CNT 4419, Sec 001, Exam, 120 minutes **NAME:**_____

Instructions: All the instructions that are on the syllabus. For example, this exam is closed everything, including phones, notes, AI, and other students. Do not talk to another student during the exam. Do not look at another student's answers during the exam. Do not ask a question during the exam 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 exam is 7 pages.

1. [12 points] What are best practices for creating, using, modifying, and deleting session IDs, as discussed in class? Hit all the main points discussed in class. [3-5 sentences]

2. [5 points] Explain intuitively why all access-control policies are safety properties. [2-3 sentences]

3. [3 points] In which program segments can buffer overflows occur? [1 sentence]

4. [4 points] Explain, at the level of detail discussed in class, the spectrum of CFI enforcement, from coarse to fine grained. [1-3 sentences]

5. [7 points] Based on our in-class discussion, explain how an active attacker can defeat Diffie-Hellman key exchange. Include a diagram in your response, like the other sorts of diagrams we discussed in class, to show how the attack proceeds. [2-5 sentences + diagram]

6. [6 points] Suppose an Inventory table has columns for SKU and Name, and a Purchases table has columns for SKU and Purchaser. Write a SQL query to find and return the total number of purchases containing items not currently in inventory.

7. [5 points] As shown in class, write a short snippet of code to illustrate the creation and execution of a parameterized query with JDBC. Be sure your example query contains the same types of parameters as the example from class. Don't worry about the auxiliary code to create a database connection, and don't worry about getting the method and type names exactly right—we just want to see that you know all the basic ideas of creating and executing a parameterized query with JDBC.

8. [18 points] [Essay] As discussed in class, what is the primary characteristic of an application that makes it vulnerable to a code-injection attack? Identify all the different kinds of code-injection attacks discussed in class (for example, SQLIAs are one kind of code-injection attack), and for each kind, briefly explain that kind of attack at the level of detail discussed in class.

9. [20 points]
Consider the following code, assuming a 32-bit architecture, that all needed #include's are present, that each int and char is stored in 1 byte, and that getUInt securely inputs an unsigned int from the user. Also assume an optimized memory layout, where system calls are not given their own stack frames. According to its documentation, the 2nd argument to fgets, here an unsigned int, "is the maximum number of characters to be read (including the final null-character)".

```
0       #define MAX 128
1       void getMessage(char *name, unsigned int size) {
2          char msg[MAX];
3          printf(name);
4          if(size+1 <= MAX) { //add 1 to be sure there's enough space for the null-char
5               fgets(msg, size, stdin);
6          }
7       }
8       int main() {
9          char name[MAX];
10         unsigned int size;
11         fgets(name, MAX, stdin);
12         size = getUInt();
13         getMessage(name, size);
14      }
```

a) Describe how an attacker could overflow a buffer in this code. E.g., which buffer could be overflowed, on which line of code, which inputs would the attacker provide, and what would those inputs enable?

b) Using your basic technique from Part (a), explain a classic stack-smashing attack on this code. Assume that the system lacks ASLR, NX bits, and StackGuard.

c) Describe a format-string attack on this code in detail, assuming the system uses ASLR, NX bits, and StackGuard with 4-byte canaries. Describe the information an attacker gains from this attack.

d) Referencing Parts (a)-(c) as helpful, describe a maximally effective attack on this code assuming ASLR, NX bits, and StackGuard are in use.

10. [20 points] For this exam, define the concatenation of two properties G_1 and G_2 as follows.

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

Note that this new definition of property concatenation differs from the old definitions on Tests 1 and 2. Using this new definition of property concatenation, **prove or disprove**:
a) Liveness properties are closed under concatenation.
b) Safety properties are closed under concatenation.