

1. Explain, in your own words, how a buffer overflow occurs in memory.

2. Describe a real-world situation (that was not discussed in class) in which a person can stuff more into a container than the designer initially intended. What are the potential consequences?

3. Classify each of the following instances as a Threat, Vulnerability, or an Exploit.

A) Unlocked Door _____

B) Thief _____

C) Burglary _____

4. You have discovered a buffer overflow vulnerability that will allow you to run 1 program of your choosing only once, and never again.

A) Which program should you choose?

a) Facebook.com

b) World of Warcraft

c) Terminal

d) Email

Why?

B) What privilege level would you ideally run this program with?

SHOW ALL WORK

5. Given the following variable declarations and stack diagram (no padding) for a game:

	Address	Value	Variable
<code>char user[16];</code>	0xbffff800->		<-nickname
<code>int highscore;</code>	0xbffff80c->	1000	<-highscore
<code>char nickname[x];</code>	0xbffff810->	Flynn	<-user
	0xbffff820->	0xbffff840	<-SFP
	0xbffff824->	0x080483e0	<-Return Address

This program allows you to enter your nickname when you run it from the command line.

```
> ./game nickname
```

A) What is the value of 'x'? (# of bytes allocated for **nickname**)

B) How many bytes must you put into **nickname** to overwrite **highscore**?

C) How could you change the above variable declarations to ensure **highscore** could not be overwritten?

D) How many bytes must you put into **nickname** to crash the program?

Extra Credit) What string can you put into **nickname** to overwrite **highscore** with value 16705?

ALL: Compile **game_of_chance.c** on page 103 and play a few rounds. You've earned it!