Git Got Good

I solved this challenge using a combination of static and dynamic analysis. First, I ran checksec and determined that partial relro is enabled, which means that the .got.plt table can be written to. There is also a function called run_cmd in the binary and puts is called after the program asks for input. Thus, I can conclude that the .got.plt table should be edited so that calls to puts are redirected to run_cmd. Next, I used dynamic analysis to figure out how to send in the payload. My first argument was the address where puts is located in the .got.plt table - 8. This is because the assembly increments by 8 and then overwrites the value at that address with the second argument we send. Since we want to redirect calls to puts to run_cmd, the second argument should be the start address of run_cmd. Finally, the last argument should be "/bin/sh" because rdi is the register that stores the first argument to functions, and passing "/bin/sh" as the last argument will allow it to be stored in rdi, since it gets passed from rax to rdi. Now run_cmd will run, having "/bin/sh" as its argument, which will give us the shell and thus the challenge is solved.