Arjun Gupta
EECE 5698
April 25th 2019

Lab 7 Writeup

The retrieved secret was
"**596rokwrzg8pnl1fweopyat13m3w7r6saau6khb6puna4aigpqebov49xsjxjzbavw9owxlzof4**"

Stack canaries are bad when the attacker has an idea for how a program's stack is laid out. If they have access to an executable binary then they can disassemble the binary to understand how the program is laid out. Function parameters are allocated first on the stack, and then local variables are allocated next. Since the process fork to handle client connections creates a stack canary based on the value passed in to the forked function, this creates the situation that we exploited, since the attacker can manipulate the stack cookie value overflowing both the secret and cookie to an arbitrary number. A way to prevent buffer overflows using canaries could involve creating a canary that started with a NULL byte, so that when an attacker tries to overflow this value, they hit an EOF and their socket send function terminates.

Buffer overflow countermeasures

1) Using ASLR to prevent arbitrary shellcode execution. This happens by randomizing stack and heap address locations for a process. Further isolation and randomization of virtual memory layout for a process also helps to prevent return-oriented exploits and buffer overflows.
2) Making the stack not-executable will also greatly help to prevent buffer overflows.