CSE 484 In-Class Worksheet #5 – Spring 2019

Name: _____ UWNetID: _____ Date: _____

Email address: _____

Partner names for this activity: _____

Will you want to pick up your worksheet later? Circle one: Yes / No

Q1: Consider again the following vulnerable function:

```
foo() {
      char buf[…];
      strncpy(buf, readUntrustedInput(), sizeof(buf));
      printf(buf); //vulnerable
}
```

And suppose **readUntrustedInput()** provides an attack string of the form:

```
… attackString%n … <shellcode> …
```

Your goal is to set up the attackString such that printf's internal stack pointer points to the saved RET on the stack when the %n is processed. What value do you want written when %n is processed? In other words, the number of characters in "attackString" must be equal to… what? What are the challenges with that idea, and how might you overcome it?

Q2: What might an attacker be able to accomplish even if they cannot execute code on the stack?

Q3: What might be a good value for a stack canary?