Meng-Chieh Lin lin1055@purdue.edu

CS526: Project 1

Solution 1.

(1) **Briefly describe the behavior of the program.**
The program receives two arguments and copy second argument to a fixed size buffer.

(2) **Identify and describe the vulnerability as well as its implications.**
This program has a problem that it does not check the size of the argument in function, which may cause a overflow and let others to rewrite return address.

(3) **Discuss how your program or script exploits the vulnerability and describe the structure of your attack.**
My program use a NOP sled and some padding characters to enclose a shellcode. Then through exploiting $strcpy()$ by injecting a designed value into the $\$eip$ register, my program can redirect to root shell with that enclosed shellcode.

(4) **Provide your attack as a self-contained program written in Python.**
The code is in sol1.py.

(5) **Suggest a fix for the vulnerability. How might you systematically eliminate vulnerabilities of this type?**
Use a function to check both input sizes and buffer sizes when there is any read/write related function.

Solution 2.

(1) **Briefly describe the behavior of the program.**
The program copy the argument and to the fixed size buffer.

(2) **Identify and describe the vulnerability as well as its implications.**
The memory location of $*p$ and $'a'$ are concatenated with the buffer, and the $strncpy()$ allows user to write bytes without checking the buffer size. This is the vulnerable of this program.

(3) **Discuss how your program or script exploits the vulnerability and describe the structure of your attack.**
My program uses the assignment statement to store the address of the shellcode to return address.

(4) **Provide your attack as a self-contained program written in Python.**
The code is in sol2.py.

(5) **Suggest a fix for the vulnerability. How might you systematically eliminate vulnerabilities of this type?**
Use a function to check whether bytes to write is equal or less than the buffer sizes.

Solution 5.

(1) **Briefly describe the behavior of the program.**
The program basically does the same thing in vulnerable1.

(2) **Identify and describe the vulnerability as well as its implications.**
This program has a problem that it lets user to write as many bytes as he wants, while the return address of main function is stable.

(3) **Discuss how your program or script exploits the vulnerability and describe the structure of your attack.**
My program put the return address of main function to the return address of vulnerable function, and writes the shellcode over return address. Thus, the instruction pointer will go the shellcode every time.

(4) **Provide your attack as a self-contained program written in Python.** The code is in sol5.py.

(5) **Suggest a fix for the vulnerability. How might you systematically eliminate vulnerabilities of this type?**
Make the buffer size smaller so that user cannot write to main address.