Applying Process Observation Techniques

Lab Manual

Objectives

- Use soft hooking with Python Debugger
- Use hard hooking with Immunity Debugger
- Demonstrate Python to IDA Pro API
-

Table of Contents

1.

Background Reading

Textbook:

Grey Hat Python, chapter 6

Other resources:

https://www.sans.org/reading-room/whitepapers/malicious/basic-reverse-engineering-immunity-debugger-36982
https://sgros-students.blogspot.ca/2014/05/immunity-debugger-basics-part-1.html

Notes common to all lab and home assignment problems

For every lab and home assignment, all work should go into your personal repository, subdirectory named mXX, where XX stands for the module number. For each problem, carefully name the program as described. The programs are extracted from your repository by a Python script, and errors in the program name will result in the instructor never seeing your program, and your mark for it will be ZERO!

Anything to record and report in this lab is to be written in plain text file (Word document is not a plain text file). The file MUST be named mXXlabrep.txt. Mis-naming this file, or not having it in the proper location will result in mark ZERO for anything to be recorded or reported.

There are always many ways how to solve a programming problem, and usually one or two ways which are fast, compact and elegant.

Make sure to push your work to the server often, and have pushed the working version of the program by the deadline specified. The script extracting your programs from your repository will be run at any time after the deadline.

Lab specific intro

In this lab we will explore the Immunity Debugger further, and demonstrate soft and hard hooking techniques. Read the Grey Hat Python book, chapter 6.1, which describes soft hooking with PyDbg debugger. In our first problem, we will place soft hook in Firefox.

The PyCommands are found in:

“/c/Program Files/Immunity Inc/Immunity Debugger/PyCommands”

To be able to edit files at that location, start your code editor (e.g., emacs) with admin privileges.

Problem 1

The task is to intercept non-encrypted login and passwords in Firefox. We will access encrypted web

page at https://itss.biomea.com/login.html [replace with sait local website]. The page requests user id and a password, and the Firefox browser sends this information encrypted over the internet. We want to exploit the username and password information, before it is encrypted and sent to the server. The general concept follows the chapter 6.1 in the Grey Hat Python book, however, you will write the exploit as Immunity Debugger PyCommand script, named **`m08p01.py`**. The result should be plaintext username and password in the Immunity Debugger log file (right click the Log window, choose Log to File, enter filename **`m08p01.log`** at the appropriate module 8 location within your git repository).

Hints:

The dll where **`PR_Write()`** function lives is called **`nss3.dll`**, and is part of the Firefox distribution. There are several very useful functions within the Immunity Debugger libraries:

`callStack()`

`getProcedure()`

`getStackDump()`

`readMemory()`

`getAllModules()`

`getName()`

`setLoggingBreakpoint()`

`getAllSymbolsFromModule()`

Applying Process Observation Techniques

Home Assignment Manual

Objectives

- Use soft hooking with Python Debugger
- Use hard hooking with Immunity Debugger
- Demonstrate Python to IDA Pro API

Table of Contents

1.

Background Reading

Textbook:

Grey Hat Python, chapter 6

Other resources:

https://www.sans.org/reading-room/whitepapers/malicious/basic-reverse-engineering-immunity-debugger-36982
https://sgros-students.blogspot.ca/2014/05/immunity-debugger-basics-part-1.html

Notes common to all lab and home assignment problems

For every lab and home assignment, all work should go into your personal repository, subdirectory named mXX, where XX stands for the module number. For each problem, carefully name the program as described. The programs are extracted from your repository by a Python script, and errors in the program name will result in the instructor never seeing your program, and your mark for it will be ZERO!

Anything to record and report in this lab is to be written in plain text file (Word document is not a plain text file). The file MUST be named mXXlabrep.txt. Mis-naming this file, or not having it in the proper location will result in mark ZERO for anything to be recorded or reported.

There are always many ways how to solve a programming problem, and usually one or two ways which are fast, compact and elegant.

Make sure to push your work to the server often, and have pushed the working version of the program by the deadline specified. The script extracting your programs from your repository will be run at any time after the deadline.

Problem 2

Study the Grey Hat chapter 6.2 in detail. Implement the code **`hippie_easy.py`** and observe its behaviour. Answer the following questions and submit your answers in file m08lab02.txt:

1. What is the difference between STDCALLFastLogHook() and FastLogHook()
2. What does function imm.getKnowledge() return? (is it an integer, a string, a dictionary…?)
3. What is the second parameter to function imm.disasmBackward() ?