**Student Name:** **Student ID:**

# Objectives

* Create a remote thread.
* Inject code into DLL.
* Evaluate file hiding techniques.
* Write backdoor code.
* Compile Python script into a Windows executable.

# Important Information

* For *every* lab and home assignment, store all your work in your personal repository in a subdirectory named **mXX**, where XX is the module number. Carefully name the program as described in each problem.
* Your programs are extracted from your repository by a Python script. If there are any errors in the program name, then your instructor will never see your program, and you will receive a mark of zero.
* Push your work to the server often, and ensure that you push the final version of a program by the deadline specified, because the script extracting them can be run at any time after the deadline.

# Instructions

1. Read chapter 7 in the *Gray Hat Python* textbook. The following links are also useful:
   * <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>
2. Complete Problems 4, 5 and 6.

**Note:** Problem numbering continues from the module’s lab.

## Problem 4

You will create a backdoor to a computer by writing a code which will masquerade as a benign Windows application (in this case, calc.exe).

1. Move the original **calc.exe** file to different location, because you will need to execute the real calculator (since this is what the user is expecting).
2. Use the code snippets in section 7.2.2 in the *Grey Hat Python* textbook to create a complete exploit.
3. Submit the complete exploit as a file named **m09p04.py**.

## Problem 5

1. Convert the exploit you created in Python to a Windows executable file, following the process as described in section 7.2.3 of the textbook.
2. Install the **py2exe** Python library on your Windows machine, write the **setup.py** script to control the build, and then execute it.

## Problem 6

1. Code the **backdoor\_shell.py** server, which will listen on port 4444.
2. Run the program in a separate virtual machine, and then execute the exploit you created in Problem 5 on the victim machine.
3. Copy your interaction with the victim machine into your lab report.