The vulnerability in server.c was with regards to the snprintf() function call in the if clause of the vulnerable() function. The cause of the vulnerability is that snprintf takes a string formatting function, but the call to it in vulnerable does not include a string formatter. So, if the input string contains a sequence of characters that are format strings (%s,%p,%x,etc), they are evaluated as a command. So for example if %s is passed as an input string to snprintf, it will try to interpret it as a pointer to a string starting from the buffer location and retrieve character strings from the stack. Since the parameters for vulnerable are a pointer to the guess and the secret word, we know that the pointer to the secret word is in the stack frame above vulnerable()'s (where function arguments are placed in x86). So, as input for a guess, I needed to exploit the snprintf call in vulnerable by passing in format strings, so the program starts reading from memory starting in the callframe for snprintf. So I would need enough "padding" to reach the correct memory location of where the secret is, then utilize the %s string formatter to read the string. Lastly, I needed to modify the input with some scripting so that the values read from the padding formatters are not also output.