CSE 484 In-Class Worksheet #4 – Spring 2019

Name: ________________ UWNetID: _____________ Date: __________________

Email address: ____

Partner names for this activity: ____________________________________

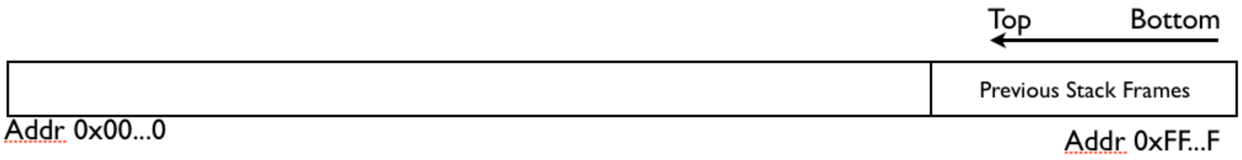
Will you want to pick up your worksheet later? Circle one: Yes / No

Q1: Consider this code:

```
void mycopy(char *input) {
      char buffer[512]; int i;

      for (i=0; i<=512; i++)
            buffer[i] = input[i];
}
void main(int argc, char *argv[]) {
      if (argc==2)
            mycopy(argv[1]);
}
```

Is this code exploitable? If not, why not? If so, why? You may use the diagram below to help answer this question, if you wish.



Q2: Consider the following function:

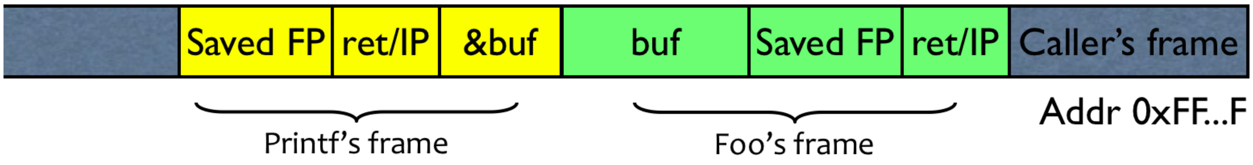
```
foo() {
      char buf[…];
      strncpy(buf, readUntrustedInput(), sizeof(buf));
      printf(buf); //vulnerable
}
```

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

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

How might we be able to use %n to overwrite the saved EIP (aka RET) on the stack? (You don't need to give the exact attack; just brainstorm about the general approach you might try.)

As a reminder, here's what the stack looks like for this program:


Saved FP
ret/IP
&buf
buf
Saved FP
ret/IP
Caller's frame
Printf's frame
Foo's frame
Addr 0xFF...F