Assignment 4 – Intro to Heap Overflows
Due Tuesday 06/08/21

For this project you are required to understand one or two vulnerable programs and develop exploits for those programs

Setup: The binaries and related source code for this assignment are available on Sakai.

You will need to get the binaries into your virtual machine. You are highly encouraged to get an early start on understanding the binaries by reading the source, running the binaries in your virtual machine and interacting with then. In addition to deducing the behavior of the binaries by interacting with them and reading the source code, you are free to disassemble the binaries using any tools at your disposal (such as objdump or ghidra) and reading through the disassembly to enhance your understanding of the binaries' behavior. You may of course try to deduce the nature of the vulnerabilities by sending unusual inputs, attempting to crash the binaries, and observing crashes in a debugger. For this project, randomization will be turned off and the stack will be executable.

Grading: Providing the deliverables below for either one of the two binaries shall be good for 80% overall credit. Providing the deliverables for both binaries shall earn 100% credit. Turn all answers and other materials in to Sakai by 1700 on the due date.

Deliverables: For each binary that you successfully exploit, answer the following questions and develop the exploit described below.

1. How does this program accept input from a user?
2. Describe the vulnerability present in this program.
3. Discuss any restrictions on the user's input that must be taken into account when attempting to trigger the vulnerability.
4. Discuss the structure of a user input that will successfully avoid any restrictions and allow you to successfully take control of this program. Be as detailed as possible (size, format, content…).
5. Develop an exploit that results in an interactive shell being available to the attacker.
6. For the exploit you develop make diagrams of the heap layout following each malloc and free operation. The diagrams must clearly show each allocated and free block in the heap. The diagram need not be to scale; I am particularly interested in the sequence of the blocks in the heap, including their exact sizes and addresses.
7. Describe all locations that are candidates for the target of your arbitrary write. List only those locations that might successfully be used to transfer control to your payload. Describe each location in as much detail as possible including information such as the address of the location, the purpose of the location, and the program section in which the location resides.
8. Turn in your attack script(s)/source code/command line(s).
9. Provide detailed instructions for executing your attack. If you use a bind shell you must indicate what port the shell will bind to. If you use a callback you must indicate what port your payload will call back to as well as the exact byte offset in your payload at which the 4 byte callback IP may be found so that it may be replaced as necessary. Provide any additional details that you think someone running your exploit might find useful.
10. Arrange to demonstrate your exploit for the instructor NLT 1700 Friday 11 Jun 2021.