PeanutPower: CS557 Project 2

Thomas Le Baron
Computer Science
Worcester Polytechnic Institute
Worcester MA, USA
tlebaron@wpi.edu

Brittany Lewis
Computer Science
Worcester Polytechnic Institute
Worcester MA, USA
bfgradel@wpi.edu

***Abstract*—In this work we introduce a vulnerable binary "PeanutPower" which has a format string vulnerability. We also prove that this binary can be explioted to launch bin shh through execve. We do this through overwriting the global offset table using a format string vulnerabilitiy. We are then able to use return-oriented programming to utilize a series of gadgets that allow us to launch a shell.**

I. Design

A. Format string vulnerability

The core of our vulnerability exists in that in the fuction Peanut we pass an arbitrary buffer of user input (gathered using fgets) to the function printf. This introduces a format string vulnerability that will allow us to "Write What Where" using the special format string %n. In our exploit, we will use this vulnerability to overwrite the Global Offset Table (GOT).

B. Exit and the Global Offset Table

Our vulnerable function, Peanut contains a call to Exit before the end. This means that we cannot overwrite the return pointer since the return pointer will never be called (we will exit before then). However, exit will be called from the Global Offset Table. This means that if we can overwrite the address to exit in the GOT, we will be able to redirect the programs control flow

C. Execve

In order give a big "Look over here" hint as well as make our exploit easier, we have embedded the arguments to call execve onto the stack. This will make it easier to call bin shh from execve. We believe that it could also be called by embedding those arguments in the buffer.

Note that we did explore calling MProtect to make the stack executible to call our shell code instead of calling execve directly using gadgets. While we got this working in GDB, we could not do so outside of the debug environment as Mprotect needed to know the exact start of the stack page and it's exact size in order to work correctly, and we were unable to find this information with our avaliable program. This is something we would like to attempt again in future work.