CSE4501 – Vulnerability Research: Midterm

Eric Pereira

October 24th, 2019

Contents

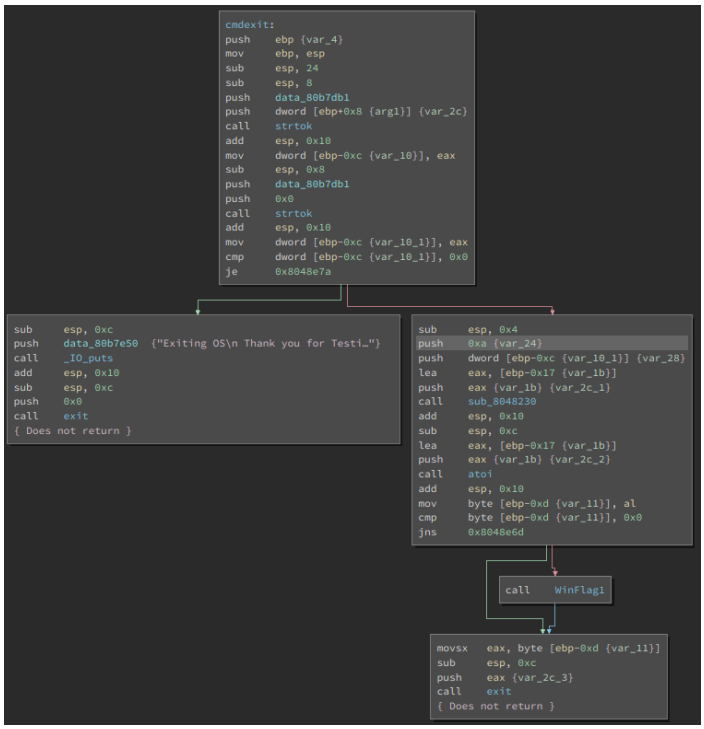
Introduction

:

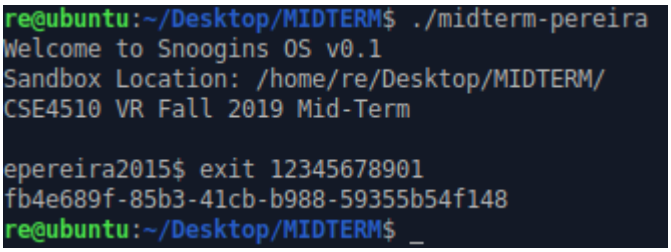
Snoogins OS is a brand new, state of the art, high tech, machine blockhain learning operating system that has incredibly useful commands. It, however, seems to possibly have at least 3 vulnerabilities. For finding at least 3 vulnerabilities I will be paid with 100 points (that's more points than I've ever seen in my life!). With that said, let's start looking for the the first vulnerability, referenced in the `WinFlag1` function.

WinFlag1

Upon Reviewing, My first instinct was to look immediately for the `WinFlag` functions and cross reference them to see where they are called. The first flag, `WinFlag1` can be found in another function, the `cmdexit` function. This function is called when you run the Snoogins OS and type the command `exit`. I then went to observe the specific code to see exactly what was happening. The graphical view of the code looks like:



What I see is right above where Winflag would be called, a variable is created with allocated memory size of 10 bytes, that would be seem near the `strtok` function. This seems to be some sort of arguement that is called in conjunction with exit, and scanned in when exit is called. If exit has a non-null arguement of some sort it goes to the path on the right, and then it seems that there is a check of some sort on the stack after 10 bytes. So, in order to get the first flag you have to give some input of at least 11 characters passed in with the `exit` command. Let's see what happens when we run this command.



There it is! The first vulnerability, represented by the UUID or GUID (or whatever term you want to call them) which is:

`fb4e689f-85b3-41cb-b988-59355b54f148`.

WinFlag2

I initially had quite a bit of trouble with the second flag, I went to use the tool Ghidra for more help as a result. I had a strange amount of difficulty cross referencing the `WinFlag2` function. Very strange. I then opened up in IDA64 only to get a notification that "There are no xrefs to WinFlga2". That, is interesting, so I supposed I am chasing the wrong flag, or I need to find some way to run the flag indirectly. I am suspecting that we need to generate the buffer of data plus the address of the function in order to run it. Let's see if we can attempt to do this. In order to do this, however, we have to look and find a potential exploit where we can place the value of a specific address into a function.