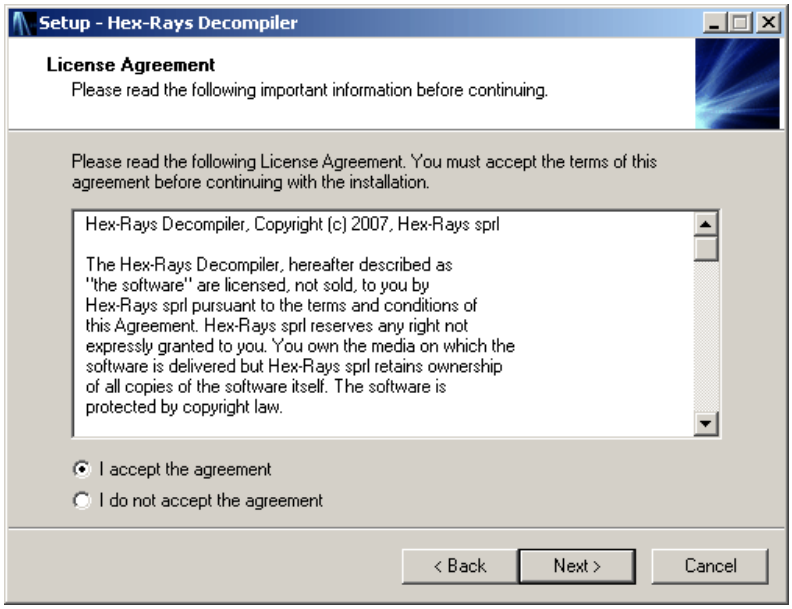
Virtually all anti-virus companies, most vulnerability research companies, many of the large software development companies and, above everything else, three letter agencies and military organizations.

3. Installation Of IDA Pro

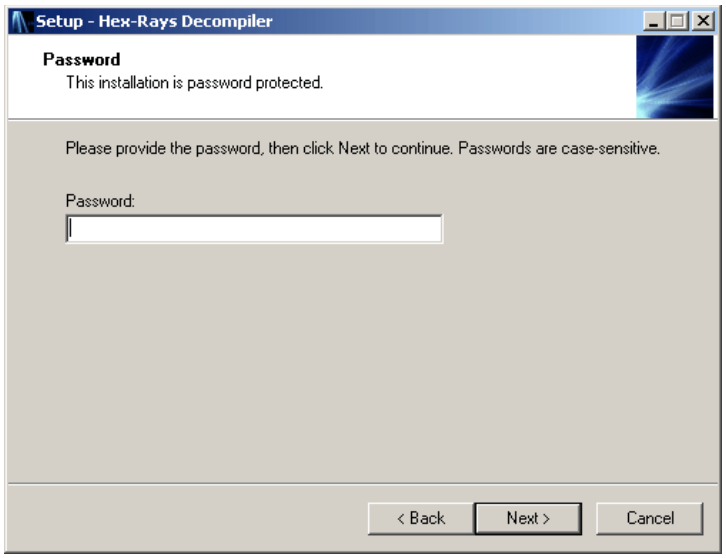
The installation procedure is pretty standard and should not pose any problems. Normally all your decompilers will be bundled with your copy of IDA and installed together. In some special cases (for example, if your support plan for IDA has expired but for the decompiler has not) you may receive a separate installer. Below are the screenshots illustrating the most important installation steps in detail.

1. Run hexrays_setup_....exe. The following initial screen should appear:

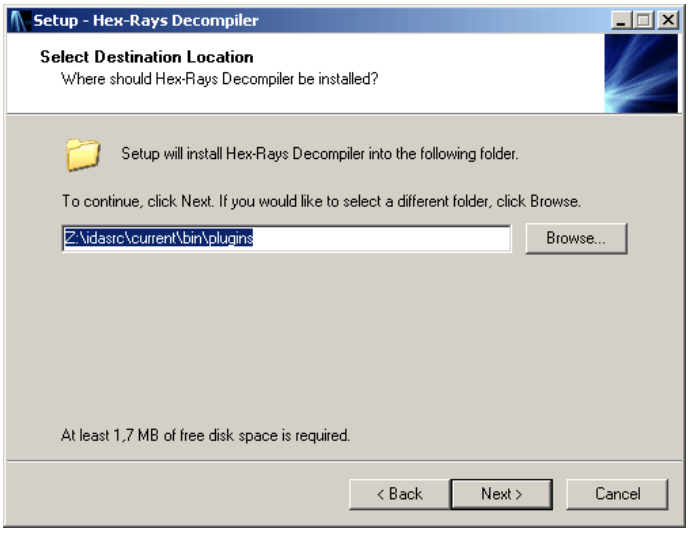
2. Click next, read the license and accept it:



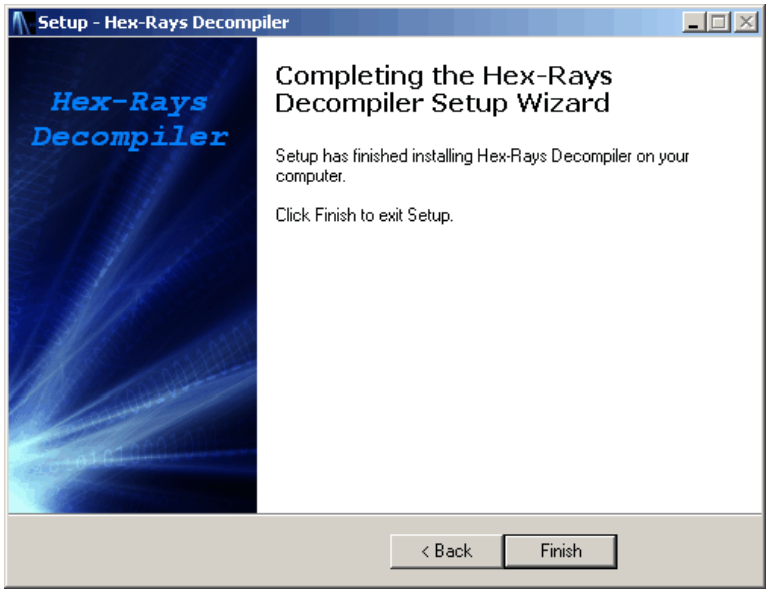
3. Click next and enter the installation password. You can find it in the email message about your order. We recommend to copy and paste it.



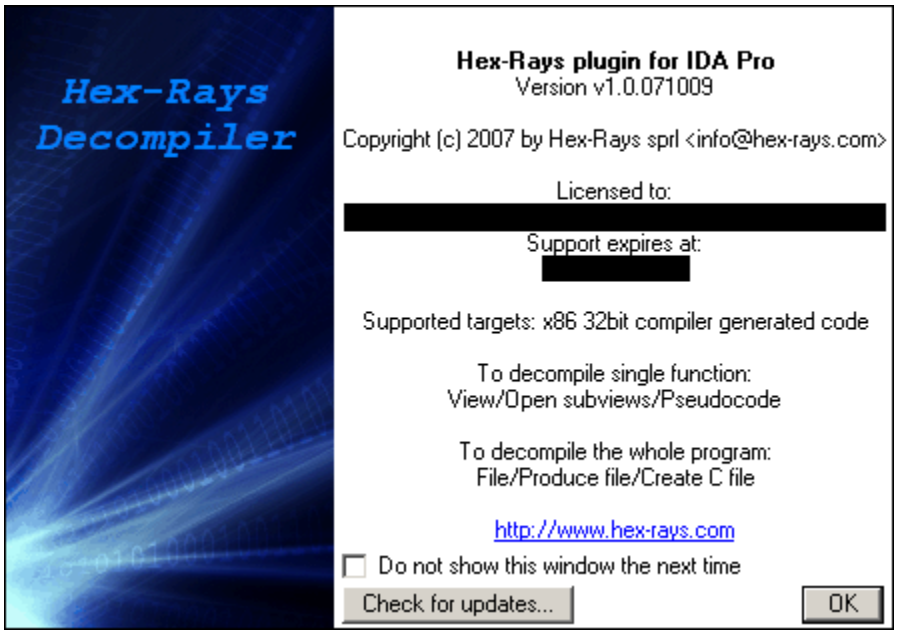
4. Click next and specify the directory to install the plugin. The directory to install in the "plugins" subdirectory of your IDA installation. The setup tries to guess it but it can fail. In this case, click Browse and specify the IDA directory. The setup will append the "plugins" subdirectory to the selected path. If the specified directory is not an IDA directory, the setup will display an error message.



5. Click next a few more times to install the plugin. At the end the following dialog will be displayed:



6. Restart IDA and load any 32bit x86 file. The decompiler will display the following splash screen:



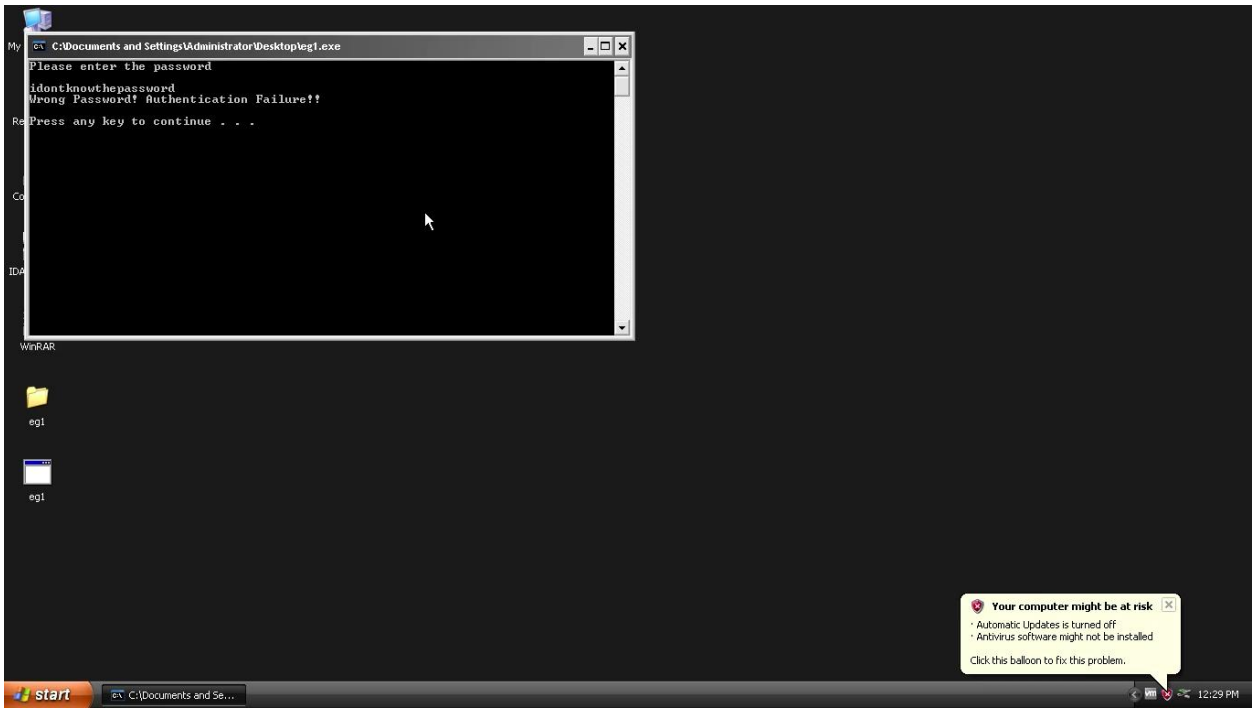
The IDA Pro decompiler is now ready for use.

4. Demonstration

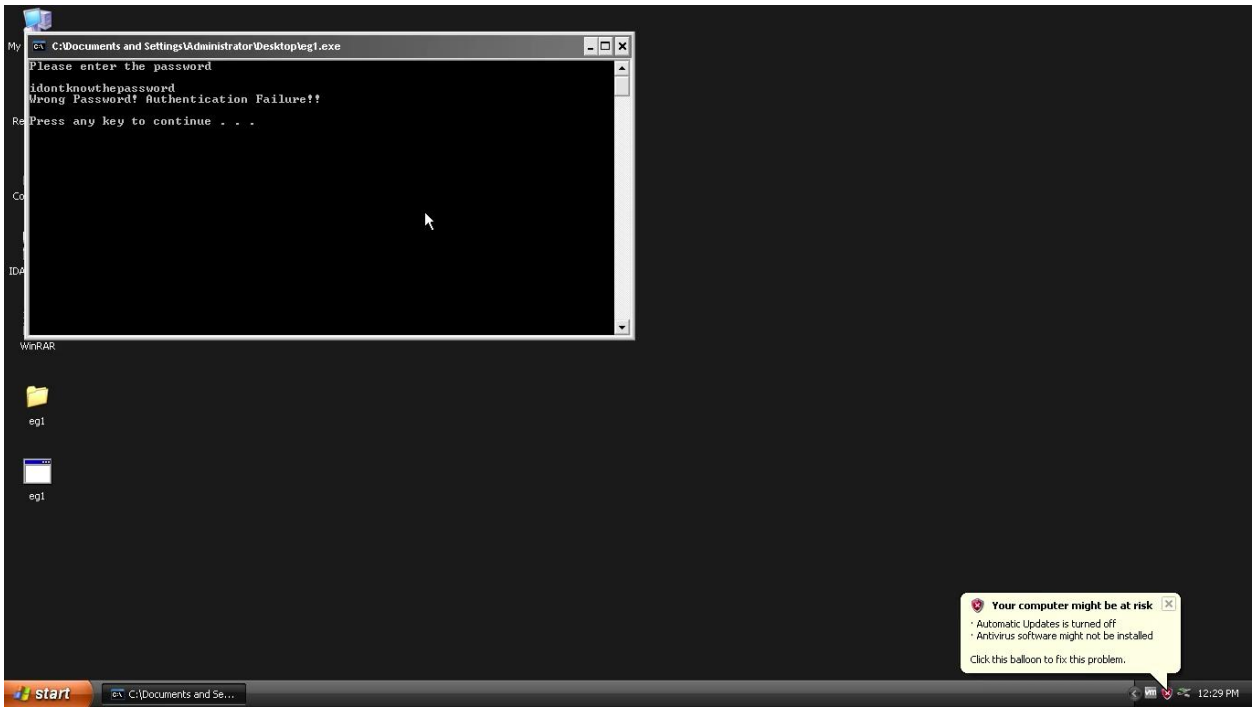
For the purposes of demonstration a simple password check binary is used and the process of reversing and bypassing the password check is performed using the tool.

Task 1 : Bypass password verification

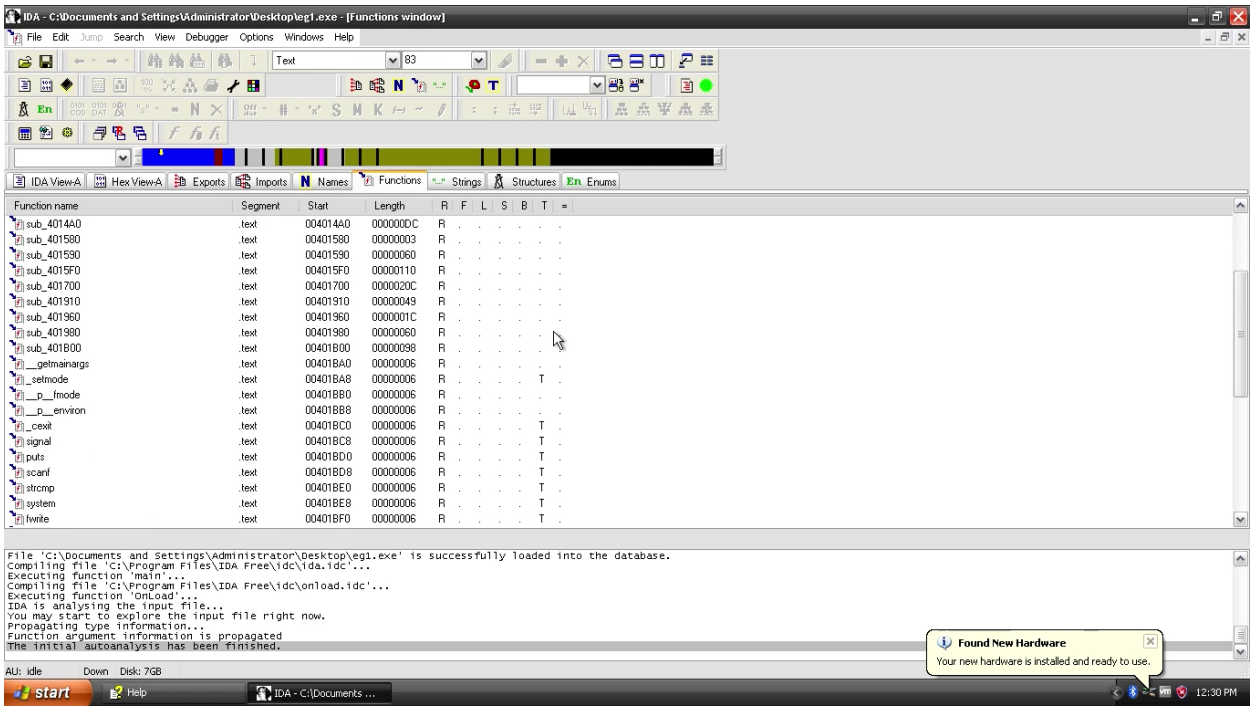
1. The binary is run and a random password is filled for demonstration to show how it fails the verification procedure

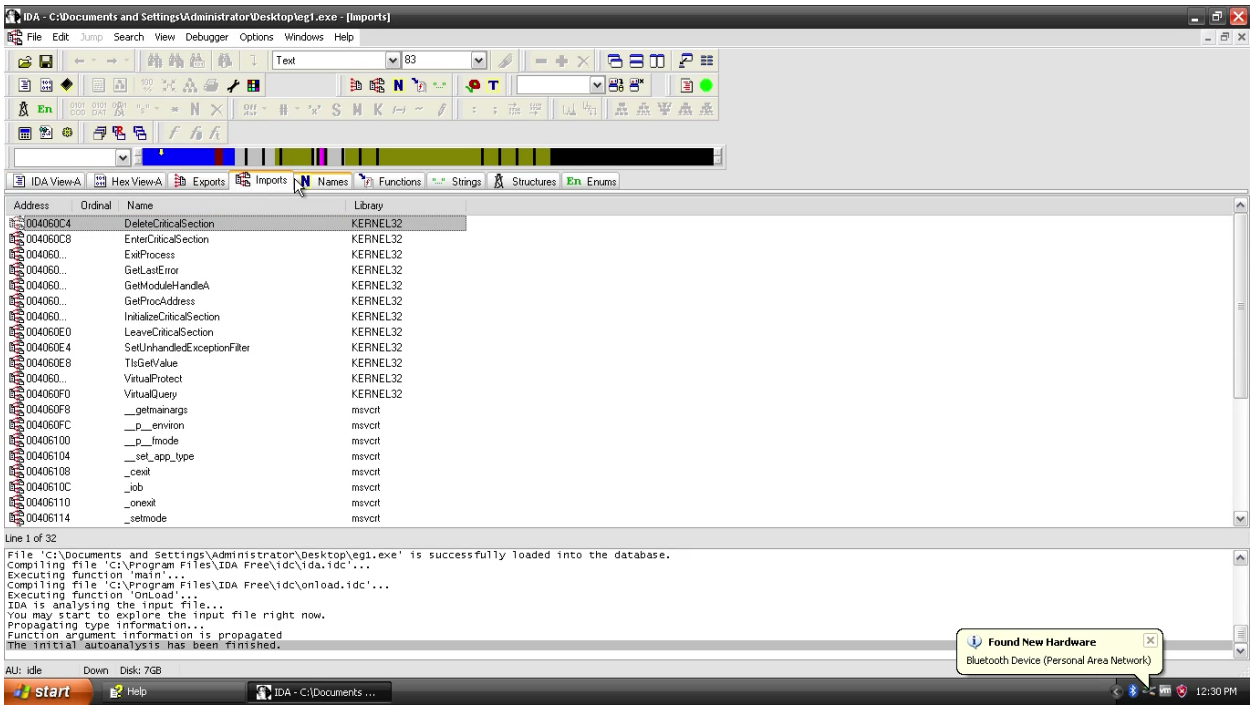


2. Opening the binary in IDA Pro:

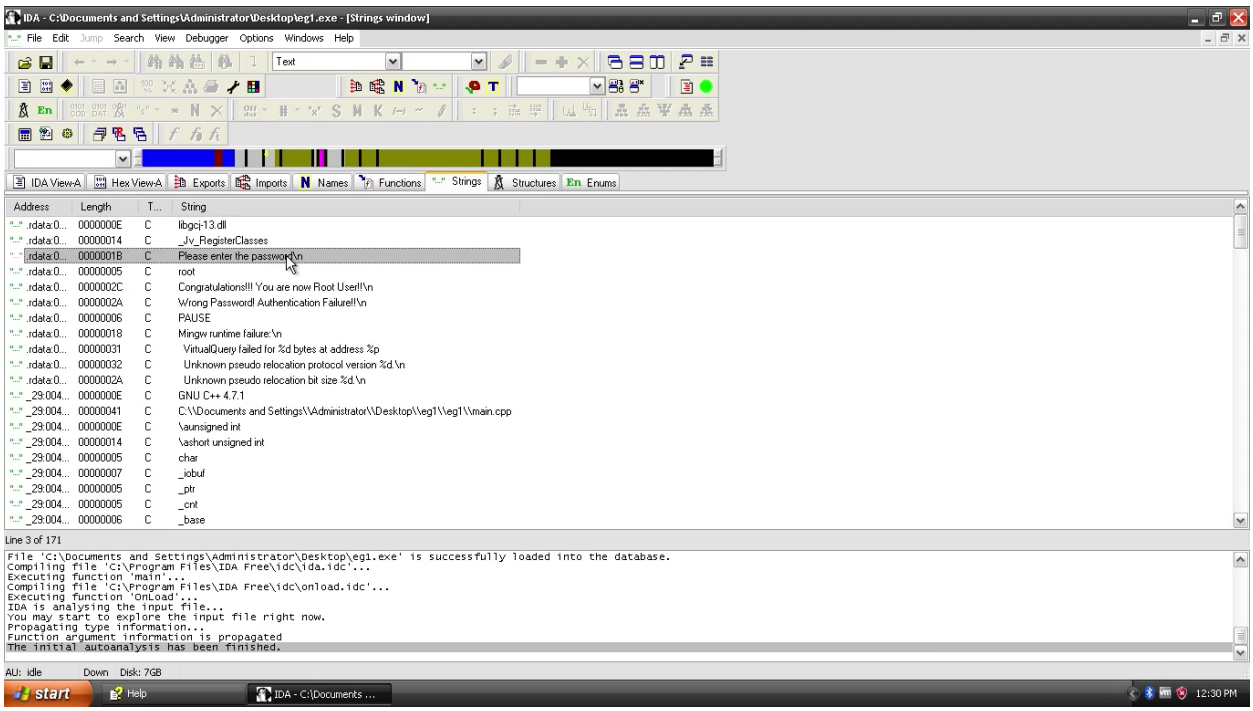


3. The different functions and imports used by the binary

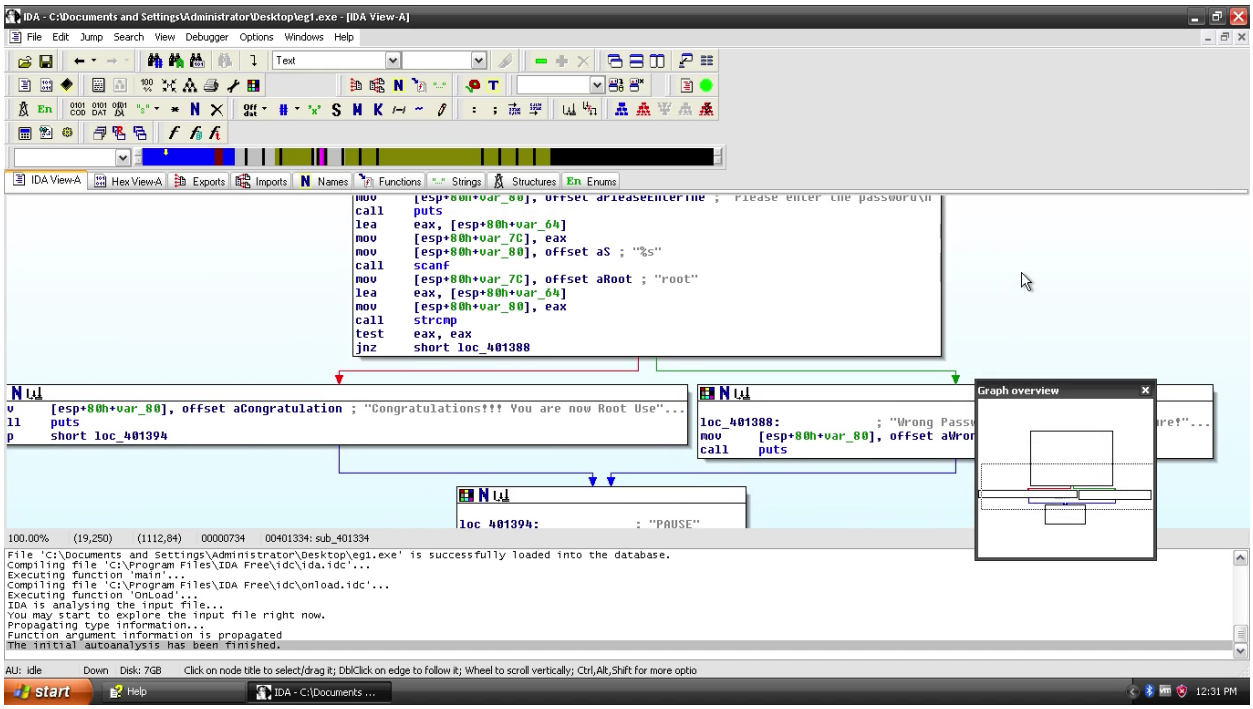




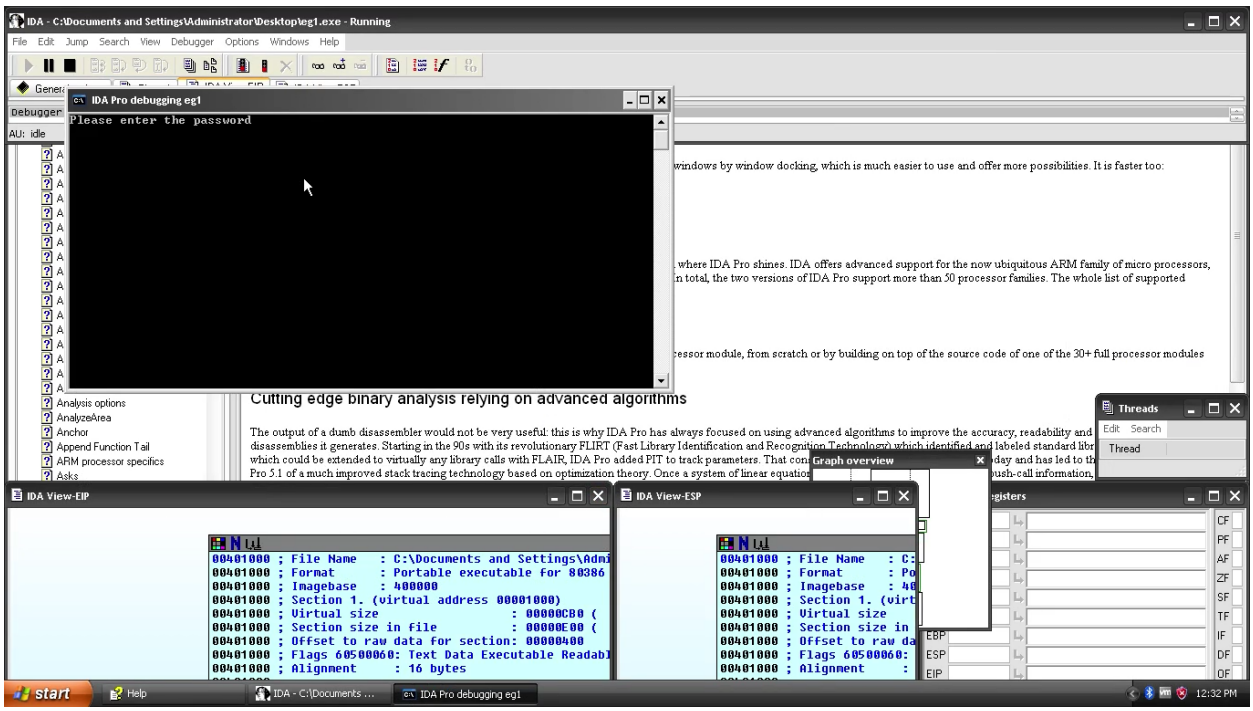
4. List of all strings as part of the Binary



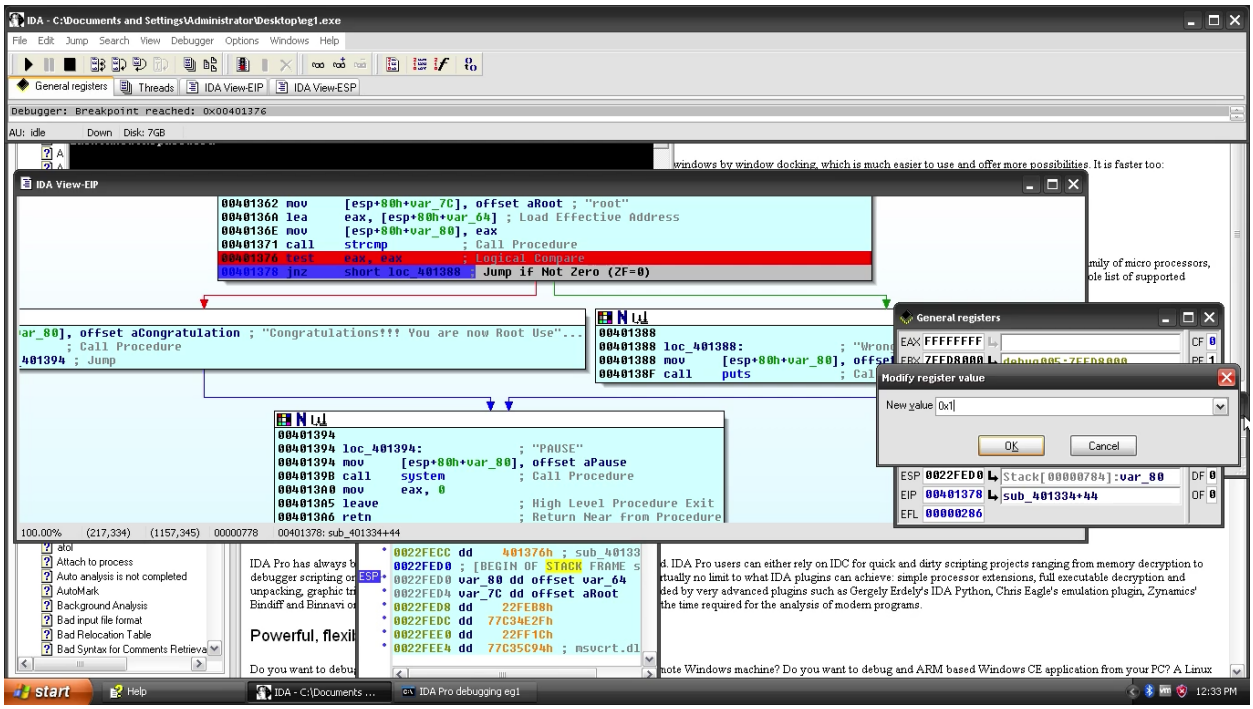
5. Locate the function where the verification of the password happens



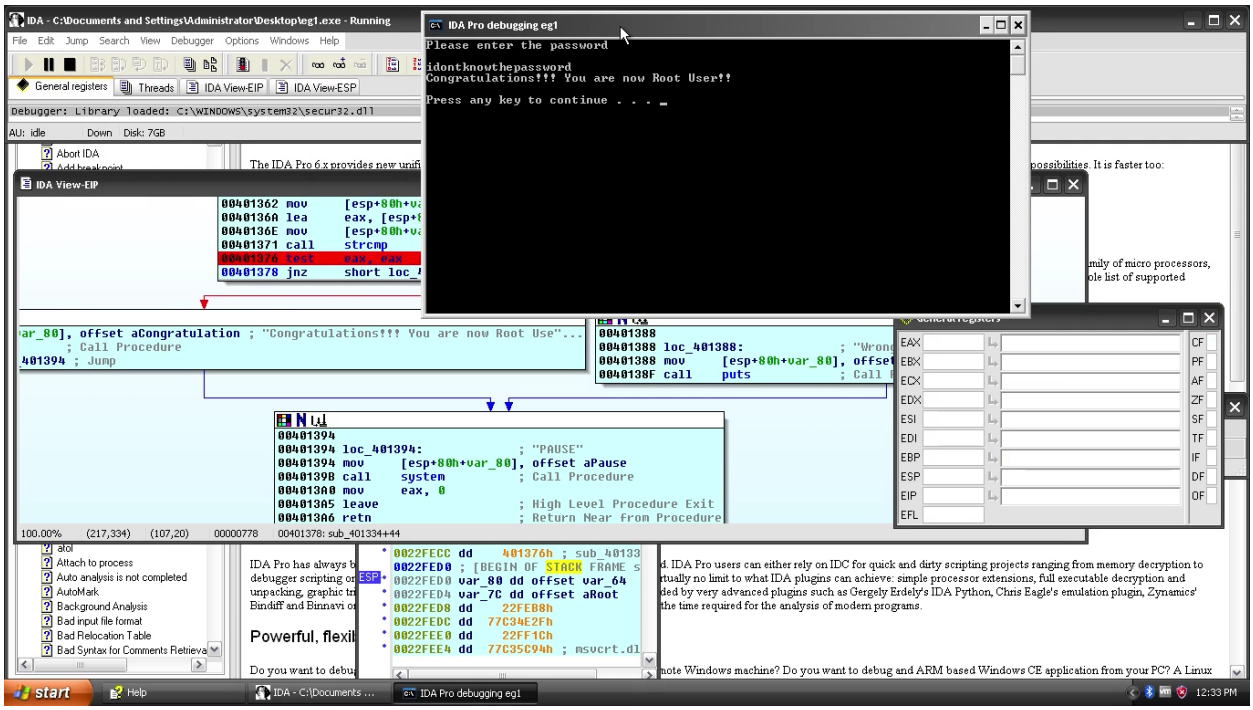
6. Enter debug mode



7. Changing the default registry values to affect the jump branch

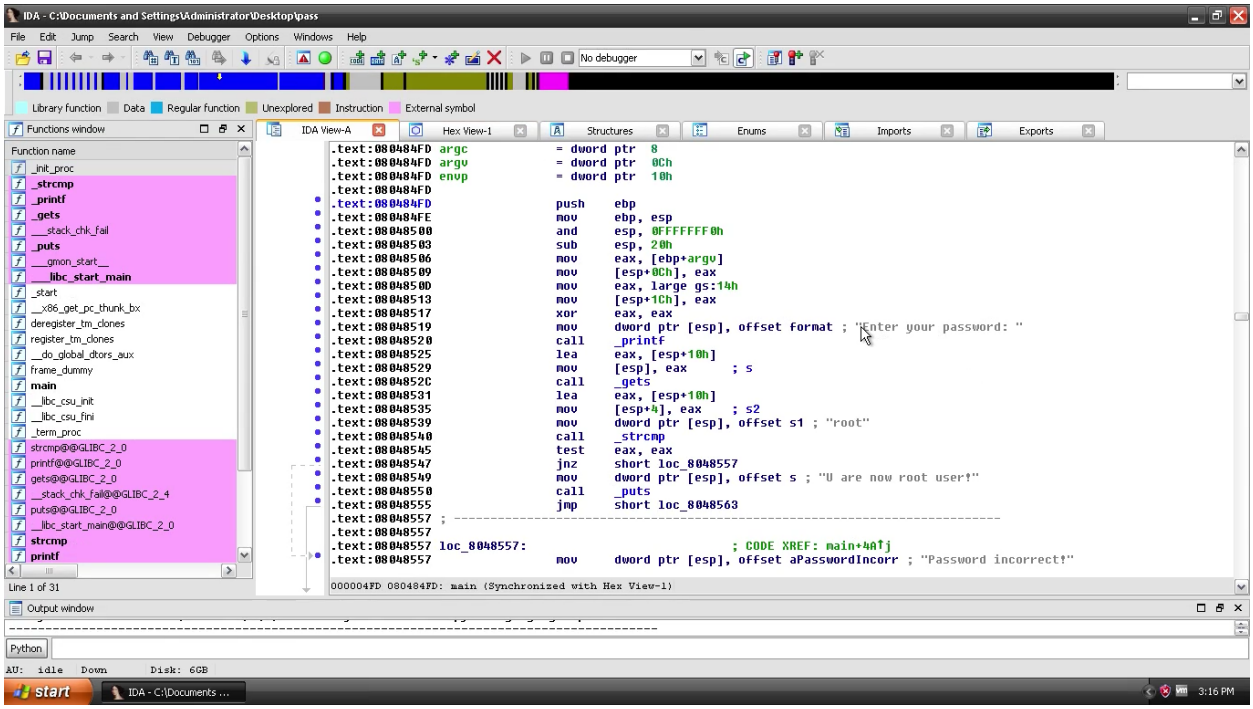


8. Continue running the program and observe the intended effect. The flow jumps to the branch we intended it to branch thus bypassing the check verification feature

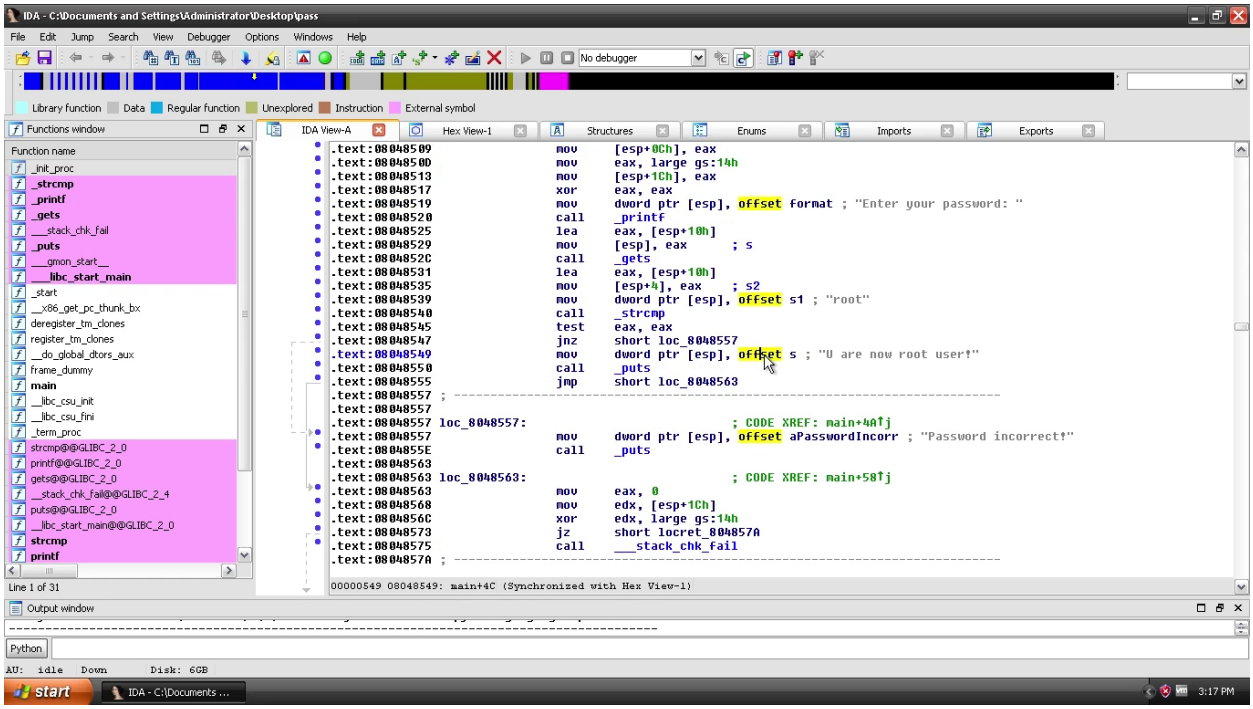


Task 2 : Patching the Binary file

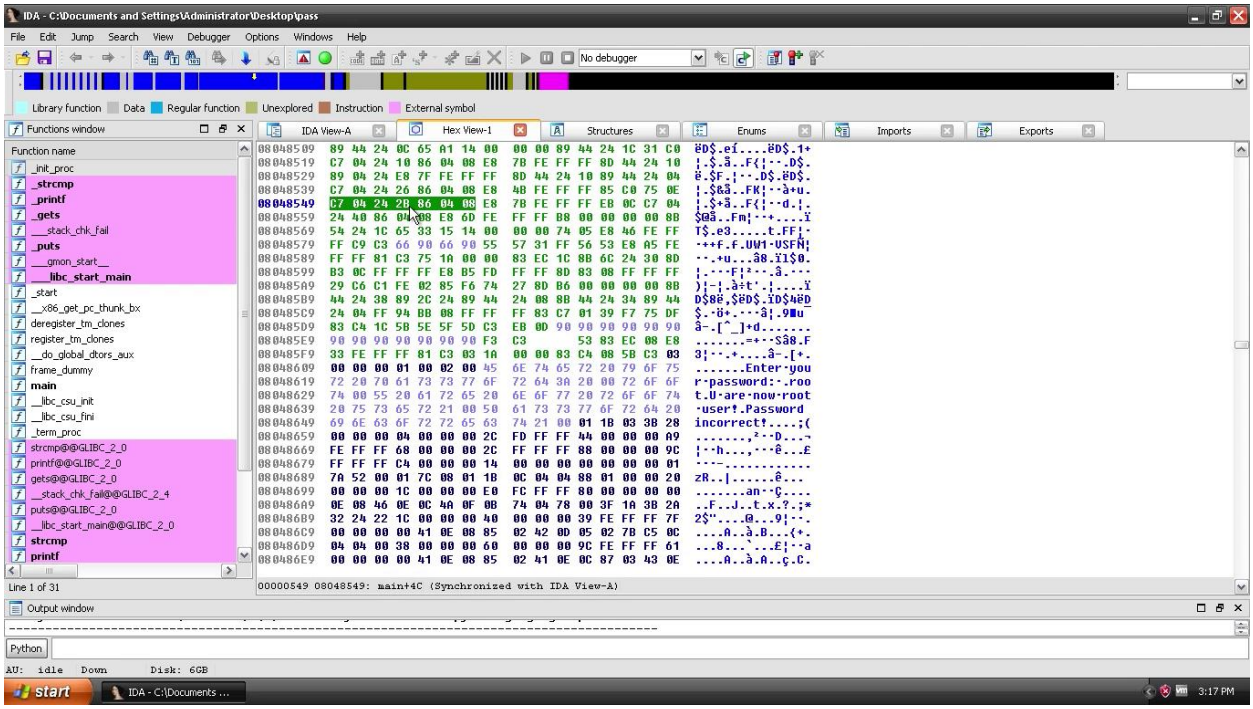
1. Open the binary file in IDA Pro



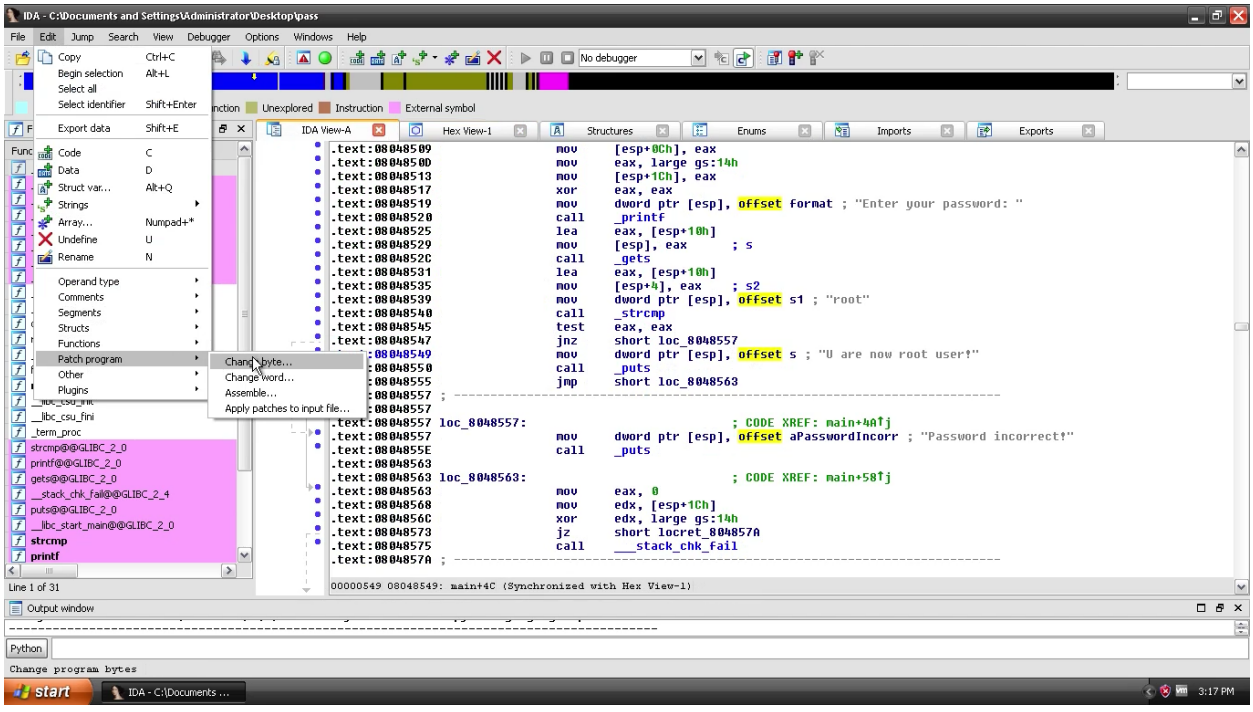
2. Locate the offset address value where we need to identify as the return address



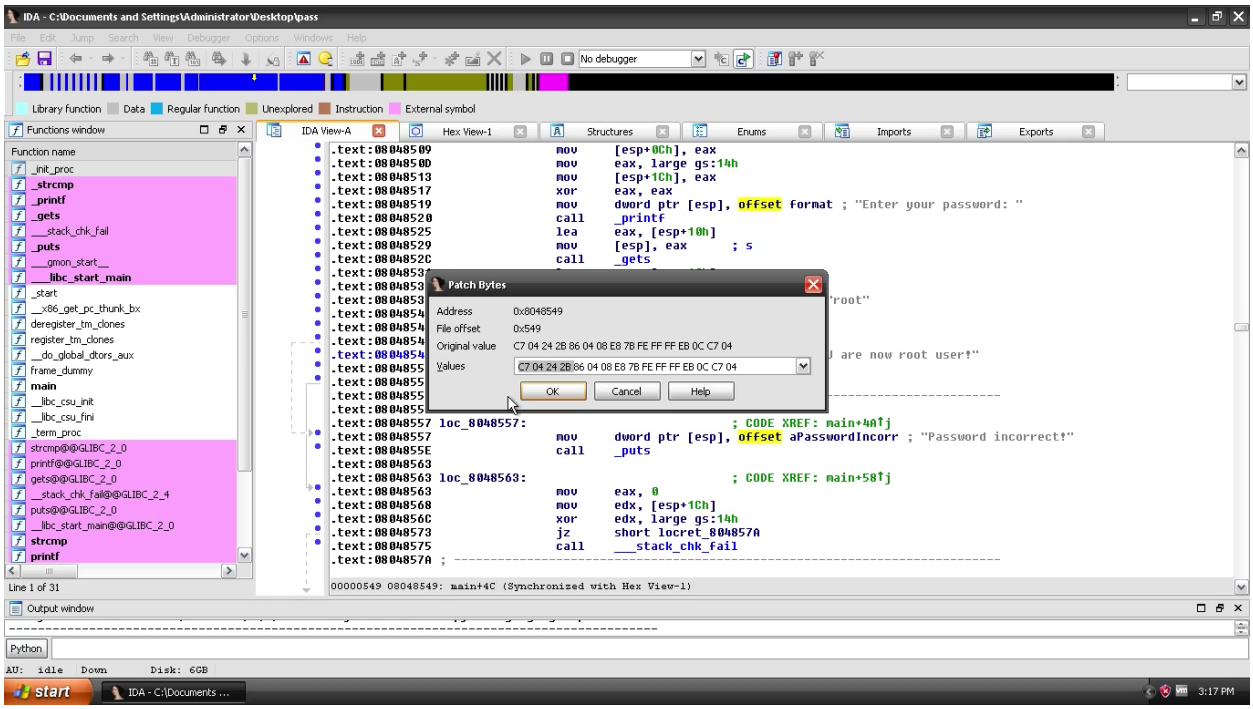
3. Hex address value



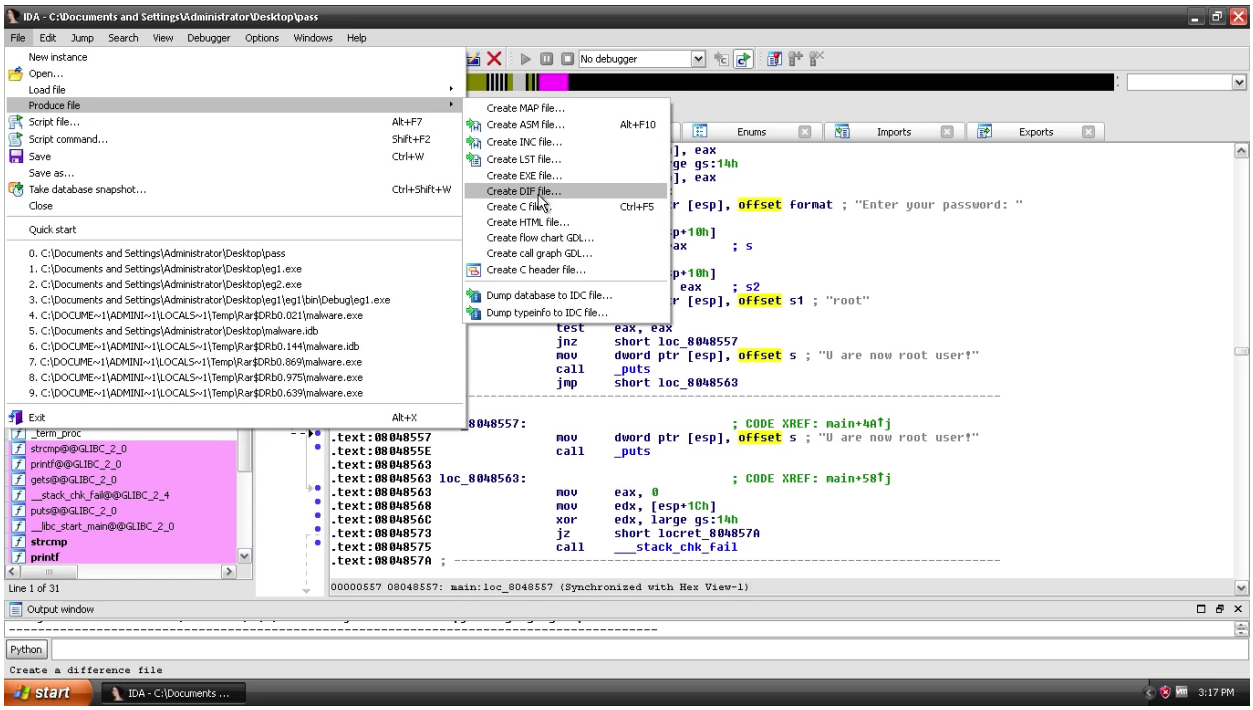
4. Choose the change byte option from patch program under edit menu



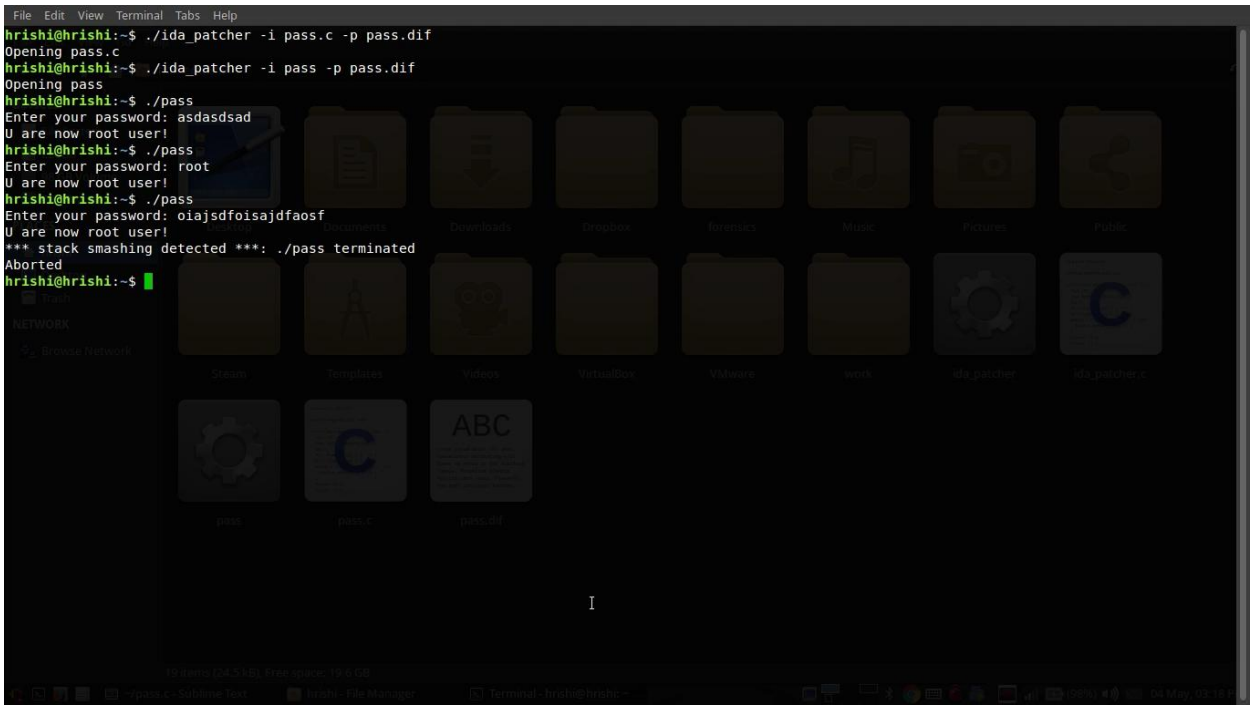
5. Copy the address that needs to be changed



6. Modify the address for the same for the offset address of the required function.
7. Choose the option as shown below and create a .dif file



8. Run the idapatcher program to patch the original binary and observe the results



4. Conclusion

IDA Pro is a well-known tool for malware reversal and forensic behavioral analysis of binary executables. This document has proposed the research on the application of this tool to collect behavioral and runtime data on an executable which can be further used to identify its behavior and reverse its malicious behavior.

References

1. IDA Pro Documentation - https://www.hex-rays.com/products/ida/support/tutorials/index.shtml
2. Christodorescu, Mihai ; Jha, Somesh: Static Analysis of Executables to Detect Malicious Patterns, Wisconsin University, Madison, Department of Computer Sciences (2006)
3. Malware Analysis - https://blog.malwarebytes.com/security-world/2012/09/so-you-want-to-be-a-malware-analyst/
4. Reverse Engineering - https://jlospinoso.github.io/developing/software/software%20engineering/reverse%20engineering/assembly/2015/03/06/reversing-with-ida.html
5. Intro to IDA Pro – http://securityxploded.com/reversing-basics-ida-pro.php