CSE127 — Midterm

Yee **Winter '02**

Name and Student ID: Answer Key

There are a total of 9 questions on 6 pages. There are 100 points possible. You might not have time to finish the entire exam — don't be discouraged. Wait until the instructor has passed out exams to everybody before you start. Advice: skim through the entire test to determine which of the problems you can solve quickly and work on those first, rather than getting stuck on a hard problem early and wasting too much of your time on it.

When you can start, you should first make sure that you have all the pages, and write your name and your student ID on the first page, and your student ID on the top of *all subsequent pages.* Pages of this exam will be separated and graded separately — if you fail to write your ID at the top of a page, you will not receive credit for answers on that page. **Write clearly**: if we cannot read your handwriting or your pencil smudges, you will not properly get credit for your answers.

This exam is closed book, closed notes.

No electronic computation aids are allowed.

Prob	1	2	3	4	5	6	7	8	9	**Total**
Score										
Poss	10	9	16	10	10	10	10	10	15	100

1 (Class basics) What does the compute security code of ethics which you signed say? (You do not need to have it memorized; a paraphrasing is okay.)

(10pts)

I will not use the knowledge learned from CSE 127 to break into computer systems uninvited. I will not share that knowledge with others unless they also agree to the Computer Security Code of Ethics.

2 (Basic Concepts) What does the KISS ("Keep It Simple, Stupid") rule mean? State it and explain its significance to computer security.

(9pts)

The idea behind the KISS rule is to keep designs as simple as possible (while still achieving the design goals), thereby making the implementation easier and verification simpler. Excess complexity leads to bugs.

3 (Basic Concepts) Give definitions of the following

A. Confidentiality

B. Integrity

C. Accountability / non-repudiation

D. Availability

(16pts)

A. Confidentiality: Only authorized users are allowed to read confidential data files; others are denied access.

B. Integrity: Only authorized users are allowed to modify (write) protected files; others are denied access.

C. Accountability: System activities are logged so that users can be made responsible for their actions.

D. Availability: Otherwise unused system resources remain available for use, rather than be tied up inappropriately.

4 (Basic Concepts) What is a Threat Model? What is Risk Management? Explain what these concepts are and how they relate to computer security.

(10pts)

Threat Models and Risk Management form part of the security model. To build a threat model, we identify the potential threats to the system — who the attackers are, what the means and methods of attack might be, e.g., sources of risks such as known as well as undiscovered bugs, operational security risks, physical security risks, etc.

After identifying the threats in threat modeling, we estimate the likelihood that the threats will materialize, determine the security assets that we wish to protect, and identify the risk mitigation strategies that we may wish to deploy and gauge their effectiveness and cost. The expected loss is the risk that must be managed — e.g., by chosing the appropriate risk mitigation technique (cost-effective security mechanisms or tighter usage policies) — so that the expected cost becomes acceptable.

5 (Defense) Suppose a security consultant to your company suggested that you should buy a new security device to protect your company's computers. The device costs $100,000 initially, and $20,000 per year to operate. The consultant assures you that the device will completely eliminate security incidents from the average of 4 per year, which costs your company about $10,000 per incident (in labor, lost customer confidence, etc). There are, of course, uncertainties for all of these estimates/claims. (1) Suppose this device operates as claimed for the forseeable future. Should you adopt this device? (2) Suppose you believe that the attackers will learn about how the device works and develop new, more sophisticated scripts that will be able to bypass it. You estimate that after 2 years of use the device will provide only partial protection, lowering the incidence rate to 2 per year and after 4 years (or 2 additional years) the device will be completely ineffective, should you adopt this device?

(10pts)

(1) Yes. If the device continues to operate as claimed for at least 5 years, it have cost the company $\$100,000 + 5 \cdot \$20,000 = \$200,000$ and saved the company $5 \cdot 4 \cdot \$10,000 = \$200,000$; this is when the cross-over occurs: at each subsequent years the company saves $\$20,000$ per year. (2) No. If the device will be of reduced effectiveness after 2 years and be useless after 4 years, then using it for the maximum lifetime of 4 years would cost $180,000, but save only $120,000 in expenses; we similarly lose money if we used the device for some shorter duration. Thus, using the device would not save any money overall — it will never recoup the initial lump sum expense.

6 (Attack) What are "buffer overflow" bugs? How can they let attackers penetrate a system? (10pts)

Buffer overflow bugs occur in languages where there is no array-bounds checking. If input can be provided to a program which causes array indexing to go out of bounds on a write operation, unintended memory changes can occur; in the common case of an input buffer array allocated in stack memory, by overflowing such an array an attacker can change control state such as return addresses to cause control to be transferred to inappropriate code (or to data that is interpreted as code).

7 (Concepts) What is the notion of the "weakest link"? Why is this important to computer security? (10pts)

The "weakest link" is the part of the defense that is easiest for an attacker to overcome. This is important because an intelligent attacker will identify the weakest part of the defenses and use that as the preferred avenue of attack. When working to improve the security of a system, identifying the weakest link and improving it will tend to yield the greatest return in increased security. (N.B.: after improving the weakest link, some other part of the system will become the (new) weakest link, and additional work should go into improving it.)

8 (Concepts) What is a Trusted Computing Base?

(10pts)

A Trusted Computing Base (TCB) is that portion of the system that must be trusted — it implements the security mechanisms which enforces the system security policy. Typically this includes the hardware, the operating system kernel, the user authentication subsystem (e.g., login program), and system administration tools. The compiler may be part of the TCB — when compiling and installing new system software from source is part of the system administration activity — or might not, e.g., when the system is used for running programs and the programs are written and compiled elsewhere.

9 (Defense) Find the loop invariant for the following function which will enable us to prove that it satisfies its specification. Mark where in the code the loop invariant should hold, and argue why the loop invariant helps to prove that the code satisfies the specifications.

```
double avg(double data[], int nelt)
{
        double sum = 0.0; int i;
        for (i = nelt; --i >= 0; )
                sum += data[i];
        return sum / nelt;
}
```

Specifications:

- precondition:

$$\mathrm{nelt} > 0$$

- postconditon:

$$\mathrm{avg}(\mathrm{d}, \mathrm{n}) = \frac{1}{\mathrm{n}} \sum_{i=0}^{\mathrm{n}-1} \mathrm{d}[i]$$

(15pts)

The invariant $\mathrm{sum} = \sum_{j=i+1}^{\mathrm{nelt}-1} \mathrm{data}[j]$ holds immediately after the boolean test `--i >= 0` and before the execution of the loop body. The invariant holds upon entry to the loop, since sum = 0.0 and $i = \mathrm{nelt} - 1$, so

$$\mathrm{sum} = \sum_{j=i+1}^{\mathrm{nelt}-1} \mathrm{data}[j] = \sum_{j=\mathrm{nelt}}^{\mathrm{nelt}-1} \mathrm{data}[j] = 0$$

is trivially true.

The body of the loop consists of the statements `sum += data[i]; --i;`, so we have $i' = i - 1$, and

$$\mathrm{sum}' = \mathrm{sum} + \mathrm{data}[i] = \left(\sum_{j=i+1}^{\mathrm{nelt}-1} \mathrm{data}[j] \right) + \mathrm{data}[i] = \sum_{j=i}^{\mathrm{nelt}-1} \mathrm{data}[j] = \sum_{j=i'+1}^{\mathrm{nelt}-1} \mathrm{data}[j]$$

and thus the invariant holds after the body executes.

The loop terminates when $i = -1$, so $\mathrm{sum} = \sum_{j=0}^{\mathrm{nelt}-1} \mathrm{data}[j]$, and the returned value is the desired $\mathrm{sum}/\mathrm{nelt} = \frac{1}{\mathrm{nelt}} \sum_{j=0}^{\mathrm{nelt}-1} \mathrm{data}[j]$.

Note: A common notational error is writing the summation with the lower bound greater than the upper bound. When this occurs the summation is by definition zero.