Topic:

I am choosing a custom topic for my project.

In my project, I will explore the factors influencing an attacker's ability to discover gadgets in executable code for return-oriented-programming (ROP) exploits.

I was initially intrigued by the thesis of the paper "The geometry of innocent flesh on the bone: return-into-libc without function calls (on the x86)", which states that sufficiently large x86 executable binaries will contain gadgets that enable arbitrary computation. From my knowledge on gadgets, the attacker's goal of finding a sequence of gadgets to get the program to act in the manner of the attacker's choosing seems nearly impossible considering the state space explosion that would result from searching for this. Manipulating a program to act as an attacker desires is difficult enough using simple and direct code injection, and the W XOR X protections make this task much more difficult. Probabilistically, the 0x3c byte will only only occur once ever 256 bytes.

Experimental Design:

I will create a number of executable binary test programs for x86. These test binaries will vary in levels of compiler optimizations, size, structure (dense loops vs sparse function calls), and defenses (ASLR on/off). I will then utilize a gadget search tool (likely ROPgadget) to scan each test binary for gadgets. I will record the number of gadgets discovered, the effect of each gadget, the density of gadgets, and the amount of time to discover the gadgets. Then, I will analyze how the independent variables of the test binaries influence these dependent variables related to gadget discovery.

I think it would be very interesting to measure not only how automated tools are able to discover gadgets, but also how these tools utilize gadgets for exploits. As I conduct my project, I will consider how to modify this experiment to analyze how these independent variables in the test binaries influence the ability of automated tools to craft a sequence of gadgets that can successfully trigger a payload (launch a command shell/make an execve() call), but this seems a bit complex and there is not a lot of information on available automated tools that can do this.

Challenges:

The biggest challenge I expect to encounter in this experiment is using gadget search tools like ROPgadget or Ropper since I have no experience using them previously. I think it also might be difficult to quantify the independent variables like structure, and there may be bias in how I group the effects of gadgets discovered. I am also unsure if time to discover gadgets is a useful metric.

Feel free to provide any feedback or modifications to this experiment. I will certainly meet with you during office hours to discuss the my approach for this project and receive some guidance.