CNT 4419, Sec 001, Test 2, 75 minutes **NAME:**_____

Instructions: All the instructions that are on the syllabus. For example, this test is closed everything, including phones, notes, AI, and other students. Do not talk to another student during the test. Do not look at another student's answers during the test. Do not ask a question during the test that gives away any part of any answer. If a response length is indicated, respond at that length and do not use bullet points or enumerated lists. Unless stated otherwise, use the same notations, definitions, and assumptions that we have been using in class. Respond at the level of detail discussed in class. This test is 6 pages.

1. [9 points] In class we discussed the classic example of a confused-deputy attack. Explain how the overarching vulnerability in this example could be considered as allowing violations of C, I, and A policies (per the CIA triad), following our discussion in class. [3-5 sentences]

2. [6 points] Summarize the Biba-Integrity model. In what context would it be used? Hit all the main points discussed in class. [2-4 sentences]

3. [12 points] Compare and contrast access-control lists and capability lists, hitting all the main points discussed in class. [2-4 sentences]

4. [8 points] Describe a path-traversal vulnerability in the SimpleWebServer, as discussed in class. Why does the vulnerability exist, how could an attacker exploit it, and how could it be mitigated? [1 paragraph]

5. [36 points] Consider the following C code. Assume that all required #includes are present; the program is valid and executed on a 32-bit architecture; none of the library functions return errors during execution; all mallocs get allocated into contiguous (adjacent) memory, each allocation immediately after the previous one, with earlier mallocs at lower addresses than later mallocs; and there are no extra security mechanisms in place (no NX bits, canaries, ASLR, or CFI).

```
#define SLEN 256
typedef struct security_flags { //a 2-byte record of security flags
  char disallowFileAccess; //if nonzero then disallow access to file system
  char disallowNetworkAccess; //if nonzero then disallow access to network
} SF;  //now the type SF refers to a record of security flags for the program
void setup_flags(SF *sf) { //implement a secure default: disallow everything
  sf->disallowFileAccess = 1;
  sf->disallowNetworkAccess = 1;
  //PROGRAM POINT 1
}
int str_len(int base, char need_nul) {
  if(need_nul) return base+1; //add 1 extra byte if a NUL-terminator is needed
  else return base;
}
int main() { //Reminder: the argument to malloc is the # of bytes to allocate
  char *s = (char *)malloc(SLEN);
  SF *sf = (SF *)malloc(sizeof(SF));
  setup_flags(sf);
  //PROGRAM POINT 2
  fgets(s, str_len(SLEN,1), stdin);
  ... //etc.
}
```

a) Draw and label 2 pictures to show all of program memory at the 2 program points marked above. When values in memory are known, show those values in your pictures. Also show the values of FP and SP.

b) Explain how the code is vulnerable to an attack, and explain the attack, including the concrete step(s) the attacker could take to exploit the vulnerability. [2-5 sentences]

If it helps, here is documentation for the function "`char *fgets(char *s, int n, FILE *stream)`": The fgets() function shall read bytes from stream into the array pointed to by s until n-1 bytes are read, or a <newline> is read and transferred to s, or an end-of-file condition is encountered. A null byte (i.e., a byte of 0s for the NUL-terminator) shall be written immediately after the last byte read into the array.

c) Would NX bits mitigate the code vulnerability? Explain. [1 sentence]

d) Would stack canaries mitigate the code vulnerability? Explain. [1 sentence]

e) Would ASLR mitigate the code vulnerability? Explain. [1 sentence]

f) Would CFI mitigate the code vulnerability? Explain. [1 sentence]

g) What are 2 logically distinct code modifications or rewrites that a programmer could implement, to mitigate the vulnerability? [1-2 sentences]

6. [4 points] As discussed in class, what are two standard components of an access-control mechanism? [1 sentence]

7. [25 points] For this test, define the concatenation of two properties $G_1$ and $G_2$ as follows.

$$G_1;G_2 = \{ t_1 \mid t_1 \text{ is infinite and } t_1 \in G_1 \} \cup \{ t_1;t_2 \mid t_1 \text{ is finite and } t_1 \in G_1 \text{ and } t_2 \in G_2 \}$$

Note that this new definition of property concatenation differs from the old definition on Test 1.
Using this new definition of property concatenation, prove or disprove:
a) Safety properties are closed under concatenation. (Hint: Consider $G_2=\emptyset$.)
b) Liveness properties are closed under concatenation.