

CSE 497B/Spring 2007 - Quiz 3  
Tuesday, April 17, 2007 — Professor Trent Jaeger

Please read the instructions and questions carefully. You will be graded for clarity and correctness. You have 25 minutes to complete this quiz, so focus on those questions whose subject matter you know well. Write legibly and check your answers before handing it in.

**Short Answer - some will be one or two words – no more than 3 sentences**

1. (3pts) List the *three guarantees* required of a correct *reference monitor*?

*answer:* (1) the reference monitor interface, mechanism, and policy are *tamperproof*; (2) the reference monitor provides *complete mediation* for all security-sensitive operations; and (3) the reference monitor interface, mechanism, and policy are *simple enough to enable verification*.

2. (3pts) How is hardware-assisted *authenticated boot* different than *secure boot*?

*answer:* Authenticated boot takes integrity measurements as each component (e.g., of code and static data) is loaded during the boot process, but does not terminate the boot process based on the integrity of some component.

3. (3pts) List three requirements in the use of a *canary* to protect a return address from a buffer overflow.

*answer:* (1) must be between the buffer and the return address on the stack; (2) must have a value that is secret from the attacker; (3) must be checked before use of that return address.

**Long Answer - no more than 2 paragraphs.**

4. (7pts) In terms of the *reference monitor guarantees*, describe the reasons why *discretionary access control systems* used by Windows and UNIX cannot prevent data leaks if a user runs malware.

*answer:* (1) the malware can *tamper* with the DAC policy enforced by the reference monitor; (2) the systems do not enforce *complete mediation* (e.g., of network communications), so malware can leak via these paths; (3) with the possible changes in policy and complexity of permission assignment it is not possible to *verify correct monitoring* (actually would fail verification for the reasons above).

**Word Problems - take your time and answer clearly and completely.**

5. (10pts) Suppose that we have a multilevel security (MLS) lattice policy consisting of four levels, *top secret*, *secret*, *confidential*, and *unclassified*, in order from most secret to least secret. Further, you have the following categories *nuclear (NUC)*, *intelligence (INTEL)*, and *defense (DEF)*. Answer the following questions relative to this MLS policy.

(a) (2pts) Can a subject cleared at *secret* with categories  $\{NUC, DEF\}$  **read** an object whose access class is *confidential*,  $\{DEF\}$ ?

(b) (2pts) Can a subject cleared at *secret* with categories  $\{NUC, DEF\}$  **write** an object whose access class is *top secret*,  $\{DEF\}$ ?

(c) (2pts) Why can't a subject cleared at *secret*,  $\{NUC\}$  **read** an object whose access class is *unclassified*  $\{NUC, DEF\}$ ?

(d) (2pts) Why can't a subject cleared at *secret*,  $\{NUC\}$  **write** an object whose access class is *unclassified*  $\{NUC, DEF\}$ ?

(e) (2pts) Assume the following MLS access classes for the variables in function *foo* below: (1) *a* is *secret*; (2) *b* is *secret*; and (3) *c* is *public*. Does the following program satisfy the Denning lattice policy? Why?

```
foo (int a) {  
    int b, c;  
    b = 1;  
  
    if (a != 1)  
        b = a;  
    else  
        c = 0;  
}
```

*answer:* (a) Yes.

(b) No.

(c) The subject must be cleared for the category *DEF* as well, otherwise DEF data is leaked to the subject by this read request.

(d) The subject cannot write to a lower level due to the  $\star$ -property. (e) No. *a* leaks to *c*.