CNT 4419: Secure Coding [Fall 2023]
Test I

NAME: _____

Instructions:

1) This test is 7 pages in length.

2) You have 75 minutes to complete and turn in this test.

3) Short-answer and essay questions include guidelines for how much to write. Respond in complete English sentences. Responses will be graded as described on the syllabus. Respond at the level of detail discussed in class. Avoid using bullet points and enumerated lists.

4) This test is closed books, notes, papers, phones, smartwatches, laptops, friends, neighbors, etc.

5) Use the backs of pages in this test packet for scratch work. If you write more than a final answer in the area next to a question, circle your final answer.

1. [10 points] For each of the following terms, explain it in 1 sentence.
a) SOC

b) Honeypot

c) Dialog fatigue

d) Conservative enforcement

e) NB

2. [4 points] Given an arbitrary set G of traces, define a property P having G as its good-trace set. Use set-builder notation, like we did in class.

3. [3 points] Given an arbitrary policy P, define a mechanism M that *soundly* enforces P.

4. [3 points] Given arbitrary policy P, define a mechanism M that *completely* enforces P.

5. [8 points]
Suppose a thief obtains the password to unlock a victim's phone, steals the phone, and successfully uses the password on the stolen phone.
a) Explain how the phone's password checker may be considered to exhibit a false negative. [1-2 sentences]

b) Explain how the phone's password checker may be considered to exhibit a true negative. [1-2 sentences]

6. [12 points] [*Short* essay] Why do we care about categorizing security policies as properties, safety, and/or liveness? As part of your response, explain: How could you try to enforce an arbitrary property in practice?

7. [60 points]
Draw a picture to show the relationships between policies, properties, safety, and liveness. Then, in your picture, pinpoint the locations of each of the following policies, based on their categorization. Prove that your categorizations are correct on the subsequent pages of this test.

For all programs p:

- p is in P_1 iff out('a') is not in any trace in p
- p is in P_2 iff in('a');out('a');in('a');out('a');... is in p
- p is in P_3 iff in('a');out('a');in('a');out('a');... is not in p
- p is in P_4 iff p has the form $\{t_1;\text{output}(k_1), t_2;\text{output}(k_2), \ldots\}$ such that $\{k_1,k_2,\ldots\} \neq K$, where K is the set of all 128-bit keys
- p is in P_5 iff p is infinite
- p is in P_6 iff all traces in p are infinite
- p is in P_7 iff p is the empty set
- p is in P_8 iff p is in P_1 and P_3
- p is in P_9 iff p is in P_1 and P_5
- p is in P_{10} iff p is in P_1 and P_6
- p is in P_{11} iff p is in P_1, P_2, …, and P_{10}
  [Hint: This last one is challenging; think carefully about the definitions. Everyone will receive enough bonus points to make P_{11} equivalent to an extra credit question.]

[This page provides additional space for Problem 7.]

[This page provides additional space for Problem 7.]

[This page provides additional space for Problem 7.]